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

Gen Li UPenn
Yuejie Chi CMU
Yuting Wei UPenn
Yuxin Chen UPenn

Abstract

This paper studies multi-agent reinforcement learning in Markov games, with the goal of learning Nash equilibria or coarse correlated equilibria (CCE) sample-optimally. All prior results suffer from at least one of the two obstacles: the curse of multiple agents and the barrier of long horizon, regardless of the sampling protocol in use. We take a step towards settling this problem, assuming access to a flexible sampling mechanism: the generative model. Focusing on non-stationary finite-horizon Markov games, we develop a fast learning algorithm called Q-FTRL and an adaptive sampling scheme that leverage the optimism principle in online adversarial learning (particularly the Follow-the-Regularized-Leader (FTRL) method). Our algorithm learns an ε-approximate CCE in a general-sum Markov game using

$$\widetilde{O}\left(\frac{H^4 S \sum_{i=1}^{m} A_i}{\varepsilon^2}\right)$$

samples, where m is the number of players, S indicates the number of states, H is the horizon, and A_i denotes the number of actions for the i-th player. This is minimax-optimal (up to log factor) when m is fixed. When applied to two-player zero-sum Markov games, our algorithm provably finds an ε-approximate Nash equilibrium with a minimal number of samples. Along the way, we derive a refined regret bound for FTRL that makes explicit the role of variance-type quantities, which might be of independent interest.

1 Introduction

The thriving field of multi-agent reinforcement learning (MARL) studies how a group of interacting agents make decisions autonomously in a shared dynamic environment [80]. The recent developments in game playing [66, 9], self-driving vehicles [58], and multi-robot control [45] 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) [59, 42], 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 strategy profiles, which are perhaps best epitomized by the concept of Nash equilibrium (NE) [49]. In a Nash equilibrium, no gain can be realized through a unilateral change — assuming no coordination between players — and hence no player has incentives to deviate from her current strategy/policy. A myriad of research has been conducted surrounding NE, which spans various aspects like existence, learnability, computational hardness, and algorithm design, among others [59, 20, 12, 53, 52, 22, 42, 28, 50, 33]. Given that finding NE is notoriously expensive in general (except for special cases like two-player zero-sum MGs) [20, 21], several more tractable solution concepts have emerged in the studies of game theory and MARL, a

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-1117

prominent example being the coarse correlated equilibrium (CCE) [47]. A key compromise made in the CCE is that it permits the players to act in an coordinated fashion, which contrasts sharply with the absence of coordination in the definition of NE.

One critical challenge impacting modern 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 dynamics and 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 NE and/or CCE with optimal sample complexity remains by and large unsettled even when it comes to the most basic setting: two-player zero-sum Markov games, as we shall discuss below.

Example: inadequacy in learning two-player zero-sum Markov games. To facilitate concrete comparisons, let us review two representative algorithms aimed at learning NE in two-player zero-sum MGs. These algorithms have been studied under two drastically different sampling protocols, and we shall discuss the shortfalls of the cutting-edge sample complexity results. In a two-player zero-sum MG, we denote by S the number of states and H the horizon or effective horizon, whereas A_1 and A_2 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 tuples), Zhang et al. [79] investigated a natural model-based algorithm, which performs planning (e.g., value iteration) on an empirical MG derived from samples produced non-adaptively by the generative model. Focusing on *stationary* discounted infinite-horizon MGs, their algorithm finds an ε-approximate NE with
$$\widetilde{O}\left(\frac{H^3SA_1A_2}{\varepsilon^2}\right) \text{ samples.} \tag{1}$$
In parallel, Liu et al. [43] studied *non-stationary* finite-horizon MGs with online exploration, and obtained similar sample complexity bounds, i.e.,
$$\widetilde{O}\left(\frac{H^4SA_1A_2}{\varepsilon^2}\right) \text{ samples} \quad \text{or} \quad \widetilde{O}\left(\frac{H^3SA_1A_2}{\varepsilon^2}\right) \text{ episodes} \tag{2}$$
for learning an ε-approximate NE. While these bounds achieve minimax-optimal dependency on the horizon H, a major drawback emerges — commonly referred to as the curse of multiple agents; namely, these results scale proportionally with the total number of *joint actions* (i.e., $\prod_{1\le i\le 2} A_i$), 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. [5], Jin et al. [31] 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
$$\widetilde{O}\left(\frac{H^6S(A_1+A_2)}{\varepsilon^2}\right) \text{ samples} \quad \text{or} \quad \widetilde{O}\left(\frac{H^5S(A_1+A_2)}{\varepsilon^2}\right) \text{ episodes,} \tag{3}$$
which effectively brings down the sample size scaling (2) from A_1A_2 (i.e., the number of joint actions) to A_1+A_2 (i.e., the sum of individual actions). It is worth pointing out, however, that this theory appears sub-optimal in terms of the horizon dependency, as it is a factor of H^2 above the minimax lower bound.

Key issues and our main contributions. While the above summary focuses on two-player zero-sum MGs, it unveils a fundamental issue surrounding the sample efficiency of learning equilibria; that is, all existing results in this front — 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 *barrier of long horizon*. A natural question to pose is:

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

To settle this favorably, both of the above hurdles need to be crossed simultaneously. Moving beyond two-player zero-sum MGs, it is not surprising to see that general-sum multi-player MGs have to grapple with the aforementioned two hurdles as well. Thus, the following question also comes into mind when learning CCE (a compromise due to the general intractability of learning NE):

Question: *can we learn a coarse correlated equilibrium in a multi-player general-sum Markov game in a sample-optimal and computation-efficient fashion?*

Note that these questions remain open regardless of the sampling scheme in use.

This paper takes a first step towards solving the problem by assuming access to the most flexible sampling protocol: the generative model (or simulator). In stark contrast to the single-agent case where uniform sampling of all state-action pairs suffices [3, 38], 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 equilibrium in a *non-stationary* finite-horizon MG, we come up with a computationally efficient learning algorithm (accompanied by an adaptive sampling strategy) that accomplishes this goal with no more than

$$\begin{cases} \widetilde{O}\Big(\frac{H^4 S(A_1+A_2)}{\varepsilon^2}\Big) \text{ samples} & \text{(learning } \varepsilon\text{-NE in two-player zero-sum MGs)} \\ \widetilde{O}\Big(\frac{H^4 S\left(\sum_{i=1}^m A_i\right)}{\varepsilon^2}\Big) \text{ samples} & \text{(learning } \varepsilon\text{-CCE in multi-player general-sum MGs)} \end{cases} \tag{4}$$

drawn from the generative model. Encouragingly, this sample complexity bound matches the minimax lower limit (up to a logarithmic factor) as long as the number of players $m \geq 2$ is a fixed constant or grows only logarithmically in problem parameters. 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 to sample-hungry applications.

The proposed algorithm is inspired by two key algorithmic ideas in RL and bandit literature: (i) optimism in the face of uncertainty (by leveraging upper confidence bounds (UCBs) in value estimation), and (ii) online and adversarial learning (particularly the Follow-the-Regularized-Leader (FTRL) algorithm). Note that the optimal design of bonus terms — typically based on certain data-driven variance estimates — is substantially more challenging than the single-agent case, as it requires intricate adaptation in response to the policy changes of one another as well as compatibility with the FTRL dynamics. Two points are worth emphasizing (which will be made precise later on):

- The efficacy of FTRL in breaking the curse of multiple agents has been illustrated in Jin et al. [31], Song et al. [63], Mao and Başar [44]. To improve horizon dependency, one needs to exploit connections between the performance of FTRL and certain variances. Towards this, we develop a refined regret bound for FTRL that unveils the role of variance-style quantities, which was previously unavailable.
- The bonus terms entail Bernstein-style variance estimates that mimic the variance-style quantities appearing in our refined FTRL regret bounds, and are carefully chosen so as to ensure certain decomposability over steps. This is crucial in optimizing the horizon dependency.

Additionally, the policy returned by our algorithm is Markovian (i.e., the action selection probability depends only on the current state s and step h), and the algorithm can be carried out in a decentralized manner without the need of directly observing the opponents' actions.

Notation. Let us also gather several convenient notation that shall be used multiple times. For any positive integer n, we write $[n] \coloneqq \{1, \cdots, n\}$. We shall abuse notation and let 1 and 0 denote the all-one vector and the all-zero vector, respectively. For a sequence $\{\alpha_k\}_{k\geq 1} \subseteq (0, 1]$, we define

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

for any $1 \leq i \leq k$. For a given vector $x \in \mathbb{R}^{SA}$ (resp. $y \in \mathbb{R}^{SAB}$), we denote by $x(s,a)$ (resp. $y(s,a,b)$) the entry of x (resp. y) associated with the state-action combination (s,a) (resp. (s,a,b)), as long as it is clear from the context. 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 that $a_i \geq b_i$ (resp. $a_i \leq b_i$) holds 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 denote by $\Delta(\mathcal{A}) = \{x \in \mathbb{R}^A \mid \sum_i x_i = 1; x \geq 0\}$ the probability simplex over $\mathcal{A}$. For any function f with domain $\mathcal{A}$ (or $\mathcal{B}$), we adopt the notation

$$\mathbb{E}_\pi[f] \coloneqq \sum\nolimits_a \pi(a) f(a) \qquad \text{and} \qquad \mathsf{Var}_\pi(f) \coloneqq \sum\nolimits_a \pi(a)\big(f(a) - \mathbb{E}_\pi[f]\big)^2. \tag{6}$$

https://doi.org/10.52202/068431-1117

2 Background and models

In this section, we introduce the basics for Markov games, as well as the solution concepts of Nash equilibrium and coarse correlated equilibrium.

Markov games. A non-stationary finite-horizon *multi-player general-sum Markov game*, denoted by $\mathcal{MG} = \big\{\mathcal{S}, \{\mathcal{A}_i\}_{1\le i\le m}, H, P, r\big\}$, involves m players competing against each other, and consists of several key elements to be formalized below. Recall that $\Delta(\mathcal{S})$ represents the probability simplex over the set $\mathcal{S}$.

- $\mathcal{S} = \{1, \cdots, S\}$ is the state space of the shared environment, which comprises S different states.
- For each $1 \le i \le m$, let $\mathcal{A}_i = \{1, \cdots, A_i\}$ represent the action space of the i-th player, which contains A_i different actions. Here and below, we denote

$$\mathcal{A} \coloneqq \mathcal{A}_1 \times \cdots \times \mathcal{A}_m \qquad \text{and} \qquad \mathcal{A}_{-i} \coloneqq \prod_{j:j\neq i} \mathcal{A}_j \quad (1 \le i \le m). \tag{7}$$

  Throughout the paper, we shall often use the boldface letter $\boldsymbol{a} \in \mathcal{A}$ (resp. $\boldsymbol{a}_{-i} \in \mathcal{A}_{-i}$) to denote a joint action profile of all players (resp. a joint action profile excluding the i-th player's action).
- H stands for the horizon length of the Markov game.
- $P = \{P_h\}_{1\le h\le H}$ — with $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ — denotes the probability transition kernel of $\mathcal{MG}$. Namely, for any $(s, \boldsymbol{a}, h, s') \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{S}$, we let $P_h(s' \mid s, \boldsymbol{a})$ indicate the probability of $\mathcal{MG}$ transitioning from state s to state s' at step h when the joint action profile taken by the players is $\boldsymbol{a}$.
- $r = \{r_{i,h}\}_{1\le h\le H, 1\le i\le m}$ — with $r_{i,h} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ — represents the (deterministic) reward function. Namely, for any $(s, \boldsymbol{a}, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, $r_{i,h}(s, \boldsymbol{a})$ stands for the immediate reward the i-th player gains in state s at step h, if the joint action profile is $\boldsymbol{a}$. Here and throughout, we assume normalized rewards in the sense that $r_{i,h}(s, \boldsymbol{a}) \in [0, 1]$ for any $(s, \boldsymbol{a}, h, i) \in \mathcal{S} \times \mathcal{A} \times [H] \times [m]$.

As an important special case, a *two-player zero-sum Markov game* — denoted by $\mathcal{MG} = \big\{\mathcal{S}, \{\mathcal{A}_1, \mathcal{A}_2\}, H, P, r\big\}$ — satisfies $r_{2,h} = -r_{1,h}$ for all $h \in [H]$. Following the convention, we assume that $r_{1,h} \ge 0$ for all $h \in [H]$,[1] and refer to the first (resp. second) player as the max-player (resp. the min-player).

Markov policies. This paper focuses on the class of Markov policies, such that the action selection strategies of the players are determined by the current state s and the step number h, without depending on previously visited states. To begin with, let $\pi_i = \{\pi_{i,h}\}_{1\le h\le H}$ represent the policy of the i-th player. Here, $\pi_{i,h}(\cdot \mid s) \in \Delta(\mathcal{A}_i)$ for any $(s, h) \in \mathcal{S} \times [H]$, where $\pi_{i,h}(a \mid s)$ indicates the probability of the i-th player selecting action a in state s at step h. The joint Markov policy can be defined analogously: we let $\pi = (\pi_1, \ldots, \pi_m) : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$ represent a joint Markov policy of all players, where the joint actions of all players in state s and step h are chosen according to the distribution specified by $\pi_h(\cdot \mid s) = (\pi_{1,h}, \ldots, \pi_{m,h})(\cdot \mid s) \in \Delta(\mathcal{A})$. For any given joint policy π, we employ π_{-i} to represent the policies of all but the i-th player, and let $\pi_{-i,h}$ denote the policies of all but the i-th player at step h. All policies are assumed throughout to be Markovian, except our brief remarks on non-Markovian policies in Section 3.2.

Additionally, a joint policy π is said to be a *product policy* if $\pi_1, \ldots, \pi_m$ are executed in a statistically independent manner (namely, under policy π the players take actions independently), and we shall adopt the notation $\pi = \pi_1 \times \cdots \times \pi_m$ to indicate that π is a product policy.

Value functions. Consider a Markovian trajectory $\{(s_h, \boldsymbol{a}_h)\}_{1\le h\le H}$, where $s_h \in \mathcal{S}$ is the state at step h and $\boldsymbol{a}_h \in \mathcal{A}$ is the joint action profile at step h. For any given joint policy π and any step $h \in [H]$, we define the value function $V_{i,h}^{\pi} : \mathcal{S} \to \mathbb{R}$ of the i-th player under policy π as follows:

$$V_{i,h}^{\pi}(s) \coloneqq \mathbb{E}\left[\sum_{t=h}^{H} r_{i,t}\big(s_t, \boldsymbol{a}_t\big) \mid s_h = s\right], \qquad \forall s \in \mathcal{S}, \tag{8}$$

[1] The careful reader might immediately note that $r_{2,h} \le 0$, thus falling outside our assumed range for the reward function. This, however, can be easily addressed by enforcing a positive global shift to $r_{2,h}$ without changing the learning process.

https://doi.org/10.52202/068431-1117

where the expectation is taken over the Markovian trajectory $\{(s_h, \boldsymbol{a}_h)\}$ with the m players jointly executing policy π; that is, conditional on s_h, we draw $\boldsymbol{a}_h \sim \pi_h(\cdot \,|\, s_h)$ and then $s_{h+1} \sim P_h(\cdot \,|\, s_h, \boldsymbol{a}_h)$.

In addition, consider the case where (i) all but the i-th player executes the joint policy π_{-i} and (ii) the i-th player executes policy π_i' *independently* from the other players; we shall denote by $V_{i,h}^{\pi_i' \times \pi_{-i}}$ the resulting value function under this joint policy $\pi_i' \times \pi_{-i}$. By optimizing over all π_i', we can further define

$$V_{i,h}^{\star,\pi_{-i}}(s) \coloneqq \max_{\pi_i' : \mathcal{S} \times [H] \to \Delta(\mathcal{A}_i)} V_{i,h}^{\pi_i' \times \pi_{-i}}(s), \qquad \forall (s,h,i) \in \mathcal{S} \times [H] \times [m]. \tag{9}$$

It is known that there exists at least one policy, denoted by $\pi_i^\star(\pi_{-i}) : \mathcal{S} \times [H] \to \Delta(\mathcal{A}_i)$ and commonly referred to as the *best-response policy*, that can simultaneously attain $V_{i,h}^{\star,\pi_{-i}}(s)$ for all $h \in [H]$ and all $s \in \mathcal{S}$. It is worth emphasizing that the best-response policy $\pi_i^\star(\pi_{-i})$ is the best among all policies of the i-th player executed independently of π_{-i}. Furthermore, if we freeze π_{-i}, then the Bellman optimality condition for the i-th player can be expressed as [8]

$$V_{i,h}^{\star,\pi_{-i}}(s) = \max_{a_i \in \mathcal{A}_i} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{a}_{-i} \sim \pi_{-i,h}(\cdot|s)} \left[r_{i,h}(s, \boldsymbol{a}) + \left\langle P_h(\cdot \,|\, s, \boldsymbol{a}),\, V_{i,h+1}^{\star,\pi_{-i}} \right\rangle \right] \right\} \tag{10}$$

for all $(s,h,i) \in \mathcal{S} \times [H] \times [m]$, where the joint action profile $\boldsymbol{a}$ is composed of a_i for the i-th player and $\boldsymbol{a}_{-i}$ for the remaining ones.

Equilibria of Markov games. In a multi-agent Markov game, each player wishes to maximize its own value function. Due to the competing objectives, finding some sorts of equilibria — e.g., the Nash equilibrium [48] and the coarse correlated equilibrium [47, 2] — becomes a central topic in the studies of Markov games. Let us introduce these solution concepts below.

- *Nash equilibrium.* A product policy $\pi = \pi_1 \times \cdots \times \pi_m$ is said to be a *(mixed-strategy) Nash equilibrium* of $\mathcal{MG}$ if the following holds:

  $$V_{i,1}^{\pi}(s) = V_{i,1}^{\star,\pi_{-i}}(s), \qquad \text{for all } (s,i) \in \mathcal{S} \times [m]. \tag{11}$$

  In other words, conditional on the opponents' current policy and the assumption that all players take actions *independently*, no player can harvest any gain by unilaterally deviating from its current policy.

- *Coarse correlated equilibrium.* A joint policy π is said to be a coarse correlated equilibrium of $\mathcal{MG}$ if

  $$V_{i,1}^{\pi}(s) \geq V_{i,1}^{\star,\pi_{-i}}(s), \qquad \text{for all } (s,i) \in \mathcal{S} \times [m]. \tag{12}$$

  While a CCE also ensures that no unilateral deviation (performed independently from others) is beneficial, its key distinction from the definition of NE lies in the fact that it allows the policy to be correlated across the players. Any NE of $\mathcal{MG}$ is, self-evidently, also a CCE.

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

$$\mathsf{gap}(\pi) \coloneqq \max_{s \in \mathcal{S}} \mathsf{gap}(\pi; s), \tag{13a}$$

where

$$\mathsf{gap}(\pi; s) \coloneqq \max_{1 \leq i \leq m} \left\{ V_{i,1}^{\star,\pi_{-i}}(s) - V_{i,1}^{\pi}(s) \right\}. \tag{13b}$$

With this sub-optimality measure in place, a *product* policy $\pi = \pi_1 \times \cdots \times \pi_m$ is said to be an ε-approximate NE — or more concisely, ε-Nash — if the resultant sub-optimality gap obeys $\mathsf{gap}(\pi) \leq \varepsilon$. Similarly, a joint (and possibly correlated) policy π is said to be an ε-approximate CCE — or more concisely, ε-CCE — if $\mathsf{gap}(\pi) \leq \varepsilon$.

Generative model / simulator. In reality, we oftentimes 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 data samples. When it comes to the data generating mechanism, this paper assumes access to a generative model (also called a simulator) [35, 34]:

https://doi.org/10.52202/068431-1117

in each call to the generative model, the learner can choose an arbitrary $(s, \boldsymbol{a}, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ and obtain an independent sample generated based on the true transition kernel:

$$s' \sim P_h(\cdot \,|\, s, \boldsymbol{a}).$$

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

3 Sample-efficient learning with a generative model

In this section, we put forward an efficient algorithm aimed at learning an ε-approximate equilibrium with the assistance of a generative model, and demonstrate its sample optimality for the full ε-range.

3.1 Algorithm description

We now describe the proposed algorithm, which is inspired by the optimism principle and the FTRL algorithm for online/adversarial learning. Following the dynamic programming approach [8], our algorithm employs backward recursion from step $h = H$ back to $h = 1$; in fact, we shall finish the sampling and learning processes for step h before moving backward to step $h - 1$. For each h, the i-th player calls the generative model for K rounds, with each round drawing SA_i independent samples; as a result, the total sample size is given by $KSH \sum_{i=1}^{m} A_i$. In what follows, let us first introduce some convenient notation that facilitates our exposition of the algorithm.

Notation. Consider any step $h \in [H]$, any player $i \in [m]$, and any data collection round $k \in [K]$. The algorithm maintains the following iterates, whose notation is gathered here with their formal definitions introduced later.

- $\widehat{V}_{i,h} \in \mathbb{R}^S$ represents the final estimate of the value function at step h by the i-th player; in particular, we set $\widehat{V}_{i,H+1} = 0$.
- $Q_{i,h}^k \in \mathbb{R}^{SA_i}$ represents the Q-function estimate of the i-th player at step h after the k-th round of data collection.
- $q_{i,h}^k \in \mathbb{R}^{SA_i}$ stands for a certain "one-step-look-ahead" Q-function estimate of the i-th player at step h using samples collected in the k-th round.
- $r_{i,h}^k \in \mathbb{R}^{SA_i}$ denotes the sample reward vector for step h received by the i-th player in the k-th round.
- $P_{i,h}^k \in \mathbb{R}^{SA_i \times S}$ denotes the empirical probability transition matrix for step h constructed using the samples collected by the i-th player in the k-th round.
- $\beta_{i,h} \in \mathbb{R}^S$ denotes the bonus vector chosen by the i-th player at step h during final value estimation.
- $\pi_{i,h}^k : \mathcal{S} \to \Delta(\mathcal{A}_i)$ denotes the policy iterate of the i-th player at step h before the beginning of the k-th round of data collection; in particular, we set $\pi_{i,h}^1$ to be uniform, namely, $\pi_{i,h}^1(a_i \,|\, s) = 1/A_i$ for any $(s, a_i) \in \mathcal{S} \times \mathcal{A}_i$.

Crucially, the above objects are all constructed from the perspective of a single player, and hence resemble those needed to operate a "single-agent" MDP (as opposed to MARL). As such, the complexity of storing/updating the above objects only scales with the aggregate size of the individual action space, rather than the size of the product action space.

Main steps of the proposed algorithm. As mentioned above, our algorithm collects multiple rounds of independent samples for each h. In what follows, let us describe the proposed procedure for the i-th player in the k-th round for step h.

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

$$s'_{k,h,s,a_i} \sim P_h\big(\,\cdot \,|\, s, \boldsymbol{a}(k,h,s,a_i)\big) \qquad \text{and} \qquad r_{k,i,h,s,a_i} = r_{i,h}\big(s, \boldsymbol{a}(k,h,s,a_i)\big), \tag{14a}$$

where $\boldsymbol{a}(k,h,s,a_i) = [a_j(k,h,s,a_i)]_{1\le j\le m} \in \mathcal{A}$ consists of independent individual actions drawn from

$$a_j(k,h,s,a_i) \overset{\text{ind.}}{\sim} \pi_{j,h}^k(\cdot \,|\, s) \quad (j \neq i) \qquad \text{and} \qquad a_i(k,h,s,a_i) = a_i. \tag{14b}$$

These samples are then employed to construct the sample reward vector $r_{i,h}^k \in \mathbb{R}^{SA_i}$ and empirical probability transition kernel $P_{i,h}^k \in \mathbb{R}^{SA_i \times S}$ such that

$$r_{i,h}^k(s,a_i) = r_{k,i,h,s,a_i} \qquad \text{and} \qquad P_{i,h}^k(s' \,|\, s,a_i) = \begin{cases} 1, & \text{if } s' = s'_{k,h,s,a_i} \\ 0, & \text{else} \end{cases} \tag{14c}$$

for all $(s,a_i,s') \in \mathcal{S} \times \mathcal{A}_i \times \mathcal{S}$. Note that the i-th player only needs to compute (14c), without the need of directly observing the other players' actions.

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

$$q_{i,h}^k = r_{i,h}^k + P_{i,h}^k \widehat{V}_{i,h+1}. \tag{15}$$

We then adopt the update rule of Q-learning:

$$Q_{i,h}^k = (1-\alpha_k) Q_{i,h}^{k-1} + \alpha_k q_{i,h}^k, \tag{16}$$

where $0 < \alpha_k < 1$ is the learning rate. Applying (16) recursively and using the quantities defined in (5), we easily arrive at the following expansion:

$$Q_{i,h}^k = \sum_{j=1}^k \alpha_j^k q_{i,h}^j. \tag{17}$$

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

$$\pi_{i,h}^{k+1}(a_i \,|\, s) = \frac{\exp\big(\eta_{k+1} Q_{i,h}^k(s,a_i)\big)}{\sum_{a' \in \mathcal{A}_i} \exp\big(\eta_{k+1} Q_{i,h}^k(s,a')\big)}, \qquad \forall (s,a_i) \in \mathcal{S} \times \mathcal{A}_i, \tag{18}$$

where $\eta_{k+1} > 0$ is another learning rate associated with policy updates (to be specified shortly). In fact, this subroutine implements the Follow-the-Regularized-Leader strategy [56]:

$$\pi_{i,h}^{k+1}(\cdot \,|\, s) = \arg\min_{\mu \in \Delta(\mathcal{A}_i)} \left\{ -\big\langle \mu, Q_{i,h}^k(s,\cdot) \big\rangle + \frac{1}{\eta_{k+1}} F(\mu) \right\}, \tag{19}$$

where the regularizer $F(\cdot)$ is chosen to be the negative entropy function $F(\mu) := \sum_{a \in \mathcal{A}_i} \mu(a) \log\big(\mu(a)\big)$.

After carrying out K rounds of the above procedure, our final policy estimate $\widehat{\pi} : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$ and the value estimate $\widehat{V}_{i,h} : \mathcal{S} \to \mathbb{R}$ for step h are taken respectively to be

$$\widehat{V}_{i,h}(s) = \min\left\{ \sum_{k=1}^K \alpha_k^K \Big\langle \pi_{i,h}^k(\cdot \,|\, s),\, q_{i,h}^k(s,\cdot) \Big\rangle + \beta_{i,h}(s),\; H-h+1 \right\} \qquad \text{and} \tag{20a}$$

$$\widehat{\pi}_h(\boldsymbol{a} \,|\, s) = \sum_{k=1}^K \alpha_k^K \prod_{i=1}^m \pi_{i,h}^k(a_i \,|\, s) \tag{20b}$$

for any $\big(s, \boldsymbol{a} = [a_1, \ldots, a_m]\big) \in \mathcal{S} \times \mathcal{A}$, where $\{\alpha_k^K\}$ is defined in (5) and $\beta_{i,h}(s) \ge 0$ is some bonus term (taking the form of some data-driven upper confidence bound) to be specified momentarily. It is worth pointing out that the final policy (20b) takes the form of *a mixture of product policies*. In the special case of two-player zero-sum MGs, we can alternatively output a product policy

$$\text{(two-player zero-sum MGs)} \qquad \widehat{\pi} = \widehat{\pi}_1 \times \widehat{\pi}_2, \tag{21}$$

where for each $i = 1, 2$, we take $\widehat{\pi}_i = \{\widehat{\pi}_{i,h}\}_{1 \le h \le H}$ with $\widehat{\pi}_{i,h} = \sum_{k=1}^K \alpha_k^K \pi_{i,h}^k$.

https://doi.org/10.52202/068431-1117

Algorithm 1: Q-FTRL.

Input: number of rounds K for each step, learning rates $\{\alpha_k\}$ (cf. (22)) and $\{\eta_{k+1}\}$ (cf. (23)).
// set initial value estimates to 0, and initial policies to uniform distributions.
Initialize: for any $i \in [m]$ and any $(s, a_i, h) \in \mathcal{S} \times \mathcal{A}_i \times [H]$, set $\widehat{V}_{i,H+1}(s) = Q^0_{i,h}(s, a_i) = 0$ and $\pi^1_{i,h}(a_i \,|\, s) = 1/A_i$.
for $h = H$ **to** 1 **do**
  for $k = 1$ **to** K **do**
    for $i = 1$ **to** m **do**
      // draw independent samples, and construct empirical models.
      $\big(r^k_{i,h}, P^k_{i,h}\big) \leftarrow \texttt{sampling}\big(i, h, \pi^k_h = \{\pi^k_{j,h}\}_{j\in[m]}\big)$. /* see Algorithm 2. */
      // update Q-estimates with upper confidence bounds.
      Compute $q^k_{i,h} = r^k_{i,h} + P^k_{i,h}\widehat{V}_{i,h+1}$, and update $Q^k_{i,h}$ according to (16).
      // update policy estimates using FTRL.
      Update $\pi^{k+1}_{i,h}$ according to (18).
  // output the final value estimate for step h.
  for $i = 1$ **to** m **do**
    Update $\widehat{V}_{i,h}$ according to (20a), where $\beta_{i,h}$ is given in (24).
if *$\mathcal{MG}$ is a two-player zero-sum Markov game* **then**
  output: $\widehat{\pi}_1 \times \widehat{\pi}_2$, where for any $i = 1, 2$, $\widehat{\pi}_i = \{\widehat{\pi}_{i,h}\}_{1\le h\le H}$ with $\widehat{\pi}_{i,h} = \sum_{k=1}^K \alpha_k^K \pi^k_{i,h}$.
if *$\mathcal{MG}$ is a multi-player general-sum Markov game* **then**
  output: $\widehat{\pi} = \{\widehat{\pi}_h\}_{1\le h\le H}$, where $\widehat{\pi}_h = \sum_{k=1}^K \alpha_k^K \big(\pi^k_{1,h} \times \cdots \times \pi^k_{m,h}\big)$.

The whole procedure is summarized in Algorithm 1.

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

$$\alpha_k = \frac{c_\alpha \log K}{k - 1 + c_\alpha \log K}, \qquad k = 1, 2, \ldots \tag{22}$$

for some constant $c_\alpha \ge 24$. In addition, the learning rates associated with policy updates are chosen to be:

$$\eta_{k+1} = \sqrt{\frac{\log K}{\alpha_k H}}, \qquad k = 1, 2, \ldots \tag{23}$$

Choices of bonus terms. It remains to specify the bonus terms, which are selected based on fairly intricate upper confidence bounds. This constitutes a key — and perhaps the most challenging — component of our algorithm design. Specifically, we take

$$\beta_{i,h}(s) = c_{\mathsf{b}} \sqrt{\frac{\log^3\big(\frac{KS\sum_i A_i}{\delta}\big)}{KH}} \sum_{k=1}^K \alpha_k^K \left\{ \mathsf{Var}_{\pi^k_{i,h}(\cdot|s)}\Big(q^k_{i,h}(s,\cdot)\Big) + H \right\} \tag{24}$$

for any $(i, s, h) \in [m] \times \mathcal{S} \times [H]$, where $c_{\mathsf{b}} > 0$ is some sufficiently large constant; see also (6) for the definition of the variance-style quantity. As in previous works, the bonus terms, which are chosen carefully in a data-driven fashion, need to compensate for the uncertainty incurred during the estimation process.

3.2 Main results

As it turns out, the proposed algorithm is tractable and provably sample-efficient. We begin by characterizing its sample complexity when learning Nash equilibria in two-player zero-sum MGs,

Algorithm 2: Auxiliary function $\texttt{sampling}\big(i, h, \pi_h = \{\pi_{j,h}\}_{j\in[m]}\big)$.

Initialize: $\overline{r} = 0 \in \mathbb{R}^{SA_i}$, and $\overline{P} = 0 \in \mathbb{R}^{SA_i \times S}$.
for $(s, a_i) \in \mathcal{S} \times \mathcal{A}_i$ **do**
  Draw an independent sample from the generative model: $s'_{s,a_i} \sim P_h\big(\cdot \,|\, s, \boldsymbol{a}(s, a_i)\big)$, where $\boldsymbol{a}(s, a_i) = [a_j(s, a_i)]_{1\le j\le m}$ is composed of independent individual actions drawn from

$$a_j(s, a_i) \overset{\text{ind.}}{\sim} \pi_{j,h}(\cdot \,|\, s) \quad (j \neq i) \qquad \text{and} \qquad a_i(s, a_i) = a_i. \tag{25}$$

  Set $\overline{r}(s, a_i) = r_{i,h}\big(s, \boldsymbol{a}(s, a_i)\big)$ and $\overline{P}\big(s'_{s,a_i} \,|\, s, a_i\big) = 1$.
Return: $\big(\overline{r}, \overline{P}\big)$.

and then shift attention to learning CCE in multi-player general-sum MGs (given the intractability of learning NEs in general).

Theorem 1 (NE for two-player zero-sum MGs)**.** *Consider a two-player zero-sum Markov game, and consider any $\varepsilon \in (0, H]$ and any $0 < \delta < 1$. Suppose that*

$$K \geq \frac{c_{\mathsf{k}} H^3 \log^4\big(\frac{KS(A_1+A_2)}{\delta}\big)}{\varepsilon^2} \tag{26}$$

for some large enough universal constant $c_{\mathsf{k}} > 0$. With probability at least $1-\delta$, the product policy $\widehat{\pi}_1 \times \widehat{\pi}_2$ computed by Algorithm 1 is an ε-approximate Nash equilibrium, i.e., its sub-optimality gap (cf. (13)) obeys $\mathsf{gap}\big(\widehat{\pi}_1 \times \widehat{\pi}_2\big) \leq \varepsilon$.

Theorem 2 (CCE for multi-player general-sum MGs)**.** *Consider an m-player general-sum Markov game, and consider any $\varepsilon \in (0, H]$ and any $0 < \delta < 1$. Suppose that*

$$K \geq \frac{c_{\mathsf{k}} H^3 \log^4\big(\frac{KS\sum_{i=1}^m A_i}{\delta}\big)}{\varepsilon^2} \tag{27}$$

for some large enough universal constant $c_{\mathsf{k}} > 0$. With probability at least $1-\delta$, the joint policy $\widehat{\pi}$ returned by Algorithm 1 is an ε-approximate CCE, i.e., its sub-optimality gap (cf. (13)) obeys $\mathsf{gap}\big(\widehat{\pi}\big) \leq \varepsilon$.

Theorems 1-2 establish sample complexity upper bounds for the proposed algorithm, which we take a moment to interpret as follows. The proofs of these two theorems are postponed to Appendix C in the supplementary material.

Sample complexity. When a generative model is available, Theorems 1-2 assert that the total number of samples (i.e., $KSH\sum_i A_i$) needed for Algorithm 1 to work is

$$\begin{cases} \widetilde{O}\big(\frac{H^4 S(A_1+A_2)}{\varepsilon^2}\big), & \text{for learning an } \varepsilon\text{-NE in two-player zero-sum MGs;} \\ \widetilde{O}\big(\frac{H^4 S\sum_{i=1}^m A_i}{\varepsilon^2}\big), & \text{for learning an } \varepsilon\text{-CCE in multi-player general-sum MGs.} \end{cases} \tag{28}$$

As far as we know, our theorems deliver the first results that uncover the plausibility of simultaneously overcoming the long-horizon barrier and the curse of multi-agents. Let us compare (28) with prior art.

- *NE in two-player zero-sum MGs.* First, consider learning ε-NE policies in two-player zero-sum MGs. In comparison to Zhang et al. [79] (cf. (1)), our result reveals that what ultimately matters is the total number of individual actions (i.e., $A_1 + A_2$) as opposed to the total number $A_1 A_2$ of possible joint actions; additionally, our results exhibit improved horizon dependency (by a factor of H^2) compared to Bai et al. [5], Jin et al. [31] (see (3)), although we remark that the online sampling protocol therein is clearly more restrictive than a generative model.
- *CCE in multi-player general-sum MGs (for a fixed m).* Similar messages carry over to the task of learning multi-player general-sum MGs when the number of players m is a fixed constant. Liu et al. [43] provided the first non-asymptotic result on learning CCE in the exploration setting; the model-based algorithm studied therein learns an ε-CCE using

$$\widetilde{O}\bigg(\frac{H^5 S^2 \prod_{i=1}^m A_i}{\varepsilon^2}\bigg) \text{ samples} \qquad \text{or} \qquad \widetilde{O}\bigg(\frac{H^4 S^2 \prod_{i=1}^m A_i}{\varepsilon^2}\bigg) \text{ episodes} \tag{29}$$

https://doi.org/10.52202/068431-1117

which is sub-optimal in terms of the dependency on both H and S and suffers from the curse of multiple agents. A more recent strand of works focused on a type of online RL algorithms called V-learning, which exploited the effectiveness of adversarial learning subroutines in overcoming the curse of multi-agents [44, 63, 31]; along this line, the state-of-the-art sample complexity bound is [31]:

$$\widetilde{O}\bigg(\frac{H^6 S \max_{1\le i\le m} A_i}{\varepsilon^2}\bigg) \text{ samples} \qquad \text{or} \qquad \widetilde{O}\bigg(\frac{H^5 S \max_{1\le i\le m} A_i}{\varepsilon^2}\bigg) \text{ episodes}, \tag{30}$$

which remains suboptimal in terms of the horizon dependency. As a drawback of these works, the policy returned by V-learning is non-Markovian, an issue that has been recently addressed by Daskalakis et al. [23] at the price of a much higher sample complexity. It is worth emphasizing that all these works assume the online exploration setting as opposed to the scenario with a generative model.

Minimax optimality. To assess the tightness of our result (28), it is helpful to look at the information-theoretic limit. Following the minimax lower bound for single-agent MDPs [3, 41], one can develop a minimax sample complexity lower bound for Markov games (w.r.t. finding either an ε-NE or an ε-CCE) that scales as

$$\text{(minimax lower bound)} \qquad \frac{H^4 S \max_{1\le i\le m} A_i}{\varepsilon^2} \tag{31}$$

modulo some logarithmic factor; see Appendix E.3 in the supplementary material for a formal statement and its proof. Taking this together with (28) confirms the minimax optimality of our algorithm (up to logarithmic terms) when the number m of players is fixed or grows only logarithmically in problem parameters.

No burn-in sample size and full ε-range. It is noteworthy that the validity of our sample complexity bound (28) 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 might be practically appealing. For instance, the output policies are Markovian in nature, which depend only on the current state s and step number h. This is enabled thanks to the availability of the generative model, which allows us to settle the sampling and learning process for step $h+1$ completely before moving backward to step h; in contrast, the online sampling protocol studied in Bai et al. [5], Jin et al. [31] cannot be implemented in this way without incurring information loss. In addition, our algorithm can be carried out in a decentralized fashion (except that the final estimate $\widehat{\pi}$ needs to aggregate policy iterates from all players), with each player acting in a symmetric yet independent manner (without the need of knowing each other's individual action). Our algorithm is also "rational" in the sense that it converges to the best-response policy of a player if all other players freeze their policies. 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 paradigm that simultaneously overcomes the curse of multiple agents and optimizes the horizon dependency when solving multi-player 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 [79, 43], whereas the availability of the generative model in conjunction with the variance-aware bonus design improves horizon dependency compared to Bai et al. [5], Jin et al. [31].

Acknowledgements: Y. Chen is supported in part by the Alfred P. Sloan Research Fellowship, the Google Research Scholar Award, the AFOSR grant FA9550-22-1-0198, the ONR grant N00014-22-1-2354, and the NSF grants CCF-2221009, CCF-1907661, IIS-2218713 and IIS-2218773. Y. Wei is supported in part by the the NSF grants CCF-2106778, DMS-2147546/2015447 and CAREER award DMS-2143215. Y. Chi are supported in part by the grants ONR N00014-19-1-2404, NSF CCF-2106778 and DMS-2134080, and CAREER award ECCS-1818571. Part of this work was done while G. Li, Y. Wei and Y. Chen were visiting the Simons Institute for the Theory of Computing.

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

[3] 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.

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

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

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

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

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

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

[10] 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.

[11] S. Cen, Y. Chi, S. Du, and L. Xiao. Faster last-iterate convergence of policy optimization in zero-sum markov games. *arXiv preprint arXiv:2210.01050*, 2022.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] Q. Cui and S. S. Du. Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. *arXiv preprint arXiv:2206.00159*, 2022.

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

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

[21] 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.

https://doi.org/10.52202/068431-1117

[22] 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.

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

[24] Z. Dou, Z. Yang, Z. Wang, and S. Du. Gap-dependent bounds for two-player markov games. In *International Conference on Artificial Intelligence and Statistics*, pages 432–455, 2022.

[25] 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.

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

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

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

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

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

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] G. Li, L. Shi, Y. Chen, Y. Gu, and Y. Chi. Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[41] 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.

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

[43] 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, 2021.

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

[45] 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.

[46] W. Mou, C. J. Li, M. J. Wainwright, P. L. Bartlett, and M. I. Jordan. On linear stochastic approximation: Fine-grained Polyak-Ruppert and non-asymptotic concentration. *arXiv preprint arXiv:2004.04719*, 2020.

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

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

[49] J. F. Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

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

[51] 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.

[52] 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.

[53] 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.

[54] 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.

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

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

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

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

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

[60] 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.

[61] 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.

 https://doi.org/10.52202/068431-1117

[62] 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.

[63] 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.

[64] 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.

[65] S. Vaswani, L. F. Yang, and C. Szepesvári. Near-optimal sample complexity bounds for constrained MDPs. *arXiv preprint arXiv:2206.06270*, 2022.

[66] 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.

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

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

[69] 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.

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

[71] 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.

[72] G. Weisz, P. Amortila, and C. Szepesvári. Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021.

[73] 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.

[74] Y. Yan, G. Li, Y. Chen, and J. Fan. Model-based reinforcement learning is minimax-optimal for offline zero-sum markov games. *arXiv preprint arXiv:2206.04044*, 2022.

[75] L. Yang and M. Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019.

[76] Y. Yang and C. Ma. $o(t^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum markov games. *arXiv preprint arXiv:2209.12430*, 2022.

[77] A. Zanette, M. J. Kochenderfer, and E. Brunskill. Almost horizon-free structure-aware best policy identification with a generative model. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[79] 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.

[80] 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.

[81] R. Zhang, Q. Liu, H. Wang, C. Xiong, N. Li, and Y. Bai. Policy optimization for Markov games: Unified framework and faster convergence. *arXiv preprint arXiv:2206.02640*, 2022.

[82] 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.

[83] 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...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-1117