On Improving Model-Free Algorithms for Decentralized Multi-Agent Reinforcement Learning

Weichao Mao [1] **Lin F. Yang** [2] **Kaiqing Zhang** [3] **Tamer Başar** [1]

Abstract

Multi-agent reinforcement learning (MARL) algorithms often suffer from an exponential sample complexity dependence on the number of agents, a phenomenon known as *the curse of multiagents*. We address this challenge by investigating sample-efficient model-free algorithms in *decentralized* MARL, and aim to improve existing algorithms along this line. For learning (coarse) correlated equilibria in general-sum Markov games, we propose *stage-based* V-learning algorithms that significantly simplify the algorithmic design and analysis of recent works, and circumvent a rather complicated no-*weighted*-regret bandit subroutine. For learning Nash equilibria in Markov potential games, we propose an independent policy gradient algorithm with a decentralized momentum-based variance reduction technique. All our algorithms are decentralized in that each agent can make decisions based on only its local information. Neither communication nor centralized coordination is required during learning, leading to a natural generalization to a large number of agents. Finally, we provide numerical simulations to corroborate our theoretical findings.

1. Introduction

Many real-world sequential decision-making problems involve the strategic interactions of multiple agents in a shared environment, which are commonly addressed with multi-agent reinforcement learning (MARL). Successful applications of MARL include playing the game of Go (Silver et al., 2016), Poker (Brown & Sandholm, 2018), real-time strategy games (Vinyals et al., 2019), autonomous driving (Shalev-Shwartz et al., 2016), and robotics (Kober et al., 2013).

Despite the empirical successes, sample-efficient solutions are still relatively lacking for MARL with a large number of agents, mostly due to the well-known challenge named *the curse of multiagents* (Jin et al., 2021): The joint action space in a MARL problem is equal to the Cartesian product of the individual action spaces of all agents, which scales exponentially in the number of agents. A typical kind of algorithms that easily fail at this challenge are those using centralized/joint learning (Boutilier, 1996; Claus & Boutilier, 1998). Specifically, centralized learning assumes the existence of a single coordinator who can access the local information of all the agents, and learns policies jointly for all of them. This centralized training (though possibly decentralized execution) approach has become a common practice in empirical MARL (Oliehoek et al., 2008; Foerster et al., 2016; Lowe et al., 2017; Rashid et al., 2018; Son et al., 2019; Mao et al., 2020a). Centralized learning essentially reduces the multi-agent problem to a single-agent one, but unfortunately suffers from the exponential dependence as it usually needs to exhaustively search the joint action space.

Such a computation bottleneck can be partially resolved by allowing communications among the agents and hence distributing the workload to each of them (Kar et al., 2013; Zhang et al., 2018; Dubey & Pentland, 2021). However, communication-based methods instead suffer from the additional communication overheads, which can be unrealistic in some real-world scenarios where communication may be expensive and/or unreliable, such as in unmanned aerial vehicle (UAV) field coverage (Pham et al., 2018).

Given the aforementioned limitations, in this paper, we are interested in a more practical setting: *decentralized* learning[1]. We focus on solutions where each agent can make

[1]Department of Electrical and Computer Engineering & Coordinated Science Laboratory, University of Illinois Urbana-Champaign. [2]Department of Electrical and Computer Engineering, University of California, Los Angeles. Part of this work done while the author was visiting DeepMind. [3]Laboratory for Information & Decision Systems, Massachusetts Institute of Technology. Part of this work done while the author was visiting Simons Institute for the Theory of Computing. Correspondence to: W. Mao <weichao2@illinois.edu>, L. F. Yang <linyang@ee.ucla.edu>, K. Zhang <kaiqing@mit.edu>, T. Başar <basar1@illinois.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1] This setting has been studied under various names in the literature, including individual learning (Leslie & Collins, 2005), decentralized learning (Arslan & Yüksel, 2016), agnostic learning (Tian et al., 2021; Wei et al., 2021), and independent learning (Claus &

decisions based on only its local information (e.g., local actions and rewards), and need not communicate with its opponents or be coordinated by any central controller during learning. In fact, in our algorithms, the agents can be completely oblivious to the presence of other agents. Under such weak assumptions, decentralized algorithms are suitable for many practical MARL scenarios (Fudenberg et al., 1998), and do not suffer from the exponential sample & computation complexity. Such algorithms are naturally model-free, as they do not maintain explicit estimates of the transition functions. Compared with model-based algorithms, model-free ones typically enjoy higher time- and space-efficiency, and are more compatible with the modern deep RL architectures (Jin et al., 2018; Zhang et al., 2020b).

In this paper, we investigate the theoretical aspects of decentralized MARL in the non-asymptotic regime. We address the curse of multiagents by presenting sample-efficient model-free algorithms that scale to a large number of agents, and aim to improve the existing algorithms along this line. Our main contributions are summarized as follows.

Contributions. 1) For general-sum Markov games (Section 3), we present algorithms that learn an ε-approximate coarse correlated equilibrium (CCE) in $\widetilde{O}(H^5 S A_{\max}/\varepsilon^2)$ episodes, and an ε-approximate correlated equilibrium (CE) in $\widetilde{O}(H^5 S A_{\max}^2/\varepsilon^2)$ episodes, where S is the number of states, $A_{\max}$ is the size of the largest individual action space, and H is the length of an episode. Our algorithms rely on a novel *stage-based* V-learning method that significantly simplifies the algorithmic design and analysis of recent works. 2) In the important special case of Markov potential games (MPGs, Section 4), we propose an independent policy gradient algorithm that learns an ε-approximate Nash equilibrium (NE) in $\propto \widetilde{O}(1/\varepsilon^{4.5})$ episodes. Our algorithm utilizes a momentum-based variance reduction technique that can be executed in a decentralized way. 3) We further provide numerical results that corroborate our theoretical findings (Section 5). All our algorithms are decentralized and model-free, and readily generalize to a large number of agents.

Related Work. A common mathematical framework of MARL is stochastic games (Shapley, 1953), which are also referred to as Markov games. Early attempts to learn NE in Markov games include Littman (1994; 2001); Hu & Wellman (2003); Hansen et al. (2013), but they either assume the transition kernel and rewards are known, or only yield asymptotic guarantees. Recently, various sample-efficient methods have been proposed (Wei et al., 2017; Bai & Jin, 2020; Sidford et al., 2020; Xie et al., 2020; Bai et al., 2020; Liu et al., 2021; Zhao et al., 2021; Guo et al., 2021), mostly for learning in two-player zero-sum Markov games. Several works have investigated zero-sum games in a *decentralized* setting as we consider here (Daskalakis et al., 2020; Tian et al., 2021; Wei et al., 2021; Sayin et al., 2021), but these results do not carry over in any way to general-sum games or MPGs. We refer the reader to Appendix A for a more detailed discussion on these related works.

For general-sum games, Rubinstein (2016) has shown a sample complexity lower bound for learning NE that is exponential in the number of agents. Recently, Liu et al. (2021) has presented a line of results on learning NE, CE, or CCE, but their algorithm is model-based, and suffers from such exponential dependence. Song et al. (2021); Jin et al. (2021); Mao & Başar (2022) have proposed V-learning based methods for learning CCE and/or CE, and our stage-based V-learning significantly simplifies the algorithmic design and analysis along this line. Learning CE and CCE has also been extensively studied in normal-form games with no state transitions (Hart & Mas-Colell, 2000; Cesa-Bianchi & Lugosi, 2006; Blum & Mansour, 2007).

Another line of research (Macua et al., 2018; Mguni et al., 2021; Ding et al., 2022) has considered learning in Markov potential games. Arslan & Yüksel (2016) has shown that decentralized Q-learning can converge to NE in weakly acyclic games, which cover potential games as a special case. Their algorithm requires a coordinated exploration phase, and only yields asymptotic guarantees. Two recent works (Zhang et al., 2021; Leonardos et al., 2021) have proposed independent policy gradient methods in MPGs, which are most relevant to ours. We improve their sample complexity dependence on ε by utilizing decentralized variance reduction, and we do not require the two-timescale framework to coordinate policy evaluation as in Zhang et al. (2021). Fox et al. (2021) has shown that independent natural policy gradient also converges to NE, though only asymptotic convergence has been established. Finally, MPGs have also been studied in Song et al. (2021), but their model-based method is not decentralized, and requires the agents to take turns to learn the policies.

2. Preliminaries

An N-player episodic Markov game is defined by a tuple $(\mathcal{N}, H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{r_i\}_{i=1}^N, P)$, where (1) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $\mathcal{S}$ is the finite state space; (4) $\mathcal{A}_i$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r_i : [H] \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function for agent i, where $\mathcal{A} = \times_{i=1}^N \mathcal{A}_i$ is the joint action (or action profile) space; and (6) $P : [H] \times \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel. We remark that both the reward function and the state transition function depend on the joint actions of all the agents. We assume for simplicity that the reward function is deterministic. Our results can be easily

Boutilier, 1998; Daskalakis et al., 2020). It also belongs to a more general category of teams/games with decentralized information structure (Ho, 1980; Nayyar et al., 2013a;b).

generalized to stochastic reward functions. Let $S = |\mathcal{S}|$, $A_i = |\mathcal{A}_i|, \forall i \in \mathcal{N}$, and $A_{\max} = \max_{i \in \mathcal{N}} A_i$.

The agents interact in an unknown environment for K episodes. We assume that the initial state s_1 of the environment follows a fixed distribution $\rho \in \Delta(\mathcal{S})$. At each time step $h \in [H]$, the agents observe the state $s_h \in \mathcal{S}$, and take actions $a_{h,i} \in \mathcal{A}_i, i \in \mathcal{N}$ simultaneously. Agent i then receives its private reward $r_{h,i}(s_h, \boldsymbol{a}_h)$, where $\boldsymbol{a}_h = (a_{h,1}, \ldots, a_{h,N})$, and the environment transitions to the next state $s_{h+1} \sim P_h(\cdot | s_h, \boldsymbol{a}_h)$. Note that the state transition here is general and not restricted to be deterministic. This makes decentralized learning considerably more challenging, as the agents cannot implicitly coordinate by enumerating/rehearsing all possible states. We focus on the *decentralized* setting, where each agent only observes the states and its own rewards and actions, but not the rewards or actions of the other agents. In fact, in our algorithms, each agent is completely oblivious of the existence of the others, and does not communicate with each other. This decentralized information structure requires each agent to learn to make decisions based on only its local information.

Policy and value function. A (Markov) policy $\pi_i : [H] \times \mathcal{S} \to \Delta(\mathcal{A}_i)$ for agent $i \in \mathcal{N}$ is a mapping from the time index and state space to a distribution over its own action space. We use Π_i to denote the space of Markov policies for agent i, and let $\Pi = \times_{i=1}^{N} \Pi_i$. Each agent seeks to find a policy that maximizes its own cumulative reward. A joint policy (or policy profile) $\pi = (\pi_1, \ldots, \pi_N)$ induces a probability measure over the sequence of states and joint actions. For notational convenience, we use the subscript $-i$ to denote the set of agents excluding agent i, i.e., $\mathcal{N} \backslash \{i\}$. For example, we can rewrite $\pi = (\pi_i, \pi_{-i})$ using this convention. For a policy profile π, and for any $h \in [H]$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$, we define the value function and the state-action value function (or Q-function) for agent i as follows:

$$V^{\pi}_{h,i}(s) \overset{\text{def}}{=} \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'}) \mid s_h = s\right], \qquad (1)$$

$$Q^{\pi}_{h,i}(s, \boldsymbol{a}) \overset{\text{def}}{=} \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'}) \mid s_h = s, \boldsymbol{a}_h = \boldsymbol{a}\right].$$

For ease of notation, we also write $V^{(\pi_i, \pi_{-i})}_{h,i}(s)$ as $V^{\pi_i, \pi_{-i}}_{h,i}(s)$, and similarly for $Q^{(\pi_i, \pi_{-i})}_{h,i}(s, a)$.

Best response and Nash equilibrium. For agent i, a policy $\pi_i^\star$ is a *best response* to π_{-i} for a given initial state s_1 if $V^{\pi_i^\star, \pi_{-i}}_{1,i}(s_1) = \sup_{\pi_i} V^{\pi_i, \pi_{-i}}_{1,i}(s_1)$. A policy profile $\pi = (\pi_i, \pi_{-i}) \in \Pi$ is a *Nash equilibrium* (NE) if π_i is a best response to π_{-i} for all $i \in \mathcal{N}$. We also have an approximate notion of Nash equilibrium as follows:

Definition 1. *(ε-approximate Nash equilibrium). For any $\varepsilon > 0$, a policy profile $\pi = (\pi_i, \pi_{-i}) \in \Pi$ is an ε-approximate Nash equilibrium for an initial state s_1 if $V^{\pi_i, \pi_{-i}}_{1,i}(s_1) \geq \sup_{\pi_{i'}} V^{\pi_{i'}, \pi_{-i}}_{1,i}(s_1) - \varepsilon$, $\forall i \in \mathcal{N}$.*

Markov potential game. One particular subclass of games that we are interested in is the Markov potential game. Specifically, an episodic Markov game is an MPG if there exists a global potential function $\Phi_s : \Pi \to [0, \Phi_{\max}]$ for every initial state $s \in \mathcal{S}$, such that for any $i \in \mathcal{N}$, any $\pi_i, \pi_{i'} \in \Pi_i$, and any $\pi_{-i} \in \Pi_{-i}$,

$$\Phi_s(\pi_i, \pi_{-i}) - \Phi_s(\pi_{i'}, \pi_{-i}) = V^{\pi_i, \pi_{-i}}_{1,i}(s) - V^{\pi_{i'}, \pi_{-i}}_{1,i}(s). \qquad (2)$$

Our definition of MPG follows Song et al. (2021), which in turn is a variant of the definitions introduced in Macua et al. (2018); Leonardos et al. (2021); Zhang et al. (2021). It follows immediately that MPGs cover Markov teams (Lauer & Riedmiller, 2000) as a special case, a cooperative setting where all agents share the same reward function.

Correlated policy. More generally, we define $\pi = \{\pi_h : \mathbb{R} \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \Delta(\mathcal{A})\}_{h \in [H]}$ as a (non-Markov) *correlated policy*, where for each $h \in [H]$, π_h maps from a random variable $z \in \mathbb{R}$ and a history of length $h - 1$ to a distribution over the joint action space. We assume that the agents following a correlated policy can access a common source of randomness (e.g., a common random seed) for the random variable z. We let π_i and π_{-i} be the proper marginal policies of π whose outputs are restricted to $\Delta(\mathcal{A}_i)$ and $\Delta(\mathcal{A}_{-i})$, respectively.

For non-Markov correlated policies, we can still define their value functions at step $h = 1$ in a sense similar to (1). A best response $\pi_i^\star$ with respect to the non-Markov policies π_{-i} is a policy (independent of the randomness of π_{-i}) that maximizes agent i's value at step 1, i.e., $V^{\pi_i^\star, \pi_{-i}}_{1,i}(s_1) = \sup_{\pi_i} V^{\pi_i, \pi_{-i}}_{1,i}(s_1)$. The best response to the non-Markov policies of the opponents is not necessarily Markov.

(Coarse) correlated equilibrium. Given the PPAD-hardness of calculating Nash equilibria in general-sum games (Daskalakis et al., 2009), we introduce two relaxed solution concepts, namely coarse correlated equilibrium (CCE) and correlated equilibrium (CE). A CCE states that no agent has the incentive to deviate from a correlated policy π by playing a different independent policy.

Definition 2. *(CCE). A correlated policy π is an ε-approximate coarse correlated equilibrium for an initial state s_1 if $V^{\pi_i^\star, \pi_{-i}}_{1,i}(s_1) - V^{\pi}_{1,i}(s_1) \leq \varepsilon, \forall i \in \mathcal{N}$.*

CCE relaxes NE by allowing possible correlations in the policies. Before introducing the definition of CE, we need to first specify the concept of a strategy modification.

Definition 3. *(Strategy modification). For agent i, a strategy modification $\psi_i = \{\psi^s_{h,i} : h \in [H], s \in \mathcal{S}\}$ is a set of mappings from agent i's action space to itself, i.e., $\psi^s_{h,i} : \mathcal{A}_i \to \mathcal{A}_i$.*

Algorithm 1: Stage-Based V-Learning for CCE (agent i)

```
1  Initialize: $\overline{V}_{h,i}(s) \leftarrow H-h+1, \tilde{V}_{h,i}(s) \leftarrow H-h+1, N_h(s) \leftarrow 0, \check{N}_h(s) \leftarrow 0, \check{r}_{h,i}(s) \leftarrow 0, \check{v}_{h,i}(s) \leftarrow 0,$
   $\check{T}_h(s) \leftarrow H, \mu_{h,i}(a \mid s) \leftarrow 1/A_i$, and $L_{h,i}(s,a) \leftarrow 0, \forall h \in [H], s \in \mathcal{S}, a \in \mathcal{A}_i$.
2  for episode $k \leftarrow 1$ to $K$ do
3      Receive $s_1$;
4      for step $h \leftarrow 1$ to $H$ do
5          $N_h(s_h) \leftarrow N_h(s_h) + 1, \check{n} \overset{\text{def}}{=} \check{N}_h(s_h) \leftarrow \check{N}_h(s_h) + 1$;
6          Take action $a_{h,i} \sim \mu_{h,i}(\cdot \mid s_h)$, and observe reward $r_{h,i}$ and next state $s_{h+1}$;
7          $\check{r}_{h,i}(s_h) \leftarrow \check{r}_{h,i}(s_h) + r_{h,i}, \check{v}_{h,i}(s_h) \leftarrow \check{v}_{h,i}(s_h) + \overline{V}_{h+1,i}(s_{h+1})$;
8          $\eta_i \leftarrow \sqrt{\iota / A_i \check{T}_h(s_h)}, \gamma_i \leftarrow \eta_i/2$;
9          $L_{h,i}(s_h, a_{h,i}) \leftarrow L_{h,i}(s_h, a_{h,i}) + \frac{[H-h+1-(r_{h,i}+\overline{V}_{h+1,i}(s_{h+1}))]/H}{\mu_{h,i}(a_{h,i}|s_h)+\gamma_i}$;
10         $\mu_{h,i}(a \mid s_h) \leftarrow \frac{\exp(-\eta_i L_{h,i}(s_h,a))}{\sum_{a' \in \mathcal{A}_i} \exp(-\eta_i L_{h,i}(s_h,a'))}, \forall a \in \mathcal{A}_i$;
11         if $N_h(s_h) \in \mathcal{L}$ then
12             //Entering a new stage
13             $\tilde{V}_{h,i}(s_h) \leftarrow \frac{\check{r}_{h,i}(s_h)}{\check{n}} + \frac{\check{v}_{h,i}(s_h)}{\check{n}} + b_{\check{n}}$, where $b_{\check{n}} \leftarrow 6\sqrt{H^2 A_i \iota / \check{n}}$;
14             $\overline{V}_{h,i}(s_h) \leftarrow \min\{\tilde{V}_{h,i}(s_h), H-h+1\}$;
15             $\check{N}_h(s_h) \leftarrow 0, \check{r}_{h,i}(s_h) \leftarrow 0, \check{v}_{h,i}(s_h) \leftarrow 0, \check{T}_h(s_h) \leftarrow \lfloor (1+\frac{1}{H})\check{T}_h(s_h) \rfloor$;
16             $\mu_{h,i}(a \mid s_h) \leftarrow 1/A_i, L_{h,i}(s_h,a) \leftarrow 0, \forall a \in \mathcal{A}_i$;
```

Given a strategy modification ψ_i, for any policy π, step h and state s, if π selects the joint action $\boldsymbol{a}_h = (a_{h,1}, \dots, a_{h,N})$, then the modified policy $\psi_i \diamond \pi$ will select $(a_{h,1}, \dots, a_{h,i-1}, \psi_{h,i}^s(a_{h,i}), a_{h,i+1}, \dots, a_{h,N})$. Let Ψ_i denote the set of all possible strategy modifications for agent i. A CE is a distribution where no agent has the incentive to deviate from a correlated policy π by using any strategy modification. It is known that {NE}⊂{CE}⊂{CCE} in general-sum games (Nisan et al., 2007).

Definition 4. *(CE). A correlated policy π is an ε-approximate correlated equilibrium for initial state s_1 if*

$$\sup_{\psi_i \in \Psi_i} V_{1,i}^{\psi_i \diamond \pi}(s_1) - V_{1,i}^{\pi}(s_1) \le \varepsilon, \forall i \in \mathcal{N}.$$

3. Stage-Based V-Learning for General-Sum Markov Games

In this section, we introduce our stage-based V-learning algorithms for learning CCE and CE in general-sum Markov games, and establish their sample complexity guarantees.

3.1. Learning CCE

The Stage-Based V-Learning for CCE algorithm run by agent $i \in \mathcal{N}$ is presented in Algorithm 1. The agent maintains upper confidence bounds on the value functions to actively explore the unknown environment, and uses a stage-based rule to independently update the value estimates.

For each step-state pair $(h, s) \in [H] \times \mathcal{S}$, we divide the visitations to this pair into multiple *stages*, where the lengths of the stages increase exponentially at a rate of $(1 + 1/H)$ (Zhang et al., 2020b). Specifically, we let $e_1 = H$, and $e_{i+1} = \lfloor (1 + 1/H)e_i \rfloor, i \ge 1$ denote the lengths of the stages, and let the partial sums $\mathcal{L} \overset{\text{def}}{=} \{\sum_{i=1}^{j} e_i \mid j = 1, 2, 3, \dots\}$ denote the set of ending times of the stages. For each (h, s) pair, we update our optimistic estimates $\overline{V}_h(s_h)$ of the value function at the end of each stage (i.e., when the total number of visitations to (s, h) lies in the set $\mathcal{L}$), using samples only from this single stage (Lines 11-16). This way, our stage-based V-learning ensures that only the most recent $O(1/H)$ fraction of the collected samples are used to calculate $\overline{V}_h(s_h)$, while the first $1 - O(1/H)$ fraction is forgotten. Such a stage-based update framework in some sense mimics the celebrated optimistic Q-learning algorithm with a learning rate of $\alpha_t = \frac{H+1}{H+t}$ (Jin et al., 2018), which also roughly uses the last $O(1/H)$ fraction of samples for value updates. Stage-based value updates also create a stage-wise stationary environment for the agents, thereby partly alleviating the well-known challenge of *non-stationarity* in MARL. As a side remark, stage-based Q-learning has also achieved near-optimal regret bounds in single-agent RL (Zhang et al., 2020b).

At each time step h and state s_h, agent i selects its action $a_{h,i}$ by following a distribution $\mu_{h,i}(\cdot \mid s_h)$, where $\mu_{h,i}(\cdot \mid s_h)$ is updated using an adversarial bandit subroutine (Lines 9-10). This is consistent with the recent works under the V-learning framework (Jin et al., 2021; Song et al., 2021; Mao & Başar, 2022), but with a vital improvement: Existing works using the celebrated $\alpha_t = \frac{H+1}{H+t}$ learning rate for V-learning inevitably entail a no-*weighted*-regret bandit problem, because such a time-varying learning rate assigns different weights to each step in the history. A few methods such as weighted follow-the-regularized-leader (Jin et al., 2021; Song et al., 2021) and stabilized online mirror descent (Mao & Başar, 2022) have been recently proposed to address such a challenge, by simultaneously dealing with a changing step size, a weighted regret, and a high-probability guarantee, at the cost of less natural algorithms and more sophisticated analyses. In contrast, our stage-based V-learning assigns uniform weights to each step in the previous stage, and hence leads to a standard no(-average)-regret bandit problem. This allows us to directly plug in any off-the-shelf adversarial bandit algorithm and its analysis to our problem. For example, Algorithm 1 utilizes a simple Exp3 (Auer et al., 2002) subroutine for policy updates, and a standard implicit exploration technique (Neu, 2015) to achieve high-probability guarantees. We provide a more detailed discussion on such an improvement in Remark 1 of Appendix C.

Based on the policy trajectories from Algorithm 1, we construct an output policy profile $\bar{\pi}$ that we will show is a CCE. For any step $h \in [H]$ of an episode $k \in [K]$ and any state $s \in \mathcal{S}$, we let $\mu^k_{h,i}(\cdot \mid s) \in \Delta(\mathcal{A}_i)$ be the distribution prescribed by Algorithm 1 at this step. Let $\check{N}^k_h(s)$ denote the value of $\check{N}_h(s)$ at the *beginning* of the k-th episode. Our construction of the output policy is presented in Algorithm 2, which follows the "certified policies" introduced in Bai et al. (2020). We further let the agents sample the episode indices using a common source of randomness, and hence the output policy is correlated by nature. Such common randomness is also termed a correlation device, and is standard in decentralized learning (Bernstein et al., 2009; Arabneydi & Mahajan, 2015; Zhang et al., 2019). In practice, this can be achieved by letting the agents agree on a common random seed at the very beginning of the game, which only requires exchanging a single scalar value. Note that the correlation device is never used during the learning process to coordinate the exploration, but is simply used to synchronize the selection of the policies after they have been generated. A common random seed is generally considered as a mild assumption and does not break the decentralized paradigm. It is also worth remarking that our stage-based update rule simplifies the generating procedure of the output policy: In the original construction of Bai et al. (2020), the certified policy plays a weighted mixture of $\{\mu^k_{h,i}(\cdot \mid s) : k \in [K]\}$, while in Algorithm 2, we only need to uniformly sample an episode index from the previous stage.

The following theorem presents the sample complexity guarantee of Algorithm 1 for learning CCE in general-sum Markov games. Our sample complexity bound improves over Mao & Başar (2022) and matches those established in Song et al. (2021); Jin et al. (2021), while significantly simplifying their algorithmic design and analysis. The proof is deferred to Appendix C due to space limitations.

Algorithm 2: Construction of the Output Policy $\bar{\pi}$

```
Input: The distribution trajectory specified by
  Algorithm 1: $\{\mu^k_{h,i} : i \in \mathcal{N}, h \in [H], k \in [K]\}$;
Uniformly sample $k$ from $[K]$;
for step $h \leftarrow 1$ to $H$ do
    Receive $s_h$;
    Take joint action $\boldsymbol{a}_h \sim \times_{i=1}^N \mu^k_{h,i}(\cdot \mid s_h)$;
    Uniformly sample $j$ from $\{1, 2, \ldots, \check{N}^k_h(s_h)\}$;
    Set $k \leftarrow \check{l}^{k'}_{h,j}$, where $\check{l}^k_{h,j}$ is the index of the episode
      such that state $s_h$ was visited the $j$-th time (among
      the total $\check{N}^k_h(s_h)$ times) in the last stage;
```

Theorem 1. *(Sample complexity of learning CCE). For any $p \in (0, 1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 1 for K episodes with $K = O(SA_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1 - p$, the output policy $\bar{\pi}$ of Algorithm 2 is an ε-approximate CCE.*

3.2. Learning CE

In this subsection, we aim at learning a more strict solution concept named correlated equilibrium. Our algorithm for learning CE (a complete description presented in Algorithm 6 of Appendix D) also relies on stage-based V-learning, but replaces the no-regret learning subroutine in Algorithm 1 with a no-swap-regret learning algorithm. Our no-swap-regret algorithm follows the generic reduction introduced in Blum & Mansour (2007), and converts a follow-the-regularized-leader (FTRL) algorithm with sublinear external regret to a no-swap-regret algorithm (Jin et al., 2021). A detailed description of such a no-swap-regret FTRL subroutine as well as its regret analysis is presented in Appendix D. Again, due to the stage-based update rule, we can avoid the additional complication of dealing with a weighted swap regret as faced by recent works (Jin et al., 2021; Song et al., 2021). The construction of the output policy $\bar{\pi}$ is the same as Algorithm 2 and thus omitted. The following theorem shows that our sample complexity guarantee for learning CE improves over Song et al. (2021) and matches the best known result in the literature (Jin et al., 2021). The proof of the theorem can also be found in Appendix D.

Theorem 2. *(Sample complexity of learning CE). For any $p \in (0, 1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 6 for K episodes with $K = O(SA^2_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1 - p$, the output policy $\bar{\pi}$ is an ε-approximate CE.*

As a final remark, notice that both the V-learning and the no-regret learning components of our algorithms are decentralized, which can be implemented using only the states observed and the local action and reward information, without any communication or central coordination among the agents. In addition, the sample complexity of our algorithms only depend on $A_{\max}$ instead of $\prod_{i=1}^{N} A_i$. This allows our methods to easily generalize to a large number of agents.

4. Learning NE in Markov Potential Games

In this section, we present an algorithm for learning Nash equilibria in decentralized Markov potential games, an important subclass of Markov games. Motivated by Leonardos et al. (2021); Zhang et al. (2021), we utilize a policy gradient method, where each agent independently runs a projected gradient ascent (PGA) algorithm to update their policies. We start from the case where the policy gradients can be calculated exactly (using an infinite number of samples), and then move to the more practical case where the gradients are estimated using finite samples.

We first introduce a few notations for ease of presentation. Let $d_{h,\rho}^{\pi}(s)$ be the probability of visiting state s at step h by following policy π starting from the initial state distribution ρ, i.e., $d_{h,\rho}^{\pi}(s) \overset{\text{def}}{=} \mathbb{P}^{\pi}(s_h = s \mid s_1 \sim \rho)$. We also overload the notations of the value function and the potential function, and write $V_{1,i}^{\pi}(\rho) \overset{\text{def}}{=} \mathbb{E}_{s_1 \sim \rho}\left[V_{1,i}^{\pi}(s_1)\right]$ and $\Phi_\rho(\pi) = \mathbb{E}_{s\sim\rho}\left[\Phi_s(\pi)\right]$. We further introduce the following variant of the distribution mismatch coefficient (Agarwal et al., 2021) to characterize the difficulty of exploration.

Definition 5. *(Finite-horizon distribution mismatch coefficient). Given two policies $\pi, \pi' \in \Pi$ and an initial state distribution $\rho \in \Delta(\mathcal{S})$, we define*

$$\left\|\frac{d_{\rho}^{\pi'}}{d_{\rho}^{\pi}}\right\|_{\infty} \overset{\text{def}}{=} \max_{h\in[H], s_h \in \mathcal{S}} \frac{d_{h,\rho}^{\pi'}(s_h)}{d_{h,\rho}^{\pi}(s_h)}, \text{ and } D \overset{\text{def}}{=} \max_{\pi,\pi'} \left\|\frac{d_{\rho}^{\pi'}}{d_{\rho}^{\pi}}\right\|_{\infty}.$$

4.1. Exact Gradient Estimates

The PGA algorithm updates the policy as follows:

$$\pi_i^{(t+1)} \leftarrow \mathrm{Proj}_{\Pi_i}\left(\pi_i^{(t)} + \eta \nabla_{\pi_i} V_{1,i}^{\pi^{(t)}}(\rho)\right), \qquad (3)$$

where $\pi_i^{(t)}$ is the policy of agent i at the t-th iteration, Proj_{Π_i} denotes the Euclidean projection onto Π_i, and $\eta > 0$ is the step size. Here, we use the direct parameterization of the policy (Agarwal et al., 2021), where $\pi_{h,i}(a \mid s) = \theta_{h,i}^{s,a}$ for some $\theta_{h,i}^{s,a} \geq 0$ and $\sum_{a\in\mathcal{A}_i} \theta_{h,i}^{s,a} = 1, \forall i \in \mathcal{N}, h \in [H], s \in \mathcal{S}, a \in \mathcal{A}_i$. We assume for now that the policy gradients $\nabla_{\pi_i} V_{1,i}^{\pi^{(t)}}(\rho)$ can be calculated exactly, and such an assumption will be relaxed in the next subsection.

Before presenting the analysis of PGA, we first introduce the following definition of an approximate stationary point.

Definition 6. *For any $\varepsilon > 0$, a policy profile $\pi = (\pi_1, \ldots, \pi_N)$ is a (first-order) ε-approximate stationary point of a function $\Phi_\rho : \Pi \to [0, \Phi_{\max}]$ if for any $\delta_1 \in \mathbb{R}^{A_1}, \ldots, \delta_N \in \mathbb{R}^{A_N}$, such that $\sum_{i\in\mathcal{N}} \|\delta_i\|_2^2 \leq 1$ and $\pi_i + \delta_i \in \Delta(\mathcal{A}_i), \forall i \in \mathcal{N}$, it holds that*

$$\sum_{i\in\mathcal{N}} \delta_i^{\intercal} \nabla_{\pi_i} \Phi_\rho(\pi) \leq \varepsilon.$$

Intuitively, π is an approximate stationary point if the function Φ_ρ cannot increase by more than ε along any direction that lies in the intersection of the policy space and the neighborhood of π. The following lemma establishes the equivalence between stationary points and NE.

Lemma 1. *Let $\pi = (\pi_1, \ldots, \pi_N)$ be an ε-approximate stationary point of the potential function Φ_ρ of an MPG for some $\varepsilon > 0$. Then, π is a $D\sqrt{SH}\varepsilon$-approximate NE.*

The proof of Lemma 1 relies on a gradient domination property that has been shown in single-agent RL (Agarwal et al., 2021). Its multi-agent counterpart has been studied in Zhang et al. (2021); Leonardos et al. (2021), though to the best of our knowledge, a gradient domination property in finite-horizon episodic MDPs/MPGs is still missing in the literature. For completeness, we derive such a result, together with the finite-horizon variants of the policy gradient theorem (Sutton et al., 2000) and performance difference lemma (Kakade & Langford, 2002) in Appendix E. With the above results, we arrive at the convergence guarantee of PGA in the exact gradient case. The proof of Theorem 3 is also deferred to Appendix E.

Theorem 3. *For any initial state distribution $\rho \in \Delta(\mathcal{S})$, let the agents independently run the projected gradient ascent updates (3) with a step size $\eta = \frac{1}{4NA_{\max}H^3}$ for $T = \frac{32NSA_{\max}D^2H^4\Phi_{\max}}{\varepsilon^2}$ iterations. Then, there exists $t \in [T]$, such that $\pi^{(t)}$ is an ε-approximate Nash equilibrium policy profile for the MPG.*

4.2. Finite-Sample Gradient Estimates

When the exact policy gradients are not given, we need to replace $\nabla_{\pi_i} V_{1,i}^{\pi^{(t)}}(\rho)$ in (3) with an estimate $\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)})$ that is calculated from a finite number of samples. For any policy π used in the t-iteration of PGA, we use a standard REINFORCE (Williams, 1992) gradient estimator

$$\hat{\nabla}_{\pi_i}^{(t)}(\pi) = R_i^{(t)} \sum_{h=1}^{H} \nabla \log \pi_{h,i}(a_{h,i}^{(t)} \mid s_h^{(t)}), \qquad (4)$$

where $R_i^{(t)} = \sum_{h=1}^{H} r_{h,i}^{(t)}(s_h^{(t)}, \boldsymbol{a}_h^{(t)})$ is the sum of rewards obtained at iteration t, and $s_1^{(t)} \sim \rho$.

To ensure that the variance of the gradient estimator is bounded, we let each agent use an epsilon-greedy variant of direct policy parameterization. Specifically, each agent i selects its actions according to a policy π_i, such that $\pi_{h,i}(a \mid s) = (1-\tilde{\varepsilon})\theta_{h,i}^{s,a} + \tilde{\varepsilon}/A_i$, where $\theta_{h,i}^{s,a} \geq 0$, $\sum_{a \in \mathcal{A}_i} \theta_{h,i}^{s,a} = 1$, and $\tilde{\varepsilon} > 0$ is the exploration parameter. In the following lemma, we show that the gradient estimator (4) under $\tilde{\varepsilon}$-greedy exploration is unbiased, has a bounded variance, and is mean-squared smooth. The first two properties have appeared in Daskalakis et al. (2020); Leonardos et al. (2021), while the third property is new and is used to derive an improved sample complexity result in our analysis.

Lemma 2. *For any agent $i \in \mathcal{N}$ and any iteration $t \in [T]$, the REINFORCE gradient estimator (4) with $\tilde{\varepsilon}$-greedy exploration is an unbiased estimator with a bounded variance:*

$$\mathbb{E}_{\pi^{(t)}}\left[\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)})\right] = \nabla_{\pi_i} V_{1,i}^{\pi^{(t)}}(\rho),$$

$$\mathbb{E}_{\pi^{(t)}}\left\|\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)}) - \nabla_{\pi_i} V_{1,i}^{\pi^{(t)}}(\rho)\right\|_2^2 \leq \frac{A_{\max}^2 H^4}{\tilde{\varepsilon}}.$$

Further, it is mean-squared smooth, i.e., for any $\pi'^{(t)} \in \Pi_i$,

$$\mathbb{E}_{\pi^{(t)}}\left\|\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)}) - \hat{\nabla}_{\pi_i'}^{(t)}(\pi'^{(t)})\right\|_2^2 \leq \frac{A_{\max}^3 H^3}{\tilde{\varepsilon}^3}\|\pi^{(t)} - \pi'^{(t)}\|_2^2.$$

Each agent now runs (projected) stochastic gradient ascent (SGA) to update its policy, where the gradient estimator is given by (4). In the following, we present the analysis of a generic stochastic gradient descent method that might be of independent interest, and the SGA policy update rule is simply an instantiation of such a generic method.

Consider a generic stochastic non-convex optimization problem as follows: We are given an objective function $F : \mathbb{R}^n \to \mathbb{R}$, and our goal is to find a point $x \in \mathcal{X} \subseteq \mathbb{R}^n$ such that $\nabla F(x)$ is close to 0, where $\mathcal{X}$ is the feasible region. We do not have accurate information about the function F, and can only access it through a stochastic sampling oracle $f(\cdot, \xi)$, where the random variable ξ represents the "randomness" of the oracle. We introduce the following assumptions that are standard in smooth non-convex optimization (Arjevani et al., 2019).

Assumption 1. *1. We have access to a stream of random variables $\xi_1, \ldots, \xi_T$, such that the gradient estimators are unbiased and have bounded variances: $\nabla \mathbb{E}_{\xi_t}[f(x, \xi_t)] = \nabla F(x)$, and $\mathbb{E}[\|\nabla f(x, \xi_t) - \nabla F(x)\|_2^2] \leq \sigma^2$ for some $\sigma > 0$ for all $t \in [T]$ and $x \in \mathcal{X}$.*

2. The objective F has bounded initial sub-optimality and is L-smooth: $F(x_0) - \inf_{x \in \mathcal{X}} F(x) < \infty$, and $\|\nabla F(x) - \nabla F(y)\|_2 \leq L \cdot \|x - y\|_2, \forall x, y \in \mathbb{R}^n$ for some $L > 0$. The stochastic oracle is mean-squared smooth for the same constant L: $\mathbb{E}[\|\nabla f(x,\xi) - \nabla f(y,\xi)\|_2^2] \leq L^2 \cdot \|x - y\|_2^2, \forall x, y \in \mathbb{R}^n$.

For an improved sample complexity bound, we utilize a momentum-based stochastic gradient descent (SGD) method with variance reduction (Johnson & Zhang, 2013; Allen-Zhu & Hazan, 2016; Reddi et al., 2016). Our method is a variant of the non-adaptive STOchastic Recursive Momentum (STORM) algorithm proposed in Cutkosky & Orabona (2019), and is formally described in Algorithm 3. It achieves an optimal convergence rate of $O(1/T^{1/3})$, which improves over the standard convergence rate $O(1/T^{1/4})$ of SGD with no variance reduction (e.g., Ghadimi & Lan (2013)). The key advantage of this method is to apply variance reduction in a *decentralized* way: Compared with other SGD methods with variance reduction (e.g., Allen-Zhu & Hazan (2016); Reddi et al. (2016); Fang et al. (2018)), our momentum-based algorithm does not require a batch of samples to compute checkpoint gradients. The agents hence do not need to coordinate on when to stop updating policies and to collect a batch of samples for a fixed policy profile, a common behavior when using batch-based methods.

Algorithm 3: Stochastic Recursive Momentum with Projections

```
$d_1 \leftarrow \nabla f(x_1, \xi_1)$;
for $t \leftarrow 1$ to $T$ do
    $\eta_t \leftarrow \frac{k}{(w+\sigma^2 t)^{1/3}}$;
    $x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{X}}(x_t - \eta_t d_t)$;
    $a_{t+1} \leftarrow c\eta_t^2$;
    $d_{t+1} \leftarrow \nabla f(x_{t+1},\xi_{t+1}) + (1 - a_{t+1})(d_t - \nabla f(x_t,\xi_{t+1}))$;
Output $x_\tau$ where $\tau$ is uniformly sampled from $[T]$;
```

The following result characterizes the convergence rate of Algorithm 3, and is a variant of the analysis given in Cutkosky & Orabona (2019). The proofs of Proposition 1 and its supporting lemmas are given in Appendix G.

Proposition 1. *Suppose Assumption 1 holds, and let $x_{t+1}^+ = \mathit{Proj}_{\mathcal{X}}(x_t - \eta_t \nabla F(x_t))$. For any $b > 0$, let $k = \frac{b\sigma^{\frac{2}{3}}}{L}, c = L^2\left(32 + 1/\left(7b^3\right)\right), w = \sigma^2 \max((4b)^3, 2, (32b + \frac{1}{7b^2})^3/64)$, and $M = 16(F(x_1) - \inf_{x \in \mathcal{X}} F(x)) + \frac{w^{1/3}\sigma^2}{2L^2 k} + \frac{k^3 c^2}{L^2}\ln(T+2)$. Then, the following result holds for Algorithm 3:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|\frac{1}{\eta_t}(x_{t+1}^+ - x_t)\right\|_2^2\right] \leq \frac{M w^{1/3}}{Tk} + \frac{M\sigma^{2/3}}{T^{2/3}k}.$$

Since we have shown in Lemma 2 that the conditions in Assumption 1 are satisfied by the potential function Φ_ρ and the REINFORCE policy gradient estimator $\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)})$, we can let each agent run an instance of Algorithm 3 and the convergence result in Proposition 1 directly applies. This leads us to the following sample complexity guarantee of learning Nash equilibria in MPGs. The proof of Theorem 4 can be found in Appendix F.

Theorem 4. *For any initial policies and any $\varepsilon > 0$, let the agents independently run SGA policy updates*

(Algorithm 3) for T iterations with $T = O(1/\varepsilon^{4.5}) \cdot poly(N, D, S, A_{\max}, H)$. Then, there exists $t \in [T]$, such that $\pi^{(t)}$ is an ε-approximate NE in expectation.

The polynomial sample complexity dependence on $A_{\max}$ is a natural benefit of decentralized learning, while centralized methods would typically have an exponential dependence $\prod_{i=1}^{N} A_i$. Such an improvement becomes more significant as the number of agents N increases. Also note that our sample complexity bound in Theorem 4 holds in expectation. To obtain a standard high-probability result that holds with probability $1 - p$, one could either apply Markov's inequality and tolerate an additional $O(1/p)$ factor of sample complexity, or replace our SGA method with one that has high-probability guarantees (Li & Orabona, 2020). We also remark that our results only guarantee the existence of a certain $t \in [T]$ ("best-iterate"), such that $\pi^{(t)}$ is an ε-approximate NE, but in general do not guarantee that $\pi^{(T)}$ ("last-iterate") is an approximate NE. This is mostly due to the nonconvexity of the potential function, and such a "best-iterate" convergence of gradient/gradient-mapping norm (i.e., stationary-point convergence) is consistent with the standard results in the nonconvex optimization literature (Ghadimi & Lan, 2013).

To obtain a sample complexity lower bound of the problem, we consider a simple instance where the agents share the same reward function (which clearly satisfies the definition of an MPG) and the action spaces of all but one agent i are singletons, i.e., $A_j = 1, \forall j \neq i$. Learning an approximate NE in such an MPG reduces to finding a near-optimal policy in a single-agent RL problem. Applying the regret lower bound of single-agent RL yields the following result for MARL in MPGs.

Corollary 1. *(Corollary of Jaksch et al. (2010)). For any algorithm, there exists a Markov potential game that takes the algorithm at least $\Omega(H^3 S A_{\max}/\varepsilon^2)$ episodes to learn an ε-approximate Nash equilibrium.*

We remark that such a lower bound might be very loose. Reducing to a single-agent RL problem evades the strategic learning behavior of the agents and the non-stationarity that such behavior causes to the environment, which in our opinion are the central difficulties of decentralized MARL. To derive a tighter lower bound in our decentralized setting, one should also utilize the additional constraint that each agent only has access to its local information, a factor that Corollary 1 apparently does not take into account. It is hence unsurprising that when comparing Theorem 4 with Corollary 1, we can see an obvious gap in the parameter dependence. We leave the tightening of both the upper and lower bounds to our future work.

4.3. Global Optimality in Smooth MPGs

We further show that our independent SGA algorithm can nearly find the globally optimal NE (i.e., the NE that maximizes the potential function, which is guaranteed to exist (Leonardos et al., 2021)) in an important subclass of MPGs named *smooth MPGs*.

Our definition of a (λ, ω)-smooth MPG is also adapted from the definition of smooth games in the literature (Roughgarden, 2009; Radanovic et al., 2019). Let $\pi^\star = (\pi_i^\star, \pi_{-i}^\star)$ be a policy that maximizes the potential function, i.e., $\Phi_\rho(\pi^\star) = \max_{\pi \in \Pi} \Phi_\rho(\pi)$. Let $V_{1,i}^\star$ denote the value function for agent i under policy $\pi^\star$. We consider the following definition of a smooth Markov potential game:

Definition 7. *(Adapted from Radanovic et al. (2019)). For $\lambda \geq 0$ and $0 < \omega < 1$, an N-player Markov potential game is (λ, ω)-smooth if for any policy profile $\pi = (\pi^i, \pi^{-i})$:*

$$V_{1,i}^{\pi_i^\star, \pi^{-i}}(s) \geq \lambda \cdot V_{1,i}^\star(s) - \omega \cdot V_{1,i}^\pi(s), \forall i \in \mathcal{N}, s \in \mathcal{S}.$$

The (λ, ω)-smoothness ensures that agent i continues doing well by playing its optimal policy even when the other agents are using slightly sub-optimal policies. It immediately follows that Algorithm 3 can nearly find the globally optimal NE in smooth MPGs.

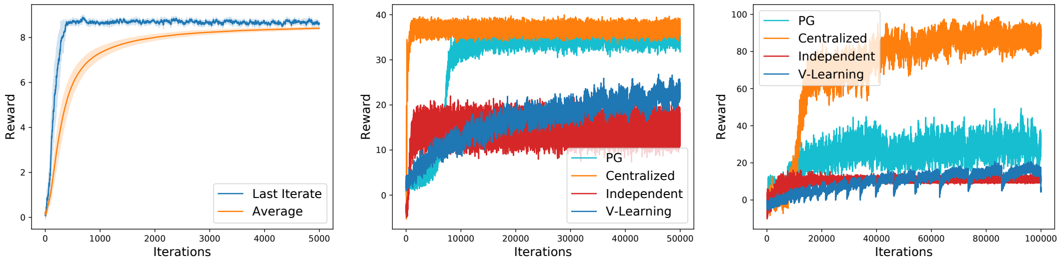
Theorem 5. *In a (λ, ω)-smooth MPG, for any initial policies and any $\varepsilon > 0$, let the agents independently run SGA policy updates (Algorithm 3) for T iterations with $T = O(1/\varepsilon^{4.5}) \cdot poly(N, D, S, A_{\max}, H)$. Then, there exists $t \in [T]$, such that*

$$\mathbb{E}\left[V_{1,i}^{\pi^{(t)}}(\rho)\right] \geq \frac{\lambda}{1+\omega} V_{1,i}^\star(\rho) - \frac{\varepsilon}{1+\omega}, \forall i \in \mathcal{N}.$$

The proof of Theorem 5 can be found in Appendix F.2. We remark that our definition of smooth MPGs generalizes that of smooth teams in Radanovic et al. (2019); Mao et al. (2020b), who assume an identical reward function of all the agents. Our approach also significantly improves the two works in that we design natural update rules for all the agents, who play symmetric roles in the self-play setting; the other two works only assign the algorithm to *one* agent, and have to *assume* that the policies of the other agent(s) change slowly.

5. Simulations

We empirically evaluate Algorithm 3 (SGA) on a classic matrix team task (Claus & Boutilier, 1998), and both Algorithms 1 and 3 on two Markov games, namely GoodState and BoxPushing (Seuken & Zilberstein, 2007). Figure 1 illustrates the performances of the algorithms in terms of the collected rewards. Detailed descriptions of the simulations are deferred to Appendix H due to space limitations.



(a) Matrix team (b) GoodState (c) BoxPushing

Figure 1. (a) Rewards of Algorithm 3 on the matrix team task, and rewards of Algorithms 1 and 3 on the (b) GoodState and (c) BoxPushing tasks. "Last Iterate" denotes the policy of the current iterate t, while "Average" represents a uniformly sampled policy from the first t iterates. "V-Learning" and "PG" denote the policies at the current iterate t of Algorithms 1 and 3, respectively. "Centralized" is an oracle that can control the actions of the agents in a centralized way. In "Independent", each agent runs a naïve single-agent Q-learning algorithm independently, by taking greedy actions with respect to its local Q-function estimates. All results are averaged over 20 runs.

For the matrix team, "Last Iterate" in Figure 1 (a) denotes the policy of the current iterate t of Algorithm 3, while "Average" represents a uniformly sampled policy from the first t iterates. We can see from Figure 1 (a) that both "Last Iterate" and "Average" converge, and obtain reward values close to 9 (where the global-optimal value is 10). This suggests that Algorithm 3 not only finds a NE in this specific task, but actually converges to a team-optimal equilibrium frequently. An encouraging observation is that, for Algorithm 1, the *actual* policy trajectories converge and achieve high rewards, even though our theoretical guarantees only hold for a "certified" output policy.

For learning in Markov games, we compare Algorithms 1 and 3 with two useful baselines, namely "Centralized" and "Independent" in Figures 1 (b) and (c). The "Centralized" oracle acts as a centralized coordinator that can control the actions of both agents. Such an oracle essentially converts the multi-agent task into a single-agent RL problem, and upper bounds the performances that our decentralized learning algorithms can possibly achieve in this task. The second baseline we consider is the naïve "Independent" Q-learning. Specifically, we let each agent run a single-agent Q-learning algorithm independently, without being aware of the existence of the other agents or the structure of the game. Each agent maintains a local optimistic Q-function, and takes greedy actions with respect to such optimistic estimates, without taking into account the other agents' actions.

Figure 3 illustrates the performances of our algorithms and the two baseline methods in terms of the collected rewards, where "V-Learning" and "PG" denote the policies at the current iterate t of Algorithms 1 and 3, respectively. Notice that the *actual* policy trajectories of both algorithms numerically converge and achieve high rewards. Further, both of our algorithms outperform the "Independent" baseline on the two tasks. In the GoodState problem, Algorithm 3 even approaches the performance of the "Centralized" oracle. The "Independent" baseline converges, albeit faster, to a clearly suboptimal value. This reiterates that the naïve idea of independent learning does not work well for MARL in general, and a careful treatment of the game structure (like our adversarial bandit subroutine) is necessary. Finally, the implemented algorithms take much fewer samples to converge than our theoretical results suggested. This indicates that the theoretical bounds might be overly conservative, and our algorithms could converge much faster in practice.

6. Concluding Remarks

In this paper, we have studied sample-efficient MARL in decentralized scenarios. We have proposed stage-based V-learning algorithms that learn CCE and CE in general-sum Markov games, and policy gradient algorithms that learn NE in Markov potential games. Our algorithms have improved existing results either through a simplified algorithmic design or a sharper sample complexity bound. An interesting future direction would be to tighten the sample complexity upper and lower bounds established in this paper. The problem of efficiently finding the globally optimal NE in generic MPGs through decentralized learning is also left open.

Acknowledgements

We thank Zihan Zhang and Chen-Yu Wei for helpful discussions and feedback. Research of W.M. and T.B. was supported in part by the ONR MURI Grant N00014-16-1-2710 and in part by the IBM-Illinois Discovery Accelerator Institute. Research of L.Y. was supported in part by DARPA grant HR00112190130. K.Z. was supported by the Simons-Berkeley Research Fellowship.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Allen-Zhu, Z. and Hazan, E. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pp. 699–707. PMLR, 2016.

Arabneydi, J. and Mahajan, A. Reinforcement learning in decentralized stochastic control systems with partial history sharing. In *American Control Conference*, pp. 5449–5456. IEEE, 2015.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Arslan, G. and Yüksel, S. Decentralized Q-learning for stochastic teams and games. *IEEE Transactions on Automatic Control*, 62(4):1545–1558, 2016.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Avner, O. and Mannor, S. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 66–81, 2014.

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, 2020.

Bai, Y., Jin, C., and Yu, T. Near-optimal reinforcement learning with self-play. *Advances in Neural Information Processing Systems*, 33, 2020.

Bernstein, D. S., Amato, C., Hansen, E. A., and Zilberstein, S. Policy iteration for decentralized control of Markov decision processes. *Journal of Artificial Intelligence Research*, 34:89–132, 2009.

Blum, A. and Mansour, Y. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Boutilier, C. Planning, learning and coordination in multiagent decision processes. In *Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 195–210, 1996.

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.

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

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Chang, W., Jafarnia-Jahromi, M., and Jain, R. Online learning for cooperative multi-player multi-armed bandits. *arXiv preprint arXiv:2109.03818*, 2021.

Claus, C. and Boutilier, C. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI Conference on Artificial Intelligence*, 1998(746-752):2, 1998.

Cohen, J., Héliou, A., and Mertikopoulos, P. Learning with bandit feedback in potential games. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6372–6381, 2017.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex SGD. *Advances in Neural Information Processing Systems*, 32:15236–15245, 2019.

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

Daskalakis, C., Foster, D. J., and Golowich, N. Independent policy gradient methods for competitive reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Ding, D., Wei, C.-Y., Zhang, K., and Jovanović, M. R. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. *arXiv preprint arXiv:2202.04129*, 2022.

Dubey, A. and Pentland, A. Provably efficient cooperative multi-agent reinforcement learning with function approximation. *arXiv preprint arXiv:2103.04972*, 2021.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. SPIDER: near-optimal non-convex optimization via stochastic path integrated differential estimator. In *International Conference on Neural Information Processing Systems*, pp. 687–697, 2018.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *International Conference on Neural Information Processing Systems*, pp. 2145–2153, 2016.

Foster, D. J., Li, Z., Lykouris, T., Sridharan, K., and Tardos, É. Learning in games: Robustness of fast convergence. In *International Conference on Neural Information Processing Systems*, pp. 4734–4742, 2016.

Fox, R., McAleer, S., Overman, W., and Panageas, I. Independent natural policy gradient always converges in Markov potential games. *arXiv preprint arXiv:2110.10614*, 2021.

Freund, Y. and Schapire, R. E. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Fudenberg, D., Drew, F., Levine, D. K., and Levine, D. K. *The Theory of Learning in Games*, volume 2. MIT press, 1998.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Guo, H., Fu, Z., Yang, Z., and Wang, Z. Decentralized single-timescale actor-critic on zero-sum two-player stochastic games. In *International Conference on Machine Learning*, pp. 3899–3909. PMLR, 2021.

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*, 60(1):1–16, 2013.

Hart, S. and Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5): 1127–1150, 2000.

Hart, S. and Mas-Colell, A. Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, 93(5):1830–1836, 2003.

Ho, Y.-C. Team decision theory and information structures. *Proceedings of the IEEE*, 68(6):644–654, 1980.

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(4), 2010.

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

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

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26:315–323, 2013.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

Kar, S., Moura, J. M. F., and Poor, H. V. QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Transactions on Signal Processing*, 61(7):1848–1862, 2013.

Kleinberg, R., Piliouras, G., and Tardos, É. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 533–542, 2009.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Lai, L., Jiang, H., and Poor, H. V. Medium access in cognitive radio networks: A competitive multi-armed bandit framework. In *Asilomar Conference on Signals, Systems and Computers*, pp. 98–102. IEEE, 2008.

Lauer, M. and Riedmiller, M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *International Conference on Machine Learning*, 2000.

Leonardos, S., Overman, W., Panageas, I., and Piliouras, G. Global convergence of multi-agent policy gradient in Markov potential games. *arXiv preprint arXiv:2106.01969*, 2021.

Leslie, D. S. and Collins, E. J. Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

Li, X. and Orabona, F. A high probability analysis of adaptive SGD with momentum. *arXiv preprint arXiv:2007.14294*, 2020.

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

Littman, M. L. Friend-or-Foe Q-learning in general-sum games. In *International Conference on Machine Learning*, pp. 322–328, 2001.

Liu, Q., Yu, T., Bai, Y., and Jin, C. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, 2021.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30:6379–6390, 2017.

Macua, S. V., Zazo, J., and Zazo, S. Learning parametric closed-loop policies for Markov potential games. *arXiv preprint arXiv:1802.00899*, 2018.

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

Mao, W., Zhang, K., Miehling, E., and Başar, T. Information state embedding in partially observable cooperative multi-agent reinforcement learning. In *IEEE Conference on Decision and Control*, pp. 6124–6131. IEEE, 2020a.

Mao, W., Zhang, K., Zhu, R., Simchi-Levi, D., and Başar, T. Near-optimal regret bounds for model-free RL in non-stationary episodic MDPs. In *International Conference on Machine Learning*, 2020b.

Marden, J. R., Arslan, G., and Shamma, J. S. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6): 1393–1407, 2009a.

Marden, J. R., Young, H. P., Arslan, G., and Shamma, J. S. Payoff-based dynamics for multiplayer weakly acyclic games. *SIAM Journal on Control and Optimization*, 48 (1):373–396, 2009b.

Menard, P., Domingues, O. D., Shang, X., and Valko, M. UCB momentum Q-learning: Correcting the bias without forgetting. *arXiv preprint arXiv:2103.01312*, 2021.

Mguni, D., Wu, Y., Du, Y., Yang, Y., Wang, Z., Li, M., Wen, Y., Jennings, J., and Wang, J. Learning in nonzero-sum stochastic games with potentials. *arXiv preprint arXiv:2103.09284*, 2021.

Nayyar, A., Gupta, A., Langbort, C., and Başar, T. Common information based Markov perfect equilibria for stochastic games with asymmetric information: Finite games. *IEEE Transactions on Automatic Control*, 59(3):555–570, 2013a.

Nayyar, A., Mahajan, A., and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013b.

Neu, G. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28:3168–3176, 2015.

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

Oliehoek, F. A., Spaan, M. T., and Vlassis, N. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Pham, H. X., La, H. M., Feil-Seifer, D., and Nefian, A. Cooperative and distributed reinforcement learning of drones for field coverage. *arXiv preprint arXiv:1803.07250*, 2018.

Radanovic, G., Devidze, R., Parkes, D., and Singla, A. Learning to collaborate in Markov decision processes. In *International Conference on Machine Learning*, pp. 5261–5270, 2019.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pp. 314–323. PMLR, 2016.

Roughgarden, T. Intrinsic robustness of the price of anarchy. In *ACM Symposium on Theory of Computing*, pp. 513–522, 2009.

Rubinstein, A. Settling the complexity of computing approximate two-player Nash equilibria. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 258–265. IEEE, 2016.

Sayin, M. O., Zhang, K., Leslie, D. S., Başar, T., and Ozdaglar, A. Decentralized Q-learning in zero-sum Markov games. *arXiv preprint arXiv:2106.02748*, 2021.

Seuken, S. and Zilberstein, S. Improved memory-bounded dynamic programming for decentralized POMDPs. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, pp. 344–351, 2007.

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

Shapley, L. S. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

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

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):484–489, 2016.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

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

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Syrgkanis, V. and Tardos, E. Composable and efficient mechanisms. In *ACM Symposium on Theory of Computing*, pp. 211–220, 2013.

Syrgkanis, V., Agarwal, A., Luo, H., and Schapire, R. E. Fast convergence of regularized learning in games. In *International Conference on Neural Information Processing Systems*, pp. 2989–2997, 2015.

Tian, Y., Wang, Y., Yu, T., and Sra, S. Online learning in unknown Markov games. *International Conference on Machine Learning*, 2021.

Verbeeck, K., Nowé, A., Lenaerts, T., and Parent, J. Learning to reach the Pareto optimal Nash equilibrium as a team. In *Australian Joint Conference on Artificial Intelligence*, pp. 407–418. Springer, 2002.

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.

Viossat, Y. and Zapechelnyuk, A. No-regret dynamics and fictitious play. *Journal of Economic Theory*, 148(2):825–842, 2013.

Wang, X. and Sandholm, T. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. *Advances in Neural Information Processing Systems*, 15: 1603–1610, 2002.

Wei, C.-Y., Hong, Y.-T., and Lu, C.-J. Online reinforcement learning in stochastic games. In *International Conference on Neural Information Processing Systems*, pp. 4994–5004, 2017.

Wei, C.-Y., Lee, C.-W., Zhang, M., and Luo, H. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive Markov games. *Annual Conference on Learning Theory*, 2021.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

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, 2020.

Yongacoglu, B., Arslan, G., and Yüksel, S. Learning team-optimality for decentralized stochastic control and dynamic games. *arXiv preprint arXiv:1903.05812*, 2019.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Başar, T. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pp. 5872–5881, 2018.

Zhang, K., Miehling, E., and Başar, T. Online planning for decentralized stochastic control with partial history sharing. In *American Control Conference*, pp. 3544–3550. IEEE, 2019.

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

Zhang, R., Ren, Z., and Li, N. Gradient play in multi-agent Markov stochastic games: Stationary points and convergence. *arXiv preprint arXiv:2106.00198*, 2021.

Zhang, Z., Zhou, Y., and Ji, X. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33, 2020b.

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

Supplementary Materials for "On Improving Model-Free Algorithms for Decentralized Multi-Agent Reinforcement Learning"

A. Detailed Discussions on Related Work

A common mathematical framework of multi-agent RL is stochastic games (Shapley, 1953), which are also referred to as Markov games. Early attempts to learn Nash equilibria in Markov games include Littman (1994; 2001); Hu & Wellman (2003); Hansen et al. (2013), but they either assume the transition kernel and rewards are known, or only yield asymptotic guarantees. More recently, various sample efficient methods have been proposed (Wei et al., 2017; Bai & Jin, 2020; Sidford et al., 2020; Xie et al., 2020; Bai et al., 2020; Liu et al., 2021; Zhao et al., 2021), mostly for learning in two-player zero-sum Markov games. Most notably, several works have investigated two-player zero-sum games in a *decentralized* environment: Daskalakis et al. (2020) have shown non-asymptotic convergence guarantees for independent policy gradient methods when the learning rates of the two agents follow a two-timescale rule. Tian et al. (2021) have studied online learning when the actions of the opponents are not observable, and have achieved the first sub-linear regret $\widetilde{O}(K^{\frac{3}{4}})$ in the decentralized setting for K episodes. More recently, Wei et al. (2021) have proposed an Optimistic Gradient Descent Ascent algorithm with a slowly-learning critic, and have shown a strong finite-time last-iterate convergence result in the decentralized/agnostic environment. Overall, these works have mainly focused on two-player zero-sum games. These results do not carry over in any way to general-sum games or MPGs that we consider in this paper.

In general-sum normal-form games, a folklore result is that when the agents independently run no-regret learning algorithms, their empirical frequency of plays converges to the set of coarse correlated equilibria (CCE) of the game (Hart & Mas-Colell, 2000). However, a CCE may suggest that the agents play obviously non-rational strategies. For example, Viossat & Zapechelnyuk (2013) have constructed an example where a CCE assigns positive probabilities only to strictly dominated strategies. On the other hand, given the PPAD completeness of finding a Nash equilibrium, convergence to NE seems hopeless in general. An impossibility result (Hart & Mas-Colell, 2003) has shown that uncoupled no-regret learning does not converge to Nash equilibrium in general, due to the informational constraint that the adjustment in an agent's strategy does not depend on the reward functions of the others. Hence, convergence to Nash equilibria is guaranteed mostly in games with special reward structures, such as two-player zero-sum games (Freund & Schapire, 1999) and potential games (Kleinberg et al., 2009; Cohen et al., 2017).

For learning in general-sum Markov games, Rubinstein (2016) has shown a sample complexity lower bound for NE that is exponential in the number of agents. Recently, Liu et al. (2021) has presented a line of results on learning NE, CE, or CCE, but their algorithm is model-based, and suffers from such exponential dependence. Song et al. (2021); Jin et al. (2021); Mao & Başar (2022) have proposed V-learning based methods for learning CCE and/or CE, which are similar to the ones that we study here, and avoid the exponential dependence. Nevertheless, our methods significantly simplify their algorithmic design and analysis, by introducing a stage-based V-learning update rule that circumvents their rather complicated no-weighted-regret bandit subroutine.

Another line of research has considered RL in Markov potential games (Macua et al., 2018; Mguni et al., 2021; Ding et al., 2022). Arslan & Yüksel (2016) has shown that decentralized Q-learning style algorithms can converge to NE in weakly acyclic games, which cover MPGs as an important special case. Their decentralized setting is similar to ours in that each agent is completely oblivious to the presence of the others. Later, such a method has been improved in Yongacoglu et al. (2019) to achieve team-optimality. However, both of them require a coordinated exploration phase, and only yield asymptotic guarantees. Decentralized learning has also been studied in single-stage weakly acyclic games (Marden et al., 2009b) or potential games (Marden et al., 2009a; Cohen et al., 2017). Two recent works (Zhang et al., 2021; Leonardos et al., 2021) have proposed independent policy gradient methods in MPGs, which are most relevant to ours. We improve their sample complexity dependence on ε by utilizing a decentralized variance reduction technique, and do not require the two-timescale framework to coordinate policy evaluation as in Zhang et al. (2021). Fox et al. (2021) has shown that independent Natural Policy Gradient also converges to NE in MPGs, though only asymptotic convergence has been established. Finally, MPGs have also been studied in Song et al. (2021), but their model-based method is not decentralized, and requires the agents to take turns to learn the policies.

MARL has also been studied in teams or cooperative games, which can be considered as a subclass of MPGs. Without enforcing a decentralized environment, Boutilier (1996) has proposed to coordinate the agents by letting them take actions in a lexicographic order. In a similar setting, Wang & Sandholm (2002) have studied optimal adaptive learning that converges to the optimal NE in Markov teams. Verbeeck et al. (2002) have presented an independent learning algorithm that achieves a Pareto optimal NE in common interest games with limited communication. These methods critically relied on communications among the agents (beforehand) or observing the teammates' actions. In contrast, the distributed Q-learning algorithm in (Lauer & Riedmiller, 2000) is decentralized and coordination-free, which, however, only works for deterministic tasks, and has no non-asymptotic guarantees.

Efficient exploration has also been widely studied in the literature of single-agent RL, see, e.g., Brafman & Tennenholtz (2002); Jaksch et al. (2010); Azar et al. (2017); Jin et al. (2018). For the tabular episodic setting, various methods (Azar et al., 2017; Zhang et al., 2020b; Menard et al., 2021) have achieved the sample complexity of $\widetilde{O}(H^3SA/\varepsilon^2)$, which matches the information-theoretical lower bound. When reduced to the bandit case, decentralized MARL is also related to the cooperative multi-armed bandit (MAB) problem (Lai et al., 2008; Avner & Mannor, 2014), originated from the literature of cognitive radio networks. The difference is that, in cooperative MAB, each agent is essentially interacting with an individual copy of the bandit, with an extra caution of action collisions; in the MARL formulation, the reward function is defined on the Cartesian product of the action spaces, which allows the agents to be coupled in more general forms. A concurrent work (Chang et al., 2021) has studied cooperative multi-player multi-armed bandits with information asymmetry. Nevertheless, (Chang et al., 2021) requires stronger conditions than our decentralized setting as their algorithm relies on playing a predetermined sequence of actions.

B. Technical Lemmas

Lemma 3. *(Bubeck et al., 2015, Lemma 3.6). Let f be a β-smooth function with a convex domain $\mathcal{X}$. For any $x \in \mathcal{X}$, let $x^+ = \text{Proj}_{\mathcal{X}}(x - \eta\nabla f(x))$ be a projected gradient descent update with $\eta = \frac{1}{\beta}$, and let $G^\eta(x) = \frac{1}{\eta}(x - x^+)$. Then, the following holds true*

$$f(x^+) - f(x) \le -\frac{1}{2\beta}\|G^\eta(x)\|_2^2.$$

Lemma 4. *(Agarwal et al., 2021, Proposition B.1). Let $f : \mathcal{X} \to \mathbb{R}$ be a β-smooth function. Define the gradient mapping as*

$$G^\eta(x) = \frac{1}{\eta}\left(\text{Proj}_{\mathcal{X}}(x + \eta\nabla f(x)) - x\right).$$

The update rule for projected gradient ascent is $x^+ = x + \eta G^\eta(x)$. If $\|G^\eta(x)\|_2 \le \varepsilon$, then

$$\max_{x+\delta\in\mathcal{X}, \|\delta\|_2^2\le 1} \delta^\intercal \nabla f(x^+) \le \varepsilon(\eta\beta + 1).$$

Lemma 5. *(Leonardos et al., 2021, Lemma D.3). Let $\Phi_\rho : \Pi \to \mathbb{R}$ be the potential function (which is β-smooth), and assume that $\pi \in \Pi$ uses ε_i-greedy parameterization. Define the gradient mapping as*

$$G^\eta(\pi) = \frac{1}{\eta}\left(\text{Proj}_{\Pi}(\pi + \eta\nabla \Phi_\rho(\pi)) - \pi\right).$$

The update rule for projected gradient ascent is $\pi^+ = \pi + \eta G^\eta(\pi)$. If $\eta\beta \le 1$ and $\|G^\eta(\pi)\|_2 \le \varepsilon$, then

$$\max_{\pi+\delta\in\Pi, \|\delta\|_2^2\le 1} \delta^\intercal \nabla \Phi_\rho(\pi^+) \le 2\varepsilon + \sqrt{NSA_{\max}^2 H^5 \varepsilon_i^2}.$$

Lemma 6. *(Leonardos et al., 2021, Claim C.2). Consider a symmetric block matrix C with $n \times n$ sub-matrices, and let C_{ij} denote the sub-matrix at the i-th and j-th column. If $\|C_{ij}\|_2 \le L$ for some $L > 0$, then it holds that $\|C\| \le nL$, i.e., if every sub-matrix of C have a spectral norm of at most L, then C has a spectral norm of at most nL.*

C. Proofs for Section 3.1

We first introduce a few notations to facilitate the analysis. For a step $h \in [H]$ of an episode $k \in [K]$, we denote by s_h^k the state that the agents observe at this time step. For any state $s \in \mathcal{S}$, we let $\mu_{h,i}^k(\cdot \mid s) \in \Delta(\mathcal{A}_i)$ be the distribution prescribed by Algorithm 1 to agent i at this step. Notice that such notations are well-defined for every $s \in \mathcal{S}$ even if s might not be the state s_h^k that is actually visited at the given step. We further let $\mu_{h,i}^k = \{\mu_{h,i}^k(\cdot \mid s) : s \in \mathcal{S}\}$, and let $a_{h,i}^k \in \mathcal{A}_i$ be the actual action taken by agent i. For any $s \in \mathcal{S}$, let $N_h^k(s)$ and $\check{N}_h^k(s)$ denote, respectively, the values of $N_h(s)$ and $\check{N}_h(s)$ at the *beginning* of the k-th episode. Note that it is proper to use the same notation to denote these values from all the agents' perspectives, because the agents maintain the same estimates of these terms as they can be calculated from the common observations (of the state-visitation). We also use $\overline{V}_{h,i}^k(s)$ and $\tilde{V}_{h,i}^k(s)$ to denote the values of $\overline{V}_{h,i}(s)$ and $\tilde{V}_{h,i}(s)$, respectively, at the beginning of the k-th episode from agent i's perspective.

Further, for a state s_h^k, let $\check{n}_h^k$ denote the number of times that state s_h^k has been visited (at the h-th step) in the stage right before the current stage, and let $\check{l}_{h,j}^k$ denote the index of the episode that this state was visited the j-th time among the $\check{n}_h^k$ times. For notational convenience, we use $\check{n}$ to denote $\check{n}_h^k$, and $\check{l}_j$ to denote $\check{l}_{h,j}^k$, whenever h and k are clear from the context. With the new notations, the update rule in Line 13 of Algorithm 1 can be equivalently expressed as

$$\tilde{V}_{h,i}(s_h) \leftarrow \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s_h, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + b_{\check{n}}. \tag{5}$$

For notational convenience, we introduce the operators $\mathbb{P}_h V(s, \boldsymbol{a}) = \mathbb{E}_{s' \sim P_h(\cdot \mid s, \boldsymbol{a})} V(s')$ for any value function V, and $\mathbb{D}_{\boldsymbol{\mu}_h} Q(s) = \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\mu}_h} Q(s, \boldsymbol{a})$. With these notations, the Bellman equations can be rewritten more succinctly as $Q_h^\pi(s, \boldsymbol{a}) = \left(r_h + \mathbb{P}_h V_{h+1}^\pi\right)(s, \boldsymbol{a})$, and $V_h^\pi(s) = \left(\mathbb{D}_{\boldsymbol{\mu}_h} Q_h^\pi\right)(s)$ for any $(s, \boldsymbol{a}, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, where $\boldsymbol{\mu}_h = \pi_h$. In the following proof, we assume without loss of generality that the initial state s_1 is fixed, i.e., ρ is a point mass distribution at s_1. Our proof can be easily generalized to the case where the initial state is drawn from a fixed distribution $\rho \in \Delta(\mathcal{S})$.

In the following, we start with an intermediate result, which justifies our choice of the bonus term.

Lemma 7. *With probability at least $1 - \frac{p}{2}$, it holds for all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ that*

$$\max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i} \times \mu_{h,-i}^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) \leq 6\sqrt{H^2 A_i \iota / \check{n}}.$$

Proof. For a fixed $(s, h, k) \in \mathcal{S} \times [H] \times [K]$, let $\mathcal{F}_j$ be the σ-algebra generated by all the random variables up to episode $\check{l}_j$. Then, $\left\{ r_{h,i}(s, \boldsymbol{a}_{h,i}^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) - \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) \right\}_{j=1}^{\check{n}}$ is a martingale difference sequence with respect to $\{\mathcal{F}_j\}_{j=1}^{\check{n}}$. From the Azuma-Hoeffding inequality, it holds with probability at least $1 - p/(4NSHK)$ that

$$\frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) \leq \sqrt{H^2 \iota / \check{n}}.$$

Therefore, we only need to bound

$$R_{\check{n}}^\star \overset{\text{def}}{=} \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i} \times \mu_{h,-i}^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s). \tag{6}$$

Notice that $R_{\check{n}}^\star$ can be considered as the averaged regret of visiting the state s with respect to the optimal policy in hindsight. Such a regret minimization problem can be handled by an adversarial multi-armed bandit problem, where the loss function at step $j \in [\check{n}]$ is defined as

$$\ell_j(a_i) = \mathbb{E}_{a_{-i} \sim \mu_{h,-i}^{\check{l}_j}}(s) \left[H - h + 1 - r_{h,i}(s, \boldsymbol{a}) - \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j}(s, \boldsymbol{a}) \right] / H.$$

Algorithm 1 applies the Exp3-IX algorithm (Neu, 2015), which ensures that with probability at least $1 - \frac{p}{4NHS}$, it holds for all $k \in [K]$ that

$$R^{\star}_{\check{n}} \leq \sqrt{\frac{8H^2 A_i \log A_i}{\check{n}}} + \left(\sqrt{\frac{2A_i}{\check{n}\log A_i}} + \frac{1}{\check{n}}\right) H \log(2/p).$$

A union bound over all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ completes the proof. □

Remark 1. *We would like to discuss the alternative of using V-learning with the celebrated learning rate $\alpha_t = \frac{H+1}{H+t}$ (Jin et al., 2018) to update $\overline{V}_h$ instead of employing stage-based updates. This is the case for several recent works also under the V-learning formulation for MARL (Bai et al., 2020; Jin et al., 2021; Song et al., 2021; Mao & Başar, 2022). Such a learning rate induces an update rule as follows:*

$$\overline{V}_{h,i}\left(s_h\right) \leftarrow \left(1-\alpha_t\right) \overline{V}_{h,i}\left(s_h\right) + \alpha_t \left(r_{h,i}\left(s_h, \boldsymbol{a}_h\right) + \overline{V}_{h+1,i}\left(s_{h+1}\right) + \beta_t\right), \tag{7}$$

where t is the number of times that s_h has been visited, and β_t is some bonus term. In this way, $\overline{V}_{h,i}(s_h)$ is updated every time the state s_h is visited. With such a learning rate, the update rule (7) of $\overline{V}_{h,i}$ can be equivalently expressed as

$$\overline{V}^{k}_{h,i}(s_h) = \alpha^0_t H + \sum_{j=1}^{t} \alpha^j_t \left[r_{h,i}\left(s, \boldsymbol{a}^{k^j}_h\right) + \overline{V}^{k^j}_{h+1,i}\left(s^{k^j}_{h+1}\right) + \beta_j \right],$$

where k^j is the index of the episode such that s_h is visited the j-th time. The weights α^j_t are given by

$$\alpha^0_t = \prod_{j=1}^{t} \left(1-\alpha_j\right), \quad \textit{and} \quad \alpha^j_t = \alpha_j \prod_{k=j+1}^{t} \left(1-\alpha_k\right), \forall 1 \leq j \leq t.$$

Compared with stage-based updates (6), we now need to upper bound a regret term of the following form:

$$R^{\star}_t(s) = \max_{\mu_{h,i}} \sum_{j=1}^{t} \alpha^j_t \mathbb{D}_{\mu_{h,i} \times \mu^{kj}_{h,-i}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{k^j}_{h+1,i} \right)(s) - \sum_{j=1}^{t} \alpha^j_t \mathbb{D}_{\mu^{kj}_{h,i} \times \mu^{kj}_{h,-i}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{k^j}_{h+1,i} \right)(s).$$

Notice that the above definition of regret induces a adversarial bandit problem with a time-varying weighted regret, where the loss at time j is assigned a weight α^j_t. As t varies, the weight α^j_t assigned to the same step j also changes over time. These weights also cannot be pre-computed, because it relies on knowing the total number of times that a certain state s_h is visited during the entire horizon, which is impossible before seeing the output of the algorithm. To address such an additional challenge, Bai et al. (2020) proposed a Follow-the-Regularized-Leader (FTRL) algorithm that simultaneously achieves with a changing step size, a weighted regret, and a high-probability guarantee, which inevitably leads to a more delicate analysis. In contrast, we have shown in (6) that our stage-based update rule leads to an adversarial bandit problem with a simple averaged regret. In our approach, it suffices to plug in any existing adversarial bandit solution with a high-probability regret bound, such as the Exp3-IX method that we used in Algorithm 1. Therefore, our stage-based update significantly simplifies both the algorithmic design and the analysis of V-learning in MARL.

Based on the trajectory of the distributions $\{\mu^k_{h,i} : i \in \mathcal{N}, h \in [H], k \in [K]\}$ specified by Algorithm 1, we construct a correlated policy $\bar{\pi}^k_h$ for each $(h, k) \in [H] \times [K]$. Our construction of the correlated policies, largely inspired by the "certified policies" (Bai et al., 2020) for learning in two-player zero-sum games, is formally presented in Algorithm 4. We further define an output policy $\bar{\pi}$ that first uniformly samples an index k from $[K]$, and then proceed with $\bar{\pi}^k_1$. A more formal description of $\bar{\pi}$ has been given in Algorithm 2. By construction of the correlated policies $\bar{\pi}^k_h$, we know that for any $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H+1] \times [K]$, the corresponding value function can be written recursively as follows:

$$V^{\bar{\pi}^k_h}_{h,i}(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu^{\check{l}_j}_h} \left(r_{h,i} + \mathbb{P}_h V^{\bar{\pi}^{\check{l}_j}_{h+1}}_{h+1,i} \right)(s),$$

Algorithm 4: Construction of the Correlated Policy $\bar{\pi}_h^k$

```
Input: The distribution trajectory $\{\mu_{h,i}^k : i \in \mathcal{N}, h \in [H], k \in [K]\}$ specified by Algorithm 1.
Initialize: $k' \leftarrow k$.
for step $h' \leftarrow h$ to $H$ do
    Receive $s_{h'}$;
    Take joint action $\boldsymbol{a}_{h'} \sim \times_{i=1}^N \mu_{h',i}^{k'}(\cdot \mid s_{h'})$;
    Uniformly sample $j$ from $\{1, 2, \ldots, \check{N}_{h'}^{k'}(s_{h'})\}$;
    Set $k' \leftarrow \check{l}_{h',j}^{k'}$, where $\check{l}_{h',j}^{k'}$ is the index of the episode such that state $s_{h'}$ was visited the $j$-th time (among the total
      $\check{N}_{h'}^{k'}(s_{h'})$ times) in the last stage;
```

and $V_{h,i}^{\bar{\pi}_h^k}(s) = 0$ if $h = H+1$ or k is in the first stage of the corresponding (h, s) pair. We also immediately obtain that

$$V_{1,i}^{\bar{\pi}}(s_1) = \frac{1}{K}\sum_{k=1}^{K} V_{1,i}^{\bar{\pi}_1^k}(s_1).$$

Only for analytical purposes, we introduce two new notations $\underline{V}$ and $\underset{\sim}{V}$ that serve as lower confidence bounds of the value estimates. Specifically, for any $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H+1] \times [K]$, we define $\underline{V}_{h,i}^k(s) = \underset{\sim}{V}_{h,i}^k(s) = 0$ if $h = H+1$ or k is in the first stage of the (h, s) pair, and

$$\underset{\sim}{V}_{h,i}^k(s) = \frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(r_{h,i}(s_h, \boldsymbol{a}_h^{\check{l}_j}) + \underline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right) - b_{\check{n}}, \text{ and } \underline{V}_{h,i}^k(s) = \max\left\{\underset{\sim}{V}_{h,i}^k(s), 0\right\}.$$

Notice that these two notations are only introduced for ease of analysis, and the agents need not explicitly maintain such values during the learning process. Further, recall that $V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s)$ is agent i's best response value against its opponents' policy $\bar{\pi}_{h,-i}^k$. Our next lemma shows that $\overline{V}_{h,i}^k(s)$ and $\underline{V}_{h,i}^k(s)$ are indeed valid upper and lower bounds of $V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s)$ and $V_{h,i}^{\bar{\pi}_h^k}(s)$, respectively.

Lemma 8. *It holds with probability at least $1 - p$ that for all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$,*

$$\overline{V}_{h,i}^k(s) \geq V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s), \textit{ and } \underline{V}_{h,i}^k(s) \leq V_{h,i}^{\bar{\pi}_h^k}(s).$$

Proof. Consider a fixed $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$. The desired result clearly holds for any state s that is in its first stage, due to our initialization of $\overline{V}_{h,i}^k(s)$ and $\underline{V}_{h,i}^k(s)$ for this special case. In the following, we only need to focus on the case where $\overline{V}_{h,i}(s)$ and $\underline{V}_{h,i}^k(s)$ have been updated at least once at the given state s before the k-th episode.

We first prove the first inequality. It suffices to show that $\tilde{V}_{h,i}^k(s) \geq V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s)$ because $\overline{V}_{h,i}^k(s) = \min\{\tilde{V}_{h,i}^k(s), H - h + 1\}$, and $V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s)$ is always less than or equal to $H - h + 1$. Our proof relies on induction on $k \in [K]$. First, the claim holds for $k = 1$ due to the aforementioned logic. For each step $h \in [H]$ and $s \in \mathcal{S}$, we consider the following two cases.

Case 1: $\tilde{V}_{h,i}(s)$ has just been updated in (the end of) episode $k - 1$. In this case,

$$\tilde{V}_{h,i}^k(s) = \frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right) + b_{\check{n}}. \tag{8}$$

By the definition of $V_h^{\star,\bar{\nu}_h^k}(s)$, it holds with probability at least $1-\frac{p}{2NSKH}$ that

$$\begin{aligned}
V_{h,i}^{\star,\bar{\pi}_{h,-i}^k}(s) \leq & \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i}\times\mu_{h,-i}^{\check{l}_j}} \left(r_{h,i}+\mathbb{P}_h V_{h+1,i}^{\star,\bar{\pi}_{h+1,-i}^{\check{l}_j}}\right)(s) \\
\leq & \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i}\times\mu_{h,-i}^{\check{l}_j}} \left(r_{h,i}+\mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j}\right)(s) \\
\leq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s,\boldsymbol{a}_h^{\check{l}_j})+\overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)+6\sqrt{H^2A_i\iota/\check{n}} \\
\leq & \tilde{V}_{h,i}^k(s),
\end{aligned} \tag{9}$$

where the second step is by the induction hypothesis, the third step holds due to Lemma 7, and the last step is by the definition of $b_{\check{n}}$.

Case 2: $\tilde{V}_{h,i}(s)$ was not updated in (the end of) episode $k-1$. Since we have excluded the case that $\tilde{V}_{h,i}$ has never been updated, we are guaranteed that there exists an episode j such that $\tilde{V}_{h,i}(s)$ has been updated in the end of episode $j-1$ most recently. In this case, $\tilde{V}_{h,i}^k(s)=\tilde{V}_{h,i}^{k-1}(s)=\cdots=\tilde{V}_{h,i}^j(s)\geq V_{h,i}^{\star,\bar{\pi}_{h,-i}^j}(s)$, where the last step is by the induction hypothesis. Finally, observe that by our definition, the value of $V_{h,i}^{\star,\bar{\pi}_{h,-i}^j}(s)$ is a constant for all episode indices j that belong to the same stage. Since we know that episode j and episode k lie in the same stage, we can conclude that $V_{h,i}^{\star,\bar{\pi}_{h,-i}^k}(s)=V_{h,i}^{\star,\bar{\pi}_{h,-i}^j}(s)\leq\tilde{V}_{h,i}^k(s)$.

Combining the two cases and applying a union bound over all $(i,s,h,k)\in\mathcal{N}\times\mathcal{S}\times[H]\times[K]$ complete the proof of the first inequality.

Next, we prove the second inequality in the statement of the lemma. Notice that it suffices to show $\underset{\sim}{V}_{h,i}^k(s)\leq V_{h,i}^{\bar{\pi}_h^k}(s)$ because $\underline{V}_{h,i}^k(s)=\max\{\underset{\sim}{V}_{h,i}^k(s),0\}$. Our proof again relies on induction on $k\in[K]$. Similar to the proof of the first inequality, the claim apparently holds for $k=1$, and we consider the following two cases for each step $h\in[H]$ and $s\in\mathcal{S}$.

Case 1: The value of $\underset{\sim}{V}_{h,i}(s)$ has just changed in (the end of) episode $k-1$. In this case,

$$\underset{\sim}{V}_{h,i}^k(s)=\frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(r_{h,i}(s,\boldsymbol{a}_h^{\check{l}_j})+\underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)-b_{\check{n}}. \tag{10}$$

By the definition of $V_{h,i}^{\bar{\pi}_h^k}(s)$, it holds with probability at least $1-\frac{p}{2NSKH}$ that

$$\begin{aligned}
V_{h,i}^{\bar{\pi}_h^k}(s) = & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i}+\mathbb{P}_h V_{h+1,i}^{\bar{\pi}_{h+1}^{\check{l}_j}}\right)(s) \\
\geq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i}+\mathbb{P}_h \underset{\sim}{V}_{h+1,i}^{\check{l}_j}\right)(s) \\
\geq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s,\boldsymbol{a}_h^{\check{l}_j})+\underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)-\sqrt{H^2\iota/\check{n}} \\
\geq & \underset{\sim}{V}_{h,i}^k(s),
\end{aligned} \tag{11}$$

where the second step is by the induction hypothesis, the third step holds due to the Azuma-Hoeffding inequality, and the last step is by the definition of $b_{\check{n}}$.

Case 2: The value of $\underset{\sim}{V}_{h,i}(s)$ has not changed in (the end of) episode $k-1$. Since we have excluded the case that $\underset{\sim}{V}_{h,i}$ has never been updated, we are guaranteed that there exists an episode j such that $\underset{\sim}{V}_{h,i}(s)$ has changed in the end of episode $j-1$ most recently. In this case, we know that indices j and k belong to the same stage, and $\underset{\sim}{V}^k_{h,i}(s) = \underset{\sim}{V}^{k-1}_{h,i}(s) = \cdots = \underset{\sim}{V}^j_{h,i}(s) \leq V^{\bar{\pi}^j_h}_{h,i}(s)$, where the last step is by the induction hypothesis. Finally, observe that by our definition, the value of $V^{\bar{\pi}^j_h}_{h,i}(s)$ is a constant for all episode indices j that belong to the same stage. Since we know that episode j and episode k lie in the same stage, we can conclude that $V^{\bar{\pi}^k_h}_{h,i}(s) = V^{\bar{\pi}^j_h}_{h,i}(s) \geq \underset{\sim}{V}^k_{h,i}(s)$.

Again, combining the two cases and applying a union bound over all $(i,s,h,k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ complete the proof. □

The following result shows that the agents have no incentive to deviate from the correlated policy $\bar{\pi}$, up to a regret term of the order $\widetilde{O}(\sqrt{H^5 S A_{\max}/K})$.

Theorem 6. *For any $p \in (0,1]$, let $\iota = \log(2NSA_{\max}KH/p)$. Suppose $K \geq \frac{SH}{A_{\max}\iota}$, with probability at least $1-p$, it holds that*

$$V^{\star,\bar{\pi}_{-i}}_{1,i}(s_1) - V^{\bar{\pi}}_{1,i}(s_1) \leq O\left(\sqrt{H^5 S A_{\max}\iota/K}\right),$$

Proof. We first recall the definitions of several notations and define a few new ones. For a state s^k_h, recall that $\check{n}^k_h$ denotes the number of visits to the state s^k_h (at the h-th step) in the stage right before the current stage, and $\check{l}^k_{h,j}$ denotes the j-th episode among the $\check{n}^k_h$ episodes. Similarly, let n^k_h be the total number of episodes that this state has been visited prior to the current stage, and let $l^k_{h,j}$ denote the index of the episode that this state was visited the j-th time among the total n^k_h times. For simplicity, we use l_j and $\check{l}_j$ to denote $l^k_{h,j}$ and $\check{l}^k_{h,j}$, and $\check{n}$ to denote $\check{n}^k_h$, whenever h and k are clear from the context.

From Lemma 8, we know that

$$\begin{aligned} V^{\star,\bar{\pi}_{-i}}_{1,i}(s_1) - V^{\bar{\pi}}_{1,i}(s_1) \leq & \frac{1}{K}\sum_{k=1}^{K}\left(V^{\star,\bar{\pi}^k_{1,-i}}_{1,i}(s_1) - V^{\bar{\pi}^k_1}_{1,i}(s_1)\right) \\ \leq & \frac{1}{K}\sum_{k=1}^{K}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right). \end{aligned}$$

We hence only need to upper bound $\frac{1}{K}\sum_{k=1}^{K}(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1))$. For a fixed agent $i \in \mathcal{N}$, we define the following notation:

$$\delta^k_h \overset{\text{def}}{=} \overline{V}^k_{h,i}(s^k_h) - \underline{V}^k_{h,i}(s^k_h).$$

The main idea of the subsequent proof is to upper bound $\sum_{k=1}^{K}\delta^k_h$ by the next step $\sum_{k=1}^{K}\delta^k_{h+1}$, and then obtain a recursive formula. From the update rule of $\overline{V}^k_{h,i}(s^k_h)$ in (5), we know that

$$\overline{V}^k_{h,i}(s^k_h) \leq \mathbb{I}[n^k_h = 0]H + \frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(r_{h,i}(s_h, \boldsymbol{a}^{\check{l}_j}_h) + \overline{V}^{\check{l}_j}_{h+1,i}(s^{\check{l}_j}_{h+1})\right) + b_{\check{n}},$$

where the $\mathbb{I}[n^k_h = 0]$ term counts for the event that the optimistic value function has never been updated for the given state.

Further recalling the definition of $\underline{V}^k_{h,i}(s^k_h)$, we have

$$\begin{aligned} \delta^k_h \leq & \mathbb{I}[n^k_h = 0]H + \frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(\overline{V}^{\check{l}_j}_{h+1,i}(s^{\check{l}_j}_{h+1}) - \underline{V}^{\check{l}_j}_{h+1,i}(s^{\check{l}_j}_{h+1})\right) + 2b_{\check{n}} \\ \leq & \mathbb{I}[n^k_h = 0]H + \frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\delta^{\check{l}_j}_{h+1} + 2b_{\check{n}}, \end{aligned} \tag{12}$$

To find an upper bound of $\sum_{k=1}^K \delta_h^k$, we proceed to upper bound each term on the RHS of (12) separately. First, notice that $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right] \leq SH$, because each fixed state-step pair (s, h) contributes at most 1 to $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right]$. Next, we turn to analyze the second term on the RHS of (12). Observe that

$$\begin{aligned}
\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} =& \sum_{k=1}^K \sum_{m=1}^K \frac{1}{\check{n}_h^k} \delta_{h+1}^m \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \\
=& \sum_{m=1}^K \delta_{h+1}^m \sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right]. \qquad (13)
\end{aligned}$$

For a fixed episode m, notice that $\sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] \leq 1$, and that $\sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] = 1$ happens if and only if $s_h^k = s_h^m$ and (m, h) lies in the previous stage of (k, h) with respect to the state-step pair (s_h^k, h). Define $\mathcal{K}_m \stackrel{\text{def}}{=} \{k \in [K] : \sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] = 1\}$. We then know that all episode indices $k \in \mathcal{K}_m$ belong to the same stage, and hence these episodes have the same value of $\check{n}_h^k$. That is, there exists an integer $N_m > 0$, such that $\check{n}_h^k = N_m, \forall k \in \mathcal{K}_m$. Further, since the stages are partitioned in a way such that each stage is at most $(1 + \frac{1}{H})$ times longer than the previous stage, we know that $|\mathcal{K}_m| \leq (1 + \frac{1}{H})N_m$. Therefore, for every m, it holds that

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \leq 1 + \frac{1}{H}. \qquad (14)$$

Combining (13) and (14) leads to the following upper bound of the second term in (12):

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} \leq (1 + \frac{1}{H}) \sum_{k=1}^K \delta_{h+1}^k. \qquad (15)$$

So far, we have obtained the following upper bound:

$$\sum_{k=1}^K \delta_h^k \leq SH^2 + (1 + \frac{1}{H}) \sum_{k=1}^K \delta_{h+1}^k + 2 \sum_{k=1}^K b_{\check{n}_h^k}.$$

Iterating the above inequality over $h = H, H-1, \ldots, 1$ leads to

$$\sum_{k=1}^K \delta_1^k \leq O\left(SH^3 + \sum_{h=1}^H \sum_{k=1}^K (1 + \frac{1}{H})^{h-1} b_{\check{n}_h^k}\right), \qquad (16)$$

where we used the fact that $(1 + \frac{1}{H})^H \leq e$. In the following, we analyze the bonus term $b_{\check{n}_h^k}$ more carefully. Recall our definitions that $e_1 = H,\ e_{i+1} = \left\lfloor (1 + \frac{1}{H}) e_i \right\rfloor, i \geq 1$, and $b_{\check{n}} = 6\sqrt{H^2 A_i \iota / \check{n}}$. For any $h \in [H]$,

$$\begin{aligned}
\sum_{k=1}^K (1 + \frac{1}{H})^{h-1} b_{\check{n}_h^k} \leq& \sum_{k=1}^K (1 + \frac{1}{H})^{h-1} 6\sqrt{H^2 A_i \iota / \check{N}_h^k} \\
=& 6\sqrt{H^2 A_i \iota} \sum_{s \in \mathcal{S}} \sum_{j \geq 1} (1 + \frac{1}{H})^{h-1} e_j^{-\frac{1}{2}} \sum_{k=1}^K \mathbb{I}\left[s_h^k = s, \check{N}_h^k(s_h^k) = e_j\right] \\
=& 6\sqrt{H^2 A_i \iota} \sum_{s \in \mathcal{S}} \sum_{j \geq 1} (1 + \frac{1}{H})^{h-1} w(s, j) e_j^{-\frac{1}{2}},
\end{aligned}$$

where we define $w(s, j) \stackrel{\text{def}}{=} \sum_{k=1}^K \mathbb{I}\left[s_h^k = s, \check{N}_h^k(s_h^k) = e_j\right]$ for any $s \in \mathcal{S}$. If we further let $w(s) \stackrel{\text{def}}{=} \sum_{j \geq 1} w(s, j)$, we can see that $\sum_{s \in \mathcal{S}} w(s) = K$. For each fixed state s, we now seek an upper bound of its corresponding j

value, denoted as J in what follows. Since each stage is $(1+\frac{1}{H})$ times longer than its previous stage, we know that $w(s,j)=\sum_{k=1}^{K}\mathbb{I}\left[s_h^k=s,\check{N}_h^k(s_h^k)=e_j\right]=\lfloor(1+\frac{1}{H})e_j\rfloor$ for any $1\le j\le J$. Since $\sum_{j=1}^{J}w(s,j)=w(s)$, we obtain that $e_J\le(1+\frac{1}{H})^{J-1}\le\frac{10}{1+\frac{1}{H}}\frac{w(s)}{H}$ by taking the sum of a geometric sequence. Therefore, by plugging in $w(s,j)=\lfloor(1+\frac{1}{H})e_j\rfloor$,

$$\sum_{j\ge 1}(1+\frac{1}{H})^{h-1}w(s,j)e_j^{-\frac{1}{2}}\le O\left(\sum_{j=1}^{J}e_j^{\frac{1}{2}}\right)\le O\left(\sqrt{w(s)H}\right),$$

where in the second step we again used the formula of the sum of a geometric sequence. Finally, using the fact that $\sum_{s\in\mathcal{S}}w(s)=K$ and applying the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}b_{\check{n}_h^k}=&O\left(\sqrt{H^4A_i\iota}\sum_{s\in\mathcal{S}}\sum_{j\ge 1}(1+\frac{1}{H})^{h-1}w(s,j)e_j^{-\frac{1}{2}}\right)\\ \le&O\left(\sqrt{SA_iKH^5\iota}\right).\end{aligned}\tag{17}$$

Summarizing the results above leads to

$$\sum_{k=1}^{K}\delta_1^k\le O\left(SH^3+\sqrt{SA_iKH^5\iota}\right).$$

In the case when K is large enough, such that $K\ge\frac{SH}{A_i\iota}$, the second term becomes dominant, and we obtain the desired result:

$$V_{1,i}^{\star,\bar{\pi}_{-i}}(s_1)-V_{1,i}^{\bar{\pi}}(s_1)\le\frac{1}{K}\sum_{k=1}^{K}\delta_1^k\le O\left(\sqrt{SA_iH^5\iota/K}\right).$$

This completes the proof of the theorem. □

An immediate corollary is that we obtain an ε-approximate CCE when $\sqrt{SA_{\max}H^5\iota/K}\le\varepsilon$, which is Theorem 1 in the main text.

Theorem 1. (Sample complexity of learning CCE). For any $p\in(0,1]$, set $\iota=\log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 1 for K episodes with $K=O(SA_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ constitutes an ε-approximate coarse correlated equilibrium.

D. Proofs for Section 3.2

We first present a no-swap-regret learning algorithm for the adversarial bandit problem, which serves as an important subroutine to achieve correlated equilibria in Markov games. We consider a standard adversarial bandit problem that lasts for T time steps. The agent has an action space of $\mathcal{A}=\{1,\dots,A\}$. At each time step $t\in[T]$, the agent specifies a distribution $p_t\in\Delta(\mathcal{A})$ over the action space, and takes an action a_t according to p_t. The adversary then selects a loss vector $l_t\in[0,1]^A$, where $l_t(a)\in[0,1]$ denotes the loss of action a at time t. We consider partial information (bandit) feedback, where the agent only receives the reward associated with the selected action a_t. The external regret measures the difference between the cumulative reward that an algorithm obtains and that of the best fixed action in hindsight. Specifically,

$$R_{\text{external}}(T)=\max_{a^\star\in\mathcal{A}}\sum_{t=1}^{T}\left(l_t(a_t)-l_t(a^\star)\right).$$

The swap regret, instead, measures the difference between the cumulative reward of an algorithm and the cumulative reward that could be achieved by swapping multiple pairs of actions of the algorithm. To be more specific, we define

Algorithm 5: No-swap-regret learning

```
Initialize: $p_1(a) \leftarrow 1/A, \forall a \in \mathcal{A}$, $\gamma \leftarrow \sqrt{\log A/T}$, and $\eta \leftarrow \sqrt{\log A/T}$.
for $t \leftarrow 1$ to $T$ do
    Take action $a_t \sim p_t(\cdot)$, and observe loss $l_t(a_t)$;
    for action $a \in \mathcal{A}$ do
        for action $a' \in \mathcal{A}$ do
            $\hat{l}_t(a' \mid a) \leftarrow p_t(a) l_t(a_t) \mathbb{I}\{a_t = a'\}/(p_t(a') + \gamma)$;
            $q_{t+1}(a' \mid a) \leftarrow \frac{\exp(-\eta \sum_{i=1}^t \hat{l}_i(a'|a))}{\sum_{b \in \mathcal{A}} \exp(-\eta \sum_{i=1}^t \hat{l}_i(b|a))}$;
    Set $p_{t+1}$ such that $p_{t+1}(\cdot) = \sum_{a \in \mathcal{A}} p_{t+1}(a) q_{t+1}(\cdot \mid a)$;
```

a strategy modification $F : \mathcal{A} \rightarrow \mathcal{A}$ to be a mapping from the action space to itself. For any action selection distribution p, we let $F \diamond p$ be the swapped distribution that takes action $a \in \mathcal{A}$ with probability $\sum_{a' \in \mathcal{A}, F(a') = a} p(a')$. The swap regret[2] is then defined as

$$R_{\text{swap}}(T) = \max_{F: \mathcal{A} \rightarrow \mathcal{A}} \sum_{t=1}^{T} \left(\langle p_t, l_t \rangle - \langle F \diamond p_t, l_t \rangle \right),$$

where recall that p_t is the distribution that the algorithm specifies at time t for action selection.

We follow the generic reduction introduced in Blum & Mansour (2007), and convert a Follow-the-Regularized-Leader algorithm with sublinear external regret to a no-swap-regret algorithm (Jin et al., 2021). The resulting algorithm is presented as Algorithm 5. The following lemma shows that Algorithm 5 is indeed a no-swap-regret learning algorithm.

Lemma 9. *(Jin et al., 2021, Theorem 26). For any $T \in \mathbb{N}$ and $p \in (0, 1)$, let $\iota = \log(A^2/p)$. With probability at least $1 - 3p$, it holds that*

$$R_{swap}(T) \leq 10\sqrt{A^2 T \iota}.$$

It is worth noting that Jin et al. (2021) presented a more general analysis with an anytime weighted swap regret guarantee. Such complication can be avoided in our algorithm, as our stage-based learning approach only entails a simple averaged swap regret analysis.

The complete Stage-Based V-Learning algorithm for CE is presented in Algorithm 6. In the following analysis, we follow the same notations as have been used in the CCE analysis. We again start with the following lemma that justifies our choice of the bonus term.

Lemma 10. *With probability at least $1 - \frac{p}{2}$, it holds for all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ that*

$$\max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi_{h,i}^s \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) \leq 11\sqrt{H^2 A_i^2 \iota / \check{n}}.$$

Proof. For a fixed $(s, h, k) \in \mathcal{S} \times [H] \times [K]$, let $\mathcal{F}_j$ be the σ-algebra generated by all the random variables up to episode $\check{l}_j$. Then, $\left\{ r_{h,i}(s, \boldsymbol{a}_{h,i}^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) - \mathbb{D}_{\boldsymbol{\mu}_{h,i}^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) \right\}_{j=1}^{\check{n}}$ is a martingale difference sequence with respect to $\{\mathcal{F}_j\}_{j=1}^{\check{n}}$. From the Azuma-Hoeffding inequality, it holds with probability at least $1 - p/(4NSHK)$ that

$$\frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) \leq \sqrt{H^2 \iota / \check{n}}.$$

[2] This is a modified version of the swap regret used in Blum & Mansour (2007), which is defined as $R_{\text{swap}}(T) = \max_{F: \mathcal{A} \rightarrow \mathcal{A}} \sum_{t=1}^{T} \left(l_t(a_t) - l_t(F(a_t)) \right)$.

Algorithm 6: Stage-Based V-Learning for CE (agent i)

```
Initialize: $\overline{V}_{h,i}(s) \leftarrow H-h+1, \tilde{V}_{h,i}(s) \leftarrow H-h+1, N_h(s) \leftarrow 0, \check{N}_h(s) \leftarrow 0, \check{r}_{h,i}(s) \leftarrow 0, \check{v}_{h,i}(s) \leftarrow 0,$
  $\check{T}_h(s) \leftarrow H, p_{h,i}(a \mid s) \leftarrow 1/A_i, L^s_{h,i}(a' \mid a) \leftarrow 0, \forall h \in [H], s \in \mathcal{S}, a, a' \in \mathcal{A}_i$.
for episode $k \leftarrow 1$ to $K$ do
    Receive $s_1$;
    for step $h \leftarrow 1$ to $H$ do
        $N_h(s_h) \leftarrow N_h(s_h) + 1, \check{n} \overset{\text{def}}{=} \check{N}_h(s_h) \leftarrow \check{N}_h(s_h) + 1$;
        Take action $a_{h,i} \sim p_{h,i}(\cdot \mid s_h)$, and observe reward $r_{h,i}$ and next state $s_{h+1}$;
        $\check{r}_{h,i}(s_h) \leftarrow \check{r}_{h,i}(s_h) + r_{h,i}, \check{v}_{h,i}(s_h) \leftarrow \check{v}_{h,i}(s_h) + \overline{V}_{h+1,i}(s_{h+1})$;
        $\eta_i \leftarrow \sqrt{\iota/\check{T}_h(s_h)}, \gamma_i \leftarrow \eta_i$;
        for action $a \in \mathcal{A}_i$ do
            for action $a' \in \mathcal{A}_i$ do
                $L^s_{h,i}(a' \mid a) \leftarrow L^s_{h,i}(a' \mid a) + \frac{p_{h,i}(a|s_h)[H-h+1-(r_{h,i}+\overline{V}_{h+1,i}(s_{h+1}))]}{H(p_{h,i}(a_{h,i}|s_h)+\gamma_i)}\mathbb{I}\{a_{h,i} = a\}$;
                $q^{s_h}_{h,i}(a' \mid a) \leftarrow \frac{\exp(-\eta_i L^{s_h}_{h,i}(a'|a))}{\sum_{b \in \mathcal{A}_i} \exp(-\eta_i L^{s_h}_{h,i}(b|a))}$;
        Set $p_{h,i}(a \mid s_h)$ such that $p_{h,i}(\cdot \mid s_h) = \sum_{a \in \mathcal{A}} p_{h,i}(a \mid s_h) q^{s_h}_{h,i}(\cdot \mid a)$;
        if $N_h(s_h) \in \mathcal{L}$ then
            //Entering a new stage
            $\tilde{V}_{h,i}(s_h) \leftarrow \frac{\check{r}_{h,i}(s_h)}{\check{n}} + \frac{\check{v}_{h,i}(s_h)}{\check{n}} + b_{\check{n}}$, where $b_{\check{n}} \leftarrow 11\sqrt{H^2 A_i^2 \iota/\check{n}}$;
            $\overline{V}_{h,i}(s_h) \leftarrow \min\{\tilde{V}_{h,i}(s_h), H-h+1\}$;
            $\check{N}_h(s_h) \leftarrow 0, \check{r}_{h,i}(s_h) \leftarrow 0, \check{v}_{h,i}(s_h) \leftarrow 0, \check{T}_h(s_h) \leftarrow \lfloor (1+\frac{1}{H})\check{T}_h(s_h) \rfloor$;
            $p_{h,i}(a \mid s_h) \leftarrow 1/A_i, L^{s_h}_{h,i}(a' \mid a) \leftarrow 0, \forall a, a' \in \mathcal{A}_i$;
```

Therefore, we only need to bound

$$R_{\text{swap}}(\check{n}) \overset{\text{def}}{=} \max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi^s_{h,i} \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{\check{l}_j}_{h+1,i} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{\check{l}_j}_{h+1,i} \right)(s).$$

Notice that $R_{\text{swap}}(\check{n})$ can be considered as the swap regret of an adversarial bandit problem at state s, where the loss function at step $j \in [\check{n}]$ is defined as

$$\ell_j(a_i) = \mathbb{E}_{a_{-i} \sim \mu^{\check{l}_j}_{h,-i}}(s) \left[H - h + 1 - r_{h,i}(s, \boldsymbol{a}) - \mathbb{P}_h \overline{V}^{\check{l}_j}_{h+1,i}(s, \boldsymbol{a}) \right] / H.$$

Such a problem can be addressed by a no-swap-regret learning algorithm as presented in Algorithm 5. Applying Lemma 9, we obtain that with probability at least $1 - \frac{p}{4NHS}$, it holds for all $k \in [K]$ that

$$R_{\text{swap}}(\check{n}) \le 10\sqrt{\frac{H^2 A_i^2 \iota}{\check{n}}}.$$

A union bound over all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ completes the proof. □

We again define the notations $\bar{\pi}^k_h, \bar{\pi}, V^{\bar{\pi}^k_h}_{h,i}, \underline{V}^k_{h,i}$, and $\underset{\sim}{V}^k_{h,i}(s)$ in the same sense as in Appendix C. The next lemma shows that $\overline{V}^k_{h,i}(s)$ and $\underline{V}^k_{h,i}(s)$ are valid upper and lower bounds.

Lemma 11. *It holds with probability at least $1-p$ that for all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$,*

$$\overline{V}^k_{h,i}(s) \ge \max_{\psi_i \in \Psi_i} V^{\psi_i \diamond \bar{\pi}^k_h}_{h,i}(s), \textit{ and } \underline{V}^k_{h,i}(s) \le V^{\bar{\pi}^k_h}_{h,i}(s).$$

Proof. Consider a fixed $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$. The desired result clearly holds for any state s that is in its first stage, due to our initialization of $\overline{V}_{h,i}^{k}(s)$ and $\underline{V}_{h,i}^{k}(s)$ for this special case. In the following, we only need to focus on the case where $\overline{V}_{h,i}(s)$ and $\underline{V}_{h,i}^{k}(s)$ have been updated at least once at the given state s before the k-th episode.

We start with the first inequality. It suffices to show that $\tilde{V}_{h,i}^{k}(s) \geq \max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s)$ because $\overline{V}_{h,i}^{k}(s) = \min\{\tilde{V}_{h,i}^{k}(s), H - h + 1\}$, and $\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s)$ is always less than or equal to $H - h + 1$. Our proof relies on induction on $k \in [K]$. First, the claim holds for $k = 1$ due to the aforementioned logic. For each step $h \in [H]$ and $s \in \mathcal{S}$, we consider the following two cases.

Case 1: $\tilde{V}_{h,i}(s)$ has just been updated in (the end of) episode $k - 1$. In this case,

$$\tilde{V}_{h,i}^{k}(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + b_{\check{n}}. \tag{18}$$

By the definition of $\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s)$, it holds with probability at least $1 - \frac{p}{2NSKH}$ that

$$\begin{aligned}
\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s) \leq & \max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi_i \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \max_{\psi_i' \in \Psi_i} V_{h+1,i}^{\psi_i' \diamond \bar{\pi}_{h+1}^{\check{l}_j}} \right)(s) \\
\leq & \max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi_i \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) \\
\leq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + 11\sqrt{H^2 A_i^2 \iota / \check{n}} \\
\leq & \tilde{V}_{h,i}^{k}(s),
\end{aligned} \tag{19}$$

where the second step is by the induction hypothesis, the third step holds due to Lemma 10, and the last step is by the definition of $b_{\check{n}}$.

Case 2: $\tilde{V}_{h,i}(s)$ was not updated in (the end of) episode $k - 1$. Since we have excluded the case that $\tilde{V}_{h,i}$ has never been updated, we are guaranteed that there exists an episode j such that $\tilde{V}_{h,i}(s)$ has been updated in the end of episode $j - 1$ most recently. In this case, $\tilde{V}_{h,i}^{k}(s) = \tilde{V}_{h,i}^{k-1}(s) = \cdots = \tilde{V}_{h,i}^{j}(s) \geq \max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^j}(s)$, where the last step is by the induction hypothesis. Finally, observe that by our definition, the value of $\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^j}(s)$ is a constant for all episode indices j that belong to the same stage. Since we know that episode j and episode k lie in the same stage, we can conclude that $\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s) = \max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^j}(s) \leq \tilde{V}_{h,i}^{k}(s)$.

Combining the two cases and applying a union bound over all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ complete the proof of the first inequality.

Next, we prove the second inequality in the statement of the lemma. Notice that it suffices to show $\underset{\sim}{V}_{h,i}^{k}(s) \leq V_{h,i}^{\bar{\pi}_h^k}(s)$ because $\underline{V}_{h,i}^{k}(s) = \max\{\underset{\sim}{V}_{h,i}^{k}(s), 0\}$. Our proof again relies on induction on $k \in [K]$. Similar to the proof of the first inequality, the claim apparently holds for $k = 1$, and we consider the following two cases for each step $h \in [H]$ and $s \in \mathcal{S}$.

Case 1: The value of $\underset{\sim}{V}_{h,i}(s)$ has just changed in (the end of) episode $k - 1$. In this case,

$$\underset{\sim}{V}_{h,i}^{k}(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) - b_{\check{n}}. \tag{20}$$

By the definition of $V_{h,i}^{\bar{\pi}_h^k}(s)$, it holds with probability at least $1-\frac{p}{2NSKH}$ that

$$\begin{aligned}
V_{h,i}^{\bar{\pi}_h^k}(s) =& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}}\left(r_{h,i}+\mathbb{P}_h V_{h+1,i}^{\bar{\pi}_{h+1}^{\check{l}_j}}\right)(s)\\
\geq& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}}\left(r_{h,i}+\mathbb{P}_h \underset{\sim}{V}_{h+1,i}^{\check{l}_j}\right)(s)\\
\geq& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \left(r_{h,i}(s,\boldsymbol{a}_h^{\check{l}_j})+\underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)-\sqrt{H^2\iota/\check{n}}\\
\geq& \underset{\sim}{V}_{h,i}^{k}(s),
\end{aligned}\tag{21}$$

where the second step is by the induction hypothesis, the third step holds due to the Azuma-Hoeffding inequality, and the last step is by the definition of $b_{\check{n}}$.

Case 2: The value of $\underset{\sim}{V}_{h,i}(s)$ has not changed in (the end of) episode $k-1$. Since we have excluded the case that $\underset{\sim}{V}_{h,i}$ has never been updated, we are guaranteed that there exists an episode j such that $\underset{\sim}{V}_{h,i}(s)$ has changed in the end of episode $j-1$ most recently. In this case, we know that indices j and k belong to the same stage, and $\underset{\sim}{V}_{h,i}^{k}(s)=\underset{\sim}{V}_{h,i}^{k-1}(s)=\cdots=\underset{\sim}{V}_{h,i}^{j}(s)\leq V_{h,i}^{\bar{\pi}_h^j}(s)$, where the last step is by the induction hypothesis. Finally, observe that by our definition, the value of $V_{h,i}^{\bar{\pi}_h^j}(s)$ is a constant for all episode indices j that belong to the same stage. Since we know that episode j and episode k lie in the same stage, we can conclude that $V_{h,i}^{\bar{\pi}_h^k}(s)=V_{h,i}^{\bar{\pi}_h^j}(s)\geq\underset{\sim}{V}_{h,i}^{k}(s)$.

Again, combining the two cases and applying a union bound over all $(i,s,h,k)\in\mathcal{N}\times\mathcal{S}\times[H]\times[K]$ complete the proof. □

Theorem 7. *For any $p\in(0,1]$, let $\iota=\log(2NSA_{\max}KH/p)$. Suppose $K\geq\frac{SH}{A_{\max}^2\iota}$. With probability at least $1-p$,*

$$\max_{\psi_i\in\Psi_i} V_{1,i}^{\psi_i\diamond\bar{\pi}}(s_1)-V_{1,i}^{\bar{\pi}}(s_1)\leq O\left(\sqrt{H^5SA_{\max}^2\iota/K}\right),$$

Proof. The proof follows a similar procedure as the proof of Theorem 6. From Lemma 11, we know that

$$\begin{aligned}
\max_{\psi_i\in\Psi_i} V_{1,i}^{\psi_i\diamond\bar{\pi}}(s_1)-V_{1,i}^{\bar{\pi}}(s_1)=&\max_{\psi_i\in\Psi_i}\frac{1}{K}\sum_{k=1}^{K}\left(V_{1,i}^{\psi_i\diamond\bar{\pi}_1^k}(s_1)-V_{1,i}^{\bar{\pi}_1^k}(s_1)\right)\\
\leq&\frac{1}{K}\sum_{k=1}^{K}\left(\max_{\psi_i\in\Psi_i}V_{1,i}^{\psi_i\diamond\bar{\pi}_1^k}(s_1)-V_{1,i}^{\bar{\pi}_1^k}(s_1)\right)\\
\leq&\frac{1}{K}\sum_{k=1}^{K}\left(\overline{V}_{1,i}^{k}(s_1)-\underline{V}_{1,i}^{k}(s_1)\right).
\end{aligned}$$

We hence only need to upper bound $\frac{1}{K}\sum_{k=1}^{K}(\overline{V}_{1,i}^{k}(s_1)-\underline{V}_{1,i}^{k}(s_1))$. For a fixed agent $i\in\mathcal{N}$, we define the following notation:

$$\delta_h^k\overset{\text{def}}{=}\overline{V}_{h,i}^{k}(s_h^k)-\underline{V}_{h,i}^{k}(s_h^k).$$

The main idea of the subsequent proof is to upper bound $\sum_{k=1}^{K}\delta_h^k$ by the next step $\sum_{k=1}^{K}\delta_{h+1}^k$, and then obtain a recursive formula. From the update rule of $\overline{V}_{h,i}^{k}(s_h^k)$ in (5), we know that

$$\overline{V}_{h,i}^{k}(s_h^k)\leq\mathbb{I}[n_h^k=0]H+\frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(r_{h,i}(s_h,\boldsymbol{a}_h^{\check{l}_j})+\overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)+b_{\check{n}},$$

where the $\mathbb{I}[n_h^k = 0]$ term counts for the event that the optimistic value function has never been updated for the given state.

Further recalling the definition of $\underline{V}_{h,i}^k(s_h^k)$, we have

$$\begin{aligned}\delta_h^k \leq & \mathbb{I}[n_h^k = 0]H + \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(\overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) - \underline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + 2b_{\check{n}} \\ \leq & \mathbb{I}[n_h^k = 0]H + \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \delta_{h+1}^{\check{l}_j} + 2b_{\check{n}}, \end{aligned} \tag{22}$$

To find an upper bound of $\sum_{k=1}^K \delta_h^k$, we proceed to upper bound each term on the RHS of (22) separately. First, notice that $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right] \leq SH$, because each fixed state-step pair (s, h) contributes at most 1 to $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right]$. Next, we turn to analyze the second term on the RHS of (22). Observe that

$$\begin{aligned}\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} = & \sum_{k=1}^K \sum_{m=1}^K \frac{1}{\check{n}_h^k} \delta_{h+1}^m \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \\ = & \sum_{m=1}^K \delta_{h+1}^m \sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right]. \end{aligned} \tag{23}$$

For a fixed episode m, notice that $\sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] \leq 1$, and that $\sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] = 1$ happens if and only if $s_h^k = s_h^m$ and (m, h) lies in the previous stage of (k, h) with respect to the state-step pair (s_h^k, h). Define $\mathcal{K}_m \overset{\text{def}}{=} \{k \in [K] : \sum_{j=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,j}^k = m] = 1\}$. We then know that all episode indices $k \in \mathcal{K}_m$ belong to the same stage, and hence these episodes have the same value of $\check{n}_h^k$. That is, there exists an integer $N_m > 0$, such that $\check{n}_h^k = N_m, \forall k \in \mathcal{K}_m$. Further, since the stages are partitioned in a way such that each stage is at most $(1 + \frac{1}{H})$ times longer than the previous stage, we know that $|\mathcal{K}_m| \leq (1 + \frac{1}{H})N_m$. Therefore, for every m, it holds that

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \leq 1 + \frac{1}{H}. \tag{24}$$

Combining (23) and (24) leads to the following upper bound of the second term in (22):

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} \leq (1 + \frac{1}{H}) \sum_{k=1}^K \delta_{h+1}^k. \tag{25}$$

So far, we have obtained the following upper bound:

$$\sum_{k=1}^K \delta_h^k \leq SH^2 + (1 + \frac{1}{H}) \sum_{k=1}^K \delta_{h+1}^k + 2 \sum_{k=1}^K b_{\check{n}_h^k}.$$

Iterating the above inequality over $h = H, H-1, \dots, 1$ leads to

$$\sum_{k=1}^K \delta_1^k \leq O\left(SH^3 + \sum_{h=1}^H \sum_{k=1}^K (1 + \frac{1}{H})^{h-1} b_{\check{n}_h^k} \right), \tag{26}$$

where we used the fact that $(1 + \frac{1}{H})^H \leq e$. In the following, we analyze the bonus term $b_{\check{n}_h^k}$ more carefully. Recall our definitions that $e_1 = H,\ e_{i+1} = \left\lfloor (1+\frac{1}{H})e_i \right\rfloor, i \geq 1$, and $b_{\check{n}} = 11\sqrt{H^2A_i^2\iota/\check{n}}$. For any $h \in [H]$,

$$\begin{aligned}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}b_{\check{n}_h^k} \leq &\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}11\sqrt{H^2A_i^2\iota/\check{N}_h^k}\\ =&11\sqrt{H^2A_i^2\iota}\sum_{s\in\mathcal{S}}\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}e_j^{-\frac{1}{2}}\sum_{k=1}^{K}\mathbb{I}\left[s_h^k = s, \check{N}_h^k(s_h^k) = e_j\right]\\ =&11\sqrt{H^2A_i^2\iota}\sum_{s\in\mathcal{S}}\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}w(s,j)e_j^{-\frac{1}{2}},\end{aligned}$$

where we define $w(s,j) \stackrel{\text{def}}{=} \sum_{k=1}^{K}\mathbb{I}\left[s_h^k = s, \check{N}_h^k(s_h^k) = e_j\right]$ for any $s \in \mathcal{S}$. If we further let $w(s) \stackrel{\text{def}}{=} \sum_{j\geq 1} w(s,j)$, we can see that $\sum_{s\in\mathcal{S}} w(s) = K$. For each fixed state s, we now seek an upper bound of its corresponding j value, denoted as J in what follows. Since each stage is $(1+\frac{1}{H})$ times longer than its previous stage, we know that $w(s,j) = \sum_{k=1}^{K}\mathbb{I}\left[s_h^k = s, \check{N}_h^k(s_h^k) = e_j\right] = \left\lfloor (1+\frac{1}{H})e_j \right\rfloor$ for any $1 \leq j \leq J$. Since $\sum_{j=1}^{J} w(s,j) = w(s)$, we obtain that $e_J \leq (1+\frac{1}{H})^{J-1} \leq \frac{10}{1+\frac{1}{H}}\frac{w(s)}{H}$ by taking the sum of a geometric sequence. Therefore, by plugging in $w(s,j) = \lfloor (1+\frac{1}{H})e_j \rfloor$,

$$\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}w(s,j)e_j^{-\frac{1}{2}} \leq O\left(\sum_{j=1}^{J} e_j^{\frac{1}{2}}\right) \leq O\left(\sqrt{w(s)H}\right),$$

where in the second step we again used the formula of the sum of a geometric sequence. Finally, using the fact that $\sum_{s\in\mathcal{S}} w(s) = K$ and applying the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}b_{\check{n}_h^k} =&O\left(\sqrt{H^4A_i^2\iota}\sum_{s\in\mathcal{S}}\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}w(s,j)e_j^{-\frac{1}{2}}\right)\\ \leq&O\left(\sqrt{SA_i^2KH^5\iota}\right). \qquad (27)\end{aligned}$$

Summarizing the results above leads to

$$\sum_{k=1}^{K}\delta_1^k \leq O\left(SH^3 + \sqrt{SA_i^2KH^5\iota}\right).$$

In the case when K is large enough, such that $K \geq \frac{SH}{A_i^2\iota}$, the second term becomes dominant, and we obtain the desired result:

$$\max_{\psi_i\in\Psi_i} V_{1,i}^{\psi_i\diamond\bar{\pi}}(s_1) - V_{1,i}^{\bar{\pi}}(s_1) \leq \frac{1}{K}\sum_{k=1}^{K}\delta_1^k \leq O\left(\sqrt{SA_i^2H^5\iota/K}\right).$$

This completes the proof of the theorem. □

An immediate corollary is that we obtain an ε-approximate CE when $\sqrt{SA_{\max}^2H^5\iota/K} \leq \varepsilon$, which is Theorem 2 in the main text.

Theorem 2. (Sample complexity of learning CE). For any $p \in (0,1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 6 for K episodes with $K = O(SA_{\max}^2H^5\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ constitutes an ε-approximate correlated equilibrium.

E. Proofs for Section 4.1

We start with a multi-agent variant of the performance difference lemma (Kakade & Langford, 2002) in the finite-horizon setting.

Lemma 12. *(Performance difference lemma). For any policy $\pi = (\pi_i, \pi_{-i}) \in \Pi$ and $\pi' = (\pi'_i, \pi_{-i}) \in \Pi$, it holds for any $i \in \mathcal{N}$ that*

$$V^{\pi}_{1,i}(\rho) - V^{\pi'}_{1,i}(\rho) = \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi}_{h,\rho}} \mathbb{E}_{\boldsymbol{a}_h \sim \pi_h(\cdot|s_h)} \left[A^{\pi'}_{h,i}(s_h, \boldsymbol{a}_h) \right],$$

where $A^{\pi'}_{h,i}(s_h, \boldsymbol{a}_h) = Q^{\pi'}_{h,i}(s_h, \boldsymbol{a}_h) - V^{\pi'}_{h,i}(s_h)$ is the advantage function.

Proof. For any state-action trajectory $\tau = (s_1, \boldsymbol{a}_1, \ldots, s_H, \boldsymbol{a}_H)$, let $\mathbb{P}^{\pi}(\tau \mid \rho)$ denote the probability of observing the trajectory τ by following the policy π starting from the initial state distribution ρ. From the definition of the value function, it holds that

$$\begin{aligned}
& V^{\pi}_{1,i}(\rho) - V^{\pi'}_{1,i}(\rho) \\
=& \mathbb{E}_{\tau \sim \mathbb{P}^{\pi}(\tau|\rho)} \left[\sum_{h=1}^{H} r_h(s_h, \boldsymbol{a}_h) \right] - V^{\pi'}_{1,i}(\rho) \\
=& \mathbb{E}_{\tau \sim \mathbb{P}^{\pi}(\tau|\rho)} \left[\sum_{h=1}^{H} \left(r_h(s_h, \boldsymbol{a}_h) + V^{\pi'}_{h,i}(s_h) - V^{\pi'}_{h,i}(s_h) \right) \right] - V^{\pi'}_{1,i}(\rho) \\
=& \mathbb{E}_{\tau \sim \mathbb{P}^{\pi}(\tau|\rho)} \left[\sum_{h=1}^{H-1} \left(r_h(s_h, \boldsymbol{a}_h) + V^{\pi'}_{h+1,i}(s_{h+1}) - V^{\pi'}_{h,i}(s_h) \right) + r_H(s_H, \boldsymbol{a}_H) - V^{\pi'}_{H,i}(s_H) \right] \\
\overset{(a)}{=}& \mathbb{E}_{\tau \sim \mathbb{P}^{\pi}(\tau|\rho)} \left[\sum_{h=1}^{H-1} \left(r_h(s_h, \boldsymbol{a}_h) + \mathbb{E}\left[V^{\pi'}_{h+1,i}(s_{h+1}) \mid s_h, \boldsymbol{a}_h \right] - V^{\pi'}_{h,i}(s_h) \right) + r_H(s_H, \boldsymbol{a}_H) - V^{\pi'}_{H,i}(s_H) \right] \\
\overset{(b)}{=}& \mathbb{E}_{\tau \sim \mathbb{P}^{\pi}(\tau|\rho)} \left[\sum_{h=1}^{H} A^{\pi'}_{h,i}(s_h, \boldsymbol{a}_h) \right] = \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi}_{h,\rho}} \mathbb{E}_{\boldsymbol{a}_h \sim \pi_h(\cdot|s_h)} \left[A^{\pi'}_{h,i}(s_h, \boldsymbol{a}_h) \right],
\end{aligned}$$

where (a) uses the tower property of conditional expectation, and (b) is due to the definition of the advantage function and the fact that $Q^{\pi'}_{H,i}(s_h, \boldsymbol{a}_H) = r_H(s_H, \boldsymbol{a}_H)$. □

In the following, we reproduce a variant of the policy gradient theorem (Sutton et al., 2000) in the setting of finite-horizon MPGs.

Lemma 13. *(Policy gradient theorem). For any $i \in \mathcal{N}$, it holds that*

$$\nabla V^{\pi}_{1,i}(\rho) = \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi}_{h,\rho}} \mathbb{E}_{\boldsymbol{a}_h \sim \pi_h(\cdot|s_h)} \left[Q^{\pi}_{h,i}(s_h, \boldsymbol{a}_h) \nabla \log \pi_h(\boldsymbol{a}_h \mid s_h) \right].$$

Proof. For any fixed initial state $s_1 \in \mathcal{S}$, differentiating both sides of the Bellman equation leads to

$$\begin{aligned}
& \nabla V^{\pi}_{1,i}(s_1) \\
=& \nabla \sum_{\boldsymbol{a}_1} \pi_1(\boldsymbol{a}_1 \mid s_1) Q^{\pi}_{1,i}(s_1, \boldsymbol{a}_1) \\
=& \sum_{\boldsymbol{a}_1} \left(\nabla \pi_1(\boldsymbol{a}_1 \mid s_1) Q^{\pi}_{1,i}(s_1, \boldsymbol{a}_1) + \pi_1(\boldsymbol{a}_1 \mid s_1) \nabla Q^{\pi}_{1,i}(s_1, \boldsymbol{a}_1) \right) \\
=& \sum_{\boldsymbol{a}_1} \left(\pi_1(\boldsymbol{a}_1 \mid s_1) \nabla \log \pi_1(\boldsymbol{a}_1 \mid s_1) Q^{\pi}_{1,i}(s_1, \boldsymbol{a}_1) + \pi_1(\boldsymbol{a}_1 \mid s_1) \nabla \left(r_{1,i}(s_1, \boldsymbol{a}_1) + \mathbb{E}_{s_2 \sim P_1(\cdot|s_1, \boldsymbol{a}_1)} \left[V^{\pi}_{2,i}(s_2) \right] \right) \right) \\
=& \sum_{\boldsymbol{a}_1} \pi_1(\boldsymbol{a}_1 \mid s_1) \left(\nabla \log \pi_1(\boldsymbol{a}_1 \mid s_1) Q^{\pi}_{1,i}(s_1, \boldsymbol{a}_1) + \mathbb{E}_{s_2 \sim P_1(\cdot|s_1, \boldsymbol{a}_1)} \left[\nabla V^{\pi}_{2,i}(s_2) \right] \right).
\end{aligned}$$

From the linearity of expectation, we know that for any state distribution ρ,

$$\nabla V^\pi_{1,i}(\rho) = \nabla\left(\mathbb{E}_{s_1\sim d^\pi_{1,\rho}}\left[V^\pi_{1,i}(s_1)\right]\right) = \mathbb{E}_{s_1\sim d^\pi_{1,\rho}}\mathbb{E}_{\boldsymbol{a}_1\sim\pi_1(\cdot|s_1)}\left[Q^\pi_{1,i}(s_1,\boldsymbol{a}_1)\nabla\log\pi_1(\boldsymbol{a}_1\mid s_1)\right] + \mathbb{E}_{s_2\sim d^\pi_{2,\rho}}\left[\nabla V^\pi_{2,i}(s_2)\right].$$

Applying the above equation recursively, we obtain that

$$\nabla V^\pi_{1,i}(\rho) = \sum_{h=1}^{H}\mathbb{E}_{s_h\sim d^\pi_{h,\rho}}\mathbb{E}_{\boldsymbol{a}_h\sim\pi_h(\cdot|s_h)}\left[Q^\pi_{h,i}(s_h,\boldsymbol{a}_h)\nabla\log\pi_h(\boldsymbol{a}_h\mid s_h)\right].$$

This completes the proof of the policy gradient theorem in the finite-horizon case. □

With direct parameterization, we can further derive from the policy gradient theorem that for any $h\in[H], s\in\mathcal{S}, a\in\mathcal{A}_i$,

$$\frac{\partial V^\pi_{1,i}(\rho)}{\partial\pi_{h,i}(a\mid s)} = \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[d^\pi_{h,\rho}(s)Q^\pi_{h,i}(s,a,a_{h,-i})\right]. \tag{28}$$

In the following, we state and prove a finite-horizon variant of the gradient domination property, which has been shown in single-agent policy gradient methods (Agarwal et al., 2021) and infinite-horizon discounted MPGs (Zhang et al., 2021; Leonardos et al., 2021).

Lemma 14. *(Gradient domination). For any policy $\pi=(\pi_i,\pi_{-i})\in\Pi$ in a Markov potential game, let $\pi^\star_i$ be agent i's best response to π_{-i}, and let $\pi^\star=(\pi^\star_i,\pi_{-i})$. With direct policy parameterization, for any initial state distribution $\rho\in\Delta(\mathcal{S})$, it holds that*

$$V^{\pi^\star}_{1,i}(\rho) - V^\pi_{1,i}(\rho) \le \left\|\frac{d^{\pi^\star}_\rho}{d^\pi_\rho}\right\|_\infty \max_{\overline{\pi}_i\in\Pi_i}(\overline{\pi}-\pi)^\intercal\nabla_{\pi_i}V^\pi_{1,i}(\rho),$$

where the L^∞ norm takes the maximum over $[H]\times\mathcal{S}$.

Proof. From the performance difference lemma (Lemma 12), we know that

$$\begin{aligned}
V^{\pi^\star}_{1,i}(\rho) - V^\pi_{1,i}(\rho) &= \sum_{h=1}^{H}\mathbb{E}_{s_h\sim d^{\pi^\star}_{h,\rho}}\mathbb{E}_{a_{h,i}\sim\pi^\star_{h,i}(\cdot|s_h)}\mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[A^\pi_{h,i}(s_h,a_{h,i},a_{h,-i})\right]\\
&\le \sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d^{\pi^\star}_{h,\rho}(s_h)\max_{a^\star_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,-i}(a_{h,-i}\mid s_h)A^\pi_{h,i}(s_h,a^\star_{h,i},a_{h,-i})\\
&= \sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}}\frac{d^{\pi^\star}_{h,\rho}(s_h)}{d^\pi_{h,\rho}(s_h)}d^\pi_{h,\rho}(s_h)\max_{a^\star_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,-i}(a_{h,-i}\mid s_h)A^\pi_{h,i}(s_h,a^\star_{h,i},a_{h,-i})\\
&\le \left(\max_{h\in[H],s_h\in\mathcal{S}}\frac{d^{\pi^\star}_{h,\rho}(s_h)}{d^\pi_{h,\rho}(s_h)}\right)\sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d^\pi_{h,\rho}(s_h)\max_{a^\star_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,-i}(a_{h,-i}\mid s_h)A^\pi_{h,i}(s_h,a^\star_{h,i},a_{h,-i}),
\end{aligned}$$

where in the last step we used the fact that

$$\sum_{a_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,i}(a_{h,i}\mid s_h)\pi_{h,-i}(a_{h,-i}\mid s_h)A^\pi_{h,i}(s_h,a_{h,i},a_{h,-i}) = 0, \tag{29}$$

which in turn implies that

$$\max_{a^\star_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,-i}(a_{h,-i}\mid s_h)A^\pi_{h,i}(s_h,a^\star_{h,i},a_{h,-i})\ge 0.$$

Further, we have that

$$
\begin{aligned}
&\sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d_{h,\rho}^{\pi}(s_h)\max_{a_{h,i}^{\star}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\pi_{h,-i}(a_{h,-i}\mid s_h)A_{h,i}^{\pi}(s_h,a_{h,i}^{\star},a_{h,-i})\\
=&\sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d_{h,\rho}^{\pi}(s_h)\max_{\overline{\pi}_i\in\Pi_i}\sum_{a_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\overline{\pi}_{h,i}(a_{h,i}\mid s_h)\pi_{h,-i}(a_{h,-i}\mid s_h)A_{h,i}^{\pi}(s_h,a_{h,i},a_{h,-i})\\
\overset{(a)}{=}&\sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d_{h,\rho}^{\pi}(s_h)\max_{\overline{\pi}_i\in\Pi_i}\sum_{a_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\left(\overline{\pi}_{h,i}(a_{h,i}\mid s_h)-\pi_{h,i}(a_{h,i}\mid s_h)\right)\pi_{h,-i}(a_{h,-i}\mid s_h)A_{h,i}^{\pi}(s_h,a_{h,i},a_{h,-i})\\
\overset{(b)}{=}&\sum_{h=1}^{H}\sum_{s_h\in\mathcal{S}} d_{h,\rho}^{\pi}(s_h)\max_{\overline{\pi}_i\in\Pi_i}\sum_{a_{h,i}\in\mathcal{A}_i}\sum_{a_{h,-i}\in\mathcal{A}_{-i}}\left(\overline{\pi}_{h,i}(a_{h,i}\mid s_h)-\pi_{h,i}(a_{h,i}\mid s_h)\right)\pi_{h,-i}(a_{h,-i}\mid s_h)Q_{h,i}^{\pi}(s_h,a_{h,i},a_{h,-i})\\
\overset{(c)}{=}&\max_{\overline{\pi}_i\in\Pi_i}(\overline{\pi}-\pi)^{\intercal}\nabla_{\pi_i}V_{1,i}^{\pi}(\rho)
\end{aligned}
$$

where (a) again uses (29), and (b) relies on the fact that

$$\sum_{a_{h,i}\in\mathcal{A}_i}\left(\overline{\pi}_{h,i}(a_{h,i}\mid s_h)-\pi_{h,i}(a_{h,i}\mid s_h)\right)V_{h,i}^{\pi}(s_h)=0.$$

Equality (c) is due to the policy gradient theorem with direct parameterization (28). Finally, putting everything together, we conclude that

$$V_{1,i}^{\pi^{\star}}(\rho)-V_{1,i}^{\pi}(\rho)\leq\left\|\frac{d_{\rho}^{\pi^{\star}}}{d_{\rho}^{\pi}}\right\|_{\infty}\max_{\overline{\pi}_i\in\Pi_i}(\overline{\pi}-\pi)^{\intercal}\nabla_{\pi_i}V_{1,i}^{\pi}(\rho),$$

where the L^{∞} norm takes the maximum over the space $[H]\times\mathcal{S}$. This completes the proof of the gradient domination property. □

We are now ready to prove Lemma 1 from Section 4, which states that a stationary point of the potential function implies a NE policy.

Lemma 1. Let $\pi=(\pi_1,\dots,\pi_N)$ be an ε-approximate stationary point of the potential function Φ_{ρ} of an MPG for some $\varepsilon>0$. Then, π is a $D\sqrt{SH}\varepsilon$-approximate Nash equilibrium policy profile for this MPG.

Proof. For any $i\in\mathcal{N}$, since $\pi=(\pi_i,\pi_{-i})$ is an ε-approximate stationary point of Φ_{ρ}, we know from Definition 6 that

$$\max_{\pi_i^{\star}\in\Pi_i}(\pi_i^{\star}-\pi_i)^{\intercal}\nabla_{\pi_i}\Phi_{\rho}(\pi)=\sqrt{SH}\max_{\pi_i^{\star}\in\Pi_i}\left(\frac{\pi_i^{\star}-\pi_i}{\sqrt{SH}}\right)^{\intercal}\nabla_{\pi_i}\Phi_{\rho}(\pi)\leq\sqrt{SH}\varepsilon,$$

where we used the fact that $\left\|\frac{\pi_i^{\star}-\pi_i}{\sqrt{SH}}\right\|_2^2\leq 1$. Let $\pi^{\star}=(\pi_i^{\star},\pi_{-i})$. From the definition of the potential function, we obtain that $\nabla_{\pi_i}V_{1,i}^{\pi}(\rho)=\nabla_{\pi_i}\Phi_{\rho}(\pi)$. Further, the linearity of expectation immediately implies that $\Phi_{\rho}(\pi_i,\pi_{-i})-\Phi_{\rho}(\pi_{i'},\pi_{-i})=V_{1,i}^{\pi_i,\pi_{-i}}(\rho)-V_{1,i}^{\pi_{i'},\pi_{-i}}(\rho)$. By the gradient domination property (Lemma 14), we know that

$$
\begin{aligned}
V_{1,i}^{\pi^{\star}}(\rho)-V_{1,i}^{\pi}(\rho)\leq&\left\|\frac{d_{\rho}^{\pi^{\star}}}{d_{\rho}^{\pi}}\right\|_{\infty}\max_{\pi_i^{\star}\in\Pi_i}(\pi^{\star}-\pi)^{\intercal}\nabla_{\pi_i}V_{1,i}^{\pi}(\rho)\\
=&D\max_{\pi_i^{\star}\in\Pi_i}(\pi^{\star}-\pi)^{\intercal}\nabla_{\pi_i}\Phi_{\rho}(\pi)\\
\leq&D\sqrt{SH}\varepsilon.
\end{aligned}
$$

Since the above inequality holds for any $i\in\mathcal{N}$, we conclude that π is a $D\sqrt{SH}\varepsilon$-approximate Nash equilibrium policy profile of the MPG. □

Before proceeding to the proof of Theorem 3, we first state and prove the following supporting lemmas. The first lemma investigates the smoothness of the potential function, while the second one ensures that the projected gradient descent algorithm (3) can be executed in a decentralized way.

Lemma 15. *For any state distribution ρ, the potential function Φ_ρ is $4NA_{\max}H^3$-smooth; that is,*

$$\left\|\nabla\Phi_\rho(\pi)-\nabla\Phi_\rho(\pi')\right\|_2 \le 4NA_{\max}H^3\left\|\pi-\pi'\right\|_2.$$

Proof. It suffices to show that

$$\left\|\nabla^2\Phi_\rho\right\|_2 \le 4NA_{\max}H^3.$$

From Claim C.2 of Leonardos et al. (2021) (restated as Lemma 6 in Appendix B), we know that we only need to show

$$\left\|\nabla_{\pi_j\pi_i}V_{1,j}^{\pi}(\rho)\right\|_2 \le 4A_{\max}H^3, \forall i,j\in\mathcal{N},$$

and the desired result immediately follows.

Our proof follows a similar argument as in Agarwal et al. (2021); Leonardos et al. (2021). For a fixed policy profile π, initial state s_1, and agents $i\neq j\in\mathcal{N}$, let $s,t\ge 0$ be scalars and u,v be unit vectors such that $\pi_i+tu\in\Pi_i$ and $\pi_j+sv\in\Pi_j$. Further, define

$$V(t)=V_{1,i}^{(\pi_i+tu,\pi_{-i})}(s_1),\ \text{and}\ W(t,s)=V_{1,i}^{(\pi_i+tu,\pi_j+sv,\pi_{-i,-j})}(s_1).$$

Then, it suffices to show that

$$\max_{\|u\|_2=1}\left|\frac{d^2V(0)}{dt^2}\right|\le 4A_{\max}H^3,\ \text{and}\ \max_{\|u\|_2=1}\left|\frac{d^2W(0,0)}{dtds}\right|\le 4A_{\max}H^3. \tag{30}$$

We start with the first inequality. From the Bellman equation, we know that

$$V(t)=\sum_{a_{1,i}\in\mathcal{A}_i}\sum_{a_{1,-i}\in\mathcal{A}_{-i}}\left(\pi_{1,i}(a_{1,i}\mid s_1)+tu_1(a_{1,i}\mid s_1)\right)\prod_{j\neq i}\pi_{1,j}(a_{1,j}\mid s_1)Q_{1,i}^{(\pi_i+tu,\pi_{-i})}(s_1,\boldsymbol{a}_1),$$

and in what follows we will write $\pi_{h,-i}(a_{h,-i}\mid s_h)=\prod_{j\neq i}\pi_{h,j}(a_{h,j}\mid s_h)$ for short. Taking the second derivative on both sides,

$$\begin{aligned}\frac{d^2V(t)}{dt^2}=&2\sum_{a_{1,i}\in\mathcal{A}_i}\sum_{a_{1,-i}\in\mathcal{A}_{-i}}u_1(a_{1,i}\mid s_1)\pi_{1,-i}(a_{1,-i}\mid s_1)\frac{dQ_{1,i}^{(\pi_i+tu,\pi_{-i})}(s_1,\boldsymbol{a}_1)}{dt}\\&+\sum_{a_{1,i}\in\mathcal{A}_i}\sum_{a_{1,-i}\in\mathcal{A}_{-i}}\left(\pi_{1,i}(a_{1,i}\mid s_1)+tu_1(a_{1,i}\mid s_1)\right)\pi_{1,-i}(a_{1,-i}\mid s_1)\frac{d^2Q_{1,i}^{(\pi_i+tu,\pi_{-i})}(s_1,\boldsymbol{a}_1)}{dt^2}.\end{aligned}\tag{31}$$

In the following, we will bound each of the two terms on the RHS separately. Let $\pi(t)=(\pi_i+tu,\pi_{-i})$. From the Bellman equation, we know that for any $h\in[H]$,

$$\begin{aligned}Q_{h,i}^{\pi(t)}(s_h,\boldsymbol{a}_h)=&r_{h,i}(s_h,\boldsymbol{a}_h)+\sum_{s_{h+1}}P_h(s_{h+1}\mid s_h,\boldsymbol{a}_h)\sum_{a_{h+1,i}}\sum_{a_{h+1,-i}}\left(\pi_{h+1,i}(a_{h+1,i}\mid s_{h+1})+tu_{h+1}(a_{h+1,i}\mid s_{h+1})\right)\\&\times\pi_{h+1,-i}(a_{h+1,-i}\mid s_{h+1})Q_{h+1,i}^{\pi(t)}(s_{h+1},\boldsymbol{a}_{h+1}).\end{aligned}\tag{32}$$

Differentiating both sides of the equation,

$$\begin{aligned}\left|\frac{dQ_{h,i}^{\pi(t)}(s_h,\boldsymbol{a}_h)}{dt}\right|\le&\left|\sum_{s_{h+1}}P_h(s_{h+1}\mid s_h,\boldsymbol{a}_h)\sum_{a_{h+1,i}}\sum_{a_{h+1,-i}}\left(\pi_{h+1,i}+tu_{h+1}\right)\pi_{h+1,-i}\frac{dQ_{h+1,i}^{\pi(t)}(s_{h+1},\boldsymbol{a}_{h+1})}{dt}\right|\\&+\left|\sum_{s_{h+1}}P_h(s_{h+1}\mid s_h,\boldsymbol{a}_h)\sum_{a_{h+1,i}}\sum_{a_{h+1,-i}}u_{h+1}\pi_{h+1,-i}Q_{h+1,i}^{\pi(t)}(s_{h+1},\boldsymbol{a}_{h+1})\right|\\\le&\left|\sum_{s_{h+1}}P_h(s_{h+1}\mid s_h,\boldsymbol{a}_h)\sum_{a_{h+1,i}}\sum_{a_{h+1,-i}}\left(\pi_{h+1,i}+tu_{h+1}\right)\pi_{h+1,-i}\frac{dQ_{h+1,i}^{\pi(t)}(s_{h+1},\boldsymbol{a}_{h+1})}{dt}\right|+\sqrt{A_{\max}}H,\end{aligned}$$

where we abbreviated $\pi_{h+1,i}(a_{h+1,i} \mid s_{h+1})$ as $\pi_{h+1,i}$, $u_{h+1}(a_{h+1,i} \mid s_{h+1})$ as u_{h+1}, and $\pi_{h+1,-i}(a_{h+1,-i} \mid s_{h+1})$ as $\pi_{h+1,-i}$. The last step holds because $Q^{\pi(t)}_{h+1,i}(s_{h+1}, \boldsymbol{a}_{h+1}) \leq H$, $\sum_{a_{h+1,-i}} \pi_{h+1,-i}(a_{h+1,-i} \mid s_{h+1}) = 1$, $\sum_{a_{h+1,i}} |u_{h+1}(a_{h+1,i} \mid s_{h+1})| \leq \sqrt{A_i} \leq \sqrt{A_{\max}}$, and $\sum_{s_{h+1}} P_h(s_{h+1} \mid s_h, \boldsymbol{a}_h) = 1$. Applying the above inequality recursively over $h = H, H-1, \ldots, 1$, and recalling the facts that

$$\frac{dQ^{\pi(t)}_{H,i}(s_H, \boldsymbol{a}_H)}{dt} = \frac{dr_{H,i}(s_H, \boldsymbol{a}_H)}{dt} = 0$$

and that $\sum_{a_{h+1,i}} (\pi_{h+1,i}(a_{h+1,i} \mid s_{h+1}) + tu_{h+1}(a_{h+1,i} \mid s_{h+1})) = 1$ lead to the result that

$$\left| \frac{dQ^{\pi(t)}_{h,i}(s_h, \boldsymbol{a}_h)}{dt} \right| \leq \sqrt{A_{\max}} H^2, \forall h \in [H]. \tag{33}$$

Further, taking the second derivative on both sides of (32), we get that

$$\begin{aligned}
\left| \frac{d^2Q^{\pi(t)}_{h,i}(s_h, \boldsymbol{a}_h)}{dt^2} \right| \leq & \left| \sum_{s_{h+1}} P_h(s_{h+1} \mid s_h, \boldsymbol{a}_h) \sum_{a_{h+1,i}} \sum_{a_{h+1,-i}} (\pi_{h+1,i} + tu_{h+1})\, \pi_{h+1,-i} \frac{d^2Q^{\pi(t)}_{h+1,i}(s_{h+1}, \boldsymbol{a}_{h+1})}{dt^2} \right| \\
& + 2 \left| \sum_{s_{h+1}} P_h(s_{h+1} \mid s_h, \boldsymbol{a}_h) \sum_{a_{h+1,i}} \sum_{a_{h+1,-i}} u_{h+1} \pi_{h+1,-i} \frac{dQ^{\pi(t)}_{h+1,i}(s_{h+1}, \boldsymbol{a}_{h+1})}{dt} \right| \\
\leq & \left| \sum_{s_{h+1}} P_h(s_{h+1} \mid s_h, \boldsymbol{a}_h) \sum_{a_{h+1,i}} \sum_{a_{h+1,-i}} (\pi_{h+1,i} + tu_{h+1})\, \pi_{h+1,-i} \frac{d^2Q^{\pi(t)}_{h+1,i}(s_{h+1}, \boldsymbol{a}_{h+1})}{dt^2} \right| + 2A_{\max} H^2,
\end{aligned}$$

where in the last step we used (33), and the facts that $\sum_{a_{h+1,-i}} \pi_{h+1,-i}(a_{h+1,-i} \mid s_{h+1}) = 1$, $\sum_{a_{h+1,i}} |u_{h+1}(a_{h+1,i} \mid s_{h+1})| \leq \sqrt{A_i} \leq \sqrt{A_{\max}}$, and $\sum_{s_{h+1}} P_h(s_{h+1} \mid s_h, \boldsymbol{a}_h) = 1$. Again, applying the above inequality recursively over $h = H, H-1, \ldots, 1$, we obtain that

$$\left| \frac{d^2Q^{\pi(t)}_{h,i}(s_h, \boldsymbol{a}_h)}{dt^2} \right| \leq 2A_{\max} H^3, \forall h \in [H]. \tag{34}$$

Substituting (33) and (34) back into (31), we can conclude that

$$\left| \frac{d^2V(t)}{dt^2} \right| \leq 2A_{\max} H^2 + 2A_{\max} H^3 \leq 4A_{\max} H^3.$$

This proves the first inequality in (30). The second inequality in (30) can be shown using a similar procedure. This completes the proof of the lemma. □

Lemma 16. *For any policy profile $\pi = (\pi_1, \ldots, \pi_N)$, let $\pi^+ = Proj_{\Pi}\left(\pi + \eta \nabla_\pi \Phi_\rho(\pi)\right)$ be a PGA update step on the potential function, where $\eta > 0$ is the step size. For each agent $i \in \mathcal{N}$, let $\pi_i^+ = Proj_{\Pi_i}\left(\pi_i + \eta \nabla_{\pi_i} V^{\pi}_{1,i}(\rho)\right)$ be an independent PGA update on its own value function with the same step size. Then, $\pi^+ = (\pi_1^+, \ldots, \pi_N^+)$. That is, running PGA on the potential function as a whole is equivalent to running PGA independently on each agent's value function.*

Proof. By the definition of the projection operator,

$$\begin{aligned}\pi^+ =& \mathrm{Proj}_\Pi\left(\pi + \eta \nabla_\pi \Phi_\rho(\pi)\right) = \operatorname*{argmin}_{x\in\Pi} \|\pi + \eta\nabla_\pi \Phi_\rho(\pi) - x\|_2^2 \\ =& \operatorname*{argmin}_{(x_1,\ldots,x_N)\in\Pi_1\times\cdots\times\Pi_N} \sum_{i=1}^N \|\pi_i + \eta\nabla_{\pi_i}\Phi_\rho(\pi) - x_i\|_2^2 \\ \overset{(a)}{=}& \operatorname*{argmin}_{(x_1,\ldots,x_N)\in\Pi_1\times\cdots\times\Pi_N} \sum_{i=1}^N \left\|\pi_i + \eta\nabla_{\pi_i}V_{1,i}^\pi(\rho) - x_i\right\|_2^2 \\ =& \left(\operatorname*{argmin}_{x_1\in\Pi_1}\left\|\pi_1 + \eta\nabla_{\pi_1}V_{1,1}^\pi(\rho) - x_1\right\|_2^2, \ldots, \operatorname*{argmin}_{x_N\in\Pi_N}\left\|\pi_N + \eta\nabla_{\pi_N}V_{1,N}^\pi(\rho) - x_N\right\|_2^2\right) \\ =& \left(\pi_1^+,\ldots,\pi_N^+\right),\end{aligned}$$

where (a) is due to the fact that $\nabla_{\pi_i}V_{1,i}^\pi(\rho) = \nabla_{\pi_i}\Phi_\rho(\pi)$. □

With Lemma 16 at hand, we only need to analyze the behavior of running PGA on the potential function, as it is equivalent to the case where each agent runs PGA independently on its own value function, i.e., the update rule given in (3). We are now ready to prove Theorem 3.

Theorem 3. For any initial state distribution $\rho \in \Delta(\mathcal{S})$, let the agents independently run the projected gradient ascent algorithm (3) with step size $\eta = \frac{1}{4NA_{\max}H^3}$ for $T = \frac{32NSA_{\max}D^2H^4\Phi_{\max}}{\varepsilon^2}$ iterations. Then, there exists $t \in [T]$, such that $\pi^{(t)}$ is an ε-approximate Nash equilibrium policy profile.

Proof. Let $\pi^{(t)}$ be the policy profile at the beginning of the t-th iteration of the PGA algorithm. Since we have shown in Lemma 15 that Φ_ρ is $4NA_{\max}H^3$-smooth, we can apply a standard sufficient ascent property for smooth functions (Lemma 3 in Appendix B) to conclude that

$$\Phi_\rho(\pi^{(t+1)}) - \Phi_\rho(\pi^{(t)}) \geq \frac{1}{8NA_{\max}H^3}\left\|\frac{1}{\eta}\left(\pi^{(t+1)} - \pi^{(t)}\right)\right\|_2^2.$$

Summing over $t = 1, 2, \ldots, T$, we know that

$$\sum_{t=1}^T \left\|\frac{1}{\eta}\left(\pi^{(t+1)} - \pi^{(t)}\right)\right\|_2^2 \leq 8NA_{\max}H^3\left(\Phi_\rho(\pi^{(T+1)}) - \Phi_\rho(\pi^{(1)})\right) \leq 8NA_{\max}H^3\Phi_{\max}.$$

When T is large enough such that $T \geq \frac{32NSA_{\max}D^2H^4\Phi_{\max}}{\varepsilon^2}$, we know that there exists a time step $t \in [T]$ that satisfies $\left\|\frac{1}{\eta}\left(\pi^{(t+1)} - \pi^{(t)}\right)\right\|_2 \leq \frac{\varepsilon}{2D\sqrt{SH}}$. Then, from a standard gradient mapping property (Lemma 4 in Appendix B), we know that $\pi^{(t+1)}$ is a $\frac{\varepsilon}{D\sqrt{SH}}$-approximate stationary point of the potential function. Finally, invoking the result that an ε-approximate stationary point implies an $D\sqrt{SH}\varepsilon$-approximate Nash equilibrium policy profile, we can conclude that $\pi^{(t+1)}$ is an ε-approximate NE policy profile. □

F. Proofs for Section 4.2

F.1. Proof of Theorem 4

Lemma 2. For any agent $i \in \mathcal{N}$ and any iteration $t \in [K]$, the REINFORCE gradient estimator (4) with $\tilde{\varepsilon}$-greedy exploration is an unbiased estimator with a bounded variance:

$$\mathbb{E}_{\pi^{(t)}}\left[\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)})\right] = \nabla_{\pi_i}V_{1,i}^{\pi^{(t)}}(\rho), \text{ and } \mathbb{E}_{\pi^{(t)}}\left[\left\|\hat{\nabla}_{\pi_i}^{(t)}(\pi^{(t)}) - \nabla_{\pi_i}V_{1,i}^{\pi^{(t)}}(\rho)\right\|_2^2\right] \leq \frac{A_{\max}^2H^4}{\tilde{\varepsilon}}.$$

Further, it is mean-squared smooth, such that for any policy $\pi'^{(t)} \in \Pi_i$,

$$\mathbb{E}_{\pi^{(t)}}\left[\left\|\hat{\nabla}^{(t)}_{\pi_i}(\pi^{(t)}) - \hat{\nabla}^{(t)}_{\pi_i'}(\pi'^{(t)})\right\|_2^2\right] \leq \frac{A_{\max}^3 H^3}{\tilde{\varepsilon}^3}\left\|\pi^{(t)} - \pi'^{(t)}\right\|_2^2.$$

Proof. In this proof, we omit the iteration index t in the superscripts of the notations as there is no ambiguity. We first show that the gradient estimator is unbiased. For any state-action trajectory $\tau = (s_1, \boldsymbol{a}_1, \ldots, s_H, \boldsymbol{a}_H)$, let $\mathbb{P}^\pi(\tau)$ denote the probability of observing the trajectory τ by following the policy π starting from the state distribution ρ, and let $R(\tau) = \sum_{h=1}^H r_h(s_h, \boldsymbol{a}_h)$ denote the total reward of the trajectory. Then, we know that

$$\mathbb{P}^\pi(\tau) = \prod_{h=1}^H \pi_{h,i}(a_{h,i} \mid s_h)\pi_{h,-i}(a_{h,-i} \mid s_h)P(s_{h+1} \mid s_h, \boldsymbol{a}_h).$$

Therefore, by the definition of the value function,

$$\begin{aligned}
\nabla_{\pi_i} V^\pi_{1,i}(\rho) =& \nabla_{\pi_i}\sum_\tau R(\tau)\mathbb{P}^\pi(\tau) = \sum_\tau R(\tau)\nabla_{\pi_i}\mathbb{P}^\pi(\tau)\\
=& \sum_\tau R(\tau)\mathbb{P}^\pi(\tau)\nabla_{\pi_i}\log\mathbb{P}^\pi(\tau)\\
=& \sum_\tau R(\tau)\mathbb{P}^\pi(\tau)\nabla_{\pi_i}\left(\sum_{h=1}^H \log\pi_{h,i}(a_{h,i} \mid s_h)\right)\\
=& \mathbb{E}_\pi\left[\left(\sum_{h=1}^H r_h(s_h, \boldsymbol{a}_h)\right)\sum_{h=1}^H \nabla_{\pi_i}\log\pi_{h,i}(a_{h,i} \mid s_h)\right]\\
=& \mathbb{E}_\pi\left[\hat{\nabla}_{\pi_i}(\pi)\right]
\end{aligned}$$

Next, we proceed to bound the variance of the gradient estimator. Since the gradient estimator is unbiased,

$$\begin{aligned}
\mathbb{E}_\pi\left[\left\|\hat{\nabla}_{\pi_i}(\pi) - \nabla_{\pi_i}V^\pi_{1,i}(\rho)\right\|_2^2\right] \leq& \mathbb{E}_\pi\left[\left\|\hat{\nabla}_{\pi_i}(\pi)\right\|_2^2\right] = \mathbb{E}_\pi\left[\left\|R_i\sum_{h=1}^H \nabla_{\pi_i}\log\pi_{h,i}(a_{h,i} \mid s_h)\right\|_2^2\right]\\
\leq& H^2\mathbb{E}_\pi\left[\left\|\sum_{h=1}^H \nabla_{\pi_i}\log\pi_{h,i}(a_{h,i} \mid s_h)\right\|_2^2\right]\\
\leq& H^3\mathbb{E}_\pi\left[\sum_{h=1}^H \left\|\nabla_{\pi_i}\log\pi_{h,i}(a_{h,i} \mid s_h)\right\|_2^2\right]\\
=& H^3\mathbb{E}_\pi\left[\sum_{h=1}^H\sum_{s,a_i}(1-\tilde{\varepsilon})^2\mathbb{I}\{s = s_h, a_i = a_{h,i}\}\frac{1}{\pi^2_{h,i}(a_i \mid s)}\right],
\end{aligned}$$

where the last step is a consequence of direct parameterization with $\tilde{\varepsilon}$-greedy exploration. We further upper bound the above term as

$$\begin{aligned}
\mathbb{E}_\pi\left[\left\|\hat{\nabla}_{\pi_i}(\pi)-\nabla_{\pi_i}V_{1,i}^{\pi}(\rho)\right\|_2^2\right] \leq& H^3\mathbb{E}_\pi\left[\sum_{h=1}^{H}\sum_{s,a_i}\mathbb{I}\{s=s_h,a_i=a_{h,i}\}\frac{1}{\pi_{h,i}^2(a_i\mid s)}\right]\\
=& H^3\mathbb{E}_\pi\left[\sum_{h=1}^{H}\sum_{s,a_i}\mathbb{I}\{s=s_h\}\frac{1}{\pi_{h,i}(a_i\mid s)}\right]\\
\leq& \frac{A_{\max}H^3}{\tilde{\varepsilon}}\mathbb{E}_\pi\left[\sum_{h=1}^{H}\sum_{s,a_i}\mathbb{I}\{s=s_h\}\right]\\
\leq& \frac{A_{\max}^2H^4}{\tilde{\varepsilon}}.
\end{aligned}$$

Finally, we proceed to show the mean-squared smoothness of the gradient estimator. Using a similar argument as above,

$$\begin{aligned}
\mathbb{E}_\pi\left[\left\|\hat{\nabla}_{\pi_i}(\pi)-\hat{\nabla}_{\pi_i'}(\pi')\right\|_2^2\right] \leq& H^2\mathbb{E}_\pi\left[\left\|\sum_{h=1}^{H}\left(\nabla_{\pi_i}\log\pi_{h,i}(a_{h,i}\mid s_h)-\nabla_{\pi_i'}\log\pi_{h,i}'(a_{h,i}\mid s_h)\right)\right\|_2^2\right]\\
\leq& H^3\mathbb{E}_\pi\left[\sum_{h=1}^{H}\left\|\nabla_{\pi_i}\log\pi_{h,i}(a_{h,i}\mid s_h)-\nabla_{\pi_i'}\log\pi_{h,i}'(a_{h,i}\mid s_h)\right\|_2^2\right]\\
\leq& H^3\mathbb{E}_\pi\left[\sum_{h=1}^{H}\sum_{s,a_i}\mathbb{I}\{s=s_h,a_i=a_{h,i}\}\left(\frac{1}{\pi_{h,i}(a_i\mid s)}-\frac{1}{\pi_{h,i}'(a_i\mid s)}\right)^2\right]\\
\leq& \frac{H^3A_{\max}^3}{\tilde{\varepsilon}^3}\mathbb{E}_\pi\left[\sum_{h=1}^{H}\sum_{s,a_i}\mathbb{I}\{s=s_h\}\left(\pi_{h,i}(a_i\mid s)-\pi_{h,i}'(a_i\mid s)\right)^2\right]\\
\leq& \frac{H^3A_{\max}^3}{\tilde{\varepsilon}^3}.
\end{aligned}$$

This completes the proof of the lemma. □

Theorem 4. For any initial policies, let the agents independently run SGA policy updates (Algorithm 3) for T iterations with $T=O(1/\varepsilon^{4.5})\cdot\text{poly}(N,D,S,A_{\max},H)$. Then, there exists $t\in[T]$, such that $\pi^{(t)}$ is an ε-approximate Nash equilibrium policy profile in expectation.

Proof. It suffices to verify that Assumption 1 is satisfied in our SGA policy updates, and then apply the convergence results from Proposition 1. From Lemma 2, we know that the REINFORCE gradient estimator with $\tilde{\varepsilon}$-greedy exploration is unbiased, mean-squared smooth, and has a bounded variance. We also know from Lemma 15 that the potential function is smooth (The smoothness parameter does not increase with epsilon-greedy exploration, e.g., Daskalakis et al. (2020, Proposition 3)). Hence, we can conclude that all the conditions in Assumption 1 are satisfied with the following choice of parameters:

$$\sigma^2=\frac{NA_{\max}^2H^4}{\tilde{\varepsilon}},\text{ and }L^2=\frac{N^2A_{\max}^3H^3}{\tilde{\varepsilon}^3}.$$

Applying the convergence result from Theorem 1, and setting $\tilde{\varepsilon}=\frac{\sqrt{N}\varepsilon}{2NDSH^3A_{\max}}$, we conclude that we can obtain an ε-approximate Nash equilibrium when $T=\widetilde{O}\left(\frac{N^{9/4}D^{9/2}S^3A_{\max}^{9/2}H^{12}}{\varepsilon^{9/2}}\right)$. This completes the proof of the theorem. □

F.2. Decentralized MARL in Smooth MPGs

Smooth games were first introduced in Roughgarden (2009) to study the Price of Anarchy (POA) in normal-form games. A large class of games are covered as examples of smooth games, including congestion games and many forms of auctions (Roughgarden, 2009; Syrgkanis & Tardos, 2013). The notion of smoothness was later extended to learning in normal-form games (Syrgkanis et al., 2015; Foster et al., 2016) and cooperative Markov games (Radanovic et al., 2019; Mao et al., 2020b). This concept enables *decentralized* no-regret learning dynamics to converge to near-optimality.

Theorem 5. In a (λ, ω)-smooth MPG, for any initial policies and any $\varepsilon > 0$, let the agents independently run SGA policy updates (Algorithm 3) for T iterations with $T = O(1/\varepsilon^{4.5}) \cdot \text{poly}(N, D, S, A_{\max}, H)$. Then, there exists $t \in [T]$, such that

$$\mathbb{E}\left[V_{1,i}^{\pi^{(t)}}(\rho)\right] \geq \frac{\lambda}{1+\omega} V_{1,i}^{\star}(\rho) - \frac{\varepsilon}{1+\omega}, \forall i \in \mathcal{N}.$$

Proof. Since $T = O(1/\varepsilon^{4.5}) \cdot \text{poly}(N, D, S, A_{\max}, H)$, Theorem 4 guarantees that there exists $t \in [T]$, such that $\pi^{(t)}$ is an ε-approximate NE in expectation. That is,

$$\mathbb{E}\left[V_{1,i}^{\pi^{(t)}}(\rho)\right] \geq \mathbb{E}\left[V_{1,i}^{\pi_i^{\star}, \pi_{-i}^{(t)}}(\rho)\right] - \varepsilon \geq \lambda \cdot V_{1,i}^{\star}(\rho) - \omega \cdot \mathbb{E}\left[V_{1,i}^{\pi^{(t)}}(\rho)\right] - \varepsilon,$$

where the second step is by the definition of smoothness. Rearranging the terms leads to the desired result. □

G. SGD with Variance Reduction

Before we present the convergence guarantee of Algorithm 3, we first introduce a few notations for ease of presentations. For any $t \in [T]$, we break the update rule into two steps:

$$\tilde{x}_{t+1} \overset{\text{def}}{=} x_t - \eta_t d_t, \text{ and } x_{t+1} = \text{Proj}_{\mathcal{X}}(\tilde{x}_{t+1}).$$

In addition, for each $t \in [T]$, we define $x_{t+1}^{+} \overset{\text{def}}{=} \text{Proj}_{\mathcal{X}}(x_t - \eta_t \nabla F(x_t))$ to be the next iterate updated using the full gradient $\nabla F(x_t)$, a value we do not have access to. Define $\varepsilon_t \overset{\text{def}}{=} d_t - \nabla F(x_t)$ to be the error in d_t. The high-level procedure of our proof is to seek to upper bound the value $\mathbb{E}\left[\sum_{t=1}^{T} \left\| \frac{1}{\eta_t}\left(x_{t+1}^{+} - x_t\right)\right\|^2\right]$, and then to invoke the gradient mapping property in Lemma 4 to conclude with a stationary point. This is in contrast with the unconstrained case, where Cutkosky & Orabona (2019) directly derive an upper bound of $\mathbb{E}\left[\sum_{t=1}^{T} \|\nabla F(x_t)\|^2\right]$. In the following, we start with a few technical lemmas.

Lemma 17. *Suppose $\eta_t \leq \frac{1}{4L}$ for all $t \in [T]$. Then,*

$$\mathbb{E}[F(x_{t+1}) - F(x_t)] \leq \mathbb{E}\left[-\frac{3}{16\eta_t}\left\|x_{t+1}^{+} - x_t\right\|^2 + \frac{7\eta_t}{8}\|\varepsilon_t\|^2\right].$$

Proof. From the first-order optimality condition, we know that

$$\langle x - x_{t+1}, x_{t+1} - (x_t - \eta_t d_t)\rangle \geq 0,$$

for any $x \in \mathcal{X}$. Taking $x = x_t$ leads to

$$\langle x_t - x_{t+1}, x_{t+1} - x_t\rangle + \langle x_t - x_{t+1}, \eta_t d_t\rangle \geq 0,$$

which in turn implies that

$$\langle x_{t+1} - x_t, d_t\rangle \leq -\frac{1}{\eta_t}\|x_t - x_{t+1}\|^2. \tag{35}$$

It follows that

$$\begin{aligned}\langle \nabla F(x_t), x_{t+1} - x_t\rangle =& \langle d_t - \varepsilon_t, x_{t+1} - x_t\rangle \\ \leq& -\frac{1}{\eta_t}\left\|x_t - x_{t+1}\right\|^2 - \langle \varepsilon_t, x_{t+1} - x_t\rangle \\ \leq& -\frac{1}{\eta_t}\left\|x_t - x_{t+1}\right\|^2 + \frac{\eta_t}{2}\left\|\varepsilon_t\right\|^2 + \frac{1}{2\eta_t}\left\|x_{t+1} - x_t\right\|^2 \\ =& -\frac{1}{2\eta_t}\left\|x_t - x_{t+1}\right\|^2 + \frac{\eta_t}{2}\left\|\varepsilon_t\right\|^2,\end{aligned}$$

where the first inequality uses (35), and the second inequality is due to Hölder's inequality and Young's inequality. From the smoothness of F,

$$\begin{aligned}\mathbb{E}[F(x_{t+1})] \leq& \mathbb{E}\left[F(x_t) + \langle \nabla F(x_t), x_{t+1} - x_t\rangle + \frac{L}{2}\left\|x_{t+1} - x_t\right\|^2\right] \\ \leq& \mathbb{E}\left[F(x_t) - \frac{1}{2\eta_t}\left\|x_t - x_{t+1}\right\|^2 + \frac{\eta_t}{2}\left\|\varepsilon_t\right\|^2 + \frac{L}{2}\left\|x_{t+1} - x_t\right\|^2\right] \\ \leq& \mathbb{E}\left[F(x_t) - \frac{3}{8\eta_t}\left\|x_t - x_{t+1}\right\|^2 + \frac{\eta_t}{2}\left\|\varepsilon_t\right\|^2\right],\end{aligned} \tag{36}$$

where the last step uses $\eta_t \leq \frac{1}{4L}$. From the fact that $\left\|x+y\right\|^2 \leq 2\left\|x\right\|^2 + 2\left\|y\right\|^2$, we know

$$\left\|x_{t+1}^+ - x_t\right\|^2 = \left\|x_{t+1}^+ - x_{t+1} + x_{t+1} - x_t\right\|^2 \leq 2\left\|x_{t+1}^+ - x_{t+1}\right\|^2 + 2\left\|x_{t+1} - x_t\right\|^2.$$

Rearranging the terms leads to

$$\begin{aligned}-\left\|x_t - x_{t+1}\right\|^2 \leq& \left\|x_{t+1}^+ - x_{t+1}\right\|^2 - \frac{1}{2}\left\|x_{t+1}^+ - x_t\right\|^2 \\ \leq& \left\|\text{Proj}_{\mathcal{X}}(x_t - \eta_t \nabla F(x_t)) - \text{Proj}_{\mathcal{X}}(x_t - \eta_t d_t)\right\|^2 - \frac{1}{2}\left\|x_{t+1}^+ - x_t\right\|^2 \\ \leq& \left\|(x_t - \eta_t \nabla F(x_t)) - (x_t - \eta_t d_t)\right\|^2 - \frac{1}{2}\left\|x_{t+1}^+ - x_t\right\|^2 \\ =& \eta_t^2\left\|\varepsilon_t\right\|^2 - \frac{1}{2}\left\|x_{t+1}^+ - x_t\right\|^2.\end{aligned} \tag{37}$$

The second inequality uses the definition of x_{t+1}^+. The third step holds because the projection operator is non-expansive, i.e. $\|\text{Proj}_{\mathcal{X}}(x) - \text{Proj}_{\mathcal{X}}(y)\| \leq \|x - y\|$. Substituting (37) back to (36) leads to

$$\mathbb{E}[F(x_{t+1})] \leq \mathbb{E}\left[F(x_t) - \frac{3}{16\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2 + \frac{7\eta_t}{8}\left\|\varepsilon_t\right\|^2\right].$$

Rearranging the terms completes the proof. □

Lemma 18. *(Lemma 3 in Cutkosky & Orabona (2019)). For any $t \in [T]$, it holds that*

$$\begin{aligned}&\mathbb{E}\left[\frac{(1-a_t)^2}{\eta_{t-1}}(\nabla f(x_t,\xi_t) - \nabla F(x_t)) \cdot \varepsilon_{t-1}\right] = 0, \\ &\mathbb{E}\left[\frac{(1-a_t)^2}{\eta_{t-1}}(\nabla f(x_t,\xi_t) - \nabla f(x_{t-1},\xi_t) - \nabla F(x_t) + \nabla F(x_{t-1})) \cdot \varepsilon_{t-1}\right] = 0.\end{aligned}$$

Lemma 19. *(Adapted from Lemma 5 in Cutkosky & Orabona (2019)). With the notations in Algorithm 3, we have*

$$\mathbb{E}\left[\frac{\left\|\varepsilon_t\right\|^2}{\eta_{t-1}}\right] \leq \mathbb{E}\left[2c^2\eta_{t-1}^3\sigma^2 + \frac{(4L^2\eta_{t-1}^2 + 1)(1-a_t)^2}{\eta_{t-1}}\left\|\varepsilon_{t-1}\right\|^2 + \frac{4\left(1-a_t\right)^2 L^2}{\eta_{t-1}}\left\|x_t^+ - x_{t-1}\right\|^2\right].$$

Proof. First, observe that

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)-\nabla F(x_t)+\nabla F(x_{t-1})\right\|^2\right]\\
\leq&\mathbb{E}\left[\left\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)\right\|^2+\left\|\nabla F(x_t)-\nabla F(x_{t-1})\right\|^2\right]\\
&-2\mathbb{E}\left[\langle\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t),\nabla F(x_t)-\nabla F(x_{t-1})\rangle\right]\\
\leq&\mathbb{E}\left[\left\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)\right\|^2+\left\|\nabla F(x_t)-\nabla F(x_{t-1})\right\|^2\right]\\
&-2\mathbb{E}\left[\mathbb{E}\left[\langle\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t),\nabla F(x_t)-\nabla F(x_{t-1})\rangle\mid\xi_1,\dots,\xi_{t-1}\right]\right]\\
=&\mathbb{E}\left[\left\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)\right\|^2-\left\|\nabla F(x_t)-\nabla F(x_{t-1})\right\|^2\right]\\
\leq&\mathbb{E}\left[\left\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)\right\|^2\right].
\end{aligned}\tag{38}$$

By the definition of ε_t, we have $\varepsilon_t=d_t-\nabla F(x_t)=\nabla f(x_t,\xi_t)+(1-a_t)(d_{t-1}-\nabla f(x_{t-1},\xi_t))-\nabla F(x_t)$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\frac{\|\varepsilon_t\|^2}{\eta_{t-1}}\right]=&\mathbb{E}\left[\frac{1}{\eta_{t-1}}\left\|\nabla f(x_t,\xi_t)+(1-a_t)(d_{t-1}-\nabla f(x_{t-1},\xi_t))-\nabla F(x_t)\right\|^2\right]\\
=&\mathbb{E}\left[\frac{1}{\eta_{t-1}}\| a_t(\nabla f(x_t,\xi_t)-\nabla F(x_t))+(1-a_t)(d_{t-1}-\nabla F(x_{t-1}))\right.\\
&\left.+(1-a_t)(\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)-\nabla F(x_t)+\nabla F(x_{t-1}))\|^2\right]\\
\leq&\mathbb{E}\left[2c^2\eta_{t-1}^3\left\|\nabla f\left(x_t,\xi_t\right)-\nabla F\left(x_t\right)\right\|^2+\frac{1}{\eta_{t-1}}(1-a_t)^2\left\|\varepsilon_{t-1}\right\|^2\right.\\
&\left.+\frac{2}{\eta_{t-1}}\left(1-a_t\right)^2\left\|\nabla f\left(x_t,\xi_t\right)-\nabla f\left(x_{t-1},\xi_t\right)-\nabla F\left(x_t\right)+\nabla F\left(x_{t-1}\right)\right\|^2\right],
\end{aligned}$$

where in the last step we used Lemma 18 and the simple fact that $\|x+y\|^2\leq 2\|x\|^2+2\|y\|^2$. Further applying (38) and the assumption that $\mathbb{E}\left[\|\nabla f(x_t,\xi_t)-\nabla F(x_t)\|^2\right]\leq\sigma^2$ leads to

$$\begin{aligned}
\mathbb{E}\left[\frac{\|\varepsilon_t\|^2}{\eta_{t-1}}\right]=&\mathbb{E}\left[2c^2\eta_{t-1}^3\sigma^2+\frac{(1-a_t)^2}{\eta_{t-1}}\|\varepsilon_{t-1}\|^2+\frac{2\left(1-a_t\right)^2}{\eta_{t-1}}\|\nabla f(x_t,\xi_t)-\nabla f(x_{t-1},\xi_t)\|^2\right]\\
\leq&\mathbb{E}\left[2c^2\eta_{t-1}^3\sigma^2+\frac{(1-a_t)^2}{\eta_{t-1}}\|\varepsilon_{t-1}\|^2+\frac{2\left(1-a_t\right)^2L^2}{\eta_{t-1}}\|x_t-x_{t-1}\|^2\right]\\
\leq&\mathbb{E}\left[2c^2\eta_{t-1}^3\sigma^2+\frac{(1-a_t)^2}{\eta_{t-1}}\|\varepsilon_{t-1}\|^2+\frac{4\left(1-a_t\right)^2L^2}{\eta_{t-1}}\left(\left\|x_t-x_t^+\right\|^2+\left\|x_t^+-x_{t-1}\right\|^2\right)\right]\\
\leq&\mathbb{E}\left[2c^2\eta_{t-1}^3\sigma^2+\frac{(4L^2\eta_{t-1}^2+1)(1-a_t)^2}{\eta_{t-1}}\|\varepsilon_{t-1}\|^2+\frac{4\left(1-a_t\right)^2L^2}{\eta_{t-1}}\left\|x_t^+-x_{t-1}\right\|^2\right].
\end{aligned}$$

The first inequality is due to the L-smoothness of the function f. The second inequality again uses the fact that $\|x+y\|^2\leq 2\|x\|^2+2\|y\|^2$. The last step holds because of the non-expansiveness of the projection operator, that is,

$$\left\|x_t-x_t^+\right\|=\left\|\mathrm{Proj}_{\mathcal{X}}(x_{t-1}-\eta_{t-1}d_{t-1})-\mathrm{Proj}_{\mathcal{X}}(x_{t-1}-\eta_{t-1}\nabla F(x_{t-1}))\right\|\leq\eta_{t-1}^2\left\|\varepsilon_{t-1}\right\|^2.$$

This completes the proof of the lemma. □

Now, we are ready to present the convergence guarantee of Algorithm 3.

Proposition 1. (Adapted from Theorem 2 in Cutkosky & Orabona (2019)). Suppose the conditions in Assumption 1 are satisfied. For any $b > 0$, let $k = \frac{b\sigma^{\frac{2}{3}}}{L}, c = 32L^2 + \sigma^2/(7Lk^3) = L^2\left(32 + 1/(7b^3)\right), w = \max\left((4Lk)^3, 2\sigma^2, \left(\frac{ck}{4L}\right)^3\right) = \sigma^2 \max\left((4b)^3, 2, \left(32b + \frac{1}{7b^2}\right)^3/64\right)$, and $M = 16(F(x_1) - F^\star) + \frac{w^{1/3}\sigma^2}{2L^2k} + \frac{k^3c^2}{L^2}\ln(T+2)$. Then, the following convergence guarantee holds for Algorithm 3:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|\frac{1}{\eta_t}(x_{t+1}^+ - x_t)\right\|^2\right] \le \frac{Mw^{1/3}}{Tk} + \frac{M\sigma^{2/3}}{T^{2/3}k}.$$

Proof. First, define the Lyapunov function $\Phi_t = F(x_t) + \frac{1}{32L^2\eta_{t-1}}\left\|\varepsilon_t\right\|^2$. From Lemma 19, we can derive that

$$\begin{aligned}
&\mathbb{E}\left[\frac{\left\|\varepsilon_{t+1}\right\|^2}{\eta_t} - \frac{\left\|\varepsilon_t\right\|^2}{\eta_{t-1}}\right]\\
\le&\mathbb{E}\left[2c^2\eta_t^3\sigma^2 + \frac{(4L^2\eta_t^2+1)(1-a_{t+1})^2}{\eta_t}\left\|\varepsilon_t\right\|^2 + \frac{4\left(1-a_{t+1}\right)^2L^2}{\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2 - \frac{\left\|\varepsilon_t\right\|^2}{\eta_{t-1}}\right]\\
\le&\mathbb{E}\left[\underbrace{2c^2\eta_t^3\sigma^2}_{A_t} + \underbrace{\left(\frac{(4L^2\eta_t^2+1)(1-a_{t+1})^2}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\left\|\varepsilon_t\right\|^2}_{B_t} + \underbrace{\frac{4\left(1-a_{t+1}\right)^2L^2}{\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2}_{C_t}\right].
\end{aligned}$$

The first two terms A_t and B_t are exactly the same as in the proof of Theorem 2 in Cutkosky & Orabona (2019), and we refer to their results as follows:

$$\sum_{t=1}^{T}A_t \le 2k^3c^2\ln(T+2), \text{ and } \sum_{t=1}^{T}B_t \le -28L^2\sum_{t=1}^{T}\eta_t\left\|\varepsilon_t\right\|^2.$$

From $w \ge (4Lk)^3$, we know that $\eta_t \le \frac{1}{4L}$. Further, since $a_{t+1} = c\eta_t^2$, we have that $a_{t+1} \le \frac{ck}{4Lw^{1/3}} \le 1$ for all t, and hence $C_t \le \frac{4L^2}{\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2$. Putting it all together, we obtain

$$\frac{1}{32L^2}\sum_{t=1}^{T}\left(\frac{\left\|\varepsilon_{t+1}\right\|^2}{\eta_t} - \frac{\left\|\varepsilon_t\right\|^2}{\eta_{t-1}}\right) \le \frac{k^3c^2}{16L^2}\ln(T+2) + \sum_{t=1}^{T}\left(\frac{1}{8\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2 - \frac{7\eta_t}{8}\left\|\varepsilon_t\right\|^2\right). \tag{39}$$

From Lemma 17, we know that

$$\mathbb{E}\left[\Phi_{t+1} - \Phi_t\right] \le \mathbb{E}\left[-\frac{3}{16\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2 + \frac{7\eta_t}{8}\left\|\varepsilon_t\right\|^2 + \frac{1}{32L^2\eta_t}\left\|\varepsilon_{t+1}\right\|^2 - \frac{1}{32L^2\eta_{t-1}}\left\|\varepsilon_t\right\|^2\right].$$

Summing over t from 1 to T and then applying (39), we obtain

$$\begin{aligned}
\mathbb{E}\left[\Phi_{T+1} - \Phi_1\right] \le& \sum_{t=1}^{T}\mathbb{E}\left[-\frac{3}{16\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2 + \frac{7\eta_t}{8}\left\|\varepsilon_t\right\|^2 + \frac{1}{32L^2\eta_t}\left\|\varepsilon_{t+1}\right\|^2 - \frac{1}{32L^2\eta_{t-1}}\left\|\varepsilon_t\right\|^2\right]\\
\le&\mathbb{E}\left[\frac{k^3c^2}{16L^2}\ln(T+2) - \sum_{t=1}^{T}\frac{1}{16\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2\right].
\end{aligned}$$

Rearranging the terms leads to

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{\eta_t}\left\|x_{t+1}^+ - x_t\right\|^2\right] \le& \mathbb{E}\left[16(\Phi_1 - \Phi_{T+1}) + \frac{k^3c^2}{L^2}\ln(T+2)\right]\\
\le&16(F(x_1) - F^\star) + \frac{1}{2L^2\eta_0}\mathbb{E}[\left\|\varepsilon_1\right\|^2] + \frac{k^3c^2}{L^2}\ln(T+2)\\
\le&16(F(x_1) - F^\star) + \frac{w^{1/3}\sigma^2}{2L^2k} + \frac{k^3c^2}{L^2}\ln(T+2),
\end{aligned}$$

where the last step holds due to the definition that $\eta_0 = \frac{k}{w^{1/3}}$. Since η_t is decreasing in t,

$$\mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{\eta_t}\left\|x_{t+1}^{+} - x_t\right\|^2\right] = \mathbb{E}\left[\sum_{t=1}^{T} \eta_t \left\|\frac{1}{\eta_t}(x_{t+1}^{+} - x_t)\right\|^2\right] \geq \eta_T \mathbb{E}\left[\sum_{t=1}^{T} \left\|\frac{1}{\eta_t}(x_{t+1}^{+} - x_t)\right\|^2\right].$$

Dividing both sides by $T\eta_T$ and recalling the definition $M = 16(F(x_1) - F^\star) + \frac{w^{1/3}\sigma^2}{2L^2k} + \frac{k^3c^2}{L^2}\ln(T+2)$, we obtain

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \left\|\frac{1}{\eta_t}(x_{t+1}^{+} - x_t)\right\|^2\right] \leq \frac{M}{T\eta_T} = \frac{M(w+\sigma^2 T)^3}{Tk} \leq \frac{Mw^{1/3}}{Tk} + \frac{M\sigma^{2/3}}{T^{2/3}k},$$

where in the last step we used the fact that $(a+b)^{1/3} \leq a^{1/3} + b^{1/3}$. □

H. Simulations

In this section, we demonstrate the empirical performances of our algorithms, and compare their performances with various benchmarks. We evaluate Algorithm 3 (SGA) on a classic matrix team task (Claus & Boutilier, 1998), and both Algorithms 1 and 3 on two Markov games, namely GoodState and BoxPushing (Seuken & Zilberstein, 2007).

H.1. Markov Teams

We use a classic matrix team example from the literature (Claus & Boutilier, 1998; Lauer & Riedmiller, 2000), where a team problem is a special case of potential games. Its reward table is reproduced in Table 1, where agent 1 is the row player, and agent 2 is the column player, both being maximizers. The action spaces of the agents are $\mathcal{A}_1 = \{a_0, a_1, a_2\}$ and $\mathcal{B}_2 = \{b_0, b_1, b_2\}$. There are three deterministic Nash equilibria in this team, among which two of them, (a_0, b_0) and (a_2, b_2), are team-optimal. It would be preferred that the agents not only learn a NE, but also settle on the same NE out of the two team-optimal ones.

	b_0	b_1	b_2
a_0	10	0	-10
a_1	0	2	0
a_2	-10	0	10

Table 1. Reward table for the matrix team.

s_0	b_0	b_1
a_0	-2	5
a_1	2	-2

s_1	b_0	b_1
a_0	0	0
a_1	0	0

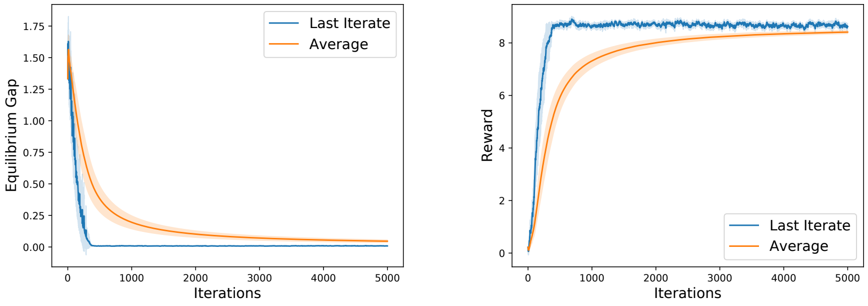
Table 2. Reward tables for GoodState.

We run Algorithm 3 on this task for $T = 5000$ rounds, and we set the step size $\eta_t = 10^{-4}$ and the momentum parameter $a_t = 0.5$. We evaluate our algorithm in terms of both the rewards it obtained and its L^2 equilibrium gap. Specifically, we define the L^2 equilibrium gap as the L^2 distance to a equilibrium point. For a pair of strategies $(\mu, \nu) \in \Delta(\mathcal{A}_1) \times \Delta(\mathcal{A}_2)$, its L^2 equilibrium gap is defined as:

$$\mathrm{Gap}(\mu, \nu) \overset{\text{def}}{=} \left\|\mu - \mu^\dagger(\nu)\right\|_2^2 + \left\|\nu - \nu^\dagger(\mu)\right\|_2^2, \tag{40}$$

where $\nu^\dagger(\mu)$ (resp. $\mu^\dagger(\nu)$) is the best response with respect to μ (resp. ν), and $\|\cdot\|_2$ is the L^2 norm. The simulation results are presented in Figure 2. All results are averaged over 20 runs. Notice that we evaluate two sets of strategy trajectories: The "Last Iterate" strategy (μ_t, ν_t) is the strategy pair used by Algorithm 3 at round t, while the "Average" strategy is to uniformly draw a random time index τ from $\{1, \ldots, t\}$ and run the strategy pair (μ_τ, ν_τ). Notice that in Theorem 4, our theoretical guarantees only hold in expectation, which correspond to the "Average" strategies.

From Figure 2(a), we can see that the equilibrium gap of both "Last Iterate" and "Average" converge to zero, indicating that they indeed find an equilibrium as the number of iterations increase. The convergence of "Average" slightly lags behind "Last Iterate" because "Average" essentially takes the time-averaged value of the actual trajectories, which requires some time to reflect the convergence behavior. A more promising result is that from Figure 2(b), we can see that the rewards collected by "Last Iterate" and "Average" converge to values close to 9. This suggests that Algorithm 3 not only finds a NE in this specific task, but actually converges to a team-optimal equilibrium most



(a) Equilibrium gap

(b) Reward

Figure 2. (a) L^2 equilibrium gaps and (b) rewards of Algorithm 3 on the matrix team task given in Table 1. "Last Iterate" denotes the actual strategy trajectory, while "Average" represents the uniformly sampled strategy pair. Shaded areas denote the standard deviations of the equilibrium gap or reward. All results are averaged over 20 runs.

of the time. It does not exactly reach the team-optimal value of 10 because it still converges to non-team-optimal NE at a rather low frequency.

H.2. Markov Games

We further evaluate both Algorithm 1 and Algorithm 3 on two Markov games, namely GoodState and BoxPushing (Seuken & Zilberstein, 2007). The GoodState task is a simple Markov team problem inspired by Yongacoglu et al. (2019). It has two states $\mathcal{S} = \{s_0, s_1\}$, where s_0 is the "good state" and s_1 is the "bad state". Each agent has two candidate actions $\mathcal{A}_1 = \{a_0, a_1\}$ and $\mathcal{A}_2 = \{b_0, b_1\}$. The reward function at each state is presented in Table 2. Specifically, at state s_1, both agents get a reward of 0 no matter what actions they select, while at state s_0, they will obtain a strictly positive reward if they either take the joint action (a_0, b_1) or the one (a_1, b_0). The state transition function is defined as follows:

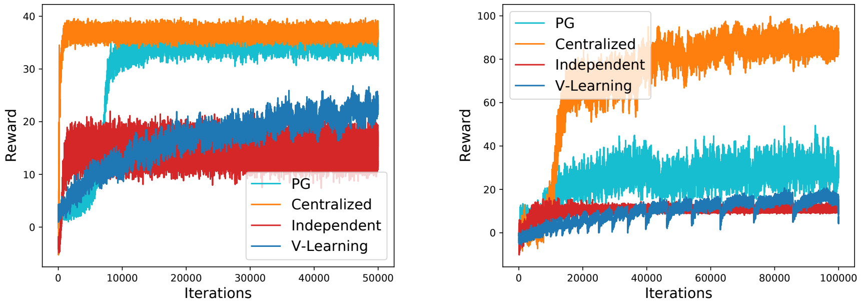
$$P_h(s_0 \mid s_0 \text{ or } s_1, a_0, b_1) = 1 - \varepsilon,\ P_h(s_1 \mid s_0 \text{ or } s_1,\ \text{not } (a_0, b_1)) = 1 - \varepsilon,\ \forall h \in [H],$$

and all the other transitions happen with probability ε. Intuitively, no matter which state the agents are in, they will transition to the good state s_0 with a high probability $1 - \varepsilon$ at the next step as long as they select the action pair (a_0, b_1). All the other joint actions will lead to the bad state s_1 with a high probability $1 - \varepsilon$. The task hence rewards the agents who learn to consistently play the action pair (a_0, b_1).

We run our two algorithms on this example for $K = 50000$ episodes, each episode containing $H = 10$ steps. We set the transition probability $\varepsilon = 0.1$. For Algorithm 1, the step size is set to be $\eta_i = \frac{1}{5\sqrt{A_i \check{T}_h(s_h)}}$, and the implicit exploration parameter is $\gamma_i = \eta_i/2$. For Algorithm 3, the step size is set to be $\eta_t = 10^{-4}$ and the momentum parameter is $a_t = 0.5$.

The BoxPushing task (Seuken & Zilberstein, 2007) is a classic DecPOMDP problem with with $\sim$100 states. It has two 2 agents, where each agent has 4 candidate actions. In the original BoxPushing problem, each agent only has a partial observation of the state. We make proper modifications to the task so that the agents can fully observe the state information and fit in our problem formulation. For Algorithm 1 on this task, the step size is set to be $\eta_i = \frac{1}{20\sqrt{A_i \check{T}_h(s_h)}}$, and the implicit exploration parameter is $\gamma_i = \eta_i/2$. For Algorithm 3, the step size is set to be $\eta_t = 5 \times 10^{-4}$ and the momentum parameter is $a_t = 0.1$.

We compare our algorithms with two meaningful benchmarks. The first benchmark is a "Centralized" oracle. This oracle acts as a centralized coordinator that can control the actions of both agents. Such an oracle essentially converts the multi-agent task into a single-agent RL problem. The (randomized) action space of the centralized agent is $\Delta(\mathcal{A}_1 \times \mathcal{A}_2)$, which is larger than the $\Delta(\mathcal{A}_1) \times \Delta(\mathcal{A}_2)$ space that we allow for Algorithm 3 in our decentralized



(a) GoodState

(b) BoxPushing

Figure 3. Rewards of Algorithms 1 and 3 on the (a) GoodState and (b) BoxPusing tasks. “V-Learning” and “PG” denote the policies at the current iterate t of Algorithms 1 and 3, respectively. “Centralized” is an oracle that can control the actions of the agents in a centralized way. In “Independent”, each agent runs a naïve single-agent Q-learning algorithm independently, by taking greedy actions w.r.t its local Q-function estimates. All results are averaged over 20 runs.

approach. “Centralized” clearly upper bounds the performances that our decentralized learning algorithms can possibly achieve in this task. In our simulations, we implement “Centralized” by using a Hoeffding-based variant of a state-of-the-art single-agent RL algorithm UCB-ADVANTAGE (Zhang et al., 2020a). This algorithm has achieved a tight sample complexity bound for single-agent RL in theory, and has also demonstrated remarkable empirical performances in practice (Mao et al., 2020b). Such an algorithm could provide a strong performance upper bound in our task. The second benchmark we consider is the naïve “Independent” Q-learning. Specifically, we let each agent run a single-agent Q-learning algorithm independently, without being aware of the existence of the other agent or the structure of the game. Each agent maintains an local optimistic Q-function, and takes greedy actions with respect to such optimistic estimates, without taking into account the other agents’ actions. Since the agents update their policies simultaneously, the stationarity assumption of the environment in single-agent RL quickly collapses, and the theoretical guarantees for single-agent Q-learning no longer hold. This is also reminiscent of the “independent learner” approach proposed in an early work (Claus & Boutilier, 1998) for learning in Markov teams. We believe that such a benchmark could provide meaningful intuitions about the consequences of not taking care of the multi-agent structure in decentralized methods. In our simulations, we implement such a benchmark by letting each agent running a variant of the single-agent UCB-ADVANTAGE (Zhang et al., 2020a) algorithm independently, where the (randomized) action spaces of the agents are $\Delta(\mathcal{A}_1)$ and $\Delta(\mathcal{A}_2)$.

Figure 3 illustrates the performances of our algorithms and the two benchmark methods in terms of the collected rewards, where “V-Learning” and “PG” denote the policies at the current iterate t of Algorithms 1 and 3, respectively. Notice that the *actual* policy trajectories of both algorithms numerically converge and achieve high rewards. This is more encouraging than our theoretical guarantees, because for Algorithm 1, our Theorem 4 only holds for a “certified” output policy but not the last-iterate policy. Further, both of our algorithms outperform the “Independent” learning benchmark on the two tasks. In the GoodState problem, Algorithm 3 even approaches the performance of the “Centralized” oracle. On the other hand, the “Independent” benchmark converges, albeit faster, to a clearly suboptimal value. This reiterates that the naïve idea of independent learning does not work well for MARL in general, and a careful treatment of the game structure (like our adversarial bandit subroutine) is necessary. Finally, the implemented algorithms take much fewer samples to converge than our theoretical results suggested. This indicates that the theoretical bounds might be overly conservative, and our algorithms could converge much faster in practice.