Minimax-Optimal Multi-Agent RL in Zero-Sum Markov Games With a Generative Model

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper is concerned with two-player zero-sum Markov games — arguably the most basic setting in multi-agent reinforcement learning — with the goal of learning a Nash equilibrium (NE) sample-optimally. All prior results suffer from at least one of the two obstacles: the curse of horizon and that of multiple agents, regardless of the sampling protocol in use. We take a first step towards settling this problem, assuming access to the most flexible sample mechanism: the generative model. Focusing on non-stationary finite-horizon Markov games, we develop a learning algorithm and an adaptive sampling scheme that leverage the optimism principle in adversarial learning (particularly the Follow-the-Regularized-Leader (FTRL) method), with a delicate design of bonus terms that ensure certain decomposability under the FTRL dynamics. Our algorithm learns an ε-approximate Markov NE policy with

$$\widetilde{O}\left(\frac{H^4 S(A+B)}{\varepsilon^2}\right)$$

samples, where S is the number of states, H is the horizon, and A (resp. B) is the number of actions for the max-player (resp. min-player). This is nearly unimprovable in a minimax sense. Along the way, we derive a refined regret bound for FTRL that makes explicit the role of variance-type quantities.

1 Introduction

The thriving field of multi-agent reinforcement learning (MARL) studies how a group of interacting agents makes decisions autonomously in a shared dynamic environment [65]. The recent developments in game playing [57, 8], self-driving vehicles [50], and multi-robot control [40] are prime examples of MARL in action. In practice, there is no shortage of situations where the agents involved have conflict of interest, and they have to act competitively in order to promote their own benefits (possibly at the expense of one another). Scenarios of this kind are frequently modeled via Markov games (MGs) [51, 37], a framework that has been a fruitful playground to formalize and stimulate the studies of competitive MARL.

In view of the irreconcilable competition between individual players, solutions of competitive MARL normally take the form of certain equilibrium states, which are perhaps best epitomized by the concept of Nash equilibrium (NE). In a Nash equilibrium, no gain can be realized through a unilateral change, and hence no player has incentives to deviate from her current strategy/policy (assuming no coordination between players). A myriad of research has been conducted surrounding NEs, which spans various aspects like existence, learnability, computational hardness, and algorithm design, among others [51, 17, 10, 45, 44, 19, 37, 24, 42, 29, 6].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Sample efficiency in zero-sum Markov games. One critical challenge impacting MARL applications is data efficiency. The players involved often have minimal knowledge about how the environment responds to their actions, and have to learn the preferable actions by probing the unknown environment. For MARL to expand into applications with enormous dimensionality and long planning horizon, the learning algorithms must manage to make efficient use of the collected data.

Nevertheless, how to learn NEs with optimal sample complexity remains by and large unsettled, even when it comes to the most basic setting — two-player zero-sum Markov games. In what follows, let us review two representative algorithms developed on this front under two drastically different sampling protocols, and discuss the shortfalls of these cutting-edge results. To facilitate precise comparisons, our discussion concentrates on zero-sum Markov games involving two players, where S is the number of states, H indicates the horizon or effective horizon, and A and B denote respectively the number of actions for the max-player and the min-player.

- *Model-based methods under either a generative model or online exploration.* Assuming access to a generative model (so that one can sample arbitrary state-action combinations), Zhang et al. [64] investigated a natural model-based algorithm, which performs planning (e.g., value iteration) on an empirical MG derived from samples produced by the generative model. Focusing on *stationary* discounted infinite-horizon MGs, their algorithm finds an ε-approximate NE with no more than

$$\widetilde{O}\left(\frac{H^3SAB}{\varepsilon^2}\right) \text{ samples.} \tag{1}$$

  In parallel, Liu et al. [38] studied *non-stationary* finite-horizon MGs with online exploration, and obtained similar sample complexity bounds, i.e.,

$$\widetilde{O}\left(\frac{H^4SAB}{\varepsilon^2}\right) \text{ samples} \qquad \text{or} \qquad \widetilde{O}\left(\frac{H^3SAB}{\varepsilon^2}\right) \text{ episodes} \tag{2}$$

  for learning ε-NEs. While these bounds achieve minimax-optimal dependency on the horizon H, a major drawback emerges as the curse of multiple agents; namely, both results scale with the number AB of *joint actions*, a quantity that blows up exponentially with the number of players.

- *V-learning for online exploration settings.* Focusing on online exploration settings, Bai et al. [4], Jin et al. [27] proposed an algorithm called V-learning that leverages the advances in online adversarial learning (e.g., adversarial bandits) to circumvent the curse of multiple agents. This algorithm provably yields an ε-approximate NE in non-stationary finite-horizon MGs using

$$\widetilde{O}\left(\frac{H^6S(A+B)}{\varepsilon^2}\right) \text{ samples} \qquad \text{or} \qquad \widetilde{O}\left(\frac{H^5S(A+B)}{\varepsilon^2}\right) \text{ episodes,} \tag{3}$$

  which effectively brings down the sample size scaling (2) from AB (i.e., the number of joint actions) to $A+B$ (i.e., the sum of individual actions). Note, however, that this theory appears sub-optimal in terms of the horizon dependency, as it is a factor of H^2 above the minimax limit.

The above results represent the most advanced sample complexity theory thus far for learning NEs in two-player zero-sum MGs. In summary, these state-of-the-art results — irrespective of the sampling mechanism in use — fall short of overcoming at least one of the two major hurdles: (i) the *curse of multiple agents*, and (ii) the *curse of horizon*. A natural question to pose is:

Question: *can we learn Nash equilibria in a two-player zero-sum Markov game in a sample-optimal and computation-efficient fashion?*

To settle this question favorably, both of the above hurdles need to be crossed simultaneously.

Main contributions. Recognizing that the above question remains open regardless of the sampling scheme in use, this paper takes a first step towards solving it assuming access to the most flexible sampler: the generative model (or simulator). In sharp contrast to the single-agent case where uniform sampling of all state-action pairs suffices [2, 34], the multi-agent scenario requires one to take samples intelligently and adaptively, a crucial step to avoid inefficient use of data (otherwise one cannot hope to break the curse of multiple agents). With the aim of computing an ε-approximate NE

in a *non-stationary* finite-horizon two-player zero-sum MG, we come up with a learning algorithm (accompanied by an adaptive sampling strategy) that accomplishes this goal with no more than

$$\widetilde{O}\left(\frac{H^4 S(A+B)}{\varepsilon^2}\right) \tag{4}$$

calls to the generative model. Encouragingly, this sample complexity bound matches the minimax lower limit (up to a logarithmic factor). Our sample complexity theory is valid for the full ε-range (i.e., any $\varepsilon \in (0, H]$); this unveils that no burn-in cost is needed for our algorithm to achieve sample optimality, which lends itself well for sample-hungry applications.

Our algorithm is inspired by two key algorithmic ideas in RL/bandit literature: (1) optimism in the face of uncertainty (by leveraging upper confidence bounds in value estimation), and (2) adversarial learning (particularly the Follow-the-Regularized-Leader (FTRL) algorithm). In particular, the design of data-driven Bernstein-style bonus terms is substantially more challenging than the single-agent case; this requires fine-grained adaptation in response to the policy changes of one another, and needs to ensure decomposability of bonus over steps in a way that is compatible with the FTRL dynamics. Additionally, the policy returned by our algorithm is Markovian (depending only on the current state and step number), and the algorithm can be carried out in a decentralized manner without the need of directly observing the opponent's actions. Along the way, we develop a refined regret bound for FTRL tailored to weighted linear optimization; our regret bound makes explicit the role of variance-style quantities, which is crucial in optimizing the horizon dependency in Markov games; see the supplementary material for this refined FTRL regret bound.

Other related works. Recent years have seen an explosion of interest in learning equilibrium solutions in MARL. While the Nash equilibrium is arguably the most compelling solution concept in Markov games, the finite-sample/finite-time studies of NEs largely concentrate on two-player zero-sum MGs (e.g., [3, 14, 39, 61, 56, 15, 67, 25]), mainly because computing NEs becomes, for the most part, computationally infeasible (i.e., PPAD-complete) when going beyond two-player zero-sum MGs [17, 18]. Roughly speaking, previous NE-finding algorithms for two-player zero-sum Markov games can be categorized into model-based algorithms [44, 64, 38], value-based algorithms [3, 4, 63, 46, 27, 13], and policy-based algorithms [9, 19, 62, 66, 12]. In particular, Bai et al. [4], Jin et al. [27] developed the first algorithms to beat the curse of multiple agents in two-player zero-sum MGs, while Jin et al. [27], Daskalakis et al. [20], Mao and Başar [39], Song et al. [55] further demonstrated how to accomplish the same goal when learning other computationally tractable solution concepts (e.g., coarse correlated equilibria) in general-sum multi-player Markov games.

Additionally, while there are multiple sampling mechanisms (e.g., online exploratory sampling, offline data) that bear practical relevance, the generative model (or simulator) serves as an idealistic sampling protocol that has received much recent attention in reinforcement learning theory, covering the design of various model-based, model-free and policy-based algorithms [31, 60, 1, 2, 34, 28, 52, 58, 59, 35, 30, 53, 43, 32, 11, 22, 62, 21]. When it comes to multi-agent RL, sample-efficient solutions with a generative model have been proposed in the recent works [54, 16, 64].

Notation. Before proceeding, let us gather several notation that will be used throughout. For any positive integer n, we set $[n] := \{1, \cdots, n\}$. We shall abuse notation and let 1 and 0 denote the all-one vector and all-zero vector, respectively. For a sequence $\{\alpha_k\}_{k\geq 1} \subseteq (0, 1]$, we define

$$\alpha_i^k := \begin{cases} \alpha_i \prod_{j=i+1}^{k}(1-\alpha_j), & \text{if } 0 < i < k \\ \alpha_k, & \text{if } i = k \end{cases} \tag{5}$$

for any $1 \leq i \leq k$. Next, consider any two vectors $a = [a_i]_{1\leq i\leq n}$ and $b = [b_i]_{1\leq i\leq n}$. We use $a \leq b$ (resp. $a \geq b$) to indicate $a_i \geq b_i$ (resp. $a_i \leq b_i$) for all i; we allow scalar functions to take vector-valued arguments in order to denote entrywise operations (e.g., $a^2 = [a_i^2]_{1\leq i\leq n}$ and $a^4 = [a_i^4]_{1\leq i\leq n}$); and we denote by $a \circ b = [a_i b_i]_{1\leq i\leq n}$ the Hadamard product. For a finite set $\mathcal{A} = \{1, \cdots, A\}$, we let $\Delta(\mathcal{A}) = \{x \in \mathbb{R}^A \mid \sum_i x_i = 1; x \geq 0\}$ be the probability simplex over $\mathcal{A}$.

2 Background and models

In this section, we introduce the basics for two-player zero-sum Markov games and the solution concept of Nash equilibrum.

Two-player Markov games. A non-stationary finite-horizon two-player Markov game, denoted by $\mathcal{MG} = \big\{\mathcal{S}, \mathcal{A}, \mathcal{B}, H, P, r\big\}$, involves two players competing against each other — with the first one called a "max-player" and the second one called a "min-player" — and consists of several key elements to be formalized below. To begin with, we denote by $\mathcal{S} = \{1, \cdots, S\}$ a shared state space comprising S different states, and let $\mathcal{A} = \{1, \cdots, A\}$ (resp. $\mathcal{B} = \{1, \cdots, B\}$) represent the action space of the max-player (resp. min-player) containing A (resp. B) different actions. The horizon length of this finite-horizon Markov game is denoted by H. The probability transition kernel of $\mathcal{MG}$ is denoted by $P = \{P_h\}_{1\le h\le H}$ with $P_h \in \mathbb{R}^{SAB\times S}$; namely, for any $(s, a, b, h, s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[H]\times\mathcal{S}$, we let $P_h(s' \,|\, s, a, b)$ indicate the probability of $\mathcal{MG}$ transitioning from state s to state s' at step h when the max-player takes action a and min-player takes action b. The function $r = \{r_h\}_{1\le h\le H}$ with $r_h \in \mathbb{R}^{SAB}$ represents the reward function; namely, for any $(s, a, b, h) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[H]$, $r_h(s, a, b)$ stands for the immediate reward gained in state s at step h if the max-player (resp. min-player) executes action a (resp. b). Given that our focal point is the family of zero-sum Markov games, it suffices to introduce a single reward function r. We shall also assume normalized rewards throughout this paper in the sense that $r_h(s, a, b) \in [0, 1]$ for any (s, a, b, h).

Value functions under product policies. Let $\mu = \{\mu_h\}_{1\le h\le H}$ (resp. $\nu = \{\nu_h\}_{1\le h\le H}$) denote the policy of the max-player (resp. min-player), where $\mu_h(\cdot \,|\, s) \in \Delta(\mathcal{A})$ and $\nu_h(\cdot \,|\, s) \in \Delta(\mathcal{B})$ for any $s \in \mathcal{S}$. More specifically, $\mu_h(a \,|\, s)$ indicates the probability of the max-player selecting action a in state s at step h; and $\nu_h(b \,|\, s)$ is defined analogously. Consider a Markovian trajectory $\{(s_t, a_t, b_t, r_t)\}_{1\le t\le H}$, where s_t is the state at time t, a_t (resp. b_t) is the action of the max-player (resp. min-player) at time t, and r_t is the immediate reward observed at time t. For any given policy μ (resp. ν) for the max-player (resp. min-player) and any step h, we define the value function $V_h^{\mu,\nu} : \mathcal{S} \to \mathbb{R}$ under the product policy $\mu \times \nu$ as follows:

$$V_h^{\mu,\nu}(s) \coloneqq \mathbb{E}\left[\sum\nolimits_{t=h}^{H} r_t\big(s_t, a_t, b_t\big) \mid s_h = s\right], \qquad \forall s \in \mathcal{S}, \tag{6}$$

where the expectation is over the Markovian trajectory $\{(s_t, a_t, b_t, r_t)\}$ with the max-player and the min-player executing policies μ and ν, respectively, in an independent fashion; that is, conditional on s_t, we draw $a_t \sim \mu_t(\cdot \,|\, s_t)$ and $b_t \sim \nu_t(\cdot \,|\, s_t)$ *independently*, and then $s_{t+1} \sim P_t(\cdot \,|\, s_t, a_t, b_t)$. In addition, conditional on the min-player executing policy ν, the optimal value function $V^{\star,\nu} = \{V_h^{\star,\nu}\}_{1\le h\le H}$ (with $V_h^{\star,\nu} : \mathcal{S} \to \mathbb{R}$) of the max-player is defined as

$$V_h^{\star,\nu}(s) \coloneqq \max_{\mu:\, \mathcal{S}\times[H]\to\Delta(\mathcal{A})} V_h^{\mu,\nu}(s), \qquad \forall (s, h) \in \mathcal{S}\times[H]; \tag{7a}$$

similarly, when the max-player chooses to execute policy μ, the optimal value function $V^{\mu,\star} = \{V_h^{\mu,\star}\}_{1\le h\le H}$ (with $V_h^{\mu,\star} : \mathcal{S} \to \mathbb{R}$) of the min-player is defined as

$$V_h^{\mu,\star}(s) \coloneqq \min_{\nu:\, \mathcal{S}\times[H]\to\Delta(\mathcal{B})} V_h^{\mu,\nu}(s), \qquad \forall (s, h) \in \mathcal{S}\times[H]. \tag{7b}$$

Furthermore, if we freeze the policy of the min-player to ν, then the Bellman optimality condition for the max-player can be expressed as [7]

$$V_h^{\star,\nu}(s) = \max_{a\in\mathcal{A}} \left\{ \mathop{\mathbb{E}}_{b\sim\nu_h(\cdot|s)} \left[r_h(s, a, b) + \Big\langle P_h(\cdot \,|\, s, a, b),\, V_{h+1}^{\star,\nu} \Big\rangle \right] \right\}, \qquad \forall (s, h) \in \mathcal{S}\times[H]. \tag{8}$$

Solution concept: Nash equilibrium. In the zero-sum two-player Markov game, the max-player wishes to maximize the value function, while the min-player aims to minimize the value function instead. Due to the competing objectives, finding some sort of equilibrium — particularly the Nash equilibrium [41] — becomes a central topic in Markov games. More precisely, a policy pair $(\mu^\star, \nu^\star) \in \Delta(\mathcal{A})^{SH} \times \Delta(\mathcal{B})^{SH}$ is said to be a *(mixed-strategy) Nash equilibrium* of $\mathcal{MG}$ if the resulting product policy $\mu^\star \times \nu^\star$ obeys

$$V_1^{\mu^\star,\nu^\star}(s) = V_1^{\star,\nu^\star}(s) \qquad \text{and} \qquad V_1^{\mu^\star,\nu^\star}(s) = V_1^{\mu^\star,\star}(s), \qquad \text{for all } s \in \mathcal{S}. \tag{9}$$

In other words, conditional on the opponent's current policy and the assumption that two players act *independently*, no player can harvest any gain by unilaterally deviating from its current policy.

In practice, it might be challenging to compute an "exact" Nash equilibrium, and instead one would seek to find approximate solutions. Towards this end, let us first define the sub-optimality gap of a policy pair (μ, ν) as follows (measured in an ℓ_∞-based manner)

$$\mathsf{NE\text{-}gap}(\mu, \nu) \coloneqq \max_{s\in\mathcal{S}} \mathsf{NE\text{-}gap}(\mu, \nu; s), \tag{10a}$$

where

$$\mathsf{NE\text{-}gap}(\mu,\nu;s) \coloneqq \max\Big\{V_1^{\star,\nu}(s)-V_1^{\mu,\nu}(s),\, V_1^{\mu,\nu}(s)-V_1^{\star,\nu}(s)\Big\}. \tag{10b}$$

With this measure in place, a policy pair (μ,ν) is said to be an ε-approximate Nash-equilibrium — or more concisely, ε-Nash — if the resultant sub-optimality gap of the product policy $\mu\times\nu$ obeys $\mathsf{NE\text{-}gap}(\mu,\nu)\le\varepsilon$.

Generative model / simulator. In reality, we often do not have access to perfect descriptions (e.g., accurate knowledge of the transition kernel P) of the Markov game under consideration; instead, one has to learn the true model on the basis of a set of data samples. When it comes to the data generating mechanism, this paper assumes access to a generative model (also called a simulator) [31, 30]: in each call to the generative model, the learner can choose an arbitrary triple $(a,b,h)\in\mathcal{A}\times\mathcal{B}\times[H]$ and obtain an independent sample generated based on the true transition kernel:

$$s'\sim P_h(\cdot\,|\,s,a,b).$$

In words, a generative model facilitates query of arbitrary state-action-step combinations, which helps alleviate the sampling constraints arising in online episodic settings. The goal of the current paper is to compute an ε-approximate Nash equilibrium of $\mathcal{MG}$ with as few samples as possible, i.e., using a minimal number of calls to the generative model.

3 Algorithm and main theory

In this section, we put forward an algorithm that seeks to find an ε-Nash policy pair with the assistance of a generative model, and demonstrate its sample optimality for the full ε-range.

3.1 Algorithm

We now describe the proposed algorithm, which is inspired by the optimism principle and the FTRL algorithm for adversarial learning. Following the dynamic programming approach [7], our algorithm employs backward recursion from step $h=H$ back to $h=1$; in fact, we shall finish estimating the values and policies for step h before moving backward to step $h-1$. For each h, the max-player (resp. min-player) calls the generative model for K rounds, with each round drawing SA (resp. SB) independent samples; as a result, the total sample size is given by $KS(A+B)H$. In what follows, let us first introduce some convenient notation that facilitates our presentation of the algorithm.

Notation. Consider any step $h\in[H]$, and any data collection round $k\in[K]$. The algorithm maintains the following iterates, whose notation is gathered here.

- $\overline{V}_h\in\mathbb{R}^S$ (resp. $\underline{V}_h\in\mathbb{R}^S$) represents the final estimate of the value function at step h by the max-player (resp. min-player); in particular, we set $\overline{V}_{H+1}=\underline{V}_{H+1}=0$.
- $\overline{Q}_h^k\in\mathbb{R}^{SA}$ (resp. $\underline{Q}_h^k\in\mathbb{R}^{SB}$) represents the Q-function estimate of the max-player (resp. min-player) at step h after the k-th round of data collection.
- $\overline{q}_h^k\in\mathbb{R}^{SA}$ (resp. $\underline{q}_h^k\in\mathbb{R}^{SB}$) represents a certain "one-step-look-ahead" Q-function estimate of the max-player (resp. min-player) at step h using samples collected in the k-th round.
- $\overline{r}_h^k\in\mathbb{R}^{SA}$ (resp. $\underline{r}_h^k\in\mathbb{R}^{SB}$) denotes the sample reward vector for step h received by the max-player (resp. min-player) in the k-th round.
- $\overline{P}_h^k\in\mathbb{R}^{SA\times S}$ (resp. $\underline{P}_h^k\in\mathbb{R}^{SB\times S}$) denotes the empirical probability transition matrix for step h constructed using the samples collected by the max-player (resp. min-player) in the k-th round.
- $\overline{\beta}_{h,Q}^k\in\mathbb{R}^{SA}$ (resp. $\underline{\beta}_{h,Q}^k\in\mathbb{R}^{SB}$) denotes the bonus (resp. penalty) vector for step h chosen by the max-player (resp. min-player) in the k-th round for Q-estimation.
- $\overline{\beta}_{h,V}\in\mathbb{R}^S$ (resp. $\underline{\beta}_{h,V}\in\mathbb{R}^S$) denotes the bonus (resp. penalty) vector chosen by the max-player (resp. min-player) for final value estimation.

In addition, for any policies $\mu \in \Delta(\mathcal{A})$ and $\nu \in \Delta(\mathcal{B})$, any vector $f \in \mathbb{R}^A$, and any function $g(\cdot,\cdot)$ with domain $\mathcal{A} \times \mathcal{B}$, we adopt the convenient notation

$$\mathbb{E}_\pi[f] \coloneqq \sum_{a\in\mathcal{A}} \pi(a) f(a) \qquad \text{and} \qquad \mathsf{Var}_\pi(f) \coloneqq \sum_{a\in\mathcal{A}} \pi(a)\big(f(a) - \mathbb{E}_\pi[f]\big)^2, \tag{11a}$$

$$\langle g, \mu\times\nu\rangle \coloneqq \mathop{\mathbb{E}}_{(a,b)\sim\mu\times\nu}\big[g(a,b)\big] = \sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} \mu(a)\nu(b) g(a,b). \tag{11b}$$

Main steps of the proposed algorithm. As mentioned above, our algorithm collects multiple rounds of independent samples for each h. Let us describe below the proposed procedure for the max-player in the k-th round for step h; the procedure for the min-player proceeds analogously.

1. *Sampling and model estimation.* For each $(s,a) \in \mathcal{S}\times\mathcal{A}$, draw an *independent* sample as follows

$$b_{k,h,s,a} \sim \nu_h^k(\cdot \mid s), \qquad s'_{k,h,s,a} \sim P_h(\cdot \mid s, a, b_{k,h,s,a}) \tag{12a}$$

and receive the reward $r_{k,h,s,a} = r_h\big(s,a,b_{k,h,s,a}\big)$. These samples are then employed to construct the sample reward vector $\overline{r}_h^k \in \mathbb{R}^{SA}$ and empirical transition kernel $\overline{P}_h^k \in \mathbb{R}^{SA\times S}$ such that

$$\overline{r}_h^k(s,a) = r_{k,h,s,a} \qquad \text{and} \qquad \overline{P}_h^k(s' \mid s,a) = \begin{cases} 1, & \text{if } s' = s'_{k,h,s,a} \\ 0, & \text{else} \end{cases} \tag{12b}$$

for all $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$. Note that each player only needs to build (12b), which does not require direct observation of the opponent's actions.

2. *Q-function estimation.* Following the dynamic programming approach, we first compute the "one-step-look-ahead" Q-function estimate as follows

$$\overline{q}_h^k = \overline{r}_h^k + \overline{P}_h^k \overline{V}_{h+1}. \tag{13}$$

We then adopt the update rule of synchronous Q-learning with certain bonus terms (via upper confidence bounds) incorporated:

$$\overline{Q}_h^k = (1-\alpha_k)\overline{Q}_h^{k-1} + \alpha_k \overline{q}_h^k + \alpha_k \overline{\beta}_{h,Q}^k, \tag{14}$$

where $0 < \alpha_k < 1$ is the learning rate, and $\overline{\beta}_{h,Q}^k$ is some bonus vector to be specified shortly. Applying (14) recursively and using the quantities defined in (5), we arrive at

$$\overline{Q}_h^k = \sum_{i=1}^k \alpha_i^k \big(\overline{q}_h^i + \overline{\beta}_{h,Q}^i\big). \tag{15}$$

3. *Policy updates.* Once the Q-estimates are updated, we adopt the exponential weights strategy to update the policy iterate of the max-player as follows

$$\mu_h^{k+1}(a \mid s) = \frac{\exp\big(\overline{\eta}_{k+1}(s)\overline{Q}_h^k(s,a)\big)}{\sum_{a'\in\mathcal{A}} \exp\big(\overline{\eta}_{k+1}(s)\overline{Q}_h^k(s,a')\big)}, \qquad \forall (s,a) \in \mathcal{S}\times\mathcal{A}, \tag{16}$$

where $\overline{\eta}_k(s) > 0$ denotes another learning rate associated with policy updates. In fact, this subroutine implements the Follow-the-Regularized-Leader update rule [48]

$$\mu_h^{k+1}(\cdot \mid s) = \arg\min_{\pi\in\Delta(\mathcal{A})} \left\{ -\big\langle \pi, \overline{Q}_h^k(s,\cdot)\big\rangle + \frac{1}{\overline{\eta}_{k+1}} F(\pi) \right\} \tag{17}$$

with the regularizer $F(\cdot)$ chosen to be the negative entropy function $F(\pi) = \sum_a \pi(a)\log\big(\pi(a)\big)$.

After carrying out K rounds of the above procedure, our final policy estimate $\widehat{\mu} \in \Delta(\mathcal{A})^{SH}$ and the value estimate $\overline{V}_h : \mathcal{S} \to \mathbb{R}$ for step h are taken to be

$$\widehat{\mu}_h = \sum\nolimits_{k=1}^K \alpha_k^K \mu_h^k \tag{18a}$$

$$\overline{V}_h(s) = \min\left\{ \sum\nolimits_{k=1}^K \alpha_k^K \big\langle \mu_h^k(\cdot \mid s), \overline{q}_h^k(s,\cdot)\big\rangle + \overline{\beta}_{h,V}(s),\ H-h+1 \right\} \tag{18b}$$

with $\{\alpha_k^K\}$ defined in (5), where $\overline{\beta}_{h,V}(s) \geq 0$ is another bonus term to be specified momentarily. In particular, the final policy (18a) is a mixture of policy iterates and is Markovian in nature (depending only on the current state s and step h). The whole procedure is summarized in Algorithm 1.

Choices of learning rates. Thus far, we have not yet specified the learning rates, which we describe now. The learning rates associated with Q-updates are set to be rescaled linear, namely,

$$\alpha_k = \frac{c_\alpha \log K}{k - 1 + c_\alpha \log K}, \qquad k = 1, 2, \ldots \tag{19}$$

for some constant $c_\alpha \geq 24$. In addition, the learning rates associated with policy updates are chosen in a data-driven manner:

$$\overline{\eta}_{k+1}(s) = \sqrt{\frac{\log K}{\alpha_k \left[H + \sum_{i=1}^{k} \alpha_i^k \mathsf{Var}_{\mu_h^i(\cdot|s)}\left(\overline{q}_h^i(s,\cdot)\right)\right]}}, \qquad \forall s \in \mathcal{S}, \tag{20}$$

where the notation Var_π is introduced in (11a). As will be made clear in the analysis, this choice is carefully tuned based on our refined understanding of FTRL as well as the other learning rates (19).

Choices of bonus terms. It remains to specify the bonus terms, which is selected based on fairly intricate analysis and constitutes a key component of our algorithm design. Specifically, we take

$$\overline{\beta}_{h,Q}^k(s,a) = c_{\mathsf{b}} \sqrt{\frac{\log \frac{K}{\delta}}{KH}} \left\{ \left(\overline{q}_h^k(s,a) - \mathbb{E}_{\mu_h^k(\cdot|s)}\left[\overline{q}_h^k(s,\cdot)\right] \right)^2 + H \right\}, \tag{21a}$$

$$\begin{aligned} \overline{\beta}_{h,V}(s) = c_{\mathsf{b}} \sqrt{\frac{\log \frac{K}{\delta}}{KH}} \sum_{k=1}^{K} \alpha_k^K \Bigg\{ & \mathsf{Var}_{\mu_h^k(\cdot|s)}\left(\overline{q}_h^k(s,\cdot)\right) \\ & + \left(\mathbb{E}_{\mu_h^k(\cdot|s)}\left[\overline{q}_h^k(s,\cdot)\right] - \sum_{k=1}^{K} \alpha_k^K \mathbb{E}_{\mu_h^k(\cdot|s)}\left[\overline{q}_h^k(s,\cdot)\right] \right)^2 + H \Bigg\} \end{aligned} \tag{21b}$$

for any $(s,a,h,k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$, where $c_{\mathsf{b}} > 0$ is some sufficiently large constant. As in previous works, the bonus term, which is chosen carefully in a data-driven fashion, needs to dominate the uncertainty incurred during the estimation process. There are some important points that we would like to highlight (which will be made clear in the analysis).

- A critical design principle is to ensure that the bonus term is, in some sense, decomposable over steps, which is crucial in sharpening the horizon dependency.
- The standard choice of the bonus terms is based on Bernstein-type concentration bounds, which typically entails some standard deviation term. The standard deviation term, however, might be harder to decompose compared to the variance term. To circumvent this difficulty by replacing standard deviation with variance, our design takes advantage of the elementary bound $2\sqrt{\mathsf{Var}(X)} \leq \frac{\mathsf{Var}(X)}{c} + c$, which remains tight if c is taken to be on the typical size of $\sqrt{\mathsf{Var}(X)}$.

3.2 Main results

As it turns out, the proposed algorithm is provably sample-efficient, whose sample complexity is characterized by the following theorem.

Theorem 1. *Consider any $\varepsilon \in (0, H]$ and any $0 < \delta < 1$. Suppose that*

$$K \geq \frac{c_{\mathsf{k}} H^3 \log^4 \frac{KSA}{\delta}}{\varepsilon^2} \tag{24}$$

for some large enough universal constant $c_{\mathsf{k}} > 0$. With probability at least $1 - \delta$, the sub-optimality gap (cf. (10)) of the policies $(\widehat{\mu}, \widehat{\nu})$ returned by Algorithm 1 obeys

$$\mathsf{NE\text{-}gap}\left(\widehat{\mu}, \widehat{\nu}\right) \leq \varepsilon. \tag{25}$$

Theorem 1 establishes a sample complexity upper bound for the proposed algorithm, which we take a moment to interpret as follows.

Algorithm 1: Nash-Q-FTRL.

Input: number of rounds K for each step, learning rates $\{\alpha_k\}$ (see (19)).

Initialize: for all $(s,a,b,h) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[H]$, set

$\overline{V}_{H+1}(s) = \underline{V}_{H+1}(s) = \overline{Q}_h^0(s,a) = \underline{Q}_h^0(s,b) = 0$, $\mu_h^1(a\,|\,s) = 1/A$ and $\nu_h^1(b\,|\,s) = 1/B$.

for $h = H$ **to** 1 **do**

  for $k = 1$ **to** K **do**

    // draw independent samples, and construct empirical models.

    $\big(\overline{r}_h^k, \underline{r}_h^k, \overline{P}_h^k, \underline{P}_h^k\big) \leftarrow \texttt{sampling}\big(h, \mu_h^k, \nu_h^k\big)$. /* see Algorithm 2. */

    // update Q-estimates with upper/lower confidence bounds.

    Compute $\overline{q}_h^k = \overline{r}_h^k + \overline{P}_h^k \overline{V}_{h+1}$, $\underline{q}_h^k = \underline{r}_h^k + \underline{P}_h^k \underline{V}_{h+1}$, and update

$$\overline{Q}_h^k = (1-\alpha_k)\overline{Q}_h^{k-1} + \alpha_k\big(\overline{q}_h^k + \overline{\beta}_{h,Q}^k\big), \quad \underline{Q}_h^k = (1-\alpha_k)\underline{Q}_h^{k-1} + \alpha_k\big(\underline{q}_h^k - \underline{\beta}_{h,Q}^k\big),$$

    where $\overline{\beta}_{h,Q}^k$ is given in (21a), and $\underline{\beta}_{h,Q}^k$ is obtained by replacing $(\mu_h^k, \overline{q}_h^k)$ in (21a) with $(\nu_h^k, \underline{q}_h^k)$.

    // update policy estimates using FTRL.

    for $(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$ **do**

$$\mu_h^{k+1}(a\,|\,s) = \frac{\exp\big(\overline{\eta}_{k+1}(s)\overline{Q}_h^k(s,a)\big)}{\sum_{a'}\exp\big(\overline{\eta}_{k+1}(s)\overline{Q}_h^k(s,a')\big)},$$

$$\nu_h^{k+1}(b\,|\,s) = \frac{\exp\big(-\underline{\eta}_{k+1}(s)\underline{Q}_h^k(s,b)\big)}{\sum_{b'}\exp\big(-\underline{\eta}_{k+1}(s)\underline{Q}_h^k(s,b')\big)},$$

      where $\overline{\eta}_{k+1}$ is defined in (20), and $\underline{\eta}_{k+1}$ can be set by replacing $(\mu_h^i, \overline{q}_h^i)$ in (20) with $(\nu_h^i, \underline{q}_h^i)$.

  // output the final policy estimate and value estimate for step h.

  Update

$$\widehat{\mu}_h = \sum_{k=1}^K \alpha_k^K \mu_h^k, \quad \overline{V}_h(s) = \min\left\{\sum_{k=1}^K \alpha_k^K \big\langle \mu_h^k(\cdot\,|\,s),\, \overline{q}_h^k(s,\cdot)\big\rangle + \overline{\beta}_{h,V}(s),\ H-h+1\right\},\ \forall s,$$

$$\widehat{\nu}_h = \sum_{k=1}^K \alpha_k^K \nu_h^k, \quad \underline{V}_h(s) = \max\left\{\sum_{k=1}^K \alpha_k^K \big\langle \mu_h^k(\cdot\,|\,s),\, \underline{q}_h^k(s,\cdot)\big\rangle - \underline{\beta}_{h,V}(s),\ 0\right\},\ \forall s,$$

  where $\overline{\beta}_{h,V}^k$ is in (21b), and $\underline{\beta}_{h,V}^k$ is obtained by replacing $(\mu_h^k, \overline{q}_h^k)$ in (21b) with $(\nu_h^k, \underline{q}_h^k)$.

Output: $\widehat{\mu} = \{\widehat{\mu}_h\}_{1\le h\le H}$ and $\widehat{\nu} = \{\widehat{\nu}_h\}_{1\le h\le H}$.

Sample complexity. When a generative model is available, Theorem 1 asserts that the total number of samples $KS(A+B)H$ needed for Algorithm 1 to yield ε-Nash policies is at most

$$\text{(sample complexity)} \qquad \widetilde{O}\left(\frac{H^4 S(A+B)}{\varepsilon^2}\right). \tag{26}$$

This delivers the first result — for any sampling protocol — that simultaneously overcomes the curse of horizon and that of multiple agents. In comparison to Zhang et al. [64] (cf. (1)), this result reveals that what ultimately matters is the total number of individual actions (i.e., $A+B$) as opposed to the total number AB of joint actions. Additionally, our result exhibits improved horizon dependency (by a factor of H^2) compared to Bai et al. [4], Jin et al. [27] (see (3)), although we should emphasize that the online sampling protocol therein is more restrictive than a generative model.

Minimax optimality. To assess the tightness of our sample complexity bound (26), it is helpful to look at the minimax lower limit. Suppose without loss of generality that $A \ge B$. Consider the extreme scenario where $B = 1$, in which case the Nash-equilibrium reduces to the optimal

Algorithm 2: Auxiliary function $\texttt{sampling}\big(h, \mu_h, \nu_h\big)$.

Initialize: $\overline{r} = 0 \in \mathbb{R}^{SA}$, $\underline{r} = 0 \in \mathbb{R}^{SB}$, $\overline{P} = 0 \in \mathbb{R}^{SA\times S}$, and $\underline{P} = 0 \in \mathbb{R}^{SB\times S}$.

for $(s,a) \in \mathcal{S}\times\mathcal{A}$ **do**

Draw an independent sample from the generative model:

$$b_{h,s,a} \sim \nu_h(\cdot \mid s), \qquad s'_{h,s,a} \sim P_h(\cdot \mid s, a, b_{h,s,a}). \tag{22}$$

set $\overline{r}(s,a) = r_h\big(s,a,b_{h,s,a}\big)$ and $\overline{P}\big(s'_{h,s,a} \mid s,a\big) = 1$.

for $(s,b) \in \mathcal{S}\times\mathcal{B}$ **do**

Draw an independent sample from the generative model:

$$a_{h,s,b} \sim \mu_h(\cdot \mid s), \qquad s'_{h,s,a} \sim P_h(\cdot \mid s, a_{h,s,b}, b). \tag{23}$$

set $\underline{r}(s,b) = r_h\big(s,a_{h,s,b},b\big)$ and $\underline{P}\big(s'_{h,s,a} \mid s,b\big) = 1$.

Return: $\big(\overline{r}, \underline{r}, \overline{P}, \underline{P}\big)$.

policy of a single-agent MDP with S states and A actions. It is well-known that for any accuracy level $\varepsilon \in (0, H]$, one can construct a non-stationary MDP with S states and A actions such that no algorithm can learn an ε-optimal policy with fewer than $\widetilde{O}\big(\frac{H^4SA}{\varepsilon^2}\big)$ samples [2, 36]. This means that in general, the minimax lower bound (w.r.t. finding an ε-Nash policy pair) scales as

$$\text{(minimax lower bound)} \qquad \frac{H^4S(A+B)}{\varepsilon^2} \tag{27}$$

modulo some logarithmic factor. Taking this together with (26) confirms the minimax optimality of our algorithm (up to logarithmic terms).

No burn-in sample size and full ε-range. It is noteworthy that the validity of our sample complexity bound (26) is guaranteed for the entire range of ε-levels (i.e., any $\varepsilon \in (0, H]$). This feature is particularly appealing in the data-starved applications, as it implies that there is no burn-in sample size needed for our algorithm to work optimally.

Miscellaneous properties of our algorithm. Finally, we would like to remark in passing that our learning algorithm enjoys several properties that are practically appealing. For instance, the algorithm can be carried out in a decentralized fashion, with two players acting in a symmetric and independent manner; the algorithm is "rational" in the sense that it converges to the best-response policy if one of the players freezes its policy; and the output policies are inherently Markovian, which depend only on the current state and step number. All this is achieved under minimal sample complexity with the aid of the generative model.

4 Discussion

The primary contribution of this paper has been to develop a sample-optimal strategy that simultaneously overcome the curse of multiple agents and optimize the horizon dependency when solving two-player zero-sum Markov games. This goal was not accomplished in any of the previous works, regardless of the sampling mechanism in use. The adoption of the adversarial learning subroutine helps break the curse of multiple agents compared to the prior model-based approach [64, 38], whereas the availability of the generative model in conjunction with the delicate bonus design enables sharpened horizon dependency compared to Bai et al. [4], Jin et al. [27]. Our work opens further questions surrounding sample efficiency in solving Markov games. For instance, how to attain minimax-optimal sample complexity if we only have access to online sampling protocols (as opposed to the flexible generative model); and how can we optimize the horizon dependency when computing coarse correlated equilibria in multi-agent scenarios [55, 27, 20] without compromising the dependency on the size of the action spaces. In addition, our refined regret bound for FTRL (based on variance-type quantities) only covers the full-information case; it would be of interest to generalize it to partial-information setting (where only partial entries of the loss vectors are observable each time).

References

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

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

[3] Y. Bai and C. Jin. Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning*, pages 551–560. PMLR, 2020.

[4] Y. Bai, C. Jin, and T. Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

[5] A. Beck. *First-order methods in optimization*. SIAM, 2017.

[6] C. L. Beck and R. Srikant. Error bounds for constant step-size Q-learning. *Systems & control letters*, 61(12):1203–1208, 2012.

[7] D. P. Bertsekas. *Dynamic programming and optimal control (4th edition)*. Athena Scientific, 2017.

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

[9] S. Cen, Y. Wei, and Y. Chi. Fast policy extragradient methods for competitive games with entropy regularization. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] X. Chen, Y. Cheng, and B. Tang. Well-supported versus approximate Nash equilibria: Query complexity of large games. *arXiv preprint arXiv:1511.00785*, 2015.

[11] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam. Finite-sample analysis of stochastic approximation using smooth convex envelopes. *arXiv preprint arXiv:2002.00874*, 2020.

[12] Z. Chen, S. Ma, and Y. Zhou. Sample efficient stochastic policy extragradient algorithm for zero-sum Markov game. In *International Conference on Learning Representations*, 2021.

[13] Z. Chen, D. Zhou, and Q. Gu. Almost optimal algorithms for two-player Markov games with linear function approximation. *arXiv preprint arXiv:2102.07404*, 2021.

[14] Z. Chen, D. Zhou, and Q. Gu. Almost optimal algorithms for two-player zero-sum linear mixture Markov games. In *International Conference on Algorithmic Learning Theory*, pages 227–261. PMLR, 2022.

[15] Q. Cui and S. S. Du. When is offline two-player zero-sum Markov game solvable? *arXiv preprint arXiv:2201.03522*, 2022.

[16] Q. Cui and L. F. Yang. Minimax sample complexity for turn-based stochastic game. In *Uncertainty in Artificial Intelligence*, pages 1496–1504. PMLR, 2021.

[17] C. Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):1–35, 2013.

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

[19] C. Daskalakis, D. J. Foster, and N. Golowich. Independent policy gradient methods for competitive reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5527–5540, 2020.

[20] C. Daskalakis, N. Golowich, and K. Zhang. The complexity of markov equilibrium in stochastic games. *arXiv preprint arXiv:2204.03991*, 2022.

[21] S. S. Du, S. M. Kakade, R. Wang, and L. F. Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

[22] E. Even-Dar and Y. Mansour. Learning rates for Q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

[23] D. A. Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

[24] T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

[25] Z. Jia, L. F. Yang, and M. Wang. Feature-based q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*, 2019.

[26] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

[27] C. Jin, Q. Liu, Y. Wang, and T. Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent RL. *arXiv preprint arXiv:2110.14555*, 2021.

[28] Y. Jin and A. Sidford. Towards tight bounds on the sample complexity of average-reward MDPs. In *International Conference on Machine Learning*, pages 5055–5064. PMLR, 2021.

[29] Y. Jin, V. Muthukumar, and A. Sidford. The complexity of infinite-horizon general-sum stochastic games. *arXiv preprint arXiv:2204.04186*, 2022.

[30] S. Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London, 2003.

[31] M. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine learning*, 49(2-3):193–208, 2002.

[32] K. Khamaru, A. Pananjady, F. Ruan, M. J. Wainwright, and M. I. Jordan. Is temporal difference learning optimal? an instance-dependent analysis. *SIAM Journal on Mathematics of Data Science*, 3(4):1013–1040, 2021.

[33] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[34] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[35] G. Li, C. Cai, Y. Chen, Y. Gu, Y. Wei, and Y. Chi. Is Q-learning minimax optimal? a tight sample complexity analysis. *arXiv preprint arXiv:2102.06548*, 2021.

[36] G. Li, L. Shi, Y. Chen, Y. Chi, and Y. Wei. Settling the sample complexity of model-based offline reinforcement learning. *arXiv preprint arXiv:2204.05275*, 2022.

[37] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[38] Q. Liu, T. Yu, Y. Bai, and C. Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021.

[39] W. Mao and T. Başar. Provably efficient reinforcement learning in decentralized general-sum Markov games. *Dynamic Games and Applications*, pages 1–22, 2022.

[40] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

[41] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[42] A. Ozdaglar, M. O. Sayin, and K. Zhang. Independent learning in stochastic games. *arXiv preprint arXiv:2111.11743*, 2021.

[43] A. Pananjady and M. J. Wainwright. Instance-dependent ℓ_∞-bounds for policy evaluation in tabular reinforcement learning. *IEEE Transactions on Information Theory*, 67(1):566–585, 2020.

[44] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin. Approximate dynamic programming for two-player zero-sum Markov games. In *International Conference on Machine Learning*, pages 1321–1329. PMLR, 2015.

[45] A. Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 258–265, 2016.

[46] M. Sayin, K. Zhang, D. Leslie, T. Basar, and A. Ozdaglar. Decentralized Q-learning in zero-sum Markov games. *Advances in Neural Information Processing Systems*, 34, 2021.

[47] S. Shalev-Shwartz. Online learning: Theory, algorithms, and applications. 2007.

[48] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[49] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2):115–142, 2007.

[50] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[51] L. S. Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

[52] A. Sidford, M. Wang, X. Wu, L. Yang, and Y. Ye. Near-optimal time and sample complexities for solving Markov decision processes with a generative model. In *Advances in Neural Information Processing Systems*, pages 5186–5196, 2018.

[53] A. Sidford, M. Wang, X. Wu, and Y. Ye. Variance reduced value iteration and faster algorithms for solving Markov decision processes. In *Symposium on Discrete Algorithms*, pages 770–787. SIAM, 2018.

[54] A. Sidford, M. Wang, L. Yang, and Y. Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pages 2992–3002. PMLR, 2020.

[55] Z. Song, S. Mei, and Y. Bai. When can we learn general-sum Markov games with a large number of players sample-efficiently? *arXiv preprint arXiv:2110.04184*, 2021.

[56] Y. Tian, Y. Wang, T. Yu, and S. Sra. Online learning in unknown markov games. In *International conference on machine learning*, pages 10279–10288. PMLR, 2021.

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

[58] M. J. Wainwright. Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning. *arXiv preprint arXiv:1905.06265*, 2019.

[59] M. J. Wainwright. Variance-reduced Q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*, 2019.

[60] B. Wang, Y. Yan, and J. Fan. Sample-efficient reinforcement learning for linearly-parameterized mdps with a generative model. *Advances in Neural Information Processing Systems*, 34, 2021.

[61] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Online reinforcement learning in stochastic games. *Advances in Neural Information Processing Systems*, 30, 2017.

[62] C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive Markov games. In *Conference on Learning Theory*, pages 4259–4299. PMLR, 2021.

[63] Q. Xie, Y. Chen, Z. Wang, and Z. Yang. Learning zero-sum simultaneous-move Markov games using function approximation and correlated equilibrium. In *Conference on Learning Theory*, pages 3674–3682. PMLR, 2020.

[64] K. Zhang, S. Kakade, T. Basar, and L. Yang. Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity. *Advances in Neural Information Processing Systems*, 33, 2020.

[65] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

[66] Y. Zhao, Y. Tian, J. D. Lee, and S. S. Du. Provably efficient policy gradient methods for two-player zero-sum Markov games. *arXiv preprint arXiv:2102.08903*, 2021.

[67] H. Zhong, W. Xiong, J. Tan, L. Wang, T. Zhang, Z. Wang, and Z. Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. *arXiv preprint arXiv:2202.07511*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] This is a theoretical work that we do not foresee any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...