A Sharp Analysis of Model-based Reinforcement Learning with Self-Play

Qinghua Liu [1] **Tiancheng Yu** [2] **Yu Bai** [3] **Chi Jin** [1]

Abstract

Model-based algorithms—algorithms that explore the environment through building and utilizing an estimated model—are widely used in reinforcement learning practice and theoretically shown to achieve optimal sample efficiency for single-agent reinforcement learning in Markov Decision Processes (MDPs). However, for multi-agent reinforcement learning in Markov games, the current best known sample complexity for model-based algorithms is rather suboptimal and compares unfavorably against recent model-free approaches.

In this paper, we present a sharp analysis of model-based self-play algorithms for multi-agent Markov games. We design an algorithm *Optimistic Nash Value Iteration* (Nash-VI) for two-player zero-sum Markov games that is able to output an ϵ-approximate Nash policy in $\tilde{\mathcal{O}}(H^3SAB/\epsilon^2)$ episodes of game playing, where S is the number of states, A, B are the number of actions for the two players respectively, and H is the horizon length. This significantly improves over the best known model-based guarantee of $\tilde{\mathcal{O}}(H^4S^2AB/\epsilon^2)$, and is the first that matches the information-theoretic lower bound $\Omega(H^3S(A+B)/\epsilon^2)$ except for a $\min\{A, B\}$ factor. In addition, our guarantee compares favorably against the best known model-free algorithm if $\min\{A, B\} = o(H^3)$, and outputs a single Markov policy while existing sample-efficient model-free algorithms output a nested mixture of Markov policies that is in general non-Markov and rather inconvenient to store and execute. We further adapt our analysis to designing a provably efficient task-agnostic algorithm for zero-sum Markov games, and designing the first line of provably sample-efficient algorithms for multi-player general-sum Markov games.

1. Introduction

This paper is concerned with the problem of multi-agent reinforcement learning (multi-agent RL), in which multiple agents learn to make decisions in an unknown environment in order to maximize their (own) cumulative rewards. Multi-agent RL has achieved significant recent success in traditionally hard AI challenges including large-scale strategy games (such as GO) (Silver et al., 2016; 2017), real-time video games involving team play such as Starcraft and Dota2 (OpenAI, 2018; Vinyals et al., 2019), as well as behavior learning in complex social scenarios (Baker et al., 2020). Achieving human-like (or super-human) performance in these games using multi-agent RL typically requires a large number of samples (steps of game playing) due to the necessity of exploration, and how to improve the sample complexity of multi-agent RL has been an important research question.

One prevalent approach towards solving multi-agent RL is *model-based* methods, that is, to use the existing visitation data to build an estimate of the model (i.e. transition dynamics and rewards), run an offline planning algorithm on the estimated model to obtain the policy, and play the policy in the environment. Such a principle underlies some of the earliest single-agent online RL algorithms such as E3 (Kearns & Singh, 2002) and RMax (Brafman & Tennenholtz, 2002), and is conceptually appealing for multi-agent RL too since the multi-agent structure does not add complexity onto the model estimation part and only requires an appropriate multi-agent planning algorithm (such as value iteration for games (Shapley, 1953)) in a black-box fashion. On the other hand, *model-free* methods do not directly build estimates of the model, but instead directly estimate the value functions or action-value (Q) functions of the problem at the optimal/equilibrium policies, and play the greedy policies with respect to the estimated value functions. Model-free algorithms have also been well developed for multi-agent RL such as friend-or-foe Q-Learning (Littman, 2001) and Nash Q-Learning (Hu & Wellman, 2003).

While both model-based and model-free algorithms have been shown to be provably efficient in multi-agent RL in a recent line of work (Bai & Jin, 2020; Xie et al., 2020; Bai et al., 2020), a more precise understanding of the optimal sample complexities within these two types of algorithms (respectively) is still lacking. In the specific setting of two-player

[1] Princeton University, [2] Massachusetts Institute of Technology, [3] Salesforce Research. Correspondence to: Qinghua Liu <qinghual@princeton.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

zero-sum Markov games, the current best sample complexity for model-based algorithms is achieved by the VI-ULCB (Value Iteration with Upper/Lower Confidence Bounds) algorithm (Bai & Jin, 2020; Xie et al., 2020): In a tabular Markov game with S states, $\{A, B\}$ actions for the two players, and horizon length H, VI-ULCB is able to find an ϵ-approximate Nash equilibrium policy in $\tilde{\mathcal{O}}(H^4S^2AB/\epsilon^2)$ episodes of game playing. However, compared with the information-theoretic lower bound $\Omega(H^3S(A+B)/\epsilon^2)$, this rate has suboptimal dependencies on all of H, S, and A, B. In contrast, the current best sample complexity for *model-free* algorithms is achieved by Nash V-Learning (Bai et al., 2020), which finds an ϵ-approximate Nash policy in $\tilde{\mathcal{O}}(H^6S(A+B)/\epsilon^2)$ episodes. Compared with the lower bound, this is tight except for a $\mathrm{poly}(H)$ factor, which may seemingly suggest that model-free algorithms could be superior to model-based ones in multi-agent RL. However, such a conclusion would be in stark contrast to the single-agent MDP setting, where it is known that model-based algorithms are able to achieve minimax optimal sample complexities (Jaksch et al., 2010; Azar et al., 2017). It naturally arises whether model-free algorithms are indeed superior in multi-agent settings, or whether the existing analyses of model-based algorithms are not tight. This motivates us to ask the following question:

How sample-efficient are model-based algorithms in multi-agent RL?

In this paper, we advance the theoretical understandings of multi-agent RL by presenting a sharp analysis of model-based algorithms on Markov games. Our core contribution is the design of a new model-based algorithm *Optimistic Nash Value Iteration* (Nash-VI) that achieves an almost optimal sample complexity for zero-sum Markov games and improves significantly over existing model-based approaches. We summarize our main contributions as follows. A comparison between our and prior results can be found in Table 1.

- We design a new model-based algorithm *Optimistic Nash Value Iteration* (Nash-VI) that provably finds ϵ-approximate Nash equilibria for Markov games in $\tilde{\mathcal{O}}(H^3SAB/\epsilon^2)$ episodes of game playing (Section 3). This improves over the best existing model-based algorithm by $O(HS)$ and is the first algorithm that matches the sample complexity lower bound except for a $\tilde{\mathcal{O}}(\min\{A, B\})$ factor, showing that model-based algorithms can indeed achieve an almost optimal sample complexity. Further, unlike state-of-the-art model-free algorithms such as Nash V-Learning (Bai et al., 2020), this algorithm achieves in addition a $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound, and outputs a simple Markov policy (instead of a nested mixture of Markov policies as returned by Nash V-Learning).
- We design an alternative algorithm *Optimistic Value Iteration with Zero Reward* (VI-Zero) that is able to perform task-agnostic (reward-free) learning for multiple Markov games sharing the same transitions (Section 4). For $N > 1$ games with the same transition and different (known) rewards, VI-Zero can find ϵ-approximate Nash policy for all games simultaneously in $\tilde{\mathcal{O}}(H^4SAB\log N/\epsilon^2)$ episodes of game playing, which scales logarithmically in the number of games.
- We design the first line of sample-efficient algorithms for *multi-player* general-sum Markov games. In a multi-player game with M players and A_i actions per player, we show that an ϵ near-optimal policy can be found in $\tilde{\mathcal{O}}(H^4S^2\prod_{i\in[M]}A_i/\epsilon^2)$ episodes, where the desired optimality can be either one of Nash equilibrium, correlated equilibrium (CE), or coarse correlated equilibrium (CCE). We achieve this guarantee by either a multi-player version of Nash-VI or a multi-player version of reward-free value iteration (Section 5 & Appendix A).

1.1. Related work

Markov games. Markov games (or stochastic games) are proposed in the early 1950s (Shapley, 1953). They are widely used to model multi-agent RL. Learning the Nash equilibria of Markov games has been studied in Littman (1994; 2001); Hu & Wellman (2003); Hansen et al. (2013); Lee et al. (2020), where the transition matrix and reward are assumed to be known, or in the asymptotic setting where the number of data goes to infinity. These results do not directly apply to the non-asymptotic setting where the transition and reward are unknown and only a limited amount of data are available for estimating them.

Another line of works make certain strong reachability assumptions under which sophisticated exploration strategies are not required. A prevalent approach is to assume access to simulators (generative models) that enable the agent to directly sample transition and reward information for any state-action pair. In this setting, Jia et al. (2019); Sidford et al. (2019); Zhang et al. (2020a) provide non-asymptotic bounds on the number of calls to the simulator for finding an ϵ-approximate Nash equilibrium. Wei et al. (2017) study Markov games under an alternative assumption that no matter what strategy one agent sticks to, the other agent can always reach all states by playing a certain policy.

Non-asymptotic guarantees without reachability assumptions. Recent works of Bai & Jin (2020); Xie et al. (2020) provide the first line of non-asymptotic sample complexity guarantees for learning Markov games without reach-

Table 1. Sample complexity (the required number of episodes) for algorithms to find ϵ-approximate Nash equlibrium policies in zero-sum Markov games: VI-explore and VI-UCLB (Bai & Jin, 2020), OMVI-SM (Xie et al., 2020), and Nash Q/V-learning (Bai et al., 2020). The lower bound is proved by Jin et al. (2018); Domingues et al. (2020).

	Algorithm	Task-Agnostic	$\sqrt{T}$-Regret	Sample Complexity	Output Policy
Model-based	VI-explore	Yes		$\tilde{O}(H^5S^2AB/\epsilon^2)$	a single Markov policy
	VI-ULCB		Yes	$\tilde{O}(H^4S^2AB/\epsilon^2)$	
	OMVI-SM		Yes	$\tilde{O}(H^4S^3A^3B^3/\epsilon^2)$	
	Algorithm 2	Yes		$\tilde{O}(H^4SAB/\epsilon^2)$	
	Algorithm 1		Yes	$\tilde{O}(H^3SAB/\epsilon^2)$	
Model-free	Nash Q-learning			$\tilde{O}(H^5SAB/\epsilon^2)$	nested mixture of Markov policies
	Nash V-learning			$\tilde{O}(H^6S(A+B)/\epsilon^2)$	
	Lower Bound	-	-	$\Omega(H^3S(A+B)/\epsilon^2)$	-

ability assumptions. More recently, Bai et al. (2020) propose two model-free algorithms—Nash Q-Learning and Nash V-Learning with better sample complexity guarantees. In particular, the Nash V-learning algorithm achieves near-optimal dependence on S, A and B. However, the dependence on H is worse than our results and the output policy is a nested mixture, which is hard to implement. We compare our results with existing non-asymptotic guarantees in Table 1.

We remark that the classic R-max algorithm (Brafman & Tennenholtz, 2002) also provides provable guarantees for learning Markov games. However, Brafman & Tennenholtz (2002) use a weaker definition of regret (similar to the online setting in Xie et al. (2020)), and consequently their result does not imply any sample complexity guarantee for finding Nash equilibrium policies.

Adversarial MDPs. Another way to model the multi-player behavior is to use *adversarial MDPs*. Most works in this line consider the setting with adversarial reward (Zimin & Neu, 2013; Rosenberg & Mansour, 2019; Jin et al., 2019), where the reward can be manipulated by an adversary arbitrarily and the goal is to compete with the optimal (stationary) policy in hindsight. Learning adversarial MDPs with changing dynamics is computationally hard even under full-information feedback (Yadkori et al., 2013). Notice these results also do not imply provable algorithms in our setting, because the opponent in Markov games can affect both the reward and the transition.

Single-agent RL. There is a rich literature on reinforcement learning in MDPs (see e.g., Jaksch et al., 2010; Osband et al., 2014; Azar et al., 2017; Dann et al., 2017; Strehl et al., 2006; Jin et al., 2018). MDPs are special cases of Markov games, where only a single agent interacts with a stochastic environment. For the tabular episodic setting with nonstationary dynamics and no simulators, the best sample complexity is $\tilde{O}(H^3SA/\epsilon^2)$, achieved by the model-based algorithm in Azar et al. (2017) and the model-free algorithm in Zhang et al. (2020c). Both of them match the lower bound $\Omega(H^3SA/\epsilon^2)$ (Jin et al., 2018).

Reward-free learning. Jin et al. (2020) study a new paradigm of learning MDPs called reward-free learning, which is also known as the task-agnostic (Zhang et al., 2020b) or reward-agnostic setting. In this setting, the agent goes through a two-stage process. In the exploration phase the agent interacts with the environment without the guidance of any reward information, and in the planning phase the reward information is revealed and the agent computes a policy based on the transition information collected in the exploration phase and the reward information revealed in the planning phase.

2. Preliminaries

In this paper, we consider Markov Games (MGs, Shapley, 1953; Littman, 1994), which are also known as stochastic games in the literature. Markov games are the generalization of standard Markov Decision Processes (MDPs) into the multi-player setting, where each player seeks to maximize her own utility. For simplicity, in this section we describe the important special case of *two-player zero-sum games*, and return to the general formulation in Appendix A.

Formally, we consider the tabular episodic version of two-player zero-sum Markov game, which we denote as $\mathrm{MG}(H, \mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$. Here H is the number of steps in each episode, $\mathcal{S}$ is the set of states with $|\mathcal{S}| \le S$, $(\mathcal{A}, \mathcal{B})$ are the sets of actions of the max-player and the min-player respectively with $|\mathcal{A}| \le A$ and $|\mathcal{B}| \le B$, $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is a collection of transition matrices, so that $\mathbb{P}_h(\cdot|s, a, b)$ gives the distribution of the next state if action pair (a, b) is taken at state s at step h, and $r = \{r_h\}_{h\in[H]}$ is a collection of reward functions, where $r_h\colon \mathcal{S}\times\mathcal{A}\times\mathcal{B}\to[0,1]$ is the deterministic reward function at step h.[1] This reward represents both the gain of the max-player and the loss of the min-player, making the problem a zero-sum Markov game.

In each episode of this MG, we start with a *fixed initial state* s_1. At each step $h\in[H]$, both players observe state $s_h\in\mathcal{S}$, and pick their own actions $a_h\in\mathcal{A}$ and $b_h\in\mathcal{B}$ simultaneously. Then, both players observe the actions of their opponent, receive reward $r_h(s_h,a_h,b_h)$, and then the environment transitions to the next state $s_{h+1}\sim\mathbb{P}_h(\cdot|s_h,a_h,b_h)$. The episode ends when s_{H+1} is reached.

Policy, value function. A (Markov) policy μ of the max-player is a collection of H functions $\{\mu_h:\mathcal{S}\to\Delta_{\mathcal{A}}\}_{h\in[H]}$, each mapping from a state to a distribution over actions. (Here $\Delta_{\mathcal{A}}$ is the probability simplex over action set $\mathcal{A}$.) Similarly, a policy ν of the min-player is a collection of H functions $\{\nu_h:\mathcal{S}\to\Delta_{\mathcal{B}}\}_{h\in[H]}$. We use the notation $\mu_h(a|s)$ and $\nu_h(b|s)$ to represent the probability of taking action a or b for state s at step h under Markov policy μ or ν respectively.

We use $V_h^{\mu,\nu}\colon\mathcal{S}\to\mathbb{R}$ to denote the value function at step h under policy μ and ν, so that $V_h^{\mu,\nu}(s)$ gives the expected cumulative rewards received under policy μ and ν, starting from s at step h:

$$V_h^{\mu,\nu}(s):=\mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'},a_{h'},b_{h'})\middle| s_h=s\right]. \quad (1)$$

We also define $Q_h^{\mu,\nu}:\mathcal{S}\times\mathcal{A}\times\mathcal{B}\to\mathbb{R}$ to be the Q-value function at step h so that $Q_h^{\mu,\nu}(s,a,b)$ gives the cumulative rewards received under policy μ and ν, starting from (s,a,b) at step h:

$$\begin{aligned}&Q_h^{\mu,\nu}(s,a,b)\\ &:=\mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'},a_{h'},b_{h'})\big| s_h=s,a_h=a,b_h=b\right].\end{aligned} \quad (2)$$

For simplicity, we define operator $\mathbb{P}_h$ as $[\mathbb{P}_hV](s,a,b):=\mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,b)}V(s')$ for any value function V. We also use notation $[\mathbb{D}_\pi Q](s):=\mathbb{E}_{(a,b)\sim\pi(\cdot,\cdot|s)}Q(s,a,b)$ for any action-value function Q. By definition of value functions, we have the Bellman equation

$$\begin{cases}Q_h^{\mu,\nu}(s,a,b)=(r_h+\mathbb{P}_hV_{h+1}^{\mu,\nu})(s,a,b),\\ V_h^{\mu,\nu}(s)=(\mathbb{D}_{\mu_h\times\nu_h}Q_h^{\mu,\nu})(s),\end{cases}$$

for all $(s,a,b,h)\in\mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[H]$, and at the $(H+1)^{\text{th}}$ step we have $V_{H+1}^{\mu,\nu}(s)=0$ for all $s\in\mathcal{S}$.

Best response and Nash equilibrium. For any policy of the max-player μ, there exists a *best response* of the min-player, which is a policy $\nu^\dagger(\mu)$ satisfying $V_h^{\mu,\nu^\dagger(\mu)}(s)=\inf_\nu V_h^{\mu,\nu}(s)$ for any $(s,h)\in\mathcal{S}\times[H]$. We denote $V_h^{\mu,\dagger}:=V_h^{\mu,\nu^\dagger(\mu)}$. By symmetry, we can also define $\mu^\dagger(\nu)$ and $V_h^{\dagger,\nu}$. It is further known (cf. Filar & Vrieze (2012)) that there exist policies $\mu^\star$, $\nu^\star$ that are optimal against the best responses of the opponents, in the sense that

$$\begin{cases}V_h^{\mu^\star,\dagger}(s)=\sup_\mu V_h^{\mu,\dagger}(s),\\ V_h^{\dagger,\nu^\star}(s)=\inf_\nu V_h^{\dagger,\nu}(s),\end{cases}$$

for all $(s,h)\in\mathcal{S}\times[H]$. We call these optimal strategies $(\mu^\star,\nu^\star)$ the Nash equilibrium of the Markov game, which satisfies the following minimax equation [2]:

$$\sup_\mu\inf_\nu V_h^{\mu,\nu}(s)=V_h^{\mu^\star,\nu^\star}(s)=\inf_\nu\sup_\mu V_h^{\mu,\nu}(s).$$

Intuitively, a Nash equilibrium gives a solution in which no player has anything to gain by changing only her own policy. We further abbreviate the values of Nash equilibrium $V_h^{\mu^\star,\nu^\star}$ and $Q_h^{\mu^\star,\nu^\star}$ as $V_h^\star$ and $Q_h^\star$. We refer readers to Appendix B for Bellman optimality equations for (the value functions of) the best responses and the Nash equilibrium.

Learning Objective. We measure the suboptimality of any pair of general policies $(\hat\mu,\hat\nu)$ using the gap between their performance and the performance of the optimal strategy (i.e., Nash equilibrium) when playing against the best responses respectively:

$$\begin{aligned}&V_1^{\dagger,\hat\nu}(s_1)-V_1^{\hat\mu,\dagger}(s_1)\\ =&\Big[V_1^{\dagger,\hat\nu}(s_1)-V_1^\star(s_1)\Big]+\Big[V_1^\star(s_1)-V_1^{\hat\mu,\dagger}(s_1)\Big].\end{aligned}$$

Definition 1 (ϵ-approximate Nash equilibrium)**.** A pair of general policies $(\hat\mu,\hat\nu)$ is an **ϵ-approximate Nash equilibrium**, if $V_1^{\dagger,\hat\nu}(s_1)-V_1^{\hat\mu,\dagger}(s_1)\le\epsilon$.

Definition 2 (Regret)**.** Let (μ^k,ν^k) denote the policies deployed by the algorithm in the k^{th} episode. After a total of K episodes, the regret is defined as

$$\mathrm{Regret}(K)=\sum_{k=1}^{K}(V_1^{\dagger,\nu^k}-V_1^{\mu^k,\dagger})(s_1).$$

One goal of reinforcement learning is to design algorithms for Markov games that can find an ϵ-approximate Nash equilibrium using a number of episodes that is small in its

[1] We assume the rewards in $[0,1]$ for normalization. Our results directly generalize to randomized reward functions, since learning the transition is more difficult than learning the reward.

[2] The minimax theorem here is different from the one for matrix games, i.e. $\max_\phi\min_\psi\phi^\top A\psi=\min_\psi\max_\phi\phi^\top A\psi$ for any matrix A, since here $V_h^{\mu,\nu}(s)$ is in general not bilinear in μ,ν.

dependency on S, A, B, H as well as $1/\epsilon$ (PAC sample complexity bound). An alternative goal is to design algorithms for Markov games that achieves regret that is sublinear in K, and polynomial in S, A, B, H (regret bound). We remark that any sublinear regret algorithm can be directly converted to a polynomial-sample PAC algorithm via the standard online-to-batch conversion (see e.g., Jin et al., 2018).

3. Optimistic Nash Value Iteration

In this section, we present our main algorithm—Optimistic Nash Value Iteration (Nash-VI), and provide its theoretical guarantee.

3.1. Algorithm description

We describe our Nash-VI Algorithm 1. In each episode, the algorithm can be decomposed into two parts.

- Line 3-15 (Optimistic planning from the estimated model): Performs value iteration with bonus using the empirical estimate of the transition $\hat{\mathbb{P}}$, and computes a new (joint) policy π which is "greedy" with respect to the estimated value functions;

- Line 18-21 (Play the policy and update the model estimate): Executes the policy π, collects samples, and updates the estimate of the transition $\hat{\mathbb{P}}$.

At a high-level, this two-phase strategy is standard in the majority of model-based RL algorithms, and also underlies provably efficient model-based algorithms such as UCBVI for single-agent (MDP) setting (Azar et al., 2017) and VI-ULCB for the two-player Markov game setting (Bai & Jin, 2020). However, VI-ULCB has two undesirable drawbacks: the sample complexity is not tight in any of H, S, and A, B dependency, and its computational complexity is PPAD-complete (a complexity class conjectured to be computationally hard (Daskalakis, 2013)).

As we elaborate in the following, our Nash-VI algorithm differs from VI-ULCB in a few important technical aspects, which allows it to significantly improve the sample complexity over VI-ULCB, and ensures that our algorithm terminates in polynomial time.

Before digging into explanations of techniques, we remark that line 16-17 is only used for computing the output policies. It chooses policy π^{out} to be the policy in the episode with minimum gap $(\overline{V}_1 - \underline{V}_1)(s_1)$. Our final output policies $(\mu^{\mathrm{out}}, \nu^{\mathrm{out}})$ are simply the *marginal policies* of π^{out}. That is, for all $(s,h) \in \mathcal{S} \times [H]$, $\mu_h^{\mathrm{out}}(\cdot|s) := \sum_{b\in\mathcal{B}} \pi_h^{\mathrm{out}}(\cdot, b|s)$, and $\nu_h^{\mathrm{out}}(\cdot|s) := \sum_{a\in\mathcal{A}} \pi_h^{\mathrm{out}}(a, \cdot|s)$.

3.1.1. Overview of techniques

Auxiliary bonus γ. The major improvement over VI-ULCB (Bai & Jin, 2020) comes from the use of a different style of bonus term γ (line 9), in addition to the standard bonus β (line 8), in value iteration steps (line 10-11). This is also the main technical contribution of our Nash-VI algorithm. This auxiliary bonus γ is computed by applying the empirical transition matrix $\hat{\mathbb{P}}_h$ to the gap at the next step $\overline{V}_{h+1} - \underline{V}_{h+1}$, This is very different from standard bonus β, which is typically designed according to the concentration inequalities.

The main purpose of these value iteration steps (line 10-11) is to ensure that the estimated values $\overline{Q}_h$ and $\underline{Q}_h$ are with high probability the upper bound and the lower bound of the Q-value of the current policy when facing best responses (see Lemma 20 and 22 for more details) [3]. To do so, prior work (Bai & Jin, 2020) only adds bonus β, which needs to be as large as $\tilde{\Theta}(\sqrt{S/t})$. In contrast, the inclusion of auxiliary bonus γ in our algorithm allows a much smaller choice for bonus β—which scales only as $\tilde{\mathcal{O}}(\sqrt{1/t})$—while still maintaining valid confidence bounds. This technique alone brings down the sample complexity to $\tilde{\mathcal{O}}(H^4SAB/\epsilon^2)$, removing an entire S factor compared to VI-ULCB. Furthermore, the coefficient in γ is only c/H for some absolute constant c, which ensures that the introduction of error term γ would hurt the overall sample complexity only up to a constant factor.

Bernstein concentration. Our Nash-VI allows two choices of the bonus function $\beta = \textsc{Bonus}(t, \hat{\sigma}^2)$:

$$\begin{cases} \text{Hoeffding type:} \quad c(\sqrt{H^2\iota/t} + H^2S\iota/t), \\ \text{Bernstein type:} \quad c(\sqrt{\hat{\sigma}^2\iota/t} + H^2S\iota/t), \end{cases} \tag{3}$$

where $\hat{\sigma}^2$ is the estimated variance, ι is the logarithmic factors and c is absolute constant. The $\hat{\mathbb{V}}$ in line 8 is the empirical variance operator defined as $\widehat{\mathbb{V}}_h V = \widehat{\mathbb{P}}_h V^2 - (\widehat{\mathbb{P}}_h V)^2$ for any $V \in [0, H]^S$. The design of both bonuses stem from the Hoeffding and Bernstein concentration inequalities. Further, the Bernstein bonus uses a sharper concentration, which saves an H factor in sample complexity compared to the Hoeffding bonus (similar to the single-agent setting (Azar et al., 2017)). This further reduces the sample complexity to $\tilde{\mathcal{O}}(H^3SAB/\epsilon^2)$ which matches the lower bound in all H, S, ϵ factors.

Coarse Correlated Equilibrium (CCE). The prior algorithm VI-ULCB (Bai & Jin, 2020) computes the "greedy"

[3] We remark that the current policy is stochastic. This is different from the single-agent setting, where the algorithm only seeks to provide an upper bound of the value of the optimal policy where the optimal policy is not random. Due to this difference, the techniques of Azar et al. (2017) cannot be directly applied here.

Algorithm 1 Optimistic Nash Value Iteration (Nash-VI)

```
1: Initialize: for any $(s,a,b,h)$, $\overline{Q}_h(s,a,b) \leftarrow H$,
   $\underline{Q}_h(s,a,b) \leftarrow 0$, $\Delta \leftarrow H$, $N_h(s,a,b) \leftarrow 0$.
2: for episode $k = 1, \ldots, K$ do
3:   for step $h = H, H-1, \ldots, 1$ do
4:     for $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ do
5:       $t \leftarrow N_h(s,a,b)$.
6:       if $t > 0$ then
7:         $U_{h+1} \leftarrow (\overline{V}_{h+1} + \underline{V}_{h+1})/2$.
8:         $\beta \leftarrow \textsc{Bonus}(t, \widehat{\mathbb{V}}_h[U_{h+1}](s,a,b))$.
9:         $\gamma \leftarrow (c/H)\widehat{\mathbb{P}}_h(\overline{V}_{h+1} - \underline{V}_{h+1})(s,a,b)$.
10:        $\overline{Q}_h(s,a,b) \leftarrow \min\{(r_h + \widehat{\mathbb{P}}_h\overline{V}_{h+1})(s,a,b)$
                   $+ \gamma + \beta, H\}$.
11:        $\underline{Q}_h(s,a,b) \leftarrow \max\{(r_h + \widehat{\mathbb{P}}_h\underline{V}_{h+1})(s,a,b)$
                   $- \gamma - \beta, 0\}$.
12:    for $s \in \mathcal{S}$ do
13:      $\pi_h(\cdot,\cdot|s) \leftarrow \mathrm{CCE}(\overline{Q}_h(s,\cdot,\cdot), \underline{Q}_h(s,\cdot,\cdot))$.
14:      $\overline{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h}\overline{Q}_h)(s)$.
15:      $\underline{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h}\underline{Q}_h)(s)$.
16:  if $(\overline{V}_1 - \underline{V}_1)(s_1) < \Delta$ then
17:    $\Delta \leftarrow (\overline{V}_1 - \underline{V}_1)(s_1)$ and $\pi^{\mathrm{out}} \leftarrow \pi$.
18:  for step $h = 1, \ldots, H$ do
19:    take action $(a_h, b_h) \sim \pi_h(\cdot,\cdot|s_h)$, observe reward
       $r_h$ and next state $s_{h+1}$.
20:    add 1 to $N_h(s_h,a_h,b_h)$ and $N_h(s_h,a_h,b_h,s_{h+1})$.
21:    $\widehat{\mathbb{P}}_h(\cdot|s_h,a_h,b_h) \leftarrow$
             $N_h(s_h,a_h,b_h,\cdot)/N_h(s_h,a_h,b_h)$.
22: Output the marginal policies of $\pi^{\mathrm{out}}$: $(\mu^{\mathrm{out}}, \nu^{\mathrm{out}})$.
```

policy with respect to the estimated value functions by directly computing the Nash equilibrium for the Q-value at each step h. However, since the algorithm maintains both the upper confidence bound and lower confidence bound of the Q-value, this leads to the requirement to compute the Nash equilibrium for a two-player general-sum matrix game, which is in general PPAD-complete (Daskalakis, 2013).

To overcome this computational challenge, we compute a relaxation of the Nash equilibrium—*Coarse Correlated Equalibirum (CCE)*—instead, a technique first introduced by Xie et al. (2020) to address reinforcement learning problems in Markov Games. Formally, for any pair of matrices $\overline{Q}, \underline{Q} \in [0,H]^{A\times B}$, $\mathrm{CCE}(\overline{Q}, \underline{Q})$ returns a distribution $\pi \in \Delta_{\mathcal{A}\times\mathcal{B}}$ such that

$$\begin{cases} \mathbb{E}_{(a,b)\sim\pi}\overline{Q}(a,b) \geq \max_{a^\star} \mathbb{E}_{(a,b)\sim\pi}\overline{Q}(a^\star,b), \\ \mathbb{E}_{(a,b)\sim\pi}\underline{Q}(a,b) \leq \min_{b^\star} \mathbb{E}_{(a,b)\sim\pi}\underline{Q}(a,b^\star). \end{cases} \tag{4}$$

Intuitively, in a CCE the players choose their actions in a potentially correlated way such that no one can benefit from unilateral unconditional deviation. A CCE always exists, since Nash equilibrium is also a CCE and a Nash equilibrium always exists. Furthermore, a CCE can be computed by linear programming in polynomial time. We remark that different from Nash equilibrium where the policies of each player are independent, the policies given by CCE are in general correlated for each player. Therefore, executing such a policy (line 19) requires the cooperation of two players.

3.2. Theoretical guarantees

Now we are ready to present the theoretical guarantees for Algorithm 1. We let π^k denote the policy computed in line 13 in the k^{th} episode, and μ^k, ν^k denote the *marginal policy* of π^k for each player.

Theorem 3 (Nash-VI with Hoeffding bonus). *For any $p \in (0,1]$, letting $\iota = \log(SABT/p)$, then with probability at least $1-p$, Algorithm 1 with Hoeffding type bonus (3) (with some absolute $c > 0$) achieves:*

(a) The output policies (μ^{out}, ν^{out}) satisfy $(V_1^{\dagger,\nu^{out}} - V_1^{\mu^{out},\dagger})(s_1) \leq \epsilon$ if we choose

$$K \geq \Omega\left(\frac{H^4SAB\iota}{\epsilon^2} + \frac{H^3S^2AB\iota^2}{\epsilon}\right).$$

(b) The algorithm has regret bound

$$\mathrm{Regret}(K) = \sum_{k=1}^{K}(V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1) \leq \mathcal{O}(\sqrt{H^3SABT\iota} + H^3S^2AB\iota^2),$$

where $T = KH$ is the total number of steps played within K episodes.

Theorem 3 provides both a sample complexity bound and a regret bound for Nash-VI to find an ϵ-approximate Nash equilibrium. For small $\epsilon \leq H/(S\iota)$, the sample complexity scales as $\tilde{\mathcal{O}}(H^4SAB/\epsilon^2)$. Similarly, for large $T \geq H^3S^3AB\iota^3$, the regret scales as $\tilde{\mathcal{O}}(\sqrt{H^3SABT})$. Theorem 3 is significant in that it improves the sample complexity of the model-based algorithm in Markov games from S^2 to S (and the regret from S to $\sqrt{S}$). This is achieved by adding the new auxiliary bonus γ in value iteration steps as explained in Section 3.1. The proof of Theorem 3 can be found in Appendix D.1.

Our next theorem states that when using Bernstein bonus instead of Hoeffding bonus as in (3), the sample complexity of Nash-VI algorithm can be further improved by a H factor in the leading order term (and the regret improved by a $\sqrt{H}$ factor).

Theorem 4 (Nash-VI with the Bernstein bonus). *For any $p \in (0,1]$, letting $\iota = \log(SABT/p)$, then with probability at least $1-p$, Algorithm 1 with Bernstein type bonus (3) (with some absolute $c > 0$) achieves:*

- *The output policies (μ^{out}, ν^{out}) satisfy $(V_1^{\dagger,\nu^{out}} - V_1^{\mu^{out},\dagger})(s_1) \le \epsilon$ if we choose*

$$K \ge \Omega\left(\frac{H^3SAB\iota}{\epsilon^2} + \frac{H^3S^2AB\iota^2}{\epsilon}\right).$$

- *The algorithm has regret bound*

$$\begin{aligned}\mathrm{Regret}(K) &= \sum_{k=1}^{K}(V_1^{\dagger,\nu^k} - V_1^{\mu^k,\dagger})(s_1)\\ &\le \mathcal{O}(\sqrt{H^2SABT\iota} + H^3S^2AB\iota^2),\end{aligned}$$

  where $T = KH$ is the total number of steps played within K episodes.

Compared with the information-theoretic sample complexity lower bound $\Omega(H^3S(A+B)\iota/\epsilon^2)$ and regret lower bound $\Omega(\sqrt{H^2S(A+B)T})$ (Bai & Jin, 2020), when ϵ is small, Nash-VI with Bernstein bonus achieves the optimal dependency on all of H, S, ϵ up to logarithmic factors in both the sample complexity and the regret, and the only gap that remains open is a $AB/(A+B) \le \min\{A, B\}$ factor. The proof of Theorem 4 can be found in Appendix D.2.

Comparison with model-free approaches. Different from our model-based approach, a recently proposed model-free algorithm Nash V-Learning (Bai et al., 2020) achieves sample complexity $\tilde{\mathcal{O}}(H^6S(A+B)\iota/\epsilon^2)$, which has a tight $(A+B)$ dependency on A, B. However, our Nash-VI has the following important advantages over Nash V-Learning: 1. Our sample complexity has a better dependency on horizon H; 2. Our algorithm outputs a single pair of Markov policies $(\mu^{\text{out}}, \nu^{\text{out}})$ while their algorithm outputs a generic history-dependent policy that can be only written as a nested mixture of Markov policies; 3. The model-free algorithms in Bai et al. (2020) cannot be directly modified to obtain a $\sqrt{T}$-regret (so that the exploration policies can be arbitrarily poor), while our model-based algorithm has the $\sqrt{T}$-regret guarantee. We comment that although both Nash-VI and Nash V-Learning have polynomial running time, the latter enjoys a better computational complexity because Nash-VI requires to solve LPs for computing CCEs in each episode.

4. Reward-free Learning

In this section, we modify our model-based algorithm Nash-VI for the reward-free exploration setting. Formally, reward-free learning has two phases: In the exploration phase, the agent collects a dataset of transitions $\mathcal{D} = \{(s_{k,h}, a_{k,h}, b_{k,h}, s_{k,h+1})\}_{(k,h)\in[K]\times[H]}$ from a Markov game $\mathcal{M}$ without the guidance of reward information. After the exploration, in the planning phase, for each task $i \in [N]$, $\mathcal{D}$ is augmented with stochastic reward information to become $\mathcal{D}^i = \{(s_{k,h}, a_{k,h}, b_{k,h}, s_{k,h+1}, r_{k,h})\}_{(k,h)\in[K]\times[H]}$, where $r_{k,h}$ is sampled from some unknown reward distribution with expectation equal to $r_h^i(s_{k,h}, a_{k,h}, b_{k,h})$. Here, r^i denotes the unknown reward function of the i^{th} task. The goal is to compute nearly-optimal policies for N tasks under $\mathcal{M}$ simultaneously given the augmented datasets $\{\mathcal{D}^i\}_{i\in[N]}$.

Algorithm 2 Optimistic Value Iteration with Zero Reward (VI-Zero)

```
Require: Bonus $\beta_t$.
 1: Initialize: for any $(s,a,b,h)$, $\widetilde{V}_h(s,a,b) \leftarrow H$, $\Delta \leftarrow H$, $N_h(s,a,b) \leftarrow 0$.
 2: for episode $k = 1, \ldots, K$ do
 3:     for step $h = H, H-1, \ldots, 1$ do
 4:         for $(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$ do
 5:             $t \leftarrow N_h(s,a,b)$.
 6:             if $t > 0$ then
 7:                 $\widetilde{Q}_h(s,a,b) \leftarrow \min\{(\widehat{\mathbb{P}}_h\widetilde{V}_{h+1})(s,a,b) + \beta_t, H\}$.
 8:         for $s \in \mathcal{S}$ do
 9:             $\pi_h(s) \leftarrow \arg\max_{(a,b)\in\mathcal{A}\times\mathcal{B}} \widetilde{Q}_h(s,a,b)$.
10:             $\widetilde{V}_h(s) \leftarrow (\mathbb{D}_{\pi_h}\widetilde{Q}_h)(s)$.
11:     if $\widetilde{V}_1(s_1) < \Delta$ then
12:         $\Delta \leftarrow \widetilde{V}_1(s_1)$ and $\widehat{\mathbb{P}}^{\text{out}} \leftarrow \widehat{\mathbb{P}}$.
13:     for step $h = 1, \ldots, H$ do
14:         take action $(a_h, b_h) \sim \pi_h(\cdot,\cdot|s_h)$, observe next state $s_{h+1}$.
15:         add 1 to $N_h(s_h,a_h,b_h)$ and $N_h(s_h,a_h,b_h,s_{h+1})$.
16:         $\widehat{\mathbb{P}}_h(\cdot|s_h,a_h,b_h) \leftarrow N_h(s_h,a_h,b_h,\cdot)/N_h(s_h,a_h,b_h)$.
17: Output $\widehat{\mathbb{P}}^{\text{out}}$.
```

There are strong practical motivations for considering the reward-free setting. First, in applications such as robotics, we face multiple tasks in sequential systems with shared transition dynamics (i.e. the world) but very different rewards. There, we prefer to learn the underlying transition independent of reward information. Second, from the algorithm design perspective, decoupling exploration and planning (i.e. performing exploration without reward information) can be valuable for designing new algorithms in more challenging settings (e.g., with function approximation).

4.1. Algorithm description

We now describe our algorithm for reward-free learning in zero-sum Markov games.

Exploration phase. In the first phase of reward-free learning, we deploy algorithm Optimistic Value Iteration with Zero Reward (VI-Zero, Algorithm 2). This algorithm differs from the reward-aware Nash-VI (Algorithm 1) in two important aspects. First, we use zero reward in the exploration phase (Line 7), and only maintain an upper bound of the (reward-free) value function instead of both upper and lower bounds. Second, our exploration policy is the maximizing (instead of CCE) policy of the value function (Line 9). We remark that the $\widetilde{Q}_h(s,a,b)$ maintained in the algorithm 2 is no longer an upper bound for any actual value function (as it has no reward), but rather a measure of uncertainty or suboptimality that the agent may suffer—if she takes action (a,b) at state s and step h, and makes decisions by utilizing the empirical estimate $\widehat{\mathbb{P}}$ in the remaining steps (see a rigorous version of this statement in Lemma 27). Finally, the empirical transition $\widehat{\mathbb{P}}$ of the episode that minimizes $\widetilde{V}_1(s_1)$ is outputted and passed to the planning phase.

Planning phase. After obtaining the estimate of tranisiton $\widehat{\mathbb{P}}$, our planning algorithm is rather simple. For the i^{th} task, let $\widehat{r}^i$ be the empirical estimate of r^i computed using the i^{th} augmented dataset $\mathcal{D}^i$. Then we compute the Nash equilibrium of the Markov game $\mathcal{M}(\widehat{\mathbb{P}},\widehat{r}^i)$ with estimated transition $\widehat{\mathbb{P}}$ and reward $\widehat{r}^i$. Since both $\widehat{\mathbb{P}}$ and $\widehat{r}^i$ are known exactly, this is a pure computation problem without any sampling error and can be efficiently solved by simple planning algorithms such as the vanilla Nash value iteration without optimism (see Appendix E.2 for more details).

4.2. Theoretical guarantee

Now we are ready to state our theoretical guarantee for reward-free learning. It claims that the empirical transition $\widehat{\mathbb{P}}^{\text{out}}$ output by VI-Zero is close to the true transition $\mathbb{P}$, in the sense that any Nash equilibrium of the $\mathcal{M}(\widehat{\mathbb{P}},\widehat{r}^i)$ $(i \in [N])$ is also an approximate Nash equilibrium of the true underlying Markov game $\mathcal{M}(\mathbb{P},r^i)$, where $\widehat{r}^i$ is the empirical estimate of r^i computed using $\mathcal{D}^i$.

Theorem 5 (Sample complexity of VI-Zero)**.** *There exists an absolute constant c, for any $p \in (0,1]$, $\epsilon \in (0,H]$, $N \in \mathbb{N}$, if we choose bonus $\beta_t = c(\sqrt{H^2\iota/t} + H^2S\iota/t)$ with $\iota = \log(NSABT/p)$ and $K \geq c(H^4SAB\iota/\epsilon^2 + H^3S^2AB\iota^2/\epsilon)$, then with probability at least $1-p$, the output $\widehat{\mathbb{P}}^{\text{out}}$ of Algorithm 2 satisfies: For any N fixed reward functions $r^1,\ldots,r^N$, a Nash equilibrium of Markov game $\mathcal{M}(\widehat{\mathbb{P}}^{\text{out}},\widehat{r}^i)$ is also an ϵ-approximate Nash equilibrium of the true Markov game $\mathcal{M}(\mathbb{P},r^i)$ for all $i \in [N]$.*

Theorem 5 shows that, when ϵ is small, VI-Zero only needs $\tilde{\mathcal{O}}(H^4SAB/\epsilon^2)$ samples to learn an estimate of the transition $\widehat{\mathbb{P}}^{\text{out}}$, which is accurate enough to learn the approximate Nash equilibrium for any N fixed rewards. The most important advantage of reward-free learning comes from the sample complexity only scaling polylogarithmically with respect to the number of tasks or reward functions N. This is in sharp contrast to the reward-aware algorithms (e.g. Nash-VI), where the algorithm has to be rerun for each different task, and the total sample complexity must scale linearly in N. In exchange for this benefit, compared to Nash-VI, VI-Zero loses a factor of H in the leading term of sample complexity since we cannot use Bernstein bonus anymore due to the lack of reward information. VI-Zero also does not have a regret guarantee, since again without reward information, the exploration policies are naturally sub-optimal. The proof of Theorem 5 can be found in Appendix E.1.

Connections with reward-free learning in MDPs. Since MDPs are special cases of Markov games, our algorithm VI-Zero directly applies to the single-agent setting, and yields a sample complexity similar to existing results (Zhang et al., 2020b; Wang et al., 2020). However, distinct from existing results which require both the exploration algorithm and the planning algorithm to be specially designed to work together, our algorithm allows an arbitrary planning algorithm as long as it computes the Nash equilibrium of a Markov game with *known* transition and reward. Therefore, our results completely decouple the exploration and the planning.

Lower bound for reward-free learning. Finally, we comment that despite the sample complexity in Theorem 5 scaling as AB instead of $A+B$, our next theorem states that unlike the general reward-aware setting, this AB scaling is unavoidable in the reward-free setting. This reveals an intrinsic gap between the reward-free and reward-aware learning: An $A+B$ dependency is only achievable via sampling schemes that are reward-aware. A similar lower bound is also presented in Zhang et al. (2020a) for the discounted setting with a different hard instance construction.

Theorem 6 (Lower bound for reward-free learning of Markov games)**.** *There exists an absolute constant $c > 0$ such that for any $\epsilon \in (0,c]$, there exists a family of Markov games $\mathfrak{M}(\epsilon)$ satisfying that: for any reward-free algorithm $\mathfrak{A}$ using $K \leq cH^2SAB/\epsilon^2$ episodes, there exists a Markov game $\mathcal{M} \in \mathfrak{M}(\epsilon)$ such that if we run $\mathfrak{A}$ on $\mathcal{M}$ and output policies $(\hat{\mu},\hat{\nu})$, then with probability at least $1/4$, we have $(V_1^{\dagger,\hat{\nu}} - V_1^{\hat{\mu},\dagger})(s_1) \geq \epsilon$.*

This lower bound shows that the sample complexity in Theorem 5 is optimal in S, A, B, and ϵ. The proof of Theorem 6 can be found in Appendix E.3.

5. Multi-player general-sum games

We adapt our analysis to multi-player general-sum games and present the first lines of provably efficient algorithms. Concretely, we design two model-based algorithms Multi-Nash-VI and Multi-VI-Zero (Algorithm 3 and Algorithm 4) that can find an (ϵ-approximate) {NASH, CE, CCE} equilibrium for any multi-player general-sum Markov game in $\tilde{\mathcal{O}}(H^4S^2\prod_{i=1}^m A_i/\epsilon^2)$ episodes of game playing, where A_i is the number of actions for player $i \in \{1,\ldots,m\}$ (Theorem 15 and Theorem 16). Due to space limit, we defer the detailed setups, algorithms and results to Appendix A.

6. Conclusion

In this paper, we provided a sharp analysis of model-based algorithms for Markov games. Our new algorithm Nash-VI can find an ϵ-approximate Nash equilibrium of a zero-sum Markov game in $\tilde{\mathcal{O}}(H^3SAB/\epsilon^2)$ episodes of game playing, which almost matches the sample complexity lower bound except for the AB vs. $A+B$ dependency. We also applied our analysis to derive new efficient algorithms for task-agnostic game playing, as well as the first line of multi-player general-sum Markov games. There are a number of compelling future directions to this work. For example, can we achieve $A+B$ instead of AB sample complexity for zero-sum games using model-based approaches (thus closing the gap between lower and upper bounds)? How can we design more efficient algorithms for general-sum games with better sample complexity (e.g., $\mathcal{O}(S)$ instead of $\mathcal{O}(S^2)$)? We leave these problems as future work.

References

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272, 2017.

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

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

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkxpxJBKwS.

Berg, K. and Sandholm, T. Exclusion method for finding nash equilibrium in multi-player games. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1417–1418, 2016.

Brafman, R. I. and Tennenholtz, M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct): 213–231, 2002.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5713–5723, 2017.

Daskalakis, C. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):23, 2013.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. *arXiv preprint arXiv:2010.03531*, 2020.

Filar, J. and Vrieze, K. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Hansen, T. D., Miltersen, P. B., and Zwick, U. 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.

Hu, J. and Wellman, M. P. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

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

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4868–4878, 2018.

Jin, C., Jin, T., Luo, H., Sra, S., and Yu, T. Learning adversarial markov decision processes with bandit feedback and unknown transition. *arXiv preprint arXiv:1912.01192*, 2019.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. *arXiv preprint arXiv:2002.02794*, 2020.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

Lattimore, T. and Szepesvári, C. Bandit algorithms. 2018.

Lee, C.-W., Luo, H., Wei, C.-Y., and Zhang, M. Linear last-iterate convergence for matrix games and stochastic games. *arXiv preprint arXiv:2006.09517*, 2020.

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

Littman, M. L. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pp. 322–328, 2001.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. *Algorithmic Game Theory*. Cambridge University Press, 2007.

OpenAI. Openai five. https://blog.openai.com/openai-five/, 2018.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

Rosenberg, A. and Mansour, Y. Online convex optimization in adversarial markov decision processes. *arXiv preprint arXiv:1905.07773*, 2019.

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

Sidford, A., Wang, M., Yang, L. F., and Ye, Y. Solving discounted stochastic two-player games with near-optimal time and sample complexity. *arXiv preprint arXiv:1908.11071*, 2019.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, pp. 881–888, 2006.

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

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*, 2020.

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

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

Yadkori, Y. A., Bartlett, P. L., Kanade, V., Seldin, Y., and Szepesvári, C. Online learning in markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems*, pp. 2508–2516, 2013.

Zhang, K., Kakade, S. M., Başar, T., and Yang, L. F. Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461*, 2020a.

Zhang, X., Ma, Y., and Singla, A. Task-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2006.09497*, 2020b.

Zhang, Z., Zhou, Y., and Ji, X. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *arXiv preprint arXiv:2004.10019*, 2020c.

Zimin, A. and Neu, G. Online learning in episodic markovian decision processes by relative entropy policy search. In *Advances in neural information processing systems*, pp. 1583–1591, 2013.