Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Learning Zero-Sum Simultaneous-Move Markov Games Using Function Approximation and Correlated Equilibrium

Qiaomin XieYudong Chen, Zhaoran Wang, Zhuoran Yang

informs
http://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–30
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Learning Zero-Sum Simultaneous-Move Markov Games Using Function Approximation and Correlated Equilibrium

Qiaomin Xie[a,*] Yudong Chen,[b] Zhaoran Wang,[c] Zhuoran Yang[d]

[a] Department of Industrial Systems and Engineering, University of Wisconsin-Madison, Madison, Wisconsin 53706; [b] Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin 53706; [c] Department of Industrial Engineering and Management Sciences, Northwestern University Evanston, Illinois 60208; [d] Department of Statistics and Data Science, Yale University, New Haven, Connecticut 06511
*Corresponding author

Contact: qiaomin.xie@wisc.edu, https://orcid.org/0000-0003-2834-6866 (QX); yudong.chen@wisc.edu, https://orcid.org/0000-0002-6416-5635 (YC); zhaoranwang@gmail.com, https://orcid.org/0000-0002-1824-2580 (ZW); zhuoran.yang@yale.edu, https://orcid.org/0000-0001-5269-9958 (ZY)

Received: May 29, 2020
Revised: September 30, 2021
Accepted: February 12, 2022
Published Online in Articles in Advance: June 29, 2022

MSC2000 subject classification: Primary: 68T05; secondary: 91A15

https://doi.org/10.1287/moor.2022.1268

Abstract. We develop provably efficient reinforcement learning algorithms for two-player zero-sum finite-horizon Markov games with simultaneous moves. To incorporate function approximation, we consider a family of Markov games where the reward function and transition kernel possess a linear structure. Both the offline and online settings of the problems are considered. In the offline setting, we control both players and aim to find the Nash equilibrium by minimizing the duality gap. In the online setting, we control a single player playing against an arbitrary opponent and aim to minimize the regret. For both settings, we propose an optimistic variant of the least-squares minimax value iteration algorithm. We show that our algorithm is computationally efficient and provably achieves an $\widetilde{O}(\sqrt{d^3H^3T})$ upper bound on the duality gap and regret, where d is the linear dimension, H the horizon and T the total number of timesteps. Our results do not require additional assumptions on the sampling model. Our setting requires overcoming several new challenges that are absent in Markov decision processes or turn-based Markov games. In particular, to achieve optimism with simultaneous moves, we construct both upper and lower confidence bounds of the value function, and then compute the optimistic policy by solving a general-sum matrix game with these bounds as the payoff matrices. As finding the Nash equilibrium of a general-sum game is computationally hard, our algorithm instead solves for a coarse correlated equilibrium (CCE), which can be obtained efficiently. To our best knowledge, such a CCE-based scheme for optimism has not appeared in the literature and might be of interest in its own right.

Funding: Q. Xie is partially supported by the National Science Foundation [Grant CNS-1955997] and by J.P. Morgan. Y. Chen is partially supported by the National Science Foundation [Grants CCF-1657420, CCF-1704828, and CCF-2047910]. Z. Wang acknowledges the National Science Foundation [Grants 2048075, 2008827, 2015568, and 1934931], the Simons Institute (Theory of Reinforcement Learning), Amazon, J.P. Morgan, and Two Sigma for their support.

Keywords: Markov games • reinforcement learning • function approximation • correlated equilibrium

1. Introduction

Reinforcement learning (Sutton and Barto [83]) is typically modeled as a Markov decision process (MDP) (Puterman [71]), where an agent aims to learn the optimal decision-making rule via interaction with the environment. In multiagent reinforcement learning (MARL), several agents interact with each other and with the underlying environment, and their goal is to optimize their individual returns. This problem is often formulated under the framework of Markov games (Shapley [76]), which is a generalization of the MDP model. Powered by function approximation techniques such as deep neural networks (Goodfellow et al. [33], LeCun et al. [50]), MARL has recently enjoyed tremendous empirical success across a variety of real-world applications. A partial list of such applications includes the game of Go (Silver et al. [78], [79]), real-time strategy games (Open AI [62], Vinyals et al. [86]), Texas hold'em poker (Brown and Sandholm [14], [15], Moravčík et al. [59]), autonomous driving (Shalev-Shwartz et al. [75]), and learning communication and emergent behaviors (Baker et al. [10], Bansal et al. [11], Foerster et al. [32], Jaques et al. [39], Lowe et al. [56]); see the surveys in Busoniu et al. [16] and Zhang et al. [99].

In contrast to the vibrant empirical study, theoretical understanding of MARL is relatively inadequate. Most existing work on Markov games assumes access to either a sampling oracle or a well-explored behavioral policy, which fails to capture the exploration-exploitation tradeoff that is fundamental in real-world applications of reinforcement learning. Moreover, these results mostly focus on the relatively simple turn-based setting. An exception is the work in Wei et al. [89], which extends the UCRL2 algorithm (Jaksch et al. [38]) for MDP to zero-sum simultaneous-move Markov games. However, their approach explicitly estimates the transition model and thus only works in the tabular setting. Problems with complicated state spaces and transitions necessitate the use of function approximation architectures. In this regard, a fundamental question is left open: Can we design a provably efficient reinforcement learning algorithm for Markov games under the function approximation setting?

In this paper, we provide an affirmative answer to this question for two-player zero-sum Markov games with simultaneous moves and a linear structure. In particular, we study an episodic setting, where each episode consists of H timesteps and the players act simultaneously at each timestep. Upon reaching the Hth timestep, the episode terminates and players replay the game again by starting a new episode. Here, the players have no knowledge of the system model (i.e., the transition kernel) nor access to a sampling oracle that returns the next state for an arbitrary state-action pair. Therefore, the players have to learn the system from data by playing the game sequentially through each episode and repeatedly for multiple episodes. More specifically, we study episodic Markov games under both the offline and online settings. In the offline setting, both players are controlled by a central learner, and the goal is to find an approximate Nash equilibrium of the game, with the approximation error measured by a notion of duality gap. In the online setting, we control one of the players and play against an opponent who implements an arbitrary policy. Our goal is to minimize the total regret, defined as the difference between the cumulative return of the controlled player and its optimal achievable return when the opponent plays the best response policy. Both settings are generalizations of the regret minimization problem for MDPs. Here we use "online" to emphasize the fact that we only control one player; this terminology is common in the literature (Jin et al. [42], Tian et al. [84]. Wei et al. [89]). Correspondingly, the name "offline" highlights the fact that we control both players. We remark that both settings are online in the sense that the samples are collected via iterative interaction with the environment.

Furthermore, to incorporate function approximation, we consider Markov games with a linear structure, motivated by the linear MDP model recently studied in Jin et al. [44]. In particular, we assume that both the transition kernel and the reward admit a d-dimensional linear representation with respect to a known feature mapping, which can be potentially nonlinear in its inputs. For both the online and offline settings, we propose the first provably efficient reinforcement learning algorithm without additional assumptions on the sampling model. Our algorithm is an optimistic version of minimax value iteration (OMNI-VI) with least-squares estimation—a model-free approach—which constructs upper confidence bounds of the optimal action-value function to promote exploration. We show that the OMNI-VI algorithm is computationally efficient, and it provably achieves an $\widetilde{O}(\sqrt{d^3H^3T})$ regret in the online setting and a similar duality gap guarantee in the offline setting, where T is the total number of timesteps and $\widetilde{O}$ omits logarithmic terms. Note that the bounds do not depend on the cardinalities of the state and action spaces, which can be very large or even infinite. When specialized to MDPs, our results recover the regret bounds established in Jin et al. [44] and are thus near-optimal.

We emphasize that the Markov game model poses several new and fundamental challenges that are absent in MDPs and arise due to subtle game-theoretic considerations. Addressing these challenges require several new ideas, which we summarize as follows:

1. **Optimism via general-sum games.** In the offline simultaneous-move setting, implementing the optimism principle for both players amounts to constructing both upper and lower confidence bounds (UCB and LCB) for the optimal value function of the game. Doing so requires one to find, as an algorithmic subroutine, the solution of a general-sum (matrix) game where the two players' payoff functions correspond to the upper and lower bounds for the action-value (or Q) functions of the original Markov game, even though the latter is zero-sum to begin with. This stands in sharp contrast of turn-based games (Hansen et al. [36], Jia et al. [40], Sidford et al. [77]), in which each turn only involves constructing an UCB for one player.

2. **Using correlated equilibrium.** Finding the Nash equilibrium (NE) of a general-sum matrix game, however, is computationally hard in general (Chen et al. [18], Daskalakis et al. [23]). Our second critical observation is that it suffices to find a *coarse correlated equilibrium* (CCE) (Aumann [6], Moulin and Vial [60]) of the game. Originally developed in algorithmic game theory, CCE is a tractable notion of equilibrium that strictly generalizes NE. In contrast to NE, a CCE can be found efficiently in polynomial time even for general-sum games (Blum et al. [12], Papadimitriou and Roughgarden [65]). Moreover, our analysis shows that using any CCE of the matrix general-sum game are sufficient for ensuring optimism for the original Markov game. Thus, by using CCE instead of NE, we achieve efficient exploration-exploitation balance while preserving computational tractability.

3. **Concentration and game stability.** The last challenge is more technical, arising in the analysis of the algorithm where we need to establish certain uniform concentration bounds for the CCEs. As we elaborate later, the CCEs of

a general-sum game are unstable (i.e., not Lipschitz) with respect to the payoff matrices. Therefore, standard approaches for proving uniform concentration, such as those based on covering/ϵ-net arguments, are fundamentally insufficient. We overcome this issue by carefully stabilizing the algorithm, for which we make use of an ϵ-net in the algorithm. Moreover, we show that this can be done in a computationally efficient way via rounding on the fly.

We shall discuss the aforementioned challenges and ideas in greater detail when we formally describe our algorithms. We note that our regret and duality gap bounds also imply polynomial sample complexity and Provably and Approximately Correct (PAC) guarantees for learning the NEs of simultaneous-move Markov games. Moreover, as turn-based games can be viewed as a special case of simultaneous games, where at each state the reward and transition kernel only depend on the action of one of the players, our algorithms and guarantees readily apply to the turn-based setting.

1.1. Related Work

There is a large body of literature on applying reinforcement learning methods to Markov games (also known as stochastic games). Most of existing results provide convergence guarantees that are asymptotic in nature. In particular, under the tabular setting, the work of Grau-Moya et al. [34], Greenwald et al. [35], Hu and Wellman [37], Littman [52], [53], and Littman and Szepesvári [54] extends the value iteration and Q-learning algorithms (Watkins and Dayan [88]) to zero-sum and general-sum Markov games, and that in Pérolat et al. [66] extends the actor-critic algorithm (Konda and Tsitsiklis [47]) to zero-sum Markov games; asymptotic convergence guarantees are obtained in these papers. Besides, the work in Srinivasan et al. [81] proposes a variant of actor-critic to tabular multiagent extensive-form games with finite-time convergence results.

We focus our discussion on results with nonasymptotic guarantees. In Table 1, we summarize some most related work on Markov games. One line of work assumes access to a sampling oracle. Particularly related to us is the work in Sidford et al. [77], which proposes a variance-reduced version of the minimax Q-learning algorithm with near-optimal sample complexity. We remark that the theoretical results therein require a sampling oracle, and they focus on turn-based games, a special case of simultaneous-move games. The recent work by Jia et al. [40] studies turn-based zero-sum Markov games, where the transition model is assumed to be embedded in some d-dimensional feature space, extending the MDP model in Yang and Wang [94]. Assuming a sampling oracle, they propose a variant of Q-learning algorithm that is guaranteed to find an ϵ-optimal strategy using $\widetilde{O}(d(1-\gamma)^{-4}\epsilon^{-2})$ samples, where γ is the discount factor. Recently, the work by Zhang et al. [99] considers a model-based algorithm that finds the ϵ-NE of tabular Markov games with $\widetilde{O}(SAB(1-\gamma)^{-3}\epsilon^{-2}2)$ sample complexity, assuming a sampling oracle, where S is the size of the state space, and A and B are the sizes of the actions spaces of the two players. This bound is shown to be minimax-optimal if the algorithm is reward-agnostic, meaning that the algorithm only queries state transition samples but not the reward.

Another line of work (Fan et al. [31], Lagoudakis and Parr [48], Pérolat et al. [67], [68], [69], [70]) considers function approximation techniques applied to variants of value iteration methods and establishes finite-time convergence to the NEs of two-player zero-sum Markov games. These results are based on the framework of fitted value iteration (Munos and Szepesvári [61]) and assume the availability of a well-explored behavioral policy. In summary, all of the works cited require either a sampling oracle or a well-explored behavioral policy for drawing transitions, therefore effectively bypassing the exploration issue.

Work on provably sample-efficient RL methods for Markov games without a sampling oracle or a well-explored policy is quite scarce. Before this paper, the only comparable work we are aware of is in Wei et al. [89], which proposes a model-based algorithm that extends the UCRL2 algorithm (Jaksch et al. [38]) for tabular MDPs to the game setting. Similarly to their work, we also consider both the online and offline settings and provide guarantees in terms of duality gap and regret. On the other hand, they only consider the tabular setting, a special case of our linear model. Their model-based algorithm explicitly estimates the Markov transition kernel and relies on the sophisticated technique of extended value iteration, which requires augmenting the state/action spaces. In comparison, our algorithm is model-free in the sense that it directly estimates the value functions. The computational cost of our algorithm only depends on the dimension d of the feature but not the cardinality of the state space.

Our work builds on a line of research on provably efficient methods for MDPs without additional assumptions on the sampling model. Most of the existing work focus on the tabular setting (see, e.g., Agrawal and Jia [4], Azar et al. [7], Dann et al. [20], Dong et al. [27] Jaksch et al. [38], Jin et al. [43], [45], Osband et al. [63], Rosenberg and Mansour [72], Russo [73], Simchowitz and Jamieson [80], Strehl et al. [82], Zanette and Brunskill [96] and the references therein). Under the function approximation setting, sample-efficient algorithms have been proposed using linear function approximators (Abbasi-Yadkori et al. [2], [3], Du et al. [29], Wang et al. [87], Yang and Wang [95]), as well as nonlinear ones (Dann et al. [21], Dong et al. [26], Du et al. [29], [30], Jiang et al. [41], Wen and Van Roy [91]). Among this line of work, our paper is most related to Cai et al. [17], Jin et al. [44], and Zanette

Table 1. Summary of recent work on RL algorithms for Markov games under different settings.

Algorithm	Objective	Explore	Independence	Observation	Structure
Minimax Q-learning (Sidford et al. [77])	NE	No	No	Yes	Tabular
Q-learning (Jia et al. [40])	NE	No	No	Yes	Linear
Model-based (Zhang et al. [99])	NE	No	No	Yes	Tabular
UCSG (Wei et al. [89])	NE	Yes	No	Yes	Tabular
	Value	Yes	Yes	Yes	Tabular
VI-ULCB (Bai and Jin [8])	NE	Yes	No	Yes	Tabular
Optimistic Nash Q/V-learning (Bai et al. [9])	NE	Yes	No	Yes	Tabular
Nash-VI (Liu et al. [55])	NE	Yes	No	Yes	Tabular
Policy gradient method (Daskalakis et al. [22])	NE	No	Yes	No	Tabular
OGDA (Wei et al. [90])	NE	No	No	Yes	Tabular
V-OL (Tian et al. [84])	Value	Yes	Yes	No	Tabular
GOLF_with_EXPLOITER (Jin et al. [42])	NE	Yes	No	Yes	General
	Value	Yes	Yes	Yes	General
Nash-UCRL-VTR (Chen et a. [19])	NE	Yes	No	Yes	Linear
This work	NE	Yes	No	Yes	Linear
	Value	Yes	Yes	Yes	Linear

Notes. The column "Objective" means whether the goal is to learn the NE policy (the offline setting) or compete with the NE value (the online setting). "Explore" means the algorithm needs to explore, without assuming a sampling oracle or a well-explored behavioral policy. "Independence" means the algorithm does not coordinate the learning procedures of different players. "Observation" means each player can observe the opponent's actions and/or rewards. "Structure" means the structural assumption imposed on the Markov game, including the tabular, linear and general function approximation settings. Nash-UCRL-VTR, Nash-UCRL with Value-Targeted Regression; Nash-VI, Optimistic Nash Value Iteration; NE, Nash equilibrium; OGDA, Optimistic Gradient Descent/Ascent; UCSG, Upper Confidence Stochastic Game algorithm; V-OL, Optimistic Nash V-learning for Online Learning; VI-ULCB, Value Iteration with Upper/Lower Confidence Bound.

et al. [97], which consider linear MDP models and propose optimistic and randomized variants of least-squares value iteration (LSVI) (Bradtke and Barto [13], Osband et al. [63]) as well as optimistic variants of proximal policy optimization (Schulman et al. [74]). Our linear Markov game model generalizes the MDP model considered in these papers, and our OMNI-VI algorithm can be viewed as a generalization of the optimistic LSVI method proposed in Jin et al. [44]. As mentioned earlier, the game structures in our problem pose fundamental challenges that are absent in MDPs, and thus their algorithms cannot be trivially extended to our game setting.

After the conference version of this paper was published (Xie et al. [92]), there have been several concurrent and follow-up papers on learning for Markov games. One line of work focuses on the offline setting with a finite state space (i.e., the tabular case). The work in Bai and Jin [8] develops a value iteration-based algorithm VI-ULCB with an $\widetilde{O}(H^5S^2AB/\epsilon^2)$ sample complexity for learning the ϵ-NE. However, this algorithm is not computationally efficient. Using the CCE idea developed in our paper, the follow-up work (Bai et al. [9]) designs two polynomial time algorithms with sample complexity $\widetilde{O}(H^6SAB/\epsilon^2)$ and $\widetilde{O}(H^7S(A+B)/\epsilon^2)$, respectively, where the latter matches the information-theoretic lower bound with respect to S, A, and B. The dependence on the horizon H has been improved in the recent work (Liu et al. [55]), which develops model-based algorithms with an $\widetilde{O}(H^4SAB/\epsilon^2)$ sample complexity. Another recent work (Daskalakis et al. [22]) considers the setting where each player independently selects a policy without observing the opponent's actions or rewards. Their policies need to use learning rates with two timescales to ensure convergence to the NE. Beyond the tabular setting, the work in Chen et al. [19] considers Markov games with a linear mixture structure, which is related to but different from the linear game setting in this paper. Building on the CCE idea developed in our paper, they propose an algorithm that achieves an $\widetilde{O}(d\sqrt{H^2T})$ regret, where d is the dimension of linear mixture structure. The work in Jin et al. [42] introduces a new complexity measure for Markov games called multiagent Bellman eluder dimension, which is adapted from its single-agent version. They design a self-play algorithm that can learn a Markov game with low Bellman eluder dimension using polynomially many samples. For the online setting where the actions of the opponents are unobservable, the work in Tian et al. [84] proposes an algorithm that achieves a sublinear regret bound (under the same regret definition as ours) that is independent of the size of the opponents' action spaces.

Finally, we remark that there is a line of work on robust MDPs (Lim et al. [51], Xu and Mannor [93]), where an adversary chooses the transition kernel from an uncertainty set. This problem is closely related to our online setting, where the adversary chooses an action that determines the transition kernel. One technical difference is that in their setting, the uncertainty set is known yet the choice of the adversary is not directly observable, whereas in our case the adversary's action is observed but its influence on the transition and value functions needs to be estimated from data. The algorithms are also different: they take an model-based approach that finds the worst-case transition kernel from the uncertainty set, whereas our algorithm computes empirical estimates of the worst-case value functions using data. Also, their results apply only to the tabular setting of MDPs.

2. Background and Preliminaries

In this section, we formally describe the setup for episodic two-player zero-sum Markov games with simultaneous moves. We then describe the setting for turn-based games, which can be viewed as a special case of simultaneous-moves games.

2.1. Notation

For two quantities x and y that potentially depend on problem parameter (d, H, $|\mathcal{A}|$, T, etc.), if $x \geq Cy$ holds for a universal absolute constant $C > 0$, we write $x \gtrsim y$, $x = \Omega(y)$ and $y = O(x)$. For each real number u, define the clipping operation $\Pi_H(u) = \max\{\min\{u, H\}, -H\}$. We use $\|\cdot\|$ to denote the vector ℓ_2 norm and $\|\cdot\|_F$ the matrix Frobenius norm. Given a positive semidefinite matrix A, define the weighted ℓ_2 norm $\|v\|_A := \sqrt{v^\top A v}$ for the vector v.

We sometimes need to consider a general-sum matrix (or normal form) game with payoff matrices $u_i \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ for each player $i \in \{1,2\}$. Here, if P1 and P2 take actions a and b, respectively, then player i receives a payoff $u_i(a,b)$. Denoting the two players as P1 and P2, we use the convention that P1 tries to maximize the payoff and P2 tries to minimize. A joint distribution $\sigma \in \Delta(\mathcal{A}\times\mathcal{A})$ of both players' actions is called a coarse correlated equilibrium (Aumann [6], Moulin and Vial [60]) of the game if it satisfies

$$\mathbb{E}_{(a,b)\sim\sigma}[u_1(x,a,b)] \geq \mathbb{E}_{b\sim\mathcal{P}_2\sigma}[u_1(x,a',b)], \quad \forall a' \in \mathcal{A}, \tag{1a}$$

$$\mathbb{E}_{(a,b)\sim\sigma}[u_2(x,a,b)] \leq \mathbb{E}_{a\sim\mathcal{P}_1\sigma}[u_2(x,a,b')], \quad \forall b' \in \mathcal{A}, \tag{1b}$$

where for $i \in \{1,2\}$, $\mathcal{P}_i\sigma \in \Delta(\mathcal{A})$ denotes the ith marginal of σ. In words, in a CCE the players choose their actions in a potentially correlated way such that no unilateral (unconditional) deviation from σ is beneficial.[1] Note that a CCE $\sigma = \sigma_1 \times \sigma_2$ in product form is an NE.

2.2. Simultaneous-Move Markov Games

A two-player, zero-sum, simultaneous-moves, episodic Markov game is defined by the tuple

$$(\mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, r, \mathbb{P}, H),$$

where $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is a finite set of actions that player $i \in \{1,2\}$ can take, r is reward function, $\mathbb{P}$ is transition kernel, and H is the number of steps in each episode. At each step $h \in [H]$, upon observing the state $x \in \mathcal{S}$, P1 and P2 take actions $a \in \mathcal{A}_1$ and $b \in \mathcal{A}_2$, respectively, and then both receive the reward $r_h(x,a,b)$. The system then transitions to a new state $x' \sim \mathbb{P}_h(\cdot \mid x,a,b)$ according to the transition kernel. Throughout this paper, we assume for simplicity that $\mathcal{A}_1 = \mathcal{A}_2 =: \mathcal{A}$ and that the rewards $r_h(x,a,b)$ are deterministic functions of the tuple (x, a, b) taking value in $[-1,1]$; generalization to the setting with $\mathcal{A}_1 \neq \mathcal{A}_2$ and stochastic rewards is straightforward.

Denote by $\Delta \equiv \Delta(\mathcal{A})$ the probability simplex over the action space $\mathcal{A}$. A stochastic policy of P1 is a length-H sequence of functions $\pi := (\pi_h : \mathcal{S} \to \Delta)_{h\in[H]}$. At each step $h \in [H]$ and state $x \in \mathcal{S}$, P1 takes an action sampled from the distribution $\pi_h(x)$ over $\mathcal{A}$. Similarly, a stochastic policy of P2 is given by the sequence $\nu := (\nu_h : \mathcal{S} \to \Delta)_{h\in[H]}$.

2.2.1. Value Functions.

For a fixed pair of policies (π, ν) for both players, the value and Q (a.k.a. action-value) functions for the previous game can be defined in a manner analogous to the episodic Markov decision process (MDP) setting:

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

where the expectation is over $a_t \sim \pi_t(x_t)$, $b_t \sim \nu_t(x_t)$, and $x_{t+1} \sim \mathbb{P}_t(\cdot \mid x_t,a_t,b_t)$. It is convenient to set $V_{H+1}^{\pi,\nu}(x) \equiv Q_{H+1}^{\pi,\nu}(x) \equiv 0$ for the terminal reward. Under the boundedness assumption on the reward, it is easy see that all value functions are bounded:

$$|V_h^{\pi,\nu}(x)| \leq H \quad \text{and} \quad |Q_h^{\pi,\nu}(x,\, a,\, b)| \leq H,\ \forall x,a,b,h,\pi,\nu.$$

In the zero-sum setting, for a given initial state x_1, P1 aims to maximize $V_1^{\pi,\nu}(x_1)$, whereas P2 aims to minimize it. Accordingly, we introduce the value and Q functions when P1 plays the best response to a fixed policy ν of P2:

$$V_h^{*,\nu}(x) = \max_\pi V_h^{\pi,\nu}(x) \quad \text{and} \quad Q_h^{*,\nu}(x,a,b) = \max_\pi Q_h^{\pi,\nu}(x,a,b).$$

Analogously, when P2 plays the best response to P1's policy π, we define

$$V_h^{\pi,*}(x) = \min_{\nu} V_h^{\pi,\nu}(x) \quad \text{and} \quad Q_h^{\pi,*}(x,a,b) = \min_{\nu} Q_h^{\pi,\nu}(x,a,b).$$

A Nash equilibrium (NE) of the game is a pair of stochastic policies (π^*, ν^*) that are the best response to each other; that is,

$$V_1^{\pi^*,\nu^*}(x_1) = V_1^{*,\nu^*}(x_1) = V_1^{\pi^*,*}(x_1), \qquad x_1 \in \mathcal{S}. \tag{2}$$

We assume that the game satisfies appropriate regularity conditions so that an NE exists and their value is unique.[2] Correspondingly, let $V_h^*(x) := V_h^{\pi^*,\nu^*}(x)$ and $Q_h^*(x,a,b) := Q_h^{\pi^*,\nu^*}(x,a,b)$ denote the values of the NE at step h.

Define the following shorthand for conditional expectation for the step h transition:

$$[\mathbb{P}_h V](x,a,b) := \mathbb{E}_{x' \sim \mathbb{P}_h(\cdot|x,a,b)}[V(x')] = \int V(x') \mathrm{d}\mathbb{P}_h(x' \mid x,a,b).$$

Although not explicitly needed in our analysis, we note that the value/Q functions for the NE satisfy the following Bellman equations:

$$Q_h^*(x,a,b) = r_h(x,a,b) + (\mathbb{P}_h V_{h+1}^*)(x,a,b), \tag{3a}$$

$$\text{and} \quad V_h^*(x) = \max_{A \in \Delta} \min_{B \in \Delta} \mathbb{E}_{a \sim A, b \sim B} Q_h^*(x,a,b) = \min_{B \in \Delta} \max_{A \in \Delta} \mathbb{E}_{a \sim A, b \sim B} Q_h^*(x,a,b). \tag{3b}$$

The fixed policy and best response value/Q functions, $V_h^{\pi,\nu}, V_h^{\pi,*}, V_h^{*,\nu}, Q_h^{\pi,\nu}, Q_h^{\pi,*}$, and $Q_h^{*,\nu}$, satisfy a similar set of Bellman equations; we omit the details.

The following weak duality result, which follows immediately from definition, relates the previous value and Q functions.

Proposition 1 (Weak Duality). *For each policy pair (π,ν) and each $h \in [H]$, $(x,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{A}$, we have*

$$Q_h^{\pi,*}(x,a,b) \le Q_h^*(x,a,b) \le Q_h^{*,\nu}(x,a,b), \qquad V_h^{\pi,*}(x) \le V_h^*(x) \le V_h^{*,\nu}(x),$$
$$Q_h^{\pi,*}(x,a,b) \le Q_h^{\pi,\nu}(x,a,b) \le Q_h^{*,\nu}(x,a,b), \qquad V_h^{\pi,*}(x) \le V_h^{\pi,\nu}(x) \le V_h^{*,\nu}(x).$$

2.2.2. Linear Structures.

We assume that both the reward function and transition kernel have a linear structure.

Assumption 1 (Linearity and Boundedness). *For each $(x,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{A}$ and $h \in [H]$, we have*

$$r_h(x,a,b) = \phi(x,a,b)^\top \theta_h \qquad \text{and} \qquad \mathbb{P}_h(\cdot \mid x,a,b) = \phi(x,a,b)^\top \mu_h(\cdot),$$

where $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}^d$ is a known feature map, $\theta_h \in \mathbb{R}^d$ is an unknown vector, and $\mu_h = (\mu_h^{(i)})_{i \in [d]}$ is a vector of d unknown (signed) measures on $\mathcal{S}$. We assume that $\|\phi(\cdot,\cdot,\cdot)\| \le 1$, $\|\theta_h\| \le \sqrt{d}$, and $\|\mu_h(\mathcal{S})\| \le \sqrt{d}$ for all $h \in [H]$, where $\|\cdot\|$ is the vector ℓ_2 norm.

Note that boundedness of the linear weights θ_h and μ_h allows for certain covering and concentration arguments in the analysis; also see Jin et al. [44, section 2.1] for a discussion on the specific choice of normalization in the previous assumption. It is also easy to see that the linearity assumption above implies that the Q functions are linear.

Lemma 1 (Linearity of Value Function). *Under Assumption 1, for any policy pair (π,ν) and any $h \in [H]$, there exists a vector $w_h^{\pi,\nu} \in \mathbb{R}^d$ such that*

$$Q_h^{\pi,\nu}(x,a,b) = \langle \phi(x,a,b), w_h^{\pi,\nu} \rangle, \ \forall (x,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{A}.$$

Proof. By Bellman equation and linearity of r_h and $\mathbb{P}_h$, we have

$$Q_h^{\pi,\nu}(x,a,b) = r_h(x,a,b) + \mathbb{P}_h V_{h+1}^{\pi,\nu}(x,a,b) = \phi(x,a,b)^\top \theta_h + \int V_{h+1}^{\pi,\nu}(x') \phi(x,a,b)^\top \mathrm{d}\mu_h(x').$$

Letting $w_h^{\pi,\nu} := \theta_h + \int V_{h+1}^{\pi,\nu}(x') \mathrm{d}\mu_h(x')$ proves the lemma. □

Remark 1. Since $Q_h^{\pi,*}(x,a,b) = Q_h^{\pi,\mathsf{br}(\pi)}(x,a,b)$, where $\mathsf{br}(\pi) \in \arg\min_\nu Q_h^{\pi,\nu}(x,a,b)$ is the best response policy to π, it follows immediately from Lemma 1 that $Q_h^{\pi,*}(x,a,b) = \langle \phi(x,a,b), w_h^{\pi,*} \rangle$ for some $w_h^{\pi,*} \in \mathbb{R}^d$. Similarly, we have $Q_h^{*,\nu}(x,a,b) = \langle \phi(x,a,b), w_h^{*,\nu} \rangle$ for some $w_h^{*,\nu} \in \mathbb{R}^d$.

The previous linear setting covers the tabular setting as a special case, where $d = |\mathcal{S}| \cdot |\mathcal{A}|^2$ and $\phi(x, a, b)$ is the indicator vector for the tuple (x, a, b). It is also clear that MDPs are a special case of Markov games when P2 plays a fixed and known policy. In particular, our setting covers both tabular MDPs as well as the linear MDP setting considered in the work by Jin et al. [44]. Finally, as we elaborate in Section 2.3, turn-based Markov games can also be viewed as a special case of our setting.

Remark 2. Linearity of the reward and transition kernel is a strictly stronger assumption than linearity of the value functions. Our analysis makes crucial use of this stronger assumption, which ensures that the linearity of value functions is preserved under the Bellman equation. In fact, it is likely that this assumption is essential for developing efficient algorithms, in view of recent hardness result (Du et al. [28]) that only assumes linearity of value functions of MDPs (a special case of Markov games).

2.3. Turn-Based Markov Games

In turn-based games, at each state only one player takes an action. Without loss of generality, we may partition the state space as $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, where $\mathcal{S}_i$ are the states at which it is player i's turn to play.[3] For each state $x \in \mathcal{S}$, let $I(x) \in \{1, 2\}$ indicate the current player to play, so that $x \in \mathcal{S}_{I(x)}$. At each step $h \in [H]$, player $I(x)$ observes the current state x and takes an action a; then the two players receive the reward $r_h(x, a)$, and the system transitions to a new state $x' \sim \mathbb{P}_h(\cdot \mid x, a)$.

The value/Q functions $V_h^{\pi,\nu}(x), Q_h^{\pi,\nu}(x, a)$, and so on, as well as the corresponding NE of the game, can be defined in a completely analogous way as in the simultaneous-move setting. Similarly to Assumption 1, we also assume that the game has a linear structure.

Assumption 2 (Linearity and Boundedness, Turn-Based). *For each $(x, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$, we have*

$$r_h(x, a) = \phi(x, a)^\top \theta_h \qquad \text{and} \qquad \mathbb{P}_h(\cdot \mid x, a) = \phi(x, a)^\top \mu_h(\cdot),$$

where $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is a known feature map, $\theta_h \in \mathbb{R}^d$ is an unknown vector, and $\{\mu_h^{(i)}\}_{i \in [d]}$ are d unknown (signed) measures on $\mathcal{S}$. We assume that $\|\phi(\cdot, \cdot)\| \le 1$, $\|\mu_h(\mathcal{S})\| \le \sqrt{d}$, and $\|\theta_h\| \le \sqrt{d}$ for all $h \in [H]$.

One may view a turn-based game as a special case of a simultaneous-move game, where at each state only one of the players is active and the other player's action has no influence on the reward or the transition. Formally, for each $x \in S_1$, the values of $r_h(x, a, b)$, $\mathbb{P}_h(\cdot \mid x, a, b)$, and $\phi(x, a, b)$ are independent of b; for each $x \in \mathcal{S}_2$, they are independent of a.

3. Main Results for the Offline Setting

In this section, we consider the offline setting, where a central controller controls both players. The goal of the controller is to learn a Nash equilibrium (π^*, ν^*) of the game in episodic setting. In what follows, we formally define the problem setup and objectives, and then present our algorithm and provide theoretic guarantees for its performance.

3.1. Setup and Performance Metrics

In the episodic setting, the Markov game is played for K episodes, each of which consists of H timesteps. At the beginning of the kth episode, an arbitrary initial state x_1^k is chosen. Then the players P1 and P2 play according to the policies $\pi^k = (\pi_h^k)_{h \in [H]}$ and $\nu^k = (\nu_h^k)_{h \in [H]}$, respectively, which may adapt to observations from past episodes. The game terminates after H timesteps and restarts for the $(k+1)$ th episode. Note that expected reward for P1 and P2 in the kth episode is $V_1^{\pi^k, \nu^k}(x_1^k)$.

3.1.1. Duality Gap Guarantees.

Recall the weak duality property in Proposition 1, which states the value of the NE, $V_1^*(x_1)$, is sandwiched between $V_1^{\pi^k,*}(x_1)$ and $V_1^{*,\nu^k}(x_1)$. Therefore, it is natural to use the duality gap $V_1^{*,\nu^k}(x_1) - V_1^{\pi^k,*}(x_1)$ to measure how well the policy (π^k, ν^k) in the kth episode approximates the NE. Accordingly, we aim to bound the following total duality gap:

$$\mathrm{Gap}(K) := \sum_{k=1}^{K} \left[V_1^{*,\nu^k}(x_1^k) - V_1^{\pi^k,*}(x_1^k) \right]. \tag{4}$$

Another way to interpret the previous objective is as follows. Define the exploitability (Davis et al. [24]) of P1 and P2, respectively, as

$$\mathrm{Exploit}_1(\pi^k, \nu^k) := V_1^{\pi^k,\nu^k}(x_1^k) - V_1^{\pi^k,*}(x_1^k) \quad \text{and} \quad \mathrm{Exploit}_2(\pi^k, \nu^k) := V_1^{*,\nu^k}(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k),$$

both of which are nonnegative by Proposition 1. Here, $\text{Exploit}_i(\pi^k, \nu^k)$ measures the potential loss of player $i \in \{1, 2\}$ in the kth episode if the other player unilaterally switched to the best response policy. The total duality gap can then be rewritten as

$$\text{Gap}(K) = \sum_{k=1}^{K} \Big[\text{Exploit}_1(\pi^k, \nu^k) + \text{Exploit}_2(\pi^k, \nu^k)\Big],$$

which is the sum of the exploitability of both players accumulated over K episodes. Also note that in special cases of MDPs, $\text{Gap}(K)$ reduces to the usual notion of total regret.

3.1.2. Sample Complexity and PAC Guarantees.

Another performance metric is the sample complexity for finding an approximate NE. In particular, suppose that for all episodes the initial states x_1 are sampled from the same fixed distribution. We are interested in the number of episodes K (or equivalently the number of samples $T = KH$) needed to find a policy pair (π, ν) satisfying

$$V_1^{*,\nu}(x_1) - V_1^{\pi,*}(x_1) \le \epsilon \qquad \text{with probability at least } 1 - \delta.$$

In light of Proposition 1, this inequality implies that (π, ν) is an ϵ-approximate NE in the sense that

$$V_1^{*,\nu}(x_1) - \epsilon \le V_1^{\pi,\nu}(x_1) \le V_1^{\pi,*}(x_1) + \epsilon;$$

that is, (π, ν) satisfies the definition (2) of NE up to an ϵ error. As we discuss in detail after presenting our main theorem, a bound on the total duality gap implies a bound on the sample complexity. Such a bound in turn implies a PAC-type guarantee in the sense of Kakade [46], which stipulates that an ϵ-approximate NE is played in all but a small number of timesteps.

3.2. Algorithm

We now present our algorithm, optimistic minimax value iteration (OMNI-VI) with least-squares estimation, which is given as Algorithm 1.

Algorithm 1. (Optimistic Minimax Value Iteration (Simultaneous Move, Offline))

1: **Input:** bonus parameter $\beta > 0$.
2: **for** episode $k = 1, 2, \ldots, K$ **do**
3:    Receive initial state x_1^k
4:    **for** step $h = H, H-1, \ldots, 2, 1$ **do**   ▷ update policy
5:       $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\phi(x_h^\tau, a_h^\tau, b_h^\tau)^\top + I.$
6:       $\overline{w}_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\Big[r_h(x_h^\tau, a_h^\tau, b_h^\tau) + \overline{V}_{h+1}^k(x_{h+1}^\tau)\Big].$
7:       $\underline{w}_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\Big[r_h(x_h^\tau, a_h^\tau, b_h^\tau) + \underline{V}_{h+1}^k(x_{h+1}^\tau)\Big].$
8:       $\overline{Q}_h^k(\cdot, \cdot, \cdot) \leftarrow \Pi_H\Big\{(\overline{w}_h^k)^\top \phi(\cdot, \cdot, \cdot) + \beta\sqrt{\phi(\cdot, \cdot, \cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot, \cdot, \cdot)}\Big\}.$
9:       $\underline{Q}_h^k(\cdot, \cdot, \cdot) \leftarrow \Pi_H\Big\{(\underline{w}_h^k)^\top \phi(\cdot, \cdot, \cdot) - \beta\sqrt{\phi(\cdot, \cdot, \cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot, \cdot, \cdot)}\Big\}.$
10:      For each x, let $\sigma_h^k(x) \leftarrow \texttt{FIND\_CCE}\Big(\overline{Q}_h^k, \underline{Q}_h^k, x\Big).$
11:      $\overline{V}_h^k(x) \leftarrow \mathbb{E}_{(a,b)\sim\sigma_h^k(x)} \overline{Q}_h^k(x, a, b)$ for each x.
12:      $\underline{V}_h^k(x) \leftarrow \mathbb{E}_{(a,b)\sim\sigma_h^k(x)} \underline{Q}_h^k(x, a, b)$ for each x.
13:    **end for**
14:    **for** step $h = 1, 2, \ldots, H$ **do**   ▷ execute policy
15:       Sample $(a_h^k, b_h^k) \sim \sigma_h^k(x_h^k)$.
16:       P1 takes action a_h^k; P2 takes action b_h^k.
17:       Observe next state x_{h+1}^k.
18:    **end for**
19: **end for**

In each episode k, the algorithm first constructs the policies for both players (lines 4–13), and then executes the policy to play the game (lines 14–18). The construction of the policy is done through backward induction with respect to the timestep h. In each timestep, we first compute upper/lower estimates $\overline{w}_h, \underline{w}_h \in \mathbb{R}^d$ of the linear coefficients of the Q function. This is done by approximately solving the Bellman Equation (3) using (regularized) least-squares estimation, for which we use empirical data from the previous $k-1$ episodes to estimate the unknown transition kernel $\mathbb{P}_h$ (lines 5–7). Then, to encourage exploration, we construct UCB/LCB for the Q function by adding/subtracting an appropriate bonus term (lines 8–9). The bonus takes the form $\beta\sqrt{\phi^\top(\Lambda_h^k)^{-1}\phi}$, where Λ_h^k is the regularized Gram matrix defined in line 5 of the algorithm. This form of bonus is common in the literature of linear bandits (Lattimore and Szepesvári [49]). The next and crucial step is to convert the UCB/LCB $(\overline{Q}_h, \underline{Q}_h)$ for the Q function into UCB/LCB $(\overline{V}_h, \underline{V}_h)$ for the value function (lines 10–12). This step turns out to be delicate; we elaborate next.

Note that $\overline{V}_h(x)$ and $\underline{V}_h(x)$ should correspond to the actions (a', b') that would be actually played at state x, that is, $\overline{V}_h(x) = \overline{Q}_h(x, a', b')$ (in expectation w.r.t. randomness of the stochastic policy; similarly for $\underline{V}_h(x)$), so that these upper/lower bounds can be tightened up using empirical observations from these actions. To construct these bounds, one may be tempted to let each player independently compute the maximin or minimax values and actions. That is, one may let P1 play the action $a' = \arg\max_a \min_b \overline{Q}_h^k(x, a, b)$ and P2 play $b' = \arg\min_b \max_a \underline{Q}_h^k(x, a, b)$, and then set $\overline{V}_h^k(x) \leftarrow \overline{Q}_h^k(x, a', b')$ and $\underline{V}_h^k(x) \leftarrow \underline{Q}_h^k(x, a', b')$. Unfortunately, such a $\overline{V}_h^k(x)$ is not a valid upper bound for the true value, because $\overline{Q}_h^k \neq \underline{Q}_h^k$ in general and hence $\overline{Q}_h^k(x, a', b') \neq \max_a \min_b \overline{Q}_h^k(x, a, b)$.

Instead, we must coordinate both players for their choices of actions, which is done by solving the general-sum matrix game with payoff matrices $\overline{Q}_h^k(x, \cdot, \cdot)$ and $\underline{Q}_h^k(x, \cdot, \cdot)$. Finding the NE for general-sum games gives valid UCB/LCB, but doing so is computationally intractable (Chen et al. [18], Daskalakis et al. [23]). Fortunately, computing an (approximate) CCE of the matrix game turns out to be sufficient as well. For technical reasons elaborated in the next subsection, the subroutine `FIND_CCE` for finding the CCE is implemented in a specific way as follows. Let $\mathcal{Q}$ be the class of functions $Q: \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ with the parametric form

$$Q(x, a, b) = \Pi_H\left\{\langle w, \phi(x, a, b)\rangle + \rho\beta\sqrt{\phi(x,a,b)^\top A \phi(x, a, b)}\right\}, \tag{5}$$

where the parameters $(w, A, \rho) \in \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \{\pm 1\}$ satisfy $\|w\| \le 2H\sqrt{dk}$ and $\|A\|_F \le \beta^2\sqrt{d}$. Let $\mathcal{Q}_\epsilon$ be a fixed ϵ-covering of $\mathcal{Q}$ with respect to the ℓ_∞ norm $\|Q - Q'\|_\infty := \sup_{x,a,b} |Q(x, a, b) - Q'(x, a, b)|$. With these notations, we present the subroutine `FIND_CCE` in Algorithm 2. The algorithm effectively rounds the game $(\overline{Q}_h^k(x, \cdot, \cdot), \underline{Q}_h^k(x, \cdot, \cdot))$ of interest into a nearby game in the finite ϵ-cover $\mathcal{Q}_\epsilon \times \mathcal{Q}_\epsilon$, and then uses the CCE of the latter game as an surrogate of the CCE of the original game. Furthermore, we remark that this rounding step can be implemented efficiently without explicitly computing/maintaining the (exponentially large) ϵ-net; see Appendix D for details.

Algorithm 2. `FIND_CCE`

1: **Input:** $\overline{Q}_h^k$, $\underline{Q}_h^k$, x and discretization parameter $\epsilon > 0$.

2: Pick a pair $(\widetilde{Q}, \underset{\sim}{Q})$ in $\mathcal{Q}_\epsilon \times \mathcal{Q}_\epsilon$ satisfying $\left\|\widetilde{Q} - \overline{Q}_h^k\right\|_\infty \le \epsilon$ and $\left\|\underset{\sim}{Q} - \underline{Q}_h^k\right\|_\infty \le \epsilon$.

3: For the input x, let $\widetilde{\sigma}(x)$ be the CCE (cf. Equation (1)) of the matrix game with payoff matrices

$$\widetilde{Q}(x, \cdot, \cdot) \text{ for P1 and } \underset{\sim}{Q}(x, \cdot, \cdot) \text{ for P2}.$$

4: **Output:** $\widetilde{\sigma}(x)$.

3.2.1. Technical Considerations for `FIND_CCE`.

The use of rounding and an ϵ-cover in `FIND_CCE` is motivated by the following considerations. First, note that the least-squares step of Algorithm 1 (lines 5–7) uses data from all previous episodes. This introduces complicated probabilistic dependency between the estimation target $\overline{V}_{h+1}^k$ and the linear features $\phi(x_h^\tau, a_h^\tau, b_h^\tau), \tau \in [k-1]$, as they both depend on past data. Such dependency is not present in usual least-squares estimation in supervised learning. To overcome this issue, a standard approach is to use a covering argument to establish uniform concentration bounds valid for all value functions $\overline{V}_{h+1}^k$.[4]

Whereas it is straightforward to construct a cover for the Q functions (as we have done in `FIND_CCE`), doing so for the value functions is challenging due to instability of the equilibria of general-sum games. In particular, recall that the value function is defined by the CCE value of a general-sum game with two payoff matrices given by the Q functions. The CCE value, however, is not a Lipschitz function of the payoff matrices, hence a cover for the former does not follow from a cover for the latter. Indeed, suppose that a game has payoff matrices $(\overline{Q}_h^k, \underline{Q}_h^k)$ that are ϵ-close to another game $(\widetilde{Q}, \underset{\sim}{Q})$ from the cover $\mathcal{Q}_\epsilon \times \mathcal{Q}_\epsilon$. Lemma E.1 in Appendix E shows the following:

i. The CCE values of the previous two games may be bounded away from each other.

Interestingly, general-sum matrix games satisfy another property, proved in Lemma 4, that is seemingly contradictory to property (i):

ii. The CCE policy of the game $(\widetilde{Q}, \underset{\sim}{Q})$ from the cover is a 2ϵ-approximate CCE policy for the original game $(\overline{Q}_h^k, \underline{Q}_h^k)$, and vice versa.

Here, a 2ϵ-approximate CCE policy is one that satisfies the definition (1) of CCE with an additive 2ϵ error on the right-hand side (RHS). The proof of Lemma E.1 gives an example in which properties (i) and (ii) hold simultaneously.

Due to property (i), it is unclear how to run a covering argument only in the analysis, because in this case the algorithm would use the CCE value of the original game $(\overline{Q}_h^k, \underline{Q}_h^k)$ and this value cannot be controlled. However, thanks to property (ii), it suffices to use the ϵ-cover in the algorithm, because in this case the algorithm actually uses the CCE policy of the game $(\widetilde{Q}, \underset{\sim}{Q})$ from the finite ϵ-cover, and its value can be controlled by a union bound over the cover. The small price we pay is that the resulting UCB/LCB are valid up to a 2ϵ error, which eventually goes into the regret bound. This error can be made negligible relative to the main terms in the regret by choosing a small enough ϵ.

In summary, the previous algorithmic use of ϵ-cover appears crucial under our current framework. We leave as an intriguing open problem whether this algorithmic complication is in fact necessary or can be avoided by a more clever analysis. We also remark that the previous issue does not exist in the tabular setting, in which case the value functions $(\overline{V}_h^k, \underline{V}_h^k)$ are just a pair of finite-dimensional vectors and hence one can directly build an ϵ-cover for the relevant set of vectors.

3.3. Theoretical Guarantees

In each episode k, Algorithm 2 computes a joint (correlated) policy σ_h^k. As NE requires the policies to be in product form, we marginalize σ_h^k into a pair of independent policies $\pi_h^k(x) := \mathcal{P}_1\sigma_h^k(x)$ and $\nu_h^k(x) := \mathcal{P}_2\sigma_h^k(x)$ for each player. Our main theoretical result is the following bound on the total duality gap (4) of these policy pairs. Recall that $T = KH$ is the total number of timesteps.

Theorem 1 (Offline, Simultaneous Moves). *Under Assumption 1, there exists a constant $c > 0$ such that the following holds for each fixed $p \in (0,1)$. Set $\beta = cdH\sqrt{\iota}$ with $\iota := \log(2dT/p)$ in Algorithm 1, and set $\epsilon = \frac{1}{KH}$ in Algorithm 2. Then with probability at least $1-p$, Algorithm 1 satisfies the bounds*

$$V_1^{*,\nu^k}(x_1^k) - V_1^{\pi^k,*}(x_1^k) \le \overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k) + \frac{8}{K}, \quad \forall k \in [K], \tag{6}$$

$$\sum_{k=1}^{K}\left[\overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k)\right] \lesssim \sqrt{d^3H^3T\iota^2}; \tag{7}$$

consequently, we have

$$\mathrm{Gap}(K) \lesssim \sqrt{d^3H^3T\iota^2}. \tag{8}$$

The proof is given in Section 5. Next, we provide discussion and remarks on this theorem.

3.3.1. Optimality of the Bound.

The theorem provides an (instance-independent) bound scaling with $\sqrt{T}$. As the total duality gap reduces to the usual regret in the special case MDPs, our bound is optimal in T in view of known minimax lower bounds for MDPs (Lattimore and Szepesvári [49]). Also note that our bound is independent of the cardinality $|\mathcal{S}|$ and $|\mathcal{A}|$ of the state/action spaces, but rather depends only on dimension d of the feature space, thanks to the use of function approximation. To investigate the tightness of the dependence of our bound on d and H, we recall that our setting covers the standard tabular MDPs and linear bandits as special cases. A direct reduction from the known lower bounds on tabular MDPs gives a lower bound $\Omega(\sqrt{dH^2T})$ for the case of nonstationary transitions (Azar et al. [7], Jin et al. [43]). Our bound is off by a factor of $\sqrt{H}$, which may be improved by using a Bernstein-type bonus term (Azar et al. [7], Jin et al. [43]). Results from linear bandits give the lower bound $\Omega(d\sqrt{T})$. The additional $\sqrt{d}$ factor in our bound is due to a covering argument applied to the d-dimensional feature space for establishing uniform concentration bounds.

3.3.2. Computational Complexity. Our algorithm can be implemented efficiently, with a computational complexity polynomial in H, K, d, and $|\mathcal{A}|$. In particular, note that a CCE of a general-sum game can be found in polynomial time (Blum et al. [12], Papadimitriou and Roughgarden [65]).[5] Moreover, in Algorithm 1 we do not need to compute $\overline{Q}(x,\cdot,\cdot)$, $\overline{V}(x)$ and $\widetilde{\sigma}(x)$, and so on, for all $x \in \mathcal{S}$; rather, we only need to do so for the states $\{x_h^k\}$ actually encountered in the algorithm. Similarly, we do not need to explicitly maintain the (exponentially large) ϵ-net $\mathcal{Q}_\epsilon$ in `FIND_CCE` (Algorithm 2). It suffices if we can find an element in $\mathcal{Q}_\epsilon$ that is ϵ-close to a given function in $\mathcal{Q}$, which can be done efficiently on the fly. Indeed, each function in $\mathcal{Q}$ has a succinct representation using $(w, A) \in \mathbb{R}^d \times \mathbb{R}^{d\times d}$. We can (implicitly) maintain a covering of the space of (w, A), and find a nearby element from this covering when needed, which can be done in $O(d^2)$ time. See Appendix D and Lemma D.1 therein for details. Moreover, when finding an ϵ-close element in $\mathcal{Q}_\epsilon$ for any function in $\mathcal{Q}$, it turns out that the implementation only involves a set of computations based on the parameter (w, A), and we do not need to explicitly store the ϵ-cover $\mathcal{Q}_\epsilon$. Thus, in addition to being sample and computationally efficient, our Algorithm 1 also enjoys memory efficiency in the sense that the required memory size is polynomial in d, H, and K.

3.3.3. Sample Complexity and PAC Guarantees. It is a standard fact that a regret bound as in Theorem 1 can be converted into a bound on the sample complexity. For simplicity, we assume that the initial state x_1 is fixed.[6] After K episodes, we choose, among the K policy pairs $(\pi^k,\nu^k), k \in [K]$ computed by Algorithm 1, the pair (π^{k_0},ν^{k_0}) with the minimum gap between the UCB and LCB; that is,

$$k_0 = \arg\min_{k\in[K]} \left\{\overline{V}_1^k(x_1) - \underline{V}_1^k(x_1)\right\}.$$

Note that the UCB/LCB $\overline{V}_1^k(x_1)$ and $\underline{V}_1^k(x_1)$ are computed by the algorithm and hence their values are known. This policy pair (π^{k_0},ν^{k_0}) satisfies the bound

$$
\begin{aligned}
&V_1^{*,\nu^{k_0}}(x_1) - V_1^{\pi^{k_0},*}(x_1) \\
&\le \overline{V}_1^{k_0}(x_1) - \underline{V}_1^{k_0}(x_1) + \frac{8}{K} && \text{inequality (6)}\\
&\le \frac{1}{K}\sum_{k=1}^{K}\left[\overline{V}_1^k(x_1) - \underline{V}_1^k(x_1)\right] + \frac{8}{K} && \min \le \text{average}\\
&\lesssim \sqrt{\frac{d^3H^5\iota^2}{T}}. && \text{inequality (7) divided by } K = T/H
\end{aligned}
$$

Therefore, we can find an ϵ-approximate NE (meaning that the last RHS is bounded by ϵ) with a sample complexity of $T = O\left(\frac{d^3H^5\iota^2}{\epsilon^2}\right)$. By playing the policy pair (π^{k_0},ν^{k_0}) in all subsequent episodes, we obtain a PAC-type guarantee (Kakade [46]) in the sense that an ϵ-approximate NE is played in all but $O\left(\frac{d^3H^5\iota^2}{\epsilon^2}\right)$ timesteps.

3.4. Turn-Based Games

In this section, we consider turn-based Markov games, which is a special case of simultaneous-move Markov games. Algorithm 1 can be specialized to this setting. For completeness, we provide the resulting algorithm in Algorithm A.1 in Appendix A.1. Note that for turn-based games, the `FIND_CCE` routine is simplified to the subroutines `FIND_MAX` and `FIND_MIN` given in Algorithm A.2, because each state is controlled by a single player and hence finding a CCE reduces to computing a maximizer or minimizer.

As a corollary of Theorem 1, we have the following bound on the total duality gap, which is defined in the same way as in Equation (4).

Corollary 1 (Offline, Turn-based). *Under Assumption 2, there exists a constant $c > 0$ such that, for each fixed $p \in (0,1)$, by setting $\beta = cdH\sqrt{\iota}$ with $\iota := \log(2dT/p)$ in Algorithm A.1, then with probability at least $1-p$, Algorithm A.1 satisfies bound*

$$\mathrm{Gap}(K) \lesssim \sqrt{d^3H^3T\iota^2}.$$

We prove this corollary in Appendix A.1.1.

4. Main Results for the Online Setting

In this section, we consider the online setting, where we control P1 and play against an arbitrary (and potentially adversarial) P2. Our goal is to maximize the reward of P1. Next we describe the performance metrics, followed by our algorithms and theoretical guarantees.

4.1. Setup and Performance Metrics

We consider the episodic setting as described in Section 3.1. Let $\pi = (\pi^k)$ and $\nu = (\nu^k)$ be the policy sequences for P1 and P2, respectively, where ν is arbitrary. We do not know P2's choice of ν nor the Markov model of the game a priori, and would like learn a good policy π online so as to optimize the reward $\sum_k V_1^{\pi^k,\nu^k}$ received by P1 over K episodes. To this end, we are interested in bounding, for each ν, the total (expected) regret

$$\text{Regret}_\nu(K) := \sum_{k=1}^{K} \left[V_1^*(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k) \right], \tag{9}$$

where x_1^k is the (arbitrary) initial state in the kth episode. The regret (9) is a weak notion of regret that competes against the minimax value (i.e., NE value) of the game. It serves as a conservative but practical metric in the sense that it provides an achievable upper bound on the performance of P1 against any opponent policy. Specifically, if we can obtain a bound on $\text{Regret}_\nu(K)$ that scales sublinearly with K for all ν, then we are guaranteed that regardless of ν, the reward collected by P1 is no worse (in the long run) than its optimal worst-case reward, that is, the NE value V_1^*.

We note that a special case of the previous setting is when P2 is omniscient and always plays the best response to P1's policy, that is,

$$\nu^k = \text{br}(\pi^k) \in \arg\min_{\nu' \in \Delta} V_1^{\pi^k,\nu'}(x_1^k), \quad \forall k \in [K].$$

Note that in this case, we have $V_1^{\pi^k,\nu^k}(x_1^k) = V_1^{\pi^k,*}(x_1^k)$ by definition.

Remark 3. One may wish to consider a stronger notion of regret that compares the expected reward received by P1 over K episodes against the best fixed policy of P1 in hindsight:

$$\text{Regret}^{(S)}(K) := \sup_\mu \sum_{k=1} \left[V_1^{\mu,\nu^k} - V_1^{\pi^k,\nu^k} \right]. \tag{10}$$

However, recent work has established hardness results for optimizing this regret. In particular, it has been shown in Bai et al. [9] that under standard hardness assumptions, there is no polynomial time algorithm that attains a sublinear regret in the sense of (10) when playing against adversarial opponents in Markov games. The work by Tian et al. [84] further shows that it is statistically hard to compete against the best policy in hindsight, as the regret defined in (10) can be either linear in K or exponential in H. In light of these results, in this paper we focus on the weak regret in (9) as a practical notion of performance metric.

4.2. Algorithm

We adapt the optimistic minimax value iteration algorithm to the online setting, as given in Algorithm 3. This algorithm can be viewed as a one-sided version of Algorithm 1: we compute least-squares estimate for the linear coefficients and then construct UCBs for the value functions—we do not need to construct LCBs as P2 is not controlled by us. Constructing the UCBs is done by computing the NE of the zero-sum matrix game with the payoff matrix $Q_h^k(x,\cdot,\cdot)$ (line 8 of Algorithm 3).

Algorithm 3. (Optimistic Minimax Value Iteration (Simultaneous Move, Online))

1: **Input:** bonus parameter $\beta > 0$.
2: **for** episode $k = 1, 2, \ldots, K$ **do**
3: Receive initial state x_1^k.
4: **for** step $h = H, H-1, \ldots, 2, 1$ **do** ▷ update policy
5: $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\phi(x_h^\tau, a_h^\tau, b_h^\tau)^\top + I$.
6: $w_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\left[r_h(x_h^\tau, a_h^\tau, b_h^\tau) + V_{h+1}^k(x_{h+1}^\tau)\right]$.
7: $Q_h^k(\cdot,\cdot,\cdot) \leftarrow \Pi_H\left\{(w_h^k)^\top \phi(\cdot,\cdot,\cdot) + \beta\sqrt{\phi(\cdot,\cdot,\cdot)^\top (\Lambda_h^k)^{-1}\phi(\cdot,\cdot,\cdot)}\right\}$.
8: For each x, let $(\pi_h^k(x), B_0)$ be the NE of the matrix game with payoff matrix $Q_h^k(x,\cdot,\cdot)$.
9: $V_h^k(\cdot) \leftarrow \mathbb{E}_{a\sim\pi_h^k(\cdot), b\sim B_0}[Q_h^k(\cdot, a, b)]$.
10: **end for**
11: **for** step $h = 1, 2, \ldots, H$ **do** ▷ execute policy
12: P1 take action $a_h^k \sim \pi_h^k(x_h^k)$.
13: Let P2 play; denote its action by b_h^k.
14: Observe next state x_{h+1}^k.
15: **end for**
16: **end for**

Because of the one-sided nature of the online setting, some of the difficulties in the offline setting—pertaining to general-sum games and CCE—no longer exist here. In particular, Algorithm 3 no longer requires the `FIND_CCE` subroutine that makes use of an ϵ-cover. Technically, this is due to the fact that zero-sum matrix games are more well-behaved than general-sum games. In particular, the value of a zero-sum game is Lipschitz in the payoff matrix, hence uniform concentration can be established in a more straightforward manner (cf. the discussion in Section 3.2).

4.3. Regret Bound Guarantees

We establish the following bound on the total regret (9) achieved by Algorithm 3.

Theorem 2 (Online, Simultaneous Move). *Under Assumption 1, there exists a constant $c>0$ such that the following holds for each fixed $p\in(0,1)$ and any policy sequence ν for* P2. *Set $\beta = cdH\sqrt{\iota}$ with $\iota := \log(2dT/p)$. Then, with probability at least $1-p$, Algorithm 3 achieves the regret bound*

$$\mathrm{Regret}_\nu(K) \lesssim \sqrt{d^3H^3T\iota^2}.$$

The proof is given in Appendix C. Note that the regret bound holds for any policy ν of P2 and any initial states $\{x_1^k\}$. Moreover, the bound is sublinear in T—scaling with $\sqrt{T}$ in particular—and depends polynomially on d and H. As our regret reduces to the standard regret notion in the special cases of MDPs and linear bandits, the discussion in Section 3.2 on the optimality of bounds, also applies here.

We remark that the previous bound provides a uniform guarantee for P1's performance, regardless of the policy of the opponent P2. An interesting future direction is to achieve a more refined guarantee that exploits a weak opponent. In particular, such a guarantee would involve a stronger notion of regret in which, instead of competing with the Nash value $\sum_k V_1^*(x_1^k)$ as in the current definition (9), one competes against the value $\max_\pi \sum_{k=1}^K V_1^{\pi,\nu^k}(x_1^k)$ achieved by the best fixed policy in hindsight. We believe doing so would require modifying the algorithm, which is left to future work.

4.4. Turn-Based Games

The algorithm above can be specialized to online turn-based games. For completeness, we provide the resulting algorithm in Appendix A.1 as Algorithm A.3. Note that in the turn-based setting, we only need to solve a unilateral maximization or minimization problem, rather than solving zero-sum games as is needed in the simultaneous-move setting.

As an immediate corollary of Theorem 2, we have the following regret bound for turn-based games in the online setting.

Corollary 2 (Online, Turn-based). *Under Assumption 2, there exists a constant $c>0$ such that the following holds for each fixed $p\in(0,1)$ and any policy sequence ν for* P2. *Set $\beta = cdH\sqrt{\iota}$ with $\iota := \log(2dT/p)$ in Algorithm A.3. Then, with probability at least $1-p$, Algorithm A.3 achieves the following regret bound:*

$$\mathrm{Regret}_\nu(K) \lesssim \sqrt{d^3H^3T\iota^2}.$$

We prove this corollary in Appendix A.2.1.

5. Proof of Theorem 1

In this section, we prove Theorem 1 for the offline setting of simultaneous games. We shall make use of the technical lemmas given in Appendix B. For clarity of exposition, we denote by $\phi_h^k := \phi(x_h^k, a_h^k, b_h^k)$ the feature vector encountered in the hth step of the kth episode. Our proof consists of five steps:

i. **Uniform concentration:** We begin by showing that an empirical estimate of the transition kernel $\mathbb{P}_h$, when acting on the value functions maintained by the algorithm, concentrates around its expectation. See Section 5.1.

ii. **Least-squares estimation error:** Using the concentration result in (i), we derive high probability bounds on the errors of our least-squares estimates of the true Q functions $Q_h^{\pi,\nu}$, recursively in the timestep h. See Section 5.2.

iii. **UCB and LCB:** We next show that the UCBs and LCBs constructed in the algorithms are indeed valid bounds on the true value functions $V_h^{\pi,*}$ and $V_h^{*,\nu}$. See Section 5.3.

iv. **Recursive decomposition of duality gap:** We derive a recursive formula for the difference between the UCB and LCB in terms of the timestep h. This difference in turn bounds the duality gap of interest. See Section 5.4.

v. **Establishing final bound:** Bounding each term in the recursive decomposition in (iv) in terms of the least-squares estimation errors, we establish the desired bound on the total duality gap, thereby completing the proof of the theorem. See Section 5.5.

Next, we provide the details of each step.

5.1. Uniform Concentration

The quantity $\sum_{\tau\in[k-1]}\phi_h^\tau \overline{V}_{h+1}^k(x_{h+1}^\tau)$ can be viewed as an empirical estimate of the unknown population quantity $\sum_{\tau\in[k-1]}\phi_h^\tau(\mathbb{P}_h\overline{V}_{h+1}^k)(x_h^\tau, a_h^\tau, b_h^\tau)$. To control the least-squares estimation error, we need to show that the empirical estimate concentrates around its population counterpart. The main challenge in doing so is that $\overline{V}_{h+1}^k$ is constructed using data from previous episodes and hence depends on ϕ_h^τ for all $\tau \in [k-1]$. We overcome this issue by noting that $\overline{V}_{h+1}^k$ is computed using the CCE of a finite class of games with payoff matrices in the ϵ-net $\mathcal{Q}_\epsilon \times \mathcal{Q}_\epsilon$, as is done in `FIND_CCE`. Therefore, we can prove a concentration bound valid uniformly over this class of games and thereby establish the following concentration result. Here we recall that $\|v\|_A := \sqrt{v^\top A v}$ denotes the weighted ℓ_2 norm of a vector v.

Lemma 2 (Concentration). *Under the setting of Theorem 1, for each $p \in (0,1)$, the following event $\mathfrak{E}$ holds with probability at least $1-p/2$:*

$$\left\|\sum_{\tau\in[k-1]}\phi_h^\tau\Big[\overline{V}_{h+1}^k(x_{h+1}^\tau)-\Big(\mathbb{P}_h\overline{V}_{h+1}^k\Big)(x_h^\tau,a_h^\tau,b_h^\tau)\Big]\right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)}, \qquad \forall (k,h)\in[K]\times[H],$$

$$\left\|\sum_{\tau\in[k-1]}\phi_h^\tau\Big[\underline{V}_{h+1}^k(x_{h+1}^\tau)-\Big(\mathbb{P}_h\overline{V}_{h+1}^k\Big)(x_h^\tau,a_h^\tau,b_h^\tau)\Big]\right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)}, \qquad \forall (k,h)\in[K]\times[H].$$

Proof. Fix $(k,h)\in[K]\times[H]$. Let

$$\mathcal{F}_{\tau-1} := \sigma(x_\cdot^1, a_\cdot^1, b_\cdot^1, \ldots, x_\cdot^{\tau-1}, a_\cdot^{\tau-1}, b_\cdot^{\tau-1}, x_1^\tau, a_1^\tau, b_1^\tau, \ldots, x_h^\tau, a_h^\tau, b_h^\tau) \tag{11}$$

be the σ-algebra generated by the data from the first $\tau-1$ episodes plus that from the first h steps of the τth episode. We note that as actions are randomized, they must also be included in the definition of the filtration in Equation (11), unlike in the MDP setting. Also note that $\phi_h^\tau, x_h^\tau, a_h^\tau, b_h^\tau \in \mathcal{F}_{\tau-1}$ and $x_{h+1}^\tau \in \mathcal{F}_\tau$.

Fix a pair $(\widetilde{Q}, \underset{\sim}{Q})$ in the ϵ-net $\mathcal{Q}_\epsilon\times\mathcal{Q}_\epsilon$. For each $x\in\mathcal{S}$, let $\widetilde{\sigma}(x)$ be the CCE of $(\widetilde{Q}(x,\cdot,\cdot), \underset{\sim}{Q}(x,\cdot,\cdot))$ in the sense of Equation (1), and set $\widetilde{V}(x) := \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}[\widetilde{Q}(x,a,b)]$. The random variable $\widetilde{V}(x_{h+1}^\tau) - (\mathbb{P}_h\widetilde{V})(x_h^\tau,a_h^\tau,b_h^\tau)$, when conditioned on $\mathcal{F}_{\tau-1}$, is zero-mean and H-bounded. Applying Lemma B.6 gives

$$\left\|\sum_{\tau\in[k-1]}\phi_h^\tau[\widetilde{V}(x_{h+1}^\tau)-(\mathbb{P}_h\widetilde{V})(x_h^\tau,a_h^\tau,b_h^\tau)]\right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)}$$

with probability at least $1-2^{-\Omega(d^2\log(dT/p))}$. Now note that $|\mathcal{Q}_\epsilon\times\mathcal{Q}_\epsilon| = (\mathcal{N}_\epsilon)^2 \le 4\Big(1+\frac{8H\sqrt{dk}}{\epsilon}\Big)^{2d}\Big(1+\frac{\beta^2\sqrt{d}}{\epsilon^2}\Big)^{2d^2}$ by Lemma B.5. By a union bound and the choice that $\epsilon = 1/(kH)$, the previous inequality holds for all $(\widetilde{Q}, \underset{\sim}{Q})\in\mathcal{Q}_\epsilon\times\mathcal{Q}_\epsilon$ with probability at least $1-p/2$.

Now, for the pair $(\overline{Q}_{h+1}^k, \underline{Q}_{h+1}^k)$, which is in $\mathcal{Q}\times\mathcal{Q}$ by Lemma B.2, let $(\widetilde{Q}, \underset{\sim}{Q})\in\mathcal{Q}_\epsilon\times\mathcal{Q}_\epsilon$ be the pair in the net as chosen in `FIND_CCE`. Recall that by construction we have $\|\widetilde{Q}-\overline{Q}_h^k\|_\infty \le \epsilon$, and $\|\underset{\sim}{Q}-\underline{Q}_h^k\|_\infty\le\epsilon$ and $\overline{V}_{h+1}^k(x) = \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}[\overline{Q}_{h+1}^k(x,a,b)]$. Therefore, the difference $\Delta(x) := \overline{V}_{h+1}^k(x) - \widetilde{V}(x)$ satisfies

$$\begin{aligned}|\Delta(x)| &= \left|\mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\Big[\overline{Q}_{h+1}^k(x,a,b)-\widetilde{Q}(x,a,b)\Big]\right| \\ &\le \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left|\overline{Q}_{h+1}^k(x,a,b)-\widetilde{Q}(x,a,b)\right| \le \epsilon, \ \forall x\in\mathcal{S}.\end{aligned}$$

It follows that

$$\begin{aligned}
&\left\| \sum_{\tau\in[k-1]} \phi_h^\tau \Big[\overline{V}_{h+1}^k(x_{h+1}^\tau) - \Big(\mathbb{P}_h \overline{V}_{h+1}^k\Big)(x_h^\tau, a_h^\tau, b_h^\tau)\Big]\right\|_{(\Lambda_h^k)^{-1}} \\
&\le \left\| \sum_{\tau\in[k-1]} \phi_h^\tau \Big[\widetilde{V}(x_{h+1}^\tau) - (\mathbb{P}_h \widetilde{V})(x_h^\tau, a_h^\tau, b_h^\tau)\Big]\right\|_{(\Lambda_h^k)^{-1}} + \left\| \sum_{\tau\in[k-1]} \phi_h^\tau [\Delta(x_{h+1}^\tau) - (\mathbb{P}_h \Delta)(x_h^\tau, a_h^\tau, b_h^\tau)]\right\|_{(\Lambda_h^k)^{-1}} \\
&\lesssim dH\sqrt{\log(dT/p)} + \epsilon \sum_{\tau\in[k-1]} \|\phi_h^\tau\|_{(\Lambda_h^k)^{-1}} \\
&\le dH\sqrt{\log(dT/p)} + \epsilon k,
\end{aligned}$$

where the last step follows from $\Lambda_h^k \ge I$ and $\|\phi_h^\tau\| \le 1$. Recalling our choice $\epsilon = \frac{1}{KH}$ proves the first inequality in the lemma. The second inequality can be proved in a similar fashion. □

5.2. Least-Squares Estimation Error

Here we bound the difference between the algorithm's action-value functions (without bonus) and the true action-value functions of any policy pair (π, ν), recursively in terms of the step h.

Lemma 3 (Least-squares Error Bound). *The quantities $\{\overline{w}_h^k, \underline{w}_h^k, \overline{V}_h^k, \underline{V}_h^k\}$ in Algorithm 1 satisfy the following. If $\beta = dH\sqrt{\iota}$, where $\iota = \log(2dT/p)$, then on the event $\mathfrak{E}$ in Lemma 2, we have for all $(x,a,b,h,k) \in \mathcal{S}\times\mathcal{A}\times\mathcal{A}\times[H]\times[K]$ and any policy pair (π,ν):*

$$\Big|\langle\phi(x,a,b), \overline{w}_h^k\rangle - Q_h^{\pi,\nu}(x,a,b) - \mathbb{P}_h\Big(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\Big)(x,a,b)\Big| \le \rho_h^k(x,a,b), \tag{12a}$$

$$\Big|\langle\phi(x,a,b), \underline{w}_h^k\rangle - Q_h^{\pi,\nu}(x,a,b) - \mathbb{P}_h\Big(\underline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\Big)(x,a,b)\Big| \le \rho_h^k(x,a,b), \tag{12b}$$

where $\rho_h^k(x,a,b) := \beta\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}$.

Proof. We only prove the first inequality (12a). The second inequality can be proved in a similar fashion.

By Lemma 1 and the Bellman equation we have the equality

$$(\phi_h^\tau)^\top w_h^{\pi,\nu} = Q_h^{\pi,\nu}(x_h^\tau, a_h^\tau, b_h^\tau) = r_h(x_h^\tau, a_h^\tau, b_h^\tau) + (\mathbb{P}_h V_{h+1}^{\pi,\nu})(x_h^\tau, a_h^\tau, b_h^\tau)$$

for all $\tau \in [k-1]$. Multiplying this equality by $(\Lambda_h^k)^{-1}\phi_h^\tau$ and summing over τ, we obtain that

$$\begin{aligned}
w_h^{\pi,\nu} - (\Lambda_h^k)^{-1} w_h^{\pi,\nu} &= (\Lambda_h^k)^{-1}\Bigg(\sum_{\tau\in[k-1]} \phi_h^\tau (\phi_h^\tau)^\top\Bigg) w_h^{\pi,\nu} \\
&= (\Lambda_h^k)^{-1} \sum_{\tau\in[k-1]} \phi_h^\tau \cdot [r_h(x_h^\tau, a_h^\tau, b_h^\tau) + (\mathbb{P}_h V_{h+1}^{\pi,\nu})(x_h^\tau, a_h^\tau, b_h^\tau)],
\end{aligned} \tag{12c}$$

where the first equality (12c) holds because $\sum_{\tau\in[k-1]} \phi_h^\tau(\phi_h^\tau)^\top = \Lambda_h^k - I$. On the other hand, recall that by algorithm specification we have $\overline{w}_h^k = (\Lambda_h^k)^{-1}\sum_{\tau\in[k-1]} \phi_h^\tau \cdot [r_h(x_h^\tau, a_h^\tau, b_h^\tau) + \overline{V}_{h+1}^k(x_{h+1}^\tau)]$. It follows that

$$\begin{aligned}
\overline{w}_h^k - w_h^{\pi,\nu} &= -(\Lambda_h^k)^{-1} w_h^{\pi,\nu} + (\Lambda_h^k)^{-1} \sum_{\tau\in[k-1]} \phi_h^\tau \cdot \Big[\overline{V}_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^{\pi,\nu})(x_h^\tau, a_h^\tau, b_h^\tau)\Big] \\
&= -\underbrace{(\Lambda_h^k)^{-1} w_h^{\pi,\nu}}_{q_1} + \underbrace{(\Lambda_h^k)^{-1} \sum_{\tau\in[k-1]} \phi_h^\tau \cdot \Big[\overline{V}_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h \overline{V}_{h+1}^k)(x_h^\tau, a_h^\tau, b_h^\tau)\Big]}_{q_2} \\
&\quad + \underbrace{(\Lambda_h^k)^{-1} \sum_{\tau\in[k-1]} \phi_h^\tau \cdot \Big[\mathbb{P}_h\Big(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\Big)(x_h^\tau, a_h^\tau, b_h^\tau)\Big]}_{q_3}.
\end{aligned}$$

Hence, for each (x, a, b):

$$\langle\phi(x,a,b), \overline{w}_h^k\rangle - Q_h^{\pi,\nu}(x,a,b) = \langle\phi(x,a,b), q_1 + q_2 + q_3\rangle.$$

We apply Cauchy-Schwarz to bound each RHS term:

1. First term: We have

$$\begin{aligned}|\langle\phi(x,a,b),q_1\rangle| &\le \|w_h^{\pi,\nu}\|_{(\Lambda_h^k)^{-1}}\cdot\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}\\ &\le \|w_h^{\pi,\nu}\|\cdot\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}} \lesssim H\sqrt{d}\cdot\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}},\end{aligned}$$

where the last two steps follow from $\Lambda_h^k \ge I$ and $\|w_h^{\pi,\nu}\| \lesssim H\sqrt{d}$ (Lemma B.1).

2. Second term: By Lemma 2, we have

$$|\langle\phi(x,a,b),q_2\rangle| \lesssim dH\sqrt{\log(dT/p)}\cdot\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}.$$

3. Third term: Recalling that $\sum_{\tau\in[k-1]}\phi_h^\tau(\phi_h^\tau)^\top = \Lambda_h^k - I$ and $\mathbb{P}_h(\cdot\mid x_h^\tau,a_h^\tau,b_h^\tau) = (\phi_h^\tau)^\top\mu_h(\cdot)$, we have

$$\begin{aligned}&\langle\phi(x,a,b),q_3\rangle\\ &= \left\langle\phi(x,a,b),(\Lambda_h^k)^{-1}\sum_{\tau\in[k-1]}\phi_h^\tau(\phi_h^\tau)^\top\int\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x')\mathrm{d}\mu_h(x')\right\rangle\\ &= \left\langle\phi(x,a,b),\int\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x')\mathrm{d}\mu_h(x')\right\rangle - \left\langle\phi(x,a,b),(\Lambda_h^k)^{-1}\int\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x')\mathrm{d}\mu_h(x')\right\rangle\\ &= \mathbb{P}_h\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x,a,b) + \underbrace{\left\langle\phi(x,a,b),(\Lambda_h^k)^{-1}\int\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x')\mathrm{d}\mu_h(x')\right\rangle}_{p_2}.\end{aligned}$$

Note that in this equality we make crucial use of the linearity assumption on the transition kernel. The term p_2 satisfies the bound

$$|p_2| \lesssim \|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}\cdot H\sqrt{d},$$

where we use the facts that $\Lambda_h^k \ge I$, $\|\mu_h(\mathcal{S})\| \le \sqrt{d}$, $\left|\overline{V}_{h+1}^k(\cdot)\right| \le H$, and $\left|V_{h+1}^{\pi,\nu}(\cdot)\right| \le H$.

Combining, we obtain

$$\left|\langle\phi(x,a,b),\overline{w}_h^k\rangle - Q_h^{\pi,\nu}(x,a,b) - \mathbb{P}_h\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,\nu}\right)(x,a,b)\right| \lesssim dH\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}} \le \beta\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}},$$

under our choice of $\beta \asymp dH\sqrt{\iota}$. This completes the proof of the inequality (12a) in the lemma. □

Lemma 3 can be specialized to the value functions of the best response (cf. Remark 1); for example, it holds that

$$\left|\langle\phi(x,a,b),\overline{w}_h^k\rangle - Q_h^{\pi,*}(x,a,b) - \mathbb{P}_h\left(\overline{V}_{h+1}^k - V_{h+1}^{\pi,*}\right)(x,a,b)\right| \le \rho_h^k(x,a,b).$$

We will make use of this bound and its variants in the subsequent proof.

5.3. Upper and Lower Confidence Bounds

With the previous bounds on the estimation errors, we can show that $\underline{V}_h^k$ and $\overline{V}_h^k$ constructed in the algorithm are indeed lower and upper bounds for the true value function. To this end, we state a simple lemma first.

Lemma 4 (Algorithm 2 Finds 2ϵ-CCE). *For each (k, h, x), $\sigma_h^k(x)$ is an 2ϵ-CCE of $\left(\overline{Q}_h^k(x,\cdot,\cdot),\underline{Q}_h^k(x,\cdot,\cdot)\right)$ in the sense that*

$$\begin{aligned}\mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a,b)\right] &\ge \mathbb{E}_{b\sim\mathcal{P}_2\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a',b)\right] - 2\epsilon, \quad \forall a'\in\mathcal{A},\\ \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a,b)\right] &\le \mathbb{E}_{a\sim\mathcal{P}_1\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a,b')\right] + 2\epsilon, \quad \forall b'\in\mathcal{A}.\end{aligned}$$

Proof. Let $(\widetilde{Q},\, Q)$ be the elements in the ϵ-net that are closest to $(\overline{Q}_h^k,\underline{Q}_h^k)$, as specified in Algorithm 2. This means that $|\overline{Q}_h^k(x,a,b) - \widetilde{Q}(x,a,b)| \le \epsilon$ and $|\underline{Q}_h^k(x,a,b) - \widetilde{Q}(x,a,b)| \le \epsilon$ for all (x, a, b). Fix an arbitrary $x\in\mathcal{S}$. Because $\sigma_h^k(x) = \widetilde{\sigma}(x)$ is a CCE of $\left(\widetilde{Q}(x,\cdot,\cdot),\widetilde{Q}(x,\cdot,\cdot)\right)$, we have for all $a'\in\mathcal{A}$:

$$\begin{aligned}\mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a,b)\right] &= \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left[\widetilde{Q}_h^k(x,a,b)\right] + \mathbb{E}_{(a,b)\sim\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a,b) - \widetilde{Q}_h^k(x,a,b)\right]\\ &\ge \mathbb{E}_{b\sim\mathcal{P}_2\widetilde{\sigma}(x)}\left[\widetilde{Q}_h^k(x,a',b)\right] - \epsilon\\ &= \mathbb{E}_{b\sim\mathcal{P}_2\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a',b)\right] + \mathbb{E}_{b\sim\mathcal{P}_2\widetilde{\sigma}(x)}\left[\widetilde{Q}_h^k(x,a',b) - \overline{Q}_h^k(x,a',b)\right] - \epsilon\\ &\ge \mathbb{E}_{b\sim\mathcal{P}_2\widetilde{\sigma}(x)}\left[\overline{Q}_h^k(x,a',b)\right] - 2\epsilon.\end{aligned}$$

This proves the first inequality in the lemma. The second inequality can be proved in a similar fashion. □

We can now establish the UCB and LCB properties.

Lemma 5 (UCB and LCB). *Under the setting of Theorem 1, on the event $\mathfrak{E}$ in Lemma 2, we have for each (x,a,b,k,h):*

$$\underline{Q}_h^k(x,a,b) - 2(H-h+1)\epsilon \overset{(a)}{\le} Q_h^{\pi^k,*}(x,a,b) \overset{(b)}{\le} Q_h^{*,\nu^k}(x,a,b) \overset{(c)}{\le} \overline{Q}_h^k(x,a,b) + 2(H-h+1)\epsilon,$$

and

$$\underline{V}_h^k(x) - 2(H-h+2)\epsilon \overset{(i)}{\le} V_h^{\pi^k,*}(x) \overset{(ii)}{\le} V_h^{*,\nu^k}(x) \overset{(iii)}{\le} \overline{V}_h^k(x) + 2(H-h+2)\epsilon.$$

Proof. The inequalities (b) and (ii) follow from Proposition 1. Next we only prove the upper bounds (c) and (iii). The lower bounds (a) and (i) can be proved in a similar fashion.

We fix k and perform induction on h. The base case $h = H+1$ holds because the terminal cost is zero. Now assume that the bounds (c) and (iii) hold for step $h+1$; that is, $\overline{Q}_{h+1}^k(x,a,b) \ge Q_{h+1}^{*,\nu^k}(x,a,b) - 2(H-h)\epsilon$ and $\overline{V}_{h+1}^k(x) \ge V_{h+1}^{*,\nu^k}(x) - 2(H-h+1)\epsilon$ for all (x, a, b). By inequality (12a) in Lemma 3 applied to $(\widetilde{\pi},\nu^k)$ with $\widetilde{\pi}$ being the best response to ν^k, we have for each (x, a, b):

$$\left|\langle\phi(x,a,b),\overline{w}_h^k\rangle - Q_h^{*,\nu^k}(x,a,b) - \mathbb{P}_h(\overline{V}_{h+1}^k - V_{h+1}^{*,\nu^k})(x,a,b)\right| \le \rho_h^k(x,a,b),$$

whence

$$\langle\phi(x,a,b),\overline{w}_h^k\rangle + \rho_h^k(x,a,b) \ge Q_h^{*,\nu^k}(x,a,b) + \mathbb{P}_h(\overline{V}_{h+1}^k - V_{h+1}^{*,\nu^k})(x,a,b),$$

where we recall that $\rho_h^k(x,a,b) := \beta\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}$. Under the induction hypothesis, we obtain

$$\langle\phi(x,a,b),\overline{w}_h^k\rangle + \rho_h^k(x,a,b) \ge Q_h^{*,\nu^k}(x,a,b) - 2(H-h+1)\epsilon \ge 0.$$

We can now lower-bound $\overline{Q}_h^k(x,a,b)$:

$$\begin{aligned}
&\overline{Q}_h^k(x,a,b) \\
&= \Pi_H\{\langle\phi(x,a,b),\overline{w}_h^k\rangle + \rho_h^k(x,a,b)\} && \text{by construction} \\
&\ge \Pi_H\{Q_h^{*,\nu^k}(x,a,b) - 2(H-h+1)\epsilon\} && u \ge v \Rightarrow \max\{\min\{u,H\},-H\} \ge \max\{\min\{v,H\},-H\} \\
&\ge \Pi_H\{Q_h^{*,\nu^k}(x,a,b)\} - 2(H-h+1)\epsilon && \Pi_H \text{ is non-expansive.} \\
&= Q_h^{*,\nu^k}(x,a,b) - 2(H-h+1)\epsilon. && Q_h^{*,\nu^k}(x,a,b) \in [-H,H]
\end{aligned}$$

This proves the inequality (c) for step h.

Finally, recall that $\nu_h^k(x) := \mathcal{P}_2\sigma_h^k(x)$, and let $\mathsf{br}(\nu_h^k(x))$ denote the best response to $\nu_h^k(x)$ with respect to $Q_h^{*,\nu^k}(x,\cdot,\cdot)$; that is,

$$\mathsf{br}(\nu_h^k(x)) := \arg\max_{A\in\Delta} \mathbb{E}_{a\sim A, b\sim\nu_h^k(x)}\left[Q_h^{*,\nu^k}(x,a,b)\right].$$

We then have for all x:

$$\begin{aligned}
\overline{V}_h^k(x) &:= \mathbb{E}_{(a,b)\sim\sigma_h^k(x)}\left[\overline{Q}_h^k(x,a,b)\right] && \text{by construction} \\
&\ge \mathbb{E}_{a'\sim\mathsf{br}(\nu_h^k(x)), b\sim\mathcal{P}_2\sigma_h^k(x)}\left[\overline{Q}_h^k(x,a',b)\right] - 2\epsilon && \sigma_h^k(x) \text{ is } 2\epsilon\text{-CCE by Lemma 4} \\
&\ge \mathbb{E}_{a'\sim\mathsf{br}(\nu_h^k(x)), b\sim\mathcal{P}_2\sigma_h^k(x)}\left[Q_h^{*,\nu^k}(x,a',b)\right] - 2(H-h+1)\epsilon - 2\epsilon && \text{inequality (c) we just proved} \\
&= \mathbb{E}_{a\sim\mathsf{br}(\nu_h^k(x)), b\sim\nu_h^k(x)}\left[Q_h^{*,\nu^k}(x,a,b)\right] - 2(H-h+2)\epsilon && \text{definition of } \pi_h^k(x) \text{ and } \nu_h^k(x) \\
&= V_h^{*,\nu^k}(x) - 2(H-h+2)\epsilon.
\end{aligned}$$

This proves inequality (iii) for step h. □

5.4. Recursive Decomposition of Duality Gap

Thanks to Lemma 5 established earlier, the difference of the UCB and LCB, namely $\delta_h^k := \overline{V}_h^k(x_h^k) - \underline{V}_h^k(x_h^k)$, is an (approximate) upper bound on the duality gap $V_h^{*,\nu^k}(x_h^k) - V_h^{\pi^k,*}(x_h^k)$. Setting the stage for bounding the duality gap, we show next that δ_h^k can be decomposed recursively into the sum of δ_{h+1}^k and some error terms.

Lemma 6 (Recursive Decomposition). *Define the random variables*

$$\begin{aligned}\delta_h^k &:= \overline{V}_h^k(x_h^k) - \underline{V}_h^k(x_h^k),\\ \zeta_h^k &:= \mathbb{E}[\delta_{h+1}^k \mid x_h^k, a_h^k, b_h^k] - \delta_{h+1}^k,\\ \overline{\gamma}_h^k &:= \mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\Big[\overline{Q}_h^k(x_h^k,a,b)\Big] - \overline{Q}_h^k(x_h^k,a_h^k,b_h^k),\\ \underline{\gamma}_h^k &:= \mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\Big[\underline{Q}_h^k(x_h^k,a,b)\Big] - \underline{Q}_h^k(x_h^k,a_h^k,b_h^k).\end{aligned}$$

Then on the event $\mathfrak{E}$ in Lemma 2, we have for all (k,h),

$$\delta_h^k \le \delta_{h+1}^k + \zeta_h^k + \overline{\gamma}_h^k - \underline{\gamma}_h^k + 4\beta\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}.$$

Proof. For each (x,a,b,k,h), by construction we have

$$\begin{aligned}\overline{Q}_h^k(x,a,b) - \underline{Q}_h^k(x,a,b) &= \Big[(\overline{w}_h^k)^\top\phi(x,a,b) + \beta\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}\Big] - \Big[(\underline{w}_h^k)^\top\phi(x,a,b) - \beta\,\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}\Big]\\ &= (\overline{w}_h^k - \underline{w}_h^k)^\top\phi(x,a,b) + 2\beta\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}.\end{aligned}$$

The inequalities (12a) and (12b) in Lemma 3 ensure that

$$(\overline{w}_h^k - \underline{w}_h^k)^\top\phi(x,a,b) \le \mathbb{P}_h\Big(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k\Big)(x,a,b) + 2\beta\,\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}},$$

Hence, by plugging back we obtain the bound

$$\overline{Q}_h^k(x,a,b) - \underline{Q}_h^k(x,a,b) \le \mathbb{P}_h\Big(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k\Big)(x,a,b) + 4\beta\,\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}. \tag{13}$$

On the other hand, observe that by definition,

$$\begin{aligned}\delta_h^k &:= \overline{V}_h^k(x_h^k) - \underline{V}_h^k(x_h^k)\\ &= \mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\Big[\overline{Q}_h^k(x_h^k,a,b)\Big] - \mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\Big[\underline{Q}_h^k(x_h^k,a,b)\Big]\\ &= \overline{Q}_h^k(x_h^k,a_h^k,b_h^k) - \underline{Q}_h^k(x_h^k,a_h^k,b_h^k) + \Big(\mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\Big[\overline{Q}_h^k(x_h^k,a,b)\Big] - \overline{Q}_h^k(x_h^k,a_h^k,b_h^k)\Big) - \Big(\mathbb{E}_{(a,b)\sim\sigma_h^k(x_h^k)}\underline{Q}_h^k[(x_h^k,a,b)] - \underline{Q}_h^k(x_h^k,a_h^k,b_h^k)\Big)\\ &= \overline{Q}_h^k(x_h^k,a_h^k,b_h^k) - \underline{Q}_h^k(x_h^k,a_h^k,b_h^k) + \overline{\gamma}_h^k - \underline{\gamma}_h^k.\end{aligned}$$

Applying the inequality (13), we obtain

$$\begin{aligned}\delta_h^k &\le \mathbb{P}_h\Big(\overline{V}_{h+1}^k - \underline{V}_{h+1}^k\Big)(x_h^k,a_h^k,b_h^k) + 4\beta\|\phi(x_h^k,a_h^k)\|_{(\Lambda_h^k)^{-1}} + \overline{\gamma}_h^k - \underline{\gamma}_h^k\\ &= \mathbb{E}\Big[\delta_{h+1}^k \mid x_h^k,a_h^k,b_h^k\Big] + 4\beta\|\phi_h^k\|_{(\Lambda_h^k)^{-1}} + \overline{\gamma}_h^k - \underline{\gamma}_h^k\\ &= \delta_{h+1}^k + \zeta_h^k + 4\beta\|\phi_h^k\|_{(\Lambda_h^k)^{-1}} + \overline{\gamma}_h^k - \underline{\gamma}_h^k\end{aligned}$$

as desired. □

5.5. Establishing Duality Gap Bound

We are now ready to prove Theorem 1. First, observe that on the event $\mathfrak{E}$ in Lemma 2 (which holds with probability at least $1-p/2$), we have for all $k\in[K]$:

$$\begin{aligned}V_1^{*,\nu^k}(x_1^k) - V_1^{\pi^k,*}(x_1^k) &\le \overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k) + 8H\epsilon && \text{Lemma 5}\\ &\le \overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k) + \frac{8}{K}. && \text{by the choice } \epsilon = \frac{1}{KH}\end{aligned}$$

This proves the first inequality (6) in Theorem 1.

We next bound the cumulated difference between the UCB and LCB that appear in the RHS of the last inequality. We have

$$\begin{aligned}\sum_{k=1}^K\Big[\overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k)\Big] &= \sum_{k=1}^K\delta_1^k && \text{definition of } \delta_1^k\\ &\le \sum_{k=1}^K\sum_{h=1}^H(\zeta_h^k + \overline{\gamma}_h^k - \underline{\gamma}_h^k) + 4\beta\sum_{k=1}^K\sum_{h=1}^H\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}. && \text{Lemma 6}\end{aligned}$$

We bound the first two RHS terms separately:

• For the first term, we know that $(\zeta_h^k + \overline{\gamma}_h^k - \underline{\gamma}_h^k)$ is a martingale difference sequence (with respect to both h and k), and $|\zeta_h^k + \overline{\gamma}_h^k - \underline{\gamma}_h^k| \le 6H$. Hence by Azuma-Hoeffding, we have with probability at least $1-p/2$,

$$\sum_{k=1}^{K}\sum_{h=1}^{H}(\zeta_h^k + \overline{\gamma}_h^k - \underline{\gamma}_h^k) \lesssim H\cdot\sqrt{KH\iota}.$$

• For the second term, we apply the elliptical potential Lemma B.4 to obtain

$$\begin{aligned}
\sum_{h=1}^{H}\sum_{k=1}^{K}\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k} &\le \sum_{h=1}^{H}\sqrt{K}\sqrt{\sum_{k=1}^{K}(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k} && \text{Jensen's inequality}\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2\log\left(\frac{\det\Lambda_h^K}{\det\Lambda_h^0}\right)} && \text{Lemma B.4}\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2\log\left(\frac{(\lambda + K\max_k\|\phi_h^k\|^2)^d}{\lambda^d}\right)} && \text{by construction of } \Lambda_h^k\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2d\log\left(\frac{\lambda+K}{\lambda}\right)} && \|\phi_h^k\|\le 1,\ \forall h,k \text{ by assumption}\\
&\le H\sqrt{2Kd\iota}.
\end{aligned}$$

Combining the aforementioned inequalities, we obtain that with probability at least $1-p/2$,

$$\sum_{k=1}^{K}\left[\overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k)\right] \lesssim H\sqrt{HK\iota} + 4\beta\cdot H\sqrt{2Kd\iota} \lesssim \sqrt{d^3H^3T\iota^2},$$

by our choice of $\beta \asymp dH\sqrt{\iota}$ and the fact that $T = KH$. This proves the second inequality (7) in Theorem 1.

Finally, recalling the definition of Gap(K) and combining the inequalities (6) and (7) we just proved, we obtain that with probability at least $1-p$,

$$\begin{aligned}
\mathrm{Gap}(K) &:= \sum_{k=1}^{K}\left[V_1^{*,\nu^k}(x_1^k) - V_1^{\pi^k,*}(x_1^k)\right]\\
&\le \sum_{k=1}^{K}\left[\overline{V}_1^k(x_1^k) - \underline{V}_1^k(x_1^k)\right] + 8 \lesssim \sqrt{d^3H^3T\iota^2},
\end{aligned}$$

thereby proving the third inequality (8) in Theorem 1.

6. Conclusion

In this paper, we develop provably efficient reinforcement learning methods for zero-sum Markov games with simultaneous moves and a linear structure. To ensure efficient exploration, our algorithms construct appropriate UCB/LCB for both players and make crucial use of the concept of coarse correlated equilibrium. We provide regret bounds under both the offline and online settings. Corollaries of these bounds apply to turn-based games and the tabular settings. Our results build on and generalize work on learning MDPs with linear structures, and at the same time highlight the crucial differences and new challenges in the game setting.

A number of directions are of interest for future research. An immediate step is to investigate whether the dependence on the dimension d and horizon H in our bounds can be improved and what are the optimal scaling. It would also be interesting to improve our online regret bounds to exploit a weak opponent, in the sense that we can compete with the best response to the opponent, not just competing with the NE. Generalizations to general-sum Markov games, as well as to games with more complicated, nonlinear structures, are also of great interest.

Acknowledgments

The authors thank the anonymous reviewers at the Conference on Learning Theory 2020 for their helpful comments, and Siddhartha Banerjee for inspiring discussion.

Appendix A: Algorithms and Proofs for Turn-based Games

In this section, we present our algorithms for turn-based games and prove the performance guarantees in Corollaries 1 and 2.

A.1. Offline Setting

In this, the algorithm for turn-based games is given in Algorithm A.1, which is derived by specializing the corresponding simultaneous-move Algorithm 1 to the turn-based setting.

Algorithm A.1. (Optimistic Minimax Value Iteration (Turn-Based, Offline))

Input: bonus parameter $\beta > 0$.
for episode $k = 1, 2, \ldots, K$ **do**
  Receive initial state x_1^k.
  for step $h = H, H-1, \ldots, 2, 1$ **do** ▷ update policy
    $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau)\phi(x_h^\tau, a_h^\tau)^\top + I$.
    $\overline{w}_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau)\left[r_h(x_h^\tau, a_h^\tau) + \overline{V}_{h+1}^k(x_{h+1}^\tau)\right]$.
    $\underline{w}_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau)\left[r_h(x_h^\tau, a_h^\tau) + \underline{V}_{h+1}^k(x_{h+1}^\tau)\right]$.
    $\overline{Q}_h^k(\cdot,\cdot) \leftarrow \Pi_H\left\{(\overline{w}_h^k)^\top \phi(\cdot,\cdot) + \beta\sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot,\cdot)}\right\}$
    $\underline{Q}_h^k(\cdot,\cdot) \leftarrow \Pi_H\left\{(\underline{w}_h^k)^\top \phi(\cdot,\cdot) - \beta\sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot,\cdot)}\right\}$
    Let

$$\begin{cases} \pi_h^k(\cdot) \leftarrow \texttt{FIND\_MAX}(\overline{Q}_h^k, \cdot), \overline{V}_h^k(\cdot) \leftarrow \overline{Q}_h^k(\cdot, \pi_h^k(\cdot)), \underline{V}_h^k(\cdot) \leftarrow \underline{Q}_h^k(\cdot, \pi_h^k(\cdot)) & I(\cdot) = 1 \\ \nu_h^k(\cdot) \leftarrow \texttt{FIND\_MIN}(\underline{Q}_h^k, \cdot), \overline{V}_h^k(\cdot) \leftarrow \overline{Q}_h^k(\cdot, \nu_h^k(\cdot)), \underline{V}_h^k(\cdot) \leftarrow \underline{Q}_h^k(\cdot, \nu_h^k(\cdot)) & I(\cdot) = 2 \end{cases}$$

  end for
  for step $h = 1, 2, \ldots, H$ **do** ▷ execute policy
    if $I(x_h^k) = 1$, P1 takes action $a_h^k = \pi_h^k(x_h^k)$,
    else if $I(x_h^k) = 2$, P2 takes action $a_h^k = \nu_h^k(x_h^k)$.
    Observe next state x_{h+1}^k.
  end for
end for

The algorithm involves the subroutines `FIND_MAX` and `FIND_MIN`, which are derived by specializing the `FIND_CCE` routine in Algorithm 2 to the turn-based setting. For completeness, we provide a description of these two subroutines. Let $\mathcal{Q}$ be the class of functions $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with the parametric form

$$Q(x, a) = \langle w, \phi(x, a)\rangle + \rho\beta\sqrt{\phi(x, a)^\top A \phi(x, a)},$$

where the parameter (w, A, ρ) satisfy $\|w\| \le 2H\sqrt{dk}$, $\|A\|_F \le \beta^2\sqrt{d}$, and $\rho \in \{\pm 1\}$. Let $\mathcal{Q}_\epsilon$ be a fixed ϵ-covering of $\mathcal{Q}$ with respect to the ℓ_∞ norm. With these notations, the subroutine `FIND_MAX` is given in Algorithm A.2, and the subroutine `FIND_MIN` is given by $\texttt{FIND\_MIN}(Q, x) = \texttt{FIND\_MAX}(-Q, x)$.

Algorithm A.2. (`FIND_MAX`)

Input: Q, x, and discretization parameter $\epsilon > 0$.
Pick $\widetilde{Q} \in \mathcal{Q}_\epsilon$ satisfying $\|\widetilde{Q} - Q\|_\infty \le \epsilon$.
For the input x, let $\widetilde{a} = \mathrm{argmax}_a \widetilde{Q}(x, a)$.
Output: $\widetilde{a}$.

Informally, one may simply think of $\texttt{FIND\_MAX}(Q, x)$ as $\mathrm{argmax}_a Q(x, a)$ and $\texttt{FIND\_MIN}(Q, x)$ as $\mathrm{argmin}_a Q(x, a)$. As in the simultaneous-move setting, these subroutines are introduced for the technical considerations explained in Section 3.2.1.

A.1.1. Proof of Corollary 1. We prove Corollary 1 by specializing Theorem 1 to the turn-based setting. Specifically, as argued in Section 2.3, linear turn-based game is a special case of linear simultaneous games with

$$\begin{aligned} \phi(x, a, b) &\equiv \phi(x, a), \quad r_h(x, a, b) \equiv r(x, a), \quad \mathbb{P}_h(x, a, b) \equiv \mathbb{P}_h(x, a), \quad \text{if } x \in \mathcal{S}_1, \\ \phi(x, a, b) &\equiv \phi(x, b), \quad r_h(x, a, b) \equiv r(x, b), \quad \mathbb{P}_h(x, a, b) \equiv \mathbb{P}_h(x, b), \quad \text{if } x \in \mathcal{S}_2. \end{aligned} \tag{A.1}$$

Moreover, Algorithm 1, when applied to the turn-based setting, degenerates to Algorithm A.1. To see this, note that under the degeneration of $\phi(x, a, b)$ in (A.1), the values $\overline{Q}_h^k$ and $\underline{Q}_h^k$ computed in Algorithm 1 only depend on the action of the active player; that is,

$$\begin{aligned} \overline{Q}_h^k(x, a, b) &\equiv \overline{Q}_h^k(x, a), \quad \text{if } x \in \mathcal{S}_1, \\ \overline{Q}_h^k(x, a, b) &\equiv \overline{Q}_h^k(x, b), \quad \text{if } x \in \mathcal{S}_2. \end{aligned} \tag{A.2}$$

In this case, one can verify that finding the CCE (cf. Equation (1)) as done in `FIND_CCE` degenerates to a unilateral maximization or minimization problem, namely $\operatorname{argmax}_a \widetilde{Q}(x,a)$ or $\operatorname{argmin}_a \widetilde{Q}(x,a)$. This is exactly what the subroutines `FIND_MAX` and `FIND_MIN` compute. With the previous reduction, Corollary 1 follows directly from Theorem 1.

A.2. Online Setting

In this setting, the algorithm for turn-based games is given in Algorithm A.3, which is derived by specializing the corresponding simultaneous-move Algorithm 3 to the turn-based setting.

Algorithm A.3. (Optimistic Minimax Value Iteration (Turn-Based, Online)

Input: bonus parameter $\beta > 0$.
for episode $k = 1, 2, \ldots, K$ **do**
  Receive initial state x_1^k.
  for step $h = H, H-1, \ldots, 2, 1$ **do** ▷ update policy
    $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau)\phi(x_h^\tau, a_h^\tau)^\top + I$.
    $w_h^k \leftarrow (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + V_{h+1}^k(x_{h+1}^\tau)]$.
    $Q_h^k(\cdot,\cdot) \leftarrow \Pi_H\left\{(w_h^k)^\top \phi(\cdot,\cdot) + \beta\sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot,\cdot)}\right\}$.
    $V_h^k(\cdot) \leftarrow \begin{cases} \max_a Q_{h+1}^k(\cdot, a) & \text{if } I(\cdot) = 1, \\ \min_a Q_{h+1}^k(\cdot, a) & \text{if } I(\cdot) = 2. \end{cases}$
  end for
  for step $h = 1, 2, \ldots, H$ **do** ▷ execute policy
    if $I(x_h^k) = 1$, take action $a_h^k = \arg\max_a Q_h^k(x_h^k, a)$,
    else do nothing and let P2 play.
    Observe next state x_{h+1}^k.
  end for
end for

A.2.1. Proof of Corollary 2. We prove Corollary 2 by specializing Theorem 2 to the turn-based setting. The argument is essentially the same as that in the proof of Corollary 1. We omit the details.

Appendix B: Technical Lemmas

The proofs of our main Theorems 1 and 2 involve several common steps. We summarize these steps as several lemmas, which are either proved below or are standard in the literature.

B.1. Boundedness of Linear Coefficients

We begin with two simple lemmas about boundedness of the linear coefficients of Q functions.

Lemma B.1 (True Coefficients Are Bounded). *Under Assumption 1, for each policy pair (π, ν) of* P1 *and* P2*, the linear coefficient of their action-value function $Q_h^{\pi,\nu}(x,a,b) = \langle \phi(x,a,b), w_h^{\pi,\nu}\rangle$ satisfies*

$$\|w_h^{\pi,\nu}\| \le 2H\sqrt{d}, \quad \forall h \in [H].$$

Proof. From the Bellman equation, we have

$$\begin{aligned} \phi(x,a,b)^\top w_h^{\pi,\nu} = Q_h^{\pi,\nu}(x,a,b) &= r_h(x,a,b) + (\mathbb{P}_h V_{h+1}^{\pi,\nu})(x,a,b) \\ &= \phi(x,a,b)^\top \theta_h + \int V_{h+1}^{\pi,\nu}(x')\phi(x,a,b)^\top d\mu_h(x'), \quad \forall x,a,b,h. \end{aligned}$$

Assuming that $\{\phi(x,a,b)\}$ spans $\mathbb{R}^d$ and solving the linear equation, we obtain

$$w_h^{\pi,\nu} = \theta_h + \int V_{h+1}^{\pi,\nu}(x')d\mu_h(x').$$

Under the normalization Assumption 1, we have $\|\theta_h\| \le \sqrt{d}$, $\|\mu_h(\mathcal{S})\| \le \sqrt{d}$ and $|V_{h+1}^{\pi,\nu}(x')| \le H$. It follows that

$$\|w_h^{\pi,\nu}\| \le \sqrt{d} + H\sqrt{d} \le 2H\sqrt{d},$$

as desired. □

An immediate consequence of the Lemma B.1 is that $\|w_h^{\pi,*}\| \le 2H\sqrt{d}$ and $\|w_h^{*,\nu}\| \le 2H\sqrt{d}$; cf. Remark 1.

Lemma B.2 (Algorithm Coefficients Are Bounded). *The coefficients $\{\overline{w}_h^k, \underline{w}_h^k\}$ in Algorithm 1 and the coefficients $\{w_h^k\}$ in Algorithm 3 satisfy*

$$\|\overline{w}_h^k\| \le 2H\sqrt{dk}, \quad \|\underline{w}_h^k\| \le 2H\sqrt{dk}, \quad \textit{and} \quad \|w_h^k\| \le 2H\sqrt{dk}, \ \forall (k,h) \in [K] \times [H].$$

Proof. We only prove the last inequality. The other two inequalities can be established in exactly the same way. For each k and h, we have

$$
\begin{aligned}
\|w_h^k\| &= \left\|(\Lambda_h^k)^{-1}\sum_{\tau=1}^{k-1}\phi(x_h^\tau,a_h^\tau,b_h^\tau)[r_h(x_h^\tau,a_h^\tau,b_h^\tau)+V_{h+1}^k(x_{t+1}^\tau)]\right\| \\
&\le \sum_{\tau=1}^{k-1}\left\|(\Lambda_h^k)^{-1}\phi(x_h^\tau,a_h^\tau,b_h^\tau)\right\|\cdot 2H && |r_h|\le H,\ |V_{h+1}^k|\le H \\
&\le \sum_{\tau=1}^{k-1}\left\|(\Lambda_h^k)^{-1/2}\right\|\cdot\left\|\phi(x_h^\tau,a_h^\tau,b_h^\tau)\right\|_{(\Lambda_h^k)^{-1}}\cdot 2H \\
&\le \sqrt{k\sum_{\tau=1}^{k-1}\left\|\phi(x_h^\tau,a_h^\tau,b_h^\tau)\right\|^2_{(\Lambda_h^k)^{-1}}}\cdot 2H && \Lambda_h^k\ge I \text{ and Jensen's} \\
&\le \sqrt{kd}\cdot 2H, && \text{Lemma B.3}
\end{aligned}
$$

thereby proving the last inequality in the lemma. □

B.2. Inequalities for Summations

We next state two lemmas for summations. The first lemma is from Jin et al. [44, lemma D.1].

Lemma B.3 (Simple Upper Bound). *If $\Lambda_t=\lambda I+\sum_{i\in[t]}\phi_i\phi_i^\top$, where $\phi_i\in\mathbb{R}^d$ and $\lambda>0$, then*

$$\sum_{i\in[t]}\phi_i^\top\Lambda_t^{-1}\phi_i\le d.$$

The second lemma can be found in Abbasi-Yadkori et al. [1, lemma 11] and Jin et al. [44, lemma D.2].

Lemma B.4 (Elliptical Potential Lemma). *Suppose that $\{\phi_t\}_t\ge 0$ is a sequence in $\mathbb{R}^d$ satisfying $\|\phi_t\|\le 1$, $\forall t$. Let $\Lambda_0\in\mathbb{R}^{d\times d}$ be a positive definite matrix, and $\Lambda_t=\Lambda_0+\sum_{i\in[t]}\phi_i\phi_i^\top$. If the smallest eigenvalue of Λ_0 satisfies $\lambda_{\min}(\Lambda_0)\ge 1$, then*

$$\log\left(\frac{\det\Lambda_t}{\det\Lambda_0}\right)\le\sum_{j\in[t]}\phi_j^\top\Lambda_{j-1}^{-1}\phi_j\le 2\log\left(\frac{\det\Lambda_t}{\det\Lambda_0}\right),\ \forall t.$$

B.3. Covering and Concentration Inequalities for Self-normalized Processes

The first lemma that follows is useful for establishing uniform concentration. Recall the function class $\mathcal{Q}$ defined in the text around Equation (5).

Lemma B.5 (Covering). *The ϵ-covering number of $\mathcal{Q}$ with respect to the ℓ_∞ norm satisfies*

$$\mathcal{N}_\epsilon\le 2\left(1+\frac{8H\sqrt{dk}}{\epsilon}\right)^d\left(1+\frac{8\beta^2\sqrt{d}}{\epsilon^2}\right)^{d^2}.$$

Proof. For any two functions $Q,Q'\in\mathcal{Q}$ with parameters (w,A,ρ) and (w',A',ρ), we have

$$
\begin{aligned}
&\|Q-Q'\|_\infty \\
&=\sup_{x,a,b}\left|\Pi_H\left\{\langle w,\phi(x,a,b)\rangle+\rho\beta\sqrt{\phi(x,a,b)^\top A\phi(x,a,b)}\right\}-\Pi_H\left\{\langle w',\phi(x,a,b)\rangle-\rho\beta\sqrt{\phi(x,a,b)^\top A'\phi(x,a,b)}\right\}\right| \\
&\le\sup_{\phi:\|\phi\|\le 1}\left|\langle w-w',\phi\rangle+\rho\beta\sqrt{\phi^\top A\phi}-\rho\beta\sqrt{\phi^\top A'\phi}\right| \\
&\le\sup_{\phi:\|\phi\|\le 1}|\langle w-w',\phi\rangle|+\sup_{\phi:\|\phi\|\le 1}\sqrt{|\phi^\top(A-A')\phi|} \\
&\le\|w-w'\|+\sqrt{\|A-A'\|_F},
\end{aligned}
$$

where the second to last inequality follows due to the fact that $|\sqrt{x}-\sqrt{y}|\le\sqrt{|x-y|}$ holds for any $x,y\ge 0$.

Therefore, a 0-cover $\mathcal{C}_\rho$ of $\{\pm 1\}$, an $\epsilon/2$-cover $\mathcal{C}_w$ of $\{w\in\mathbb{R}^d:\|w\|\le 2H\sqrt{dk}\}$, and an $\epsilon^2/4$-cover $\mathcal{C}_A$ of $\{A\in\mathbb{R}^{d\times d}:\|A\|_F\le\beta^2\sqrt{d}\}$ implies an ϵ-cover of $\mathcal{Q}$. It follows that

$$\mathcal{N}_\epsilon\le|\mathcal{C}_\rho||\mathcal{C}_w||\mathcal{C}_A|\le 2\left(1+\frac{8H\sqrt{dk}}{\epsilon}\right)^d\left(1+\frac{8\beta^2\sqrt{d}}{\epsilon^2}\right)^{d^2},$$

where the last step follows from standard bounds on the covering number of Euclidean balls, for example, Vershynin [85, lemma 5.2]. □

The next lemma, originally from Abbasi-Yadkori et al. [1, theorem 1], is now standard in the bandit literature.

Lemma B.6 (Concentration for Self-normalized Processes). *Suppose $\{\epsilon_t\}_{t\geq 1}$ is a scalar stochastic process generating the filtration $\{\mathcal{F}_t\}_{t\geq 0}$, and $\epsilon_t \mid \mathcal{F}_{t-1}$ is zero-mean and σ-subGaussian. Let $\{\phi_t\}_{t\geq 1}$ be an $\mathbb{R}^d$-valued stochastic process with $\phi_t \in \mathcal{F}_{t-1}$. Suppose* $\Lambda_0 \in \mathbb{R}^{d\times d}$ is positive definite, and $\Lambda_t = \Lambda_0 + \sum_{s=1}^t \phi_s\phi_s^\top$. *Then for each $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\left\|\sum_{s=1}^t \phi_s\epsilon_s\right\|^2_{\Lambda_t^{-1}} \leq 2\sigma^2 \log\left[\frac{\det(\Lambda_t)^{1/2}\det(\Lambda_0)^{-1/2}}{\delta}\right],\ \forall t\geq 0.$$

Appendix C: Proof of Theorem 2

In this section, we prove Theorem 2 for the online setting of simultaneous games. We shall make use of the technical lemmas given in Appendix B. Recall the shorthand $\phi_h^k := \phi(x_h^k, a_h^k, b_h^k)$. The proof follows a similar strategy as that for the proof of Theorem 1 in Section 5. In particular, our proof consists of five steps, as presented in the following subsections.

C.1. Uniform Concentration

In the online setting, the value function estimate $V_{h+1}^k(x)$ is computed using the NE of the zero-sum game defined by a single payoff matrix $Q_{h+1}^k(x,\cdot,\cdot)$. It is easier to establish uniform concentration in this setting. To see why, we recall the function class $\mathcal{Q}$ defined in the text around Equation (5), and introduce the related function class

$$\mathcal{V} := \left\{V : \mathcal{S}\to\mathbb{R}, V(x) = \max_{A\in\Delta}\min_{B\in\Delta}\mathbb{E}_{a\in A, b\in B} Q(x,a,b), Q\in\mathcal{Q}\right\}.$$

In words, $\mathcal{V}$ contains the possible values of the NEs of the zero-sum matrix games in $\mathcal{Q}$. As we show in Lemma C.1, an ϵ-cover of the set $\mathcal{Q}$ immediately induces an ϵ-cover of the set $\mathcal{V}$, thanks to the nonexpansiveness of the maximin operator for zero-sum games. (Note that general-sum games and their CCEs do not have such a nonexpansiveness property in general; see Appendix E for details.)

Lemma C.1 (Covering). *The ϵ-covering number of $\mathcal{V}$ with respect to the ℓ_∞ norm is upper bounded by*

$$\mathcal{N}_\epsilon \leq 2\left(1+\frac{8H\sqrt{dk}}{\epsilon}\right)^d\left(1+\frac{8\beta^2\sqrt{d}}{\epsilon^2}\right)^{d^2}.$$

Proof. For any two functions $V, V' \in \mathcal{V}$, let them take the form $V(\cdot) = \max_{A\in\Delta}\min_{B\in\Delta}\mathbb{E}_{a\in A,b\in B}Q(\cdot,a,b)$ and $V'(\cdot) = \max_{A\in\Delta}\min_{B\in\Delta}\mathbb{E}_{a\in A,b\in B}Q'(\cdot,a,b)$ with $Q, Q'\in\mathcal{Q}$. Because the maximin operator is nonexpansive, we have

$$\begin{aligned}\|V-V'\|_\infty &= \sup_x\left|\max_{A\in\Delta}\min_{B\in\Delta}\mathbb{E}_{a\in A,b\in B}Q(\cdot,a,b) - \max_{A\in\Delta}\min_{B\in\Delta}\mathbb{E}_{a\in A,b\in B}Q'(\cdot,a,b)\right| \\ &\leq \sup_{x,a,b}\left|Q(x,a,b)-Q'(x,a,b)\right| \\ &= \|Q-Q'\|_\infty.\end{aligned}$$

Therefore, an ϵ-cover of $\mathcal{Q}$ induces an ϵ-cover of $\mathcal{V}$, and hence the ϵ-covering number of $\mathcal{V}$ is upper bounded by the ϵ-covering number of $\mathcal{Q}$. Recalling that the latter number is bounded in Lemma B.5, we complete the proof of the desired bound. □

Lemma C.2 (Concentration). *Under the setting of Theorem 2, for each $p\in(0,1)$, the following event $\mathfrak{E}$ holds with probability at least $1-p/2$:*

$$\left\|\sum_{\tau\in[k-1]}\phi_h^\tau\left[V_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^k)(x_h^\tau,a_h^\tau,b_h^\tau)\right]\right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)},\ \forall (k,h)\in[K]\times[H].$$

Proof. Fix $(k,h)\in[K]\times[H]$. Define the filtration $\{\mathcal{F}_\tau\}$ as in Equation (11).

Set $\epsilon = \frac{1}{K}$ and let $\mathcal{V}_\epsilon$ be a minimal ϵ-net of $\mathcal{V}$. Fix a function $\widetilde{V}\in\mathcal{V}_\epsilon$. The random variable $\widetilde{V}(x_{h+1}^\tau) - \mathbb{P}_h\widetilde{V}(x_h^\tau)$, when conditioned on $\mathcal{F}_{\tau-1}$, is zero-mean and $2H$-bounded. Applying Lemma B.6 gives

$$\left\|\sum_{\tau\in[k-1]}\phi_h^\tau\left(\widetilde{V}(x_{h+1}^\tau) - \mathbb{P}_h\widetilde{V}(x_h^\tau,a_h^\tau,b_h^\tau)\right)\right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)},$$

with probability at least $2^{-\Omega(d^2\log(dT/p))}$. Now note that $|\mathcal{V}_\epsilon| = \mathcal{N}_\epsilon \leq 2\left(1+\frac{8H\sqrt{dk}}{\epsilon}\right)^d\left(1+\frac{\beta^2\sqrt{d}}{\epsilon^2}\right)^{d^2}$ by Lemma C.1. By a union bound, the previous inequality holds for all $\widetilde{V}\in\mathcal{V}_\epsilon$ with probability at least $1-p/2$.

Now, for each $V_{h+1}^k \in \mathcal{V}$ (the inclusion follows from Lemma B.2), let $\widetilde{V} \in \mathcal{V}_\epsilon$ be the closest point in the net. The difference $\Delta = V_{h+1}^k - \widetilde{V}$ satisfies $\|\Delta\|_\infty \le \epsilon$. It follows that

$$\begin{aligned}
&\left\| \sum_{\tau\in[k-1]} \phi_h^\tau \left[V_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^k)(x_h^\tau, a_h^\tau, b_h^\tau) \right] \right\|_{(\Lambda_h^k)^{-1}} \\
&\le \left\| \sum_{\tau\in[k-1]} \phi_h^\tau \left[\widetilde{V}(x_{h+1}^\tau) - (\mathbb{P}_h \widetilde{V})(x_h^\tau, a_h^\tau, b_h^\tau) \right] \right\|_{(\Lambda_h^k)^{-1}} + \left\| \sum_{\tau\in[k-1]} \phi_h^\tau \left[\Delta(x_{h+1}^\tau) - (\mathbb{P}_h \Delta)(x_h^\tau, a_h^\tau, b_h^\tau) \right] \right\|_{(\Lambda_h^k)^{-1}} \\
&\lesssim dH\sqrt{\log(dT/p)} + \epsilon \sum_{\tau\in[k-1]} \|\phi_h^\tau\|_{(\Lambda_h^k)^{-1}} \\
&\le dH\sqrt{\log(dT/p)} + \frac{1}{K}\cdot k,
\end{aligned}$$

where the last step follows from $\epsilon = \frac{1}{K}$, $\Lambda_h^k \ge I$ and $\|\phi_h^\tau\| \le 1$. This completes the proof of the lemma. □

C.2. Least-Squares Estimation Error

Here we bound the difference between the algorithm's value function (without bonus) and the true value function of any policy π, recursively in terms of the step h.

Lemma C.3 (Least-Squares Error Bound). *The quantities $\{w_h^k, V_h^k\}$ in Algorithm 3 satisfy the following. If $\beta = dH\sqrt{\iota}$, then on the event $\mathfrak{E}$ in Lemma C.2, we have for all (x,a,b,h,k) and any policy pair (π,ν):*

$$|\langle \phi(x,a,b), w_h^k\rangle - Q_h^{\pi,\nu}(x,a,b) - \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi,\nu})(x,a,b)| \le \rho_h^k(x,a,b), \tag{C.1}$$

where $\rho_h^k(x,a,b) := \beta\sqrt{\phi(x,a,b)^\top (\Lambda_h^k)^{-1}\phi(x,a,b)}$.

Proof. The proof is essentially identical to that of Lemma 3, except that we use the concentration result in Lemma C.2 instead of Lemma 2. □

C.3. Upper Confidence Bounds

Here we establish the desired UCB property.

Lemma C.4 (UCB). *On the event $\mathfrak{E}$ in Lemma 2, we have for all (x,a,b,k,h):*

$$Q_h^k(x,a,b) \ge Q_h^*(x,a,b), \qquad V_h^k(x) \ge V_h^*(x).$$

Proof. We fix k and perform induction on h. The base case $h = H$ holds because the terminal cost is zero. Now assume that the bounds hold for step $h+1$; that is, $Q_{h+1}^k(x,a,b) \ge Q_{h+1}^*(x,a,b)$ and $V_{h+1}^k(x) \ge V_{h+1}^*(x)$, $\forall (x,a,b)$. By construction, we have

$$Q_h^k(x,a,b) = \Pi H\left\{\left\langle \phi(x,a,b), w_h^k\right\rangle + \beta\left\|\phi(x,a,b)\right\|_{(\Lambda_h^k)^{-1}}\right\}.$$

On the other hand, note that $Q_h^* = Q_h^{\pi^*,\nu^*}$ and $V_h^* = V_h^{\pi^*,\nu^*}$, hence by inequality (C.1) in Lemma C.3 applied to $(\pi,\nu) = (\pi^*,\nu^*)$, we have

$$|\langle \phi(x,a,b), w_h^k\rangle - Q_h^*(x,a,b) - \mathbb{P}_h(V_{h+1}^k - V_{h+1}^*)(x,a,b)| \le \beta\,\|\phi(x,a,b)\|_{(\Lambda_h^k)^{-1}}.$$

Plugging back, we obtain

$$Q_h^k(x,a,b) \ge \Pi_H\{Q_h^*(x,a,b) + \mathbb{P}_h(V_{h+1}^k - V_{h+1}^*)(x,a,b)\}.$$

Under the induction hypothesis, we have $V_{h+1}^k(x) - V_{h+1}^*(x) \ge 0$ for each $x \in \mathcal{S}$, whence

$$Q_h^k(x,a,b) \ge \Pi_H\{Q_h^*(x,a,b)\} = Q_h^*(x,a,b).$$

Consequently, we have

$$\begin{aligned}
V_h^k(x) &= \max_{A\in\Delta}\min_{B\in\Delta} \mathbb{E}_{a\sim A, b\sim B}\left[Q_h^k(x,a,b)\right] && \text{algorithm specification} \\
&\ge \max_{A\in\Delta}\min_{B\in\Delta} \mathbb{E}_{a\sim A, b\sim B}\left[Q_h^*(x,a,b)\right] \\
&= V_h^*(x). && \text{definition}
\end{aligned}$$

We conclude that the bounds hold for step h. □

C.4. Recursive Regret Decomposition

Thanks to Lemma C.4, the regret $V_1^*(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k)$ of interest is upper bounded by the difference $V_1^k(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k)$ between the empirical value (with bonus) and true value of the agent's policy π^k. We next derive a recursive (in h) formula for this difference.

Lemma C.5 (Recursive Decomposition). *Define the random variables*

$$
\begin{aligned}
\delta_h^k &:= V_h^k(x_h^k) - V_h^{\pi^k,\nu^k}(x_h^k),\\
\zeta_h^k &:= \mathbb{E}\Big[\delta_{h+1}^k \mid x_h^k, a_h^k, b_h^k\Big] - \delta_{h+1}^k,\\
\gamma_h^k &:= \mathbb{E}_{a\sim\pi_h^k(x_h^k)}\big[Q_h^k(x_h^k, a, b_h^k)\big] - Q_h^k(x_h^k, a_h^k, b_h^k),\\
\widehat{\gamma}_h^k &:= \mathbb{E}_{a\sim\pi^k(x_h^k), b\sim\nu_h^k(x_h^k)}\Big[Q_h^{\pi^k,\nu^k}(x_h^k, a, b)\Big] - Q_h^{\pi^k,\nu^k}(x_h^k, a_h^k, b_h^k).
\end{aligned}
$$

Then, on the event $\mathfrak{E}$ *in* Lemma 2, *we have for all* (k, h)*:*

$$\delta_h^k \le \delta_{h+1}^k + \zeta_h^k + \gamma_h^k - \widehat{\gamma}_h^k + 2\beta\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}.$$

Proof. By algorithm specification and the fact that $(\pi_h^k(x_h^k), B_0)$ is the NE of $Q_h^k(x_h^k, \cdot, \cdot)$, we have

$$
\begin{aligned}
V_h^k(x_h^k) &= \min_b \mathbb{E}_{a\sim\pi_h^k(x_h^k)}\big[Q_h^k(x_h^k, a, b)\big]\\
&\le \mathbb{E}_{a\sim\pi_h^k(x_h^k)}\big[Q_h^k(x_h^k, a, b_h^k)\big]\\
&= Q_h^k(x_h^k, a_h^k, b_h^k) + \gamma_h^k,
\end{aligned}
$$

and by definition we have

$$
\begin{aligned}
V_h^{\pi^k,\nu^k}(x_h^k) &= \mathbb{E}_{a\sim\pi^k(x_h^k), b\sim\nu_h^k(x_h^k)}\Big[Q_h^{\pi^k,\nu^k}(x_h^k, a, b)\Big]\\
&= Q_h^{\pi^k,\nu^k}(x_h^k, a_h^k, b_h^k) + \widehat{\gamma}_h^k.
\end{aligned}
$$

It follows that

$$\delta_h^k \le Q_h^k(x_h^k, a_h^k, b_h^k) - Q_h^{\pi^k,\nu^k}(x_h^k, a_h^k, b_h^k) + \gamma_h^k - \widehat{\gamma}_h^k.$$

On the other hand, by construction of Q_h^k and Lemma B.6, we have for all (x, a, b),

$$Q_h^k(x,a,b) - Q_h^{\pi^k,\nu^k}(x,a,b) \le \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi^k,\nu^k})(x,a,b) + 2\beta\sqrt{\phi(x,a,b)^\top(\Lambda_h^k)^{-1}\phi(x,a,b)}.$$

Combining pieces, we obtain that

$$
\begin{aligned}
\delta_h^k &\le \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi^k,\nu^k})(x_h^k, a_h^k, b_h^k) + \gamma_h^k - \widehat{\gamma}_h^k + 2\beta\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}\\
&= \mathbb{E}\Big[\delta_{h+1}^k \mid x_h^k, a_h^k, b_h^k\Big] + \gamma_h^k - \widehat{\gamma}_h^k + 2\beta\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}\\
&= \delta_{h+1}^k + \zeta_h^k + \gamma_h^k - \widehat{\gamma}_h^k + 2\beta\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}
\end{aligned}
$$

as desired. □

C.5. Establishing Regret Bound

We are now ready to prove Theorem 2. First, observe that

$$
\begin{aligned}
\mathrm{Regret}(K) &:= \sum_{k=1}^K\Big[V_1^*(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k)\Big] && \text{definition}\\
&\le \sum_{k=1}^K\Big[V_1^k(x_1^k) - V_1^{\pi^k,\nu^k}(x_1^k)\Big] && V_1^k(x_1^k) \ge V_h^*(x_1^k) \text{ by Lemma C.4}\\
&= \sum_{k=1}^K \delta_1^k && \text{definition}\\
&\le \sum_{k=1}^K\sum_{h=1}^H(\zeta_h^k + \gamma_h^k - \hat{\gamma}_h^k) + 2\beta\sum_{k=1}^K\sum_{h=1}^H\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k}. && \text{Lemma C.5.}
\end{aligned}
$$

We bound the two RHS terms separately.

- For the first term, we know that $(\zeta_h^k + \gamma_h^k - \hat{\gamma}_h^k)$ is a martingale difference sequence (with respect to both h and k), and $|\zeta_h^k + \gamma_h^k - \hat{\gamma}_h^k| \le 6H$. Hence by Azuma-Hoeffding, we have with high probability.

$$\sum_{k=1}^K\sum_{h=1}^H(\zeta_h^k + \gamma_h^k - \hat{\gamma}_h^k) \lesssim H\cdot\sqrt{KH\iota} = H\sqrt{T\iota}.$$

- For the second term, we apply the elliptical potential Lemma B.4 to obtain

$$\begin{aligned}
\sum_{h=1}^{H}\sum_{k=1}^{K}\sqrt{(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k} &\le \sum_{h=1}^{H}\sqrt{K}\sqrt{\sum_{k=1}^{K}(\phi_h^k)^\top(\Lambda_h^k)^{-1}\phi_h^k} && \text{Jensen's inequality}\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2\log\left(\frac{\det\Lambda_h^K}{\det\Lambda_h^0}\right)} && \text{Lemma B.4}\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2\log\left(\frac{(\lambda+K\max_k\|\phi_h^k\|^2)^d}{\lambda^d}\right)} && \text{by construction of } \Lambda_h^k\\
&\le \sum_{h=1}^{H}\sqrt{K}\cdot\sqrt{2d\log\left(\frac{\lambda+K}{\lambda}\right)} && \|\phi_h^k\|\le 1,\ \forall h,k \text{ by assumption}\\
&\le H\sqrt{2Kd\iota}.
\end{aligned}$$

Combining, we obtain that

$$\mathrm{Regret}(K)\lesssim H\sqrt{T\iota}+\beta\cdot H\sqrt{2Kd\iota}\lesssim\sqrt{d^3H^3T\iota^2},$$

by our choice of $\beta\asymp dH\sqrt{\iota}$. This completes the proof of Theorem 2.

Appendix D: Efficient Implementation of `FIND_CCE`

The main computation step in `FIND_CCE` is to find an element in the fixed ϵ-cover $\mathcal{Q}_\epsilon$ that is close to a given function Q. Here, we discuss how to efficiently implement this procedure without explicitly maintaining the cover $\mathcal{Q}_\epsilon$.

Recall that each element in $\mathcal{Q}_\epsilon$ is defined by a pair $(w,A)\in\mathbb{R}^d\times\mathbb{R}^{d\times d}$. Therefore, $\mathcal{Q}_\epsilon$ is induced, up to scaling, by an ϵ-cover $\mathcal{C}_w$ in ℓ_2 norm of the Euclidean ball $\mathbb{B}_w:=\{w\in\mathbb{R}^d:\|w\|\le 1\}$ as well as an ϵ^2-cover $\mathcal{C}_A$ of the ball $\mathbb{B}_A:=\{A\in\mathbb{R}^{d\times d}:\|A\|_F\le 1\}$; cf. the proof of Lemma B.5. We may replace $\mathcal{C}_w$ by a cover $\mathcal{C}_{w,\infty}$ in the ℓ_∞ norm; similarly for $\mathcal{C}_A$. Clearly, an ℓ_∞ cover is also an ℓ_2 cover; moreover, an ℓ_∞ cover allows for efficient computation of near neighbors by simple rounding. The price we pay is an additional dimension factor d in the covering number, which eventually goes into the log term.

We now provide the details for covering $\mathbb{B}_w$; the idea applies similarly to covering $\mathbb{B}_A$.

Lemma D.1. *Let $\epsilon>0$ be a given accuracy parameter. There exists a set $\mathcal{C}_{w,\infty}$ satisfying the following:* (i) $\log|\mathcal{C}_{w,\infty}|\le d\log\left(1+\frac{2\sqrt{d}}{\epsilon}\right)$; (ii) *for each vector $w\in\mathbb{B}_w$, we can find, in $O(d)$ time, a vector $\widetilde{w}\in\mathcal{C}_{w,\infty}$ that satisfies $\|\widetilde{w}-w\|_\infty\le\frac{\epsilon}{\sqrt{d}}$, and hence $\|\widetilde{w}-w\|\le\epsilon$.*

Proof. Set $\epsilon_0:=\frac{\epsilon}{\sqrt{d}}$. We discretize the interval $G:=[-1,1]$ into an ϵ_0-grid as

$$G_{\epsilon_0}:=\left\{k\epsilon_0:k=-\left\lfloor\frac{1}{\epsilon_0}\right\rfloor,-\left\lfloor\frac{1}{\epsilon_0}\right\rfloor+1,\ldots,-2,-1,0,1,2,\ldots,\left\lfloor\frac{1}{\epsilon_0}\right\rfloor-1,\left\lfloor\frac{1}{\epsilon_0}\right\rfloor\right\}.$$

We then let $\mathcal{C}_{w,\infty}:=(G_{\epsilon_0})^d$. The log cardinality of $\mathcal{C}_{w,\infty}$ is

$$\log|\mathcal{C}_{w,\infty}|=\log|G_{\epsilon_0}|^d=\log\left(1+2\left\lfloor\frac{1}{\epsilon_0}\right\rfloor\right)^d\le d\log\left(1+\frac{2\sqrt{d}}{\epsilon}\right),$$

as claimed in part (i) of the lemma. Compare this bound with the log cardinality of the optimal ϵ-cover in ℓ_2 norm of $\{w\in\mathbb{R}^d:\|w\|\le 1\}$: $\log|\mathcal{C}_w|\asymp d\log\left(1+\frac{2}{\epsilon}\right)$. We see that the former is only logarithmic larger than the latter.

Moreover, for each vector w in the ball $\{w'\in\mathbb{R}^d:\|w'\|\le 1\}$, we can efficiently find a vector $\widetilde{w}\in\mathcal{C}_{w,\infty}$ that satisfies $\|\widetilde{w}-w\|_\infty\le\frac{\epsilon}{\sqrt{d}}$ and hence $\|\widetilde{w}-w\|\le\epsilon$. To do this, we simply let

$$\widetilde{w}_i=\left\lfloor\frac{|w_i|}{\epsilon_0}\right\rfloor\cdot\epsilon_0\cdot\mathrm{sign}(w_i),\qquad\text{for each } i\in[d],$$

with the convention that $\mathrm{sign}(0)=0$. Note that $\widetilde{w}$ can be computed in $O(d)$ time. Moreover, since $\|w\|\le 1$, for each $i\in[d]$ we have $|w_i|\le 1$ and hence

$$\left\lfloor\frac{|w_i|}{\epsilon_0}\right\rfloor\in\left\{0,1,\ldots,\left\lfloor\frac{1}{\epsilon_0}\right\rfloor\right\},$$

which means $\widetilde{w}_i\in G_{\epsilon_0}$. It follows that $\widetilde{w}\in(G_{\epsilon_0})^d=\mathcal{C}_{w,\infty}$ as claimed. Finally, the approximation accuracy satisfies

$$\begin{aligned}
\|\widetilde{w}-w\|_\infty &=\max_{i\in[d]}\left|\left\lfloor\frac{|w_i|}{\epsilon_0}\right\rfloor\cdot\epsilon_0\cdot\mathrm{sign}(w_i)-w_i\right|\\
&=\epsilon_0\max_{i\in[d]}\left|\left\lfloor\frac{|w_i|}{\epsilon_0}\right\rfloor\cdot\mathrm{sign}(w_i)-\frac{|w_i|}{\epsilon_0}\cdot\mathrm{sign}(w_i)\right| && w_i=|w_i|\cdot\mathrm{sign}(w_i)\\
&\le\epsilon_0\max_{i\in[d]}1\cdot|\mathrm{sign}(w_i)| && |\lfloor x\rfloor-x|\le 1\\
&\le\epsilon_0=\frac{\epsilon}{\sqrt{d}}.
\end{aligned}$$

This proves part (ii) of the lemma. □

Appendix E: Instability of the Value of General-Sum Game

In the analysis of our algorithms (in particular, in proving uniform concentration in the proof of Theorem 1), we encounter the following question: Is the value of the CCE of a general-sum game stable under perturbation to the payoff matrices? Here we show that the answer is negative in general, by demonstrating a counter example.

Consider a two-player general-sum matrix game, and recall our convention that player 1 tries to maximize and player 2 tries to minimize (cf. Section 2.1). Let $u_i : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ be the payoff matrix of player $i \in \{1,2\}$, such that player i receives the payoff $u_i(a,b)$ when players 1 and 2 take actions a and b, respectively. Let $\sigma \in \Delta(\mathcal{A} \times \mathcal{A})$ be any notion of CCE that is unique; for example., the social-optimal or max-entropy CCE. In this equilibrium, the expected payoff of player i is

$$V_i(u_1,u_2) := \mathbb{E}_{(a,b)\sim\sigma}[u_i(a,b)].$$

We say that the game value $V = (V_1, V_2)$ is a Lipschitz function of the payoff matrices $u = (u_1,u_2)$ if there exists a universal constant C such that

$$\underbrace{\max_{i\in\{1,2\}} |V_i(u_1,u_2) - V_i(u_1',u_2')|}_{\|V(u)-V(u')\|_\infty} \le C \cdot \underbrace{\max_{j\in\{1,2\}} \max_{a,b\in\mathcal{A}} |u_j(a,b) - u_j'(a,b)|}_{\|u-u'\|_\infty}, \quad \forall u,u'.$$

The following example shows that V is in general not Lipschitz in u.[7]

Lemma E.1. *For any $\epsilon > 0$, there exists a pair of games u and u', each with a unique CCE, such that*

$$\|u-u'\| \le 2\epsilon \quad \textit{and} \quad \|V(u)-V(u')\|_\infty \ge 1.$$

Proof. Consider two games u and u' with payoff matrices

$$(u_1,u_2) = \begin{pmatrix} 1+\epsilon, -1-\epsilon & \epsilon, -1 \\ 1, -\epsilon & 0,0 \end{pmatrix} \quad \text{and} \quad (u_1',u_2') = \begin{pmatrix} 1-\epsilon, -1+\epsilon & -\epsilon, -1 \\ 1,\epsilon & 0,0 \end{pmatrix},$$

where $\epsilon > 0$. Note that the two pairs of payoff matrices satisfy $\|u-u'\|_\infty = 2\epsilon$, so u and u' can be made arbitrarily close. The game u has a unique CCE, which is the deterministic policy (or pure strategy) corresponding to the top-left entry of the payoff matrices; similarly, the game u' has a unique CCE corresponding to the bottom-right entry. These two CCEs have values

$$(V_1(u_1,u_2), V_2(u_1,u_2)) = (1+\epsilon, -1-\epsilon) \quad \text{and} \quad (V_1(u_1',u_2'), V_2(u_1',u_2')) = (0,0),$$

which are bounded away from each other as claimed. □

We note that in the previous example, the CCE policy of the game u is an ϵ-approximate CCE of the game u', and vice versa, as any unilateral deviation leads to at most ϵ improvement in the payoff.

Endnotes

[1] We note in passing that there is a more restrictive notion of correlated equilibrium (CE) (Aumann [6], Moulin and Vial [60]), in which the deviation is allowed to depend on the original actions. The set of CCEs include the set of CEs, which in turn includes the set of NEs. We use CCE in this paper as it is the easiest to compute among the three.

[2] This holds, for example, when the state space is compact (Maitra and Parthasarathy [57], [58]).

[3] The assumption $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ is satisfied if one incorporates the "turn" of the player as part of the state.

[4] In the tabular setting, recent work (Agarawal et al. [5], Ding and Chen [25], Pananjady and Wainwright [64]) bypasses the use of uniform concentration by employing sophisticated leave-one-out techniques to decouple the probabilistic dependency. However, it is unclear how such techniques can be used in the function approximation setting.

[5] This can be done by linear programming—as the inequalities in the definition (1) of CCE are linear in σ—or by self-playing a no-regret algorithm (Blum et al. [12]).

[6] For the general case where x_1 is sampled from a fixed distribution, we can simply add an additional time step at the beginning of each episode.

[7] We learned the example from https://mathoverflow.net/questions/347366/perturbation-of-the-value-of-a-general-sum-game-at-a-equilibirium.

References

[1] Abbasi-Yadkori Y, Pál D, Szepesvári C (2011) Improved algorithms for linear stochastic bandits. *Adv. Neural Inform. Processing Systems* 24:2312–2320.

[2] Abbasi-Yadkori Y, Lazic N, Szepesvari C, Weisz G (2019a) Exploration-enhanced POLITEX. Preprint, submitted August 27, https://doi.org/10.48550/arXiv.1908.10479.

[3] Abbasi-Yadkori Y, Bartlett P, Bhatia K, Lazic N, Szepesvari C, Weisz G (2019b) POLITEX: Regret bounds for policy iteration using expert prediction. Chaudhuri K, Salakhutdinov R, eds. *Proc. Internat. Conf. Machine Learn.* (PMLR), 3692–3702.

[4] Agrawal S, Jia R (2017) Optimistic posterior sampling for reinforcement learning: Worst-case regret bounds. *Adv. Neural Inform. Processing Systems* 30:1184–1194.

[5] Agarwal A, Kakade S, Yang LF (2020) Model-based reinforcement learning with a generative model is minimax optimal. Abernethy J, Agarwal S, eds. *Proc. Conf. Learn. Theory* (PMLR), 67–83.

[6] Aumann RJ (1987) Correlated equilibrium as an expression of Bayesian rationality. *Econometrica* 55:1–18.

[7] Azar MG, Osband I, Munos R (2017) Minimax regret bounds for reinforcement learning. Precup D, Whye Teh Y, eds., *Proc. Internat. Conf. Machine Learn.*, vol. 70 (PMLR), 263–272.

[8] Bai Y, Jin C (2020) Provable self-play algorithms for competitive reinforcement learning. Daume III H, Singh A, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 551–560.

[9] Bai Y, Jin C, Yu T (2020) Near-optimal reinforcement learning with self-play. *Adv. Neural Inform. Processing Systems.* 33:2159–2170.

[10] Baker B, Kanitscheider I, Markov T, Wu Y, Powell G, McGrew B, Mordatch I (2019) Emergent tool use from multi-agent autocurricula. *Proc. Internat. Conf. Learn. Representations* (OpenReview).

[11] Bansal T, Pachocki J, Sidor S, Sutskever I, Mordatch I (2018) Emergent complexity via multi-agent competition. *Proc. Internat. Conf. Learn. Representations* (OpenReview).

[12] Blum A, Hajiaghayi M, Ligett K, Roth A (2008) Regret minimization and the price of total anarchy. Dwork C, ed., *Proc. 40th Annual ACM Symp. Theory Comput.* (Association for Computing Machinery, New York), 373–382.

[13] Bradtke SJ, Barto AG (1996) Linear least-squares algorithms for temporal difference learning. *Machine Learn.* 22(1–3):33–57.

[14] Brown N, Sandholm T (2018) Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359(6374):418–424.

[15] Brown N, Sandholm T (2019) Superhuman AI for multiplayer poker. *Science* 365(6456):885–890.

[16] Busoniu L, Babuska R, Schutter BD (2008) A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Systems Man Cybernetics C* 38(2):156–172.

[17] Cai Q, Yang Z, Jin C, Wang Z (2020) Provably efficient exploration in policy optimization. Daume III H, Singh A, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 1283–1294.

[18] Chen X, Deng X, Teng SH (2009) Settling the complexity of computing two-player Nash equilibria. *J. ACM* 56(3):1–57.

[19] Chen Z, Zhou D, Gu Q (2021) Almost optimal algorithms for two-player markov games with linear function approximation. Preprint, submitted February 15, https://doi.org/10.48550/arXiv.2102.07404.

[20] Dann C, Lattimore T, Brunskill E (2017) Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *Adv. Neural Inform. Processing Systems* 30:5717–5727.

[21] Dann C, Jiang N, Krishnamurthy A, Agarwal A, Langford J, Schapire RE (2018) On oracle-efficient PAC rl with rich observations. *Adv. Neural Inform. Processing Systems* 31:1422–1432.

[22] Daskalakis C, Foster DJ, Golowich N (2020) Independent policy gradient methods for competitive reinforcement learning. *Adv. Neural Inform. Processing Systems* 33:5527–5540.

[23] Daskalakis C, Goldberg PW, Papadimitriou CH (2009) The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39(1):195–259.

[24] Davis T, Burch N, Bowling M (2014) Using response functions to measure strategy strength. Kambhampati S, ed., *Proc. 28th AAAI Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA).

[25] Ding L, Chen Y (2020) Leave-one-out approach for matrix completion: Primal and dual analysis. *IEEE Trans. Inform. Theory* 66(11):7274–7301.

[26] Dong K, Peng J, Wang Y, Zhou Y (2020) $\sqrt{n}$-regret for learning in Markov decision processes with function approximation and low Bellman rank. Abernethy J, Agarwal S, eds., *Proc. Conf. Learn. Theory* (PMLR), 1554–1557.

[27] Dong K, Wang Y, Chen X, Wang L (2019) Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. *Proc. Internat. Conf. Learn. Representations* (OpenReview).

[28] Du SS, Kakade SM, Wang R, Yang LF (2020) Is a good representation sufficient for sample efficient reinforcement learning? *Proc. Internat. Conf. Learn. Representations* (OpenReview).

[29] Du SS, Luo Y, Wang R, Zhang H (2019b) Provably efficient Q-learning with function approximation via distribution shift error checking oracle. *Adv. Neural Inform. Processing Systems* 32:8058–8068.

[30] Du SS, Krishnamurthy A, Jiang N, Agarwal A, Dudik M, Langford J (2019a) Provably efficient RL with rich observations via latent state decoding. Chaudhuri K, Salakhutdinov R, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 1665–1674.

[31] Fan J, Wang Z, Xie Y, Yang Z (2020) A theoretical analysis of deep q-learning. Bayen AM, Jadbabaie A, Pappas G, Parrilo PA, Recht B, Tomlin C, Zeilinger M, eds., *Proc. Learn. Dynamics Control* (PMLR), 486–489.

[32] Foerster J, Assael IA, De Freitas N, Whiteson S (2016) Learning to communicate with deep multi-agent reinforcement learning. *Adv. Neural Inform. Processing Systems* 29:2137–2145.

[33] Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press, Cambridge, MA).

[34] Grau-Moya J, Leibfried F, Bou-Ammar H (2018) Balancing two-player stochastic games with soft Q-learning. Lang J, ed., *Proc. 27th Internat. Joint Conf. Artificial Intelligence*, (International Joint Conferences on Artificial Intelligence, California), 268–274.

[35] Greenwald A, Hall K, Serrano R (2003) Correlated Q-learning. Fawcett T, Mishra N, eds., *Proc. 20th Internat. Conf. Machine Learn.*, vol. 20 (AAAI Press, Menlo Park, CA), 242–249.

[36] Hansen TD, Miltersen PB, Zwick U (2013) Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J. ACM* 60(1):1–16.

[37] Hu J, Wellman MP (2003) Nash Q-learning for general-sum stochastic games. *J. Machine Learn. Res.* 4(Nov):1039–1069.

[38] Jaksch T, Ortner R, Auer P (2010) Near-optimal regret bounds for reinforcement learning. *J. Machine Learn. Res.* 11(Apr):1563–1600.

[39] Jaques N, Lazaridou A, Hughes E, Gulcehre C, Ortega P, Strouse D, Leibo JZ, De Freitas N (2019) Social influence as intrinsic motivation for multi-agent deep reinforcement learning. Chaudhuri K, Salakhutdinov R, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 3040–3049.

[40] Jia Z, Yang LF, Wang M (2019) Feature-based Q-learning for two-player stochastic games. Preprint, submitted June 2, https://doi.org/10.48550/arXiv.1906.00423.

[41] Jiang N, Krishnamurthy A, Agarwal A, Langford J, Schapire RE (2017) Contextual decision processes with low Bellman rank are PAC-learnable. Precup D, Whye Teh Y, eds., *Proc. 34th Internat. Conf. Machine Learn.*, vol. 70 (PMLR), 1704–1713.

[42] Jin C, Liu Q, Yu T (2021) The power of exploiter: Provable multi-agent rl in large state spaces. Preprint, https://doi.org/10.48550/arXiv.2106.03352.

[43] Jin C, Allen-Zhu Z, Bubeck S, Jordan MI (2018) Is Q-learning provably efficient? *Adv. Neural Inform. Processing Systems* 31:4863–4873.

[44] Jin C, Yang Z, Wang Z, Jordan MI (2020a) Provably efficient reinforcement learning with linear function approximation. Abernethy J, Agarwal S, eds., *Proc. Conf. Learn. Theory* (PMLR), 2137–2143.
[45] Jin C, Jin T, Luo H, Sra S, Yu T (2020b) Learning adversarial Markov decision processes with bandit feedback and unknown transition. Daume III H, Singh A, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 4860–4869.
[46] Kakade SM (2003) On the sample complexity of reinforcement learning. PhD thesis, Gatsby Computational Neuroscience Unit, University College London.
[47] Konda VR, Tsitsiklis JN (1999) Actor-critic algorithms. *Adv. Neural Inform. Processing Systems* 12:1008–1014.
[48] Lagoudakis MG, Parr R (2002) Value function approximation in zero-sum Markov games. Darwiche A, Friedman N, eds., *Proc. 18th Conf. Uncertainty Artificial Intelligence* (Morgan Kaufmann Publishers, San Francisco), 283–292.
[49] Lattimore T, Szepesvári C(2020) *Bandit Algorithms* (Cambridge University Press, Cambridge, UK).
[50] LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444.
[51] Lim SH, Xu H, Mannor S (2013) Reinforcement learning in robust Markov decision processes. *Adv. Neural Inform. Processing Systems* 26: 701–709.
[52] Littman ML (2001a) Friend-or-foe Q-learning in general sum games. Brodley CE, Danyluk AP, eds., *Proc. 18th Internat. Conf. Machine Learn.* (Morgan Kaufmann Publishers, San Francisco), 322–328.
[53] Littman ML (2001b) Value-function reinforcement learning in Markov games. *Cognitive Systems Res.* 2(1):55–66.
[54] Littman ML, Szepesvari C (1996) A generalized reinforcement-learning model: Convergence and applications. Saitta L, ed., *Proc. 13th Internat. Conf. Machine Learn.* (Morgan Kaufmann Publishers, San Francisco), 310–318.
[55] Liu Q, Yu T, Bai Y, Jin C (2021) A sharp analysis of model-based reinforcement learning with self-play. Meila M, Zhang T, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 7001–7010.
[56] Lowe R, Wu Y, Tamar A, Harb J, Abbeel OP, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inform. Processing Systems* 30:6379–6390.
[57] Maitra A, Parthasarathy T (1970) On stochastic games. *J. Optim. Theory Appl.* 5(4):289–300.
[58] Maitra A, Parthasarathy T (1971) On stochastic games, ii. *J. Optim. Theory Appl.* 8(2):154–160.
[59] Moravčík M, Schmid M, Burch N, Lisý V, Morrill D, Bard N, Davis T, Waugh K, Johanson M, Bowling M (2017) Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(6337):508–513.
[60] Moulin H, Vial JP (1978) Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *Internat. J. Game Theory* 7(3–4):201–221.
[61] Munos R, Szepesvári C (2008) Finite-time bounds for fitted value iteration. *J. Machine Learn. Res.* 9(May):815–857.
[62] Open AI (2018) OpenAI Five. Accessed June 21, 2022, https://openai.com/blog/openai-five/.
[63] Osband I, Van Roy B, Wen Z (2016) Generalization and exploration via randomized value functions. Balcan MF, Weinberger KQ, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 2377–2386.
[64] Pananjady A, Wainwright MJ (2019) Value function estimation in Markov reward processes: Instance-dependent ℓ_∞-bounds for policy evaluation. Preprint, submitted September 19, https://arxiv.org/abs/1909.08749.
[65] Papadimitriou CH, Roughgarden T (2008) Computing correlated equilibria in multi-player games. *J. ACM* 55(3):1–29.
[66] Pérolat J, Piot B, Pietquin O (2018) Actor-critic fictitious play in simultaneous move multistage games. Storkey A, Perez-Cruz F, eds., *Proc. Internat. Conf. Artificial Intelligence Statist.* (PMLR), 919–928.
[67] Pérolat J, Piot B, Scherrer B, Pietquin O (2016b) On the use of non-stationary strategies for solving two-player zero-sum Markov games. Gretton A, Robert CC, eds., *Proc. Artificial Intelligence Statist.* (PMLR), 893–901.
[68] Pérolat J, Scherrer B, Piot B, Pietquin O (2015) Approximate dynamic programming for two-player zero-sum Markov games. Bach F, Blei D, eds., *Proc. 32nd Internat. Conf. Machine Learn.*, vol. 37 (PMLR, Lille, France), 1321–1329.
[69] Pérolat J, Strub F, Piot B, Pietquin O (2017) Learning Nash equilibrium for general-sum Markov games from batch data. Singh A, Zhu J, eds., *Proc. Artificial Intelligence Statist.* (PMLR), 232–241.
[70] Pérolat J, Piot B, Geist M, Scherrer B, Pietquin O (2016a) Softened approximate policy iteration for Markov games. Balcan MF, Weinberger KQ, eds., *Proc. 33rd Internat. Conf. Machine Learn., vol. 48* (PMLR, New York), 1860–1868.
[71] Puterman ML (2014) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons, Hoboken, NJ).
[72] Rosenberg A, Mansour Y (2019) Online stochastic shortest path with bandit feedback and unknown transition function. *Adv. Neural Inform. Processing Systems* 32:2209–2218.
[73] Russo D (2019) Worst-case regret bounds for exploration via randomized value functions. *Adv. Neural Inform. Processing Systems* 32: 14410–14420.
[74] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. Preprint, submitted July 20, https://doi.org/10.48550/arXiv.1707.06347.
[75] Shalev-Shwartz S, Shammah S, Shashua A (2016) Safe, multi-agent, reinforcement learning for autonomous driving. Preprint, submitted October 11, https://doi.org/10.48550/arXiv.1610.03295.
[76] Shapley LS (1953) Stochastic games. *Proc. Natl. Acad. Sci. USA* 39(10):1095–1100.
[77] Sidford A, Wang M, Yang L, Ye Y (2020) Solving discounted stochastic two-player games with near-optimal time and sample complexity. Chiappa S, Calandra R, eds., *Proc. Internat. Conf. Artificial Intelligence Statist.* (PMLR), 2992–3002.
[78] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489.
[79] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A (2017) Mastering the game of Go without human knowledge. *Nature* 550(7676):354–359.
[80] Simchowitz M, Jamieson KG (2019) Non-asymptotic gap-dependent regret bounds for tabular MDPs. *Adv. Neural Inform. Processing Systems* 32:1151–1160.
[81] Srinivasan S, Lanctot M, Zambaldi V, Pérolat J, Tuyls K, Munos R, Bowling M (2018) Actor-critic policy optimization in partially observable multiagent environments. *Adv. Neural Inform. Processing Systems* 31:3422–3435.
[82] Strehl AL, Li L, Wiewiora E, Langford J, Littman ML (2006) PAC model-free reinforcement learning. Cohen W, Moore A, eds., *Proc. 23rd Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 881–888.

[83] Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).
[84] Tian Y, Wang Y, Yu T, Sra S (2021) Online learning in unknown Markov games. Meila M, Zhang T, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 10279–10288.
[85] Vershynin R (2012) Introduction to the non-asymptotic analysis of random matrices. Eldar YC, Kutyniok G, eds. *Compressed Sensing* (Cambridge University Press, Cambridge, UK), 210–268.
[86] Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, et al. (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575(7782):350–354.
[87] Wang Y, Wang R, Du SS, Krishnamurthy A (2021) Optimism in reinforcement learning with generalized linear function approximation. *Proc. Internat. Conf. Learn. Representations* (OpenReview).
[88] Watkins CJ, Dayan P (1992) Q-learning. *Machine Learn.* 30:279–292.
[89] Wei CY, Hong YT, Lu CJ (2017) Online reinforcement learning in stochastic games. *Proc. Adv. Neural Inform. Processing Systems* (Neural Information Processing Systems, Long Beach, CA), 4987–4997.
[90] Wei CY, Lee CW, Zhang M, Luo H (2021) Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive Markov games. Preprint, submitted February 8, https://doi.org/10.48550/arXiv.2102.04540.
[91] Wen Z, Van Roy B (2017) Efficient reinforcement learning in deterministic systems with value function generalization. *Math. Oper. Res.* 42(3):762–782.
[92] Xie Q, Chen Y, Wang Z, Yang Z (2020) Learning zero-sum simultaneous-move Markov games using function approximation and correlated equilibrium. Abernethy J, Agarwal S, eds., *Proc. 33rd Conf. Learn. Theory*, vol. 125 (PMLR), 3674–3682.
[93] Xu H, Mannor S (2012) Distributionally robust Markov decision processes. *Math. Oper. Res.* 37(2):288–300.
[94] Yang L, Wang M (2019) Sample-optimal parametric Q-learning using linearly additive features. Chaudhuri K, Salakhutdinov R, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 6995–7004.
[95] Yang L, Wang M (2020) Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. Daume III H, Singh A, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 10746–10756.
[96] Zanette A, Brunskill E (2019) Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. Chaudhuri K, Salakhutdinov R, eds., *Proc. Internat. Conf. Machine Learn.* (PMLR), 7304–7312.
[97] Zanette A, Brandfonbrener D, Brunskill E, Pirotta M, Lazaric A (2020) Frequentist regret bounds for randomized least-squares value iteration. Chiappa S, Calandra R, eds., *Proc. Internat. Conf. Artificial Intelligence Statist.* (PMLR), 1954–1964.
[98] Zhang K, Yang Z, Başar T (2019) Multi-agent reinforcement learning: A selective overview of theories and algorithms. Preprint, submitted November 24, https://doi.org/10.48550/arXiv.1911.10635.
[99] Zhang K, Kakade S, Basar T, Yang L (2020) Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity. *Adv. Neural Inform. Processing Systems* 33:1166–1178.