WHEN CAN WE LEARN GENERAL-SUM MARKOV GAMES WITH A LARGE NUMBER OF PLAYERS SAMPLE-EFFICIENTLY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reinforcement learning has made substantial empirical progresses in solving games with a large number of players. However, theoretically, the best known sample complexity for finding a Nash equilibrium in general-sum games scales exponentially in the number of players due to the size of the joint action space, and there is a matching exponential lower bound. This paper investigates what learning goals admit better sample complexities in the setting of m-player general-sum Markov games with H steps, S states, and A_i actions per player. First, we design algorithms for learning an ε-Coarse Correlated Equilibrium (CCE) in $\widetilde{\mathcal{O}}(H^5 S \max_{i\le m} A_i/\varepsilon^2)$ episodes, and an ε-Correlated Equilibrium (CE) in $\widetilde{\mathcal{O}}(H^6 S \max_{i\le m} A_i^2/\varepsilon^2)$ episodes. This is the first line of results for learning CCE and CE with sample complexities polynomial in $\max_{i\le m} A_i$. Our algorithm for learning CE integrates an adversarial bandit subroutine which minimizes a weighted swap regret, along with several novel designs in the outer loop. Second, we consider the important special case of Markov Potential Games, and design an algorithm that learns an ε-approximate Nash equilibrium within $\widetilde{\mathcal{O}}(S \sum_{i\le m} A_i/\varepsilon^3)$ episodes (when only highlighting the dependence on S, A_i, and ε), which only depends linearly in $\sum_{i\le m} A_i$ and significantly improves over the best known algorithm in the ε dependence. Overall, our results shed light on what equilibria or structural assumptions on the game may enable sample-efficient learning with many players.

1 INTRODUCTION

Multi-agent reinforcement learning (RL) has achieved substantial recent successes in solving artificial intelligence challenges such as GO (Silver et al., 2016; 2018), multi-player games with team play such as Starcraft (Vinyals et al., 2019) and Dota2 (Berner et al., 2019), behavior learning in social interactions (Baker et al., 2019), and economic simulation (Zheng et al., 2020; Trott et al., 2021). In many applications, multi-agent RL is able to yield high quality policies for multi-player games with a large number of players (Wang et al., 2016; Yang et al., 2018).

Despite these empirical progresses, theoretical understanding of when we can sample-efficiently solve multi-player games with a large number of players remains elusive, especially in the setting of multi-player Markov games. A main bottleneck here is the *exponential blow-up* of the *joint action space*—The total number of joint actions in a generic game with simultaneous plays is equal to the product of the number of actions for each player, which scales exponentially in the number of players. Such an exponential dependence is indeed known to be unavoidable in the worst-case for certain standard problems. For example, for learning an approximate Nash equilibrium from payoff queries in an one-step multi-player general-sum game, the query complexity lower bound of (Chen et al., 2015; Rubinstein, 2016) shows that at least exponentially many queries (samples) is required, even when each player only has two possible actions and the query is noiseless. Moreover, for learning Nash equilibrium in Markov games, the best existing sample complexity upper bound also scales with the size of the joint action space (Liu et al., 2021).

Nevertheless, these exponential lower bounds do not completely rule out interesting theoretical inquiries—there may well be other notions of equilibria or additional structures within the game that allow us to learn with a better sample complexity. This motivates us to ask the following

> **Question**: When can we solve general-sum Markov games with sample complexity milder than exponential in the number of players?

This paper makes steps towards answering the above question by considering multi-player general-sum Markov games (MGs) with m players, H steps, S states, and A_i actions per player. We make two lines of investigations: (1) Can we learn alternative notions of equilibria with better sample complexity than learning Nash; (2) Can the Nash equilibrium be learned with better sample complexity under additional structural assumptions on the game. This paper makes contributions on both ends, which we summarize as follows.

- We first design an algorithm that learns the ε-approximate Coarse Correlated Equilibrium (CCE) with $\widetilde{\mathcal{O}}(H^5 S \max_{i\in[m]} A_i/\varepsilon^2)$ episodes of play (Section 3). Our algorithm CCE-V-LEARNING is a multi-player adaptation of the Nash V-Learning algorithm of Bai et al. (2020).
- We design an algorithm CE-V-LEARNING which learns the stricter notion of ε-approximate Correlated Equilibrium (CE) with $\widetilde{\mathcal{O}}(H^6 \max_{i\in[m]} A_i^2/\varepsilon^2)$ episodes of play (Section 4). For Markov games, these are the first line of sample complexity results for learning CE and CCE that only scales polynomially with $\max_{i\in[m]} A_i$, and improves significantly in the A_i dependency over the current best algorithm which scales with $\prod_{i\in[m]} A_i$.
- Technically, our algorithm CE-V-LEARNING makes several major modifications over CCE-V-LEARNING in order to learn the CE (Section 4.2). Notably, inspired by the connection between CE and *low swap-regret* learning, we use a mixed-expert Follow-The-Regularized Leader algorithm within its inner loop to achieve low swap-regret for a particular adversarial bandit problem. Our analysis also contains new results for adversarial bandits on weighted swap regret and weighted regret with predicable weights, which may be of independent interest.
- Finally, we consider learning Nash equilibrium in *Markov Potential Games* (MPGs), an important subclass of general-sum Markov games. By a reduction to single-agent RL, we design an algorithm NASH-CA that achieves $\widetilde{\mathcal{O}}(\Phi_{\max} H^3 S \sum_{i\in[m]} A_i/\varepsilon^3)$ sample complexity, where $\Phi_{\max} \le Hm$ is the bound on the potential function (Section 5). Compared with the recent result of (Leonardos et al., 2021), we significantly improves the ε dependence from their $1/\varepsilon^6$.

1.1 RELATED WORK

Learning equlibria in general-sum games The sample (query) complexity of learning Nash, CE, and CCE from samples in *one-step* (i.e. normal form) general-sum names with m players and A_i actions per player has been studied extensively in literature (Hart & Mas-Colell, 2000; Hart, 2005; Stoltz, 2005; Cesa-Bianchi & Lugosi, 2006; Blum & Mansour, 2007; Fearnley et al., 2015; Babichenko & Barman, 2015; Chen et al., 2015; Fearnley & Savani, 2016; Goldberg & Roth, 2016; Babichenko, 2016; Rubinstein, 2016; Hart & Nisan, 2018). It is known that learning Nash equilibrium requires exponential in m samples in the worst case Rubinstein (2016), whereas CE and CCE admit efficient $\mathrm{poly}(m, \max_{i\le m} A_i)$-sample complexity algorithms by independent no-regret learning (Hart & Mas-Colell, 2000; Hart, 2005; Syrgkanis et al., 2015; Goldberg & Roth, 2016; Chen & Peng, 2020; Daskalakis et al., 2021). Our results for learning CE and CCE can be seen as extension of these works into Markov games. We remark that even when the game is fully known, the computational complexity for finding Nash in general-sum games is still PPAD-hard (Daskalakis, 2013).

Markov games Markov games (Shapley, 1953; Littman, 1994) is a widely used framework for game playing with sequential decision making, e.g. in multi-agent reinforcement learning. Algorithms with asymptotic convergence have been proposed in the early works of (Hu & Wellman, 2003; Littman, 2001; Hansen et al., 2013). A recent line of work studies the non-asymptotic sample complexity for learning Nash in two-player zero-sum Markov games (Bai & Jin, 2020; Xie et al., 2020; Bai et al., 2020; Zhang et al., 2020; Liu et al., 2021; Chen et al., 2021; Jin et al., 2021; Huang et al., 2021) and learning various equilibria in general-sum Markov games (Liu et al., 2021; Bai et al., 2021), building on techniques for learning single-agent Markov Decision Processes sample-efficiently (Azar et al., 2017; Jin et al., 2018). Learning the Nash equilibrium in general-sum Markov games are much harder than that in zero-sum Markov games. Liu et al. (2021) present the first line of results for learning Nash, CE, and CCE in general-sum Markov games; however their sample complexity

scales with $\prod_{i\leq m} A_i$ due to the model-based nature of their algorithm. Algorithms for learning CE in extensive-form games has been studied in (Celli et al., 2020), though we remark Markov games and extensive-form games are different frameworks and our results do not imply each other.

Markov potential games Lastly, a recent line of works considers Markov potential games (Macua et al., 2018; Leonardos et al., 2021; Zhang et al., 2021), a subset of general-sum Markov games in which the Nash equilibrium admits more efficient algorithms. Leonardos et al. (2021) gives a sample-efficient algorithm based on the policy gradient method (Agarwal et al., 2021). The special case of Markov cooperative games is studied empirically in e.g. Lowe et al. (2017); Yu et al. (2021). For one step potential games, (Kleinberg et al., 2009; Palaiopanos et al., 2017; Cohen et al., 2017a) show the convergence to Nash equilibria of no-regret dynamics.

2 Preliminaries

We present preliminaries for multi-player general-sum Markov games as well as the solution concept of (approximate) Nash equilibrium. Alternative solution concepts and other concrete subclasses of Markov games considered in this paper will be defined in the later sections.

Markov games A multi-player general sum Markov game (MG; (Shapley, 1953; Littman, 1994)) with m players can be described by a tuple $\mathrm{MG}(H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathbb{P}, \{r_i\}_{i=1}^m)$, where H is the episode length, $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$, $\mathcal{A}_i$ is the action space for the ith player with $|\mathcal{A}_i| = A_i$. Without loss of generality, we assume $\mathcal{A}_i = [A_i]$. We let $\boldsymbol{a} := (a_1, \cdots, a_m)$ denote the vector of joint actions taken by all the players and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ denote the joint action space. Throughout this paper we assume that S and A_i are finite. The transition probability $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is the collection of transition matrices, where $\mathbb{P}_h(\cdot|s, \boldsymbol{a}) \in \Delta_{\mathcal{S}}$ denotes the distribution of the next state when actions $\boldsymbol{a}$ are taken at state s at step h. The rewards $r_i = \{r_{h,i}\}_{h\in[H],i\in[m]}$ is the collection of reward functions for the ith player, where $r_{h,i}(s, \boldsymbol{a}) \in [0, 1]$ gives the deterministic[1] reward of ith player if actions $\boldsymbol{a}$ are taken at state s at step h. Without loss of generality, we assume the initial state s_1 is deterministic. A key feature of general-sum games is that the rewards r_i are in general different for each player i, and the goal of each player is to maximize her own cumulative reward.

Policy, value function A product policy is a collection of m policies $\pi := \{\pi_i\}_{i\in[m]}$ where π_i is the (potentially history-dependent) policy for the i-th player. We first focus on the case of Markov product policies, in which $\pi_i = \{\pi_{h,i} : \mathcal{S} \to \Delta_{\mathcal{A}_i}\}_{h\in[H]}$, and $\pi_{h,i}(a_i|s)$ is the probability for the ith player to take action a_i at state s at step h. For a policy π and $i \in [m]$, we use $\pi_{-i} := \{\pi_j\}_{j\in[m],j\neq i}$ to denote the policy of all but the ith player. The value function $V_{h,i}^{\pi}(s) : \mathcal{S} \to \mathbb{R}$ is defined as the expected cumulative reward for the ith player when policy π is taken starting from state s and step h:

$$V_{h,i}^{\pi}(s) := \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'})\middle| s_h = s\right]. \tag{1}$$

Definition of the Q function and Bellman equations can be found in Appendix B.

Best response & Nash equilibrium For any product policy $\pi = \{\pi_i\}_{i\in[m]}$, the *best response* for the ith player against π_{-i} is defined as any policy $\pi^{\dagger}$ such that $V_{1,i}^{\pi^{\dagger},\pi_{-i}}(s_1) = \sup_{\pi_i'} V_{1,i}^{\pi_i',\pi_{-i}}(s_1)$. For any Markov product policy, this best response is guaranteed to exist as the above maximization problem is equivalent to solving a Markov Decision Process (MDP) for the ith player. We will also use the notation $V_{1,i}^{\dagger,\pi_{-i}}(s_1)$ to denote the above value function $V_{1,i}^{\pi^{\dagger},\pi_{-i}}(s_1)$.

We say π is a *Nash equilibrium* (e.g. Nash (1951); Pérolat et al. (2017)) if all players play the best response against other players, i.e., for all $i \in [m]$,

$$V_{1,i}^{\pi}(s_1) = V_{1,i}^{\dagger,\pi_{-i}}(s_1).$$

[1] Our results can be straightforwardly generalized to Markov games with stochastic rewards.

Note that in general-sum MGs, there may exist multiple Nash equilibrium policies with different value functions, unlike in two-player zero-sum MGs (Shapley, 1953). To measure the suboptimality of any policy π, we define the NE-gap as

$$\text{NE-gap}(\pi) := \max_{i\in[m]} \left[\sup_{\mu_i} V_{1,i}^{\mu_i,\pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1) \right].$$

For any $\varepsilon \geq 0$, we say π is ε-approximate Nash equilibrium if NE-gap$(\pi) \leq \varepsilon$.

General correlated policy A general correlated policy π is a set of H maps $\pi := \{\pi_h : \mathbb{R} \times (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}}\}_{h\in[H]}$. The first argument of π_h is a random variable $z \in \mathbb{R}$, and the other arguments contain all the history information and the current state information (unlike Markov policies in which the policies only depend on the current state information). The output of π_h is a general distribution of actions in $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ (unlike product policies in which the action distribution is a product distribution). The random variable z is sampled from some underlying distribution which may be shared among all steps $h \in [H]$.

For any correlated policy $\pi = \{\pi_h\}_{h\in[H]}$ and any player i, we can define a marginal policy π_{-i} as a set of H maps $\pi_{-i} := \{\pi_{h,-i} : \mathbb{R} \times (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}_{-i}}\}_{h\in[H]}$ where $\mathcal{A}_{-i} := \mathcal{A}_1 \times \cdots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \cdots \times \mathcal{A}_m$, and the output of $\pi_{h,-i}$ is defined as the marginal distribution of the output of π_h restricted to the space $\mathcal{A}_{-i}$. For any general correlated policy π, we can define its initial state value function $V_{1,i}^{\pi}(s_1)$ similar as (1). The best response value of the ith player against π is $V_{1,i}^{\dagger,\pi_{-i}}(s_1) = \sup_{\mu_i} V_{1,i}^{\mu_i,\pi_{-i}}(s_1)$, where $V_{1,i}^{\mu_i,\pi_{-i}}(s_1)$ is the value function of the policy (μ_i, π_{-i}) (the ith player plays according to general policy μ_i, and all other players play according to π_{-i}), and the supremum is taken over all general policy μ_i of the ith player.

Learning setting Throughout this paper we consider the interactive learning (i.e. exploration) setting where algorithms are able to play episodes within the MG and observe the realized transitions and rewards. Our focus is on the PAC sample complexity (i.e. number of episodes of play) for any learning algorithm to output an approximate equilibrium.

2.1 Exponential lower bound for learning approximate Nash equilibrium

The focus of this paper is the setting where the number of players m is large. Intuitively, as the joint action space has size $|\mathcal{A}| = \prod_{i=1}^{m} A_i$ which scales exponentially in m (if each $A_i \geq 2$), naive algorithms for learning Nash equilibrium may learn all $r_i(\boldsymbol{a})$ by enumeratively querying all $\boldsymbol{a} \in \mathcal{A}$, and this costs exponential in m samples. Unfortunately, recent work shows that such exponential in m dependence is unavoidable in the worst-case for any algorithm—there is an $\exp(\Omega(m))$ sample complexity lower bound for learning approximate Nash, even in *one-step* general-sum games (Chen et al., 2015; Rubinstein, 2016):

Proposition 1 (Lower bound for learning approximate Nash (Rubinstein, 2016); Formal version deferred to Proposition A.2)**.** *There exists some absolute constant $\varepsilon_0 > 0$ and a family of m-player general-sum Markov games for any m with $H = 1$, $S = 1$, $A_i = 2$, and* deterministic *rewards, such that any algorithm that is able to find an ε_0-approximate Nash equilibrium with high probability must at least play $2^{\Omega(m)}$ samples (episodes).*

This suggests that the Nash equilibrium as a solution concept may be too hard to learn efficiently for MGs with a large number of players, and calls for alternative solution concepts or additional structural assumptions on the game in order to achieve an improved m dependence.

3 Efficient Learning of Coarse Correlated Equilibria (CCE)

Given the difficulty of learning Nash when the number of players m is large (Proposition 1), we consider learning other relaxed notions of equilibria for general-sum MGs. Two standard notions of equilibria for games are the Correlated Equilbrium (CE) and Coarse Correlated Equilibrium (CCE), and they satisfy $\{\text{Nash}\} \subset \{\text{CE}\} \subset \{\text{CCE}\}$ for general-sum MGs (Nisan et al., 2007).

We begin by considering learning CCE (most relaxed notion above) for Markov games.

Definition 2 (ε-approximate CCE for general-sum MGs)**.** *We say a (general) correlated policy π is an* ε-approximate Coarse Correlated Equilibrium (CCE) *if*

$$\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}}(s_1)-V_{1,i}^{\pi}(s_1)\right)\le\varepsilon.$$

We say π is an (exact) CCE if the above is satisfied with $\varepsilon=0$.

The following result shows that there exists an algorithm that can learn an ε-approximate CCE in general-sum Markov games within $\tilde{O}(H^5S\max_{i\in[m]}A_i/\varepsilon^2)$ episodes of play.

Theorem 3 (Learning ε-approximate CCE for general-sum MGs)**.** *There exists an algorithm* CCE-V-LEARNING *(Algorithm 4) for m-player general-sum MGs, such that running it for*

$$K\ge\mathcal{O}\left(\frac{H^5S\max_{i\in[m]}A_i\iota}{\varepsilon^2}+\frac{H^4S\iota^3}{\varepsilon}\right),$$

episodes ($\iota=\log(m\max_{i\in[m]}A_iHSK/(p\varepsilon))$ is a log factor), we have with probability at least $1-p$ that the certified policy $\widehat{\pi}$ defined in Algorithm 2 is an ε-approximate CCE, i.e. $\max_{i\in[m]}\left(V_{1,i}^{\dagger,\widehat{\pi}_{-i}}(s_1)-V_{1,i}^{\widehat{\pi}}(s_1)\right)\le\varepsilon$.

Mild dependence on action space For small enough ε, the sample complexity featured in Theorem 3 scales as $\widetilde{\mathcal{O}}(H^5S\max_{i\in[m]}A_i/\varepsilon^2)$. Most notably, this is the first algorithm that scales with $\max_{i\in[m]}A_i$, and exhibits a sharp difference in learning Nash and learning CCE in view of the $\exp(\Omega(m))$ lower bound for learning Nash in Proposition 1. Indeed, existing algorithms such as Multi-Nash-VI Algorithm with CCE subroutine (Liu et al., 2021) does require $\widetilde{\mathcal{O}}(H^3S^2\prod_{i=1}^m A_i/\varepsilon^2)$ episodes of play, which scales with $\prod_{i\in[m]}A_i$ due to its model-based nature. We achieve significantly better dependence on A_i and also S, though slightly worse H dependence.

Overview of algorithm and techniques Our CCE-V-LEARNING algorithm (deferred to Appendix C.1 due to space limit) is a multi-player adaptation of the Nash V-Learning algorithm of Bai et al. (2020); Tian et al. (2021) for learning Nash equilibria in two-player zero-sum MGs. Similar as Bai et al. (2020), we show that this algorithm enjoys a "no-regret" like guarantee for each player at each (h,s) (Lemma C.3), with choice of hyperparameters coming from Tian et al. (2021) in order to achieve a slightly better sample complexity. When combined with our "certified correlated policy" procedure (Algorithm 2), our algorithm outputs a policy that is ε-CCE. Our certified policy procedure is adapted from the certified policy of Bai et al. (2020), and differs in that ours output a *correlated* policy for all the players instead of a product of independent policies as in Bai et al. (2020). The key feature allowing this $\max_{i\in[m]}A_i$ dependence is that this algorithm uses *uncoupled learning* for each player, as opposed to the centralized model-based algorithm of Liu et al. (2021). The full proof of Theorem 3 can be found in Appendix C.

4 EFFICIENT LEARNING OF CORRELATED EQUILIBRIA (CE)

In this section, we move on to considering the harder problem of learning Correlated Equilibria (CE). We first present the definition of CE in Markov games.

Definition 4 (Strategy modification for i^{th} player)**.** *A strategy modification $\phi:=\{\phi_{h,s}\}_{(h,s)\in[H]\times\mathcal{S}}$ for player i is a set of $S\times H$ functions $\phi_{h,s}:\mathcal{A}_i\to\mathcal{A}_i$. A strategy modification ϕ can be composed with any policy π to give a modified policy $\phi\diamond\pi$ defined as follows: At any step h and state s, if π chooses to play $\boldsymbol{a}=(a_1,\ldots,a_m)$, the modified policy $\phi\diamond\pi$ will play $(a_1,\ldots,a_{i-1},\phi_{h,s}(a_i),a_{i+1},\ldots,a_m)$. We use Φ_i denote the set of all possible strategy modifications for player i.*

Definition 5 (ε-approximate CE for general-sum MGs)**.** *We say a (general) correlated policy π is an ε-approximate CE if*

$$\max_{i\in[m]}\sup_{\phi\in\Phi_i}\left(V_{1,i}^{\phi\diamond\pi}(s_1)-V_{1,i}^{\pi}(s_1)\right)\le\varepsilon.$$

We say π is an (exact) CE if the above is satisfied with $\varepsilon=0$.

Our definition of CE follows (Liu et al., 2021) and is a natural generalization of the CE for the well-studied special case of one-step (i.e. normal form) games (Nisan et al., 2007).

4.1 ALGORITHM DESCRIPTION

Our algorithm CE-V-LEARNING (Algorithm 1) builds further on top of CCE-V-LEARNING and Nash V-Learning, and makes several novel modifications in order to learn the CE. The key feature of CE-V-LEARNING is that it uses a ow weighted swap regret algorithm (mixed-expert FTRL) for all (s, h, i). At a high-level, CE-V-LEARNING consists of the following steps:

- Line 6-11 (Sample action using mixed-expert FTRL): For each (h, s) we maintain A_i "sub-experts" indexed by $b' \in [A_i]$ (Each sub-expert represents an independent "expert" that runs her own FTRL algorithm). Sub-expert b' first computes an action distribution $q^{b'}(\cdot) \in \Delta_{\mathcal{A}_i}$ via Follow-the-Regularized-Leader (FTRL; Line 8). Then we employ a two-step sampling procedure to obtain the action: First sample a sub-expert b from a suitable distribution μ computed from $\{q^{b'}\}_{b' \in [A_i]}$, then sample the actual action $a_{h,i}$ from q^b.
- Line 13-17 (Take action and record observations): Player i takes action $a_{h,i}$ and observes other player's actions, the reward, and the next state. Sub-expert b then computes a loss estimator and weight according to the observations, which will be used in future FTRL updates.
- Line 19 (Optimistic value update): Updates the optimistic estimate of the value $\overline{V}_{h,i}$ using step-size α_t and bonus $\overline{\beta}_t$.

Finally, after executing Algorithm 1 for K episodes, we use the certified correlated policy procedure (Algorithm 2) to obtain our final output policy $\widehat{\pi}$. This procedure is a direct modification of the certified policy procedure of (Bai et al., 2020) and outputs a correlated policy (because the randomly sampled k and l in line 1 and line 4 of Algorithm 2 are used by all the players) instead of product policy. The same procedure is also used for learning CCEs earlier in Section 3.

Here we specify the hyperparameters used in Algorithm 1:

$$\alpha_t = (H+1)/(H+t), \quad \eta_t = \sqrt{\iota/A_i t}, \quad \overline{\beta}_t = cH^2 A_i \sqrt{\iota/t} + 2cH^2\iota/t. \tag{2}$$

The constants α_t^j used in Algorithm 2 is defined as

$$\alpha_t^0 := \textstyle\prod_{k=1}^t (1-\alpha_k), \quad \alpha_t^j := \alpha_j \prod_{k=j+1}^t (1-\alpha_k). \tag{3}$$

Note that for any $t \geq 1$, $\{\alpha_t^j\}_{1 \leq j \leq t}$ sums to one and defines a distribution over $[t]$.

4.2 OVERVIEW OF TECHNIQUES

Here we briefly overview the techniques used in Algorithm 1.

Minimizing swap regret via mixed-expert FTRL. The key technical advance in our Algorithm 1 over CCE-V-LEARNING and Nash V-Learning is the use of mixed-expert FTRL (Line 6-11). The purpose of this is to allow the algorithm to achieve low *swap regret* at each (h, s) in a suitable sense—For one-step (normal form) games, it is known that combining low-swap-regret learning for each player leads to an approximate CE (Stoltz, 2005; Cesa-Bianchi & Lugosi, 2006). To integrate this into Markov games, we utilize a celebrated reduction from low-swap-regret learning to usual low-regret learning (Blum & Mansour, 2007), which for any bandit problem with A_i actions maintains A_i *sub-experts* each running its own FTRL algorithm. Our particular application builds upon the *two-step randomization* scheme of Ito (2020) which first samples a sub-expert b and the action from this sub-expert. The distribution $\mu(\cdot)$ for sampling the sub-expert is carefully chosen by solving a linear system (Line 10) so that μ also coincides with the (marginal) distribution of the sampled action, from which the reduction follows.

FTRL with predictable weights. Applied naively, the above reduction does not directly work for our purpose, as our analysis requires minimizing the *weighted* swap regret with weights α_t^i, whereas the reduction of Ito (2020) relies crucially on the vanilla (average) regret. We address this challenge by using a slightly modified FTRL algorithm for each sub-expert that takes in random but *predictable* weights (i.e. depending fully on prior information and "external" randomness). We present the analysis for such FTRL algorithm in Appendix G.2, and the consequent analysis for the weighted swap regret in Appendix G.1, both of which may be of independent interest.

Proposal distributions. Finally, a nuanced but important new design in CE-V-LEARNING is that all sub-experts compute a *proposal* action distribution to sample the sub-expert and the associated action.

Algorithm 1 CE-V-LEARNING for general-sum MGs (i-th player's version)

Initialize: For any (s, a, h), set $\overline{V}_{h,i}(s) \leftarrow H$, $N_h(s) \leftarrow 0$. Set $\mu_h(a|s) \leftarrow 1/A_i$, $q_h^{b'}(a|s) \leftarrow 1/A_i$, $\ell_{h,t}^{b'}(a|s) \leftarrow 0$, $N_h^{b'}(s) \leftarrow 0$ for all $(b', a, h, s, t) \in [A_i] \times [A_i] \times [H] \times \mathcal{S} \times [K]$.
for episode $k = 1, \ldots, K$ **do**
  Receive s_1.
  for step $h = 1, \ldots, H$ **do**
    // Compute *proposal* action distributions by FTRL
    Update accumulator $t := N_h(s_h) \leftarrow N_h(s_h) + 1$. Set $u_t \leftarrow \alpha_t^t / \alpha_t^1$.
    Let $t_{b'} \leftarrow N_h^{b'}(s_h)$ for all $b' \in [A_i]$ for shorthand.
    Compute the action distribution for all sub-experts $b' \in [A_i]$:

$$q_h^{b'}(a|s_h) \propto_a \exp\Big(-(\eta_{t_{b'}}/u_t) \textstyle\sum_{\tau=1}^{t_{b'}} w_{h,\tau}(b'|s_h) \ell_{h,\tau}^{b'}(a|s_h) \Big).$$

    // Sample sub-expert b and action from $q^b(\cdot)$
    Compute $\mu_h(\cdot|s_h) \in \Delta_{[A_i]}$ by solving $\mu_h(\cdot|s_h) = \sum_{b'=1}^{A_i} \mu_h(b'|s_h) q_h^{b'}(\cdot|s_h)$.
    Sample sub-expert $b \sim \mu_h(\cdot|s_h)$, and then action $a_{h,i} \sim q_h^b(\cdot|s_h)$.
    // Take action and feed the observations to sub-expert b
    Take action $a_{h,i}$, observe the action $\boldsymbol{a}_{h,-i}$ from the other players.
    Observe reward $r_{h,i} = r_{h,i}(s_h, a_{h,i}, \boldsymbol{a}_{h,-i})$ and next state s_{h+1} from the environment.
    Update accumulator for sampled sub-expert: $t_b := N_h^b(s_h) \leftarrow N_h^b(s_h) + 1$.
    Compute loss estimator

$$\ell_{h,t_b}^b(a|s_h) \leftarrow \frac{\big[H - h + 1 - (r_{h,i} + \min\{\overline{V}_{h+1,i}(s_{h+1}), H - h\})\big]/H \cdot \mathbf{1}\{a_{h,i} = a\}}{q_h^b(a_h|s_h) + \gamma_{t_b}}.$$

    Set $w_{h,t_b}(b|s_h) \leftarrow u_t$.
    // Optimistic value update
    $\overline{V}_{h,i}(s_h) \leftarrow (1 - \alpha_t)\overline{V}_{h,i}(s_h) + \alpha_t \big(r_{h,i}(s_h, a_{h,i}, \boldsymbol{a}_{h,-i}) + \overline{V}_{h+1,i}(s_{h+1}) + \overline{\beta}_t\big)$.

Algorithm 2 Certified correlated policy $\widehat{\pi}$ for general-sum MGs

Sample $k \leftarrow \text{Uniform}([K])$.
for step $h = 1, \ldots, H$ **do**
  Observe s_h, and set $t \leftarrow N_h^k(s_h)$ (the value of $N_h(s_h)$ at the beginning of the k'th episode).
  Sample $l \in [t]$ with $\mathbb{P}(l = j) = \alpha_t^j$ (c.f. Eq. (3)).
  $k \leftarrow k_h^l(s_h)$ (the episode at the end of which the state s_h is observed exactly l times).
  Jointly take action $(a_{h,1}, a_{h,2}, \ldots, a_{h,m}) \sim \prod_{i=1}^m \mu_{h,i}^k(\cdot|s_h)$, where $\mu_{h,i}^k(\cdot|s_h)$ is the policy $\mu_{h,i}(\cdot|s_h)$ at the beginning of k'th episode.

Then, only the sampled sub-expert takes this action, and all other proposal distributions are discarded. This is different from the original algorithms of (Blum & Mansour, 2007; Ito, 2020) in which the FTRL update come *after* the sub-expert sampling and only happens for the sampled sub-expert. Our design is required here as otherwise the sub-experts are required to predict the next time when it is sampled in order to compute the weighted FTRL update, which is impossible.

4.3 THEORETICAL GUARANTEE

We are now ready to present the theoretical guarantee for our CE-V-LEARNING algorithm.

Theorem 6 (Learning ε-approximate CE for general-sum MGs)**.** *Suppose we run the* CE-V-LEARNING *algorithm (Algorithm 1) for all m players for*

$$K \geq \mathcal{O}\left(\frac{H^6 S \max_{i \in [m]} A_i^2 \iota}{\varepsilon^2} + \frac{H^4 S \max_{i \in [m]} A_i \iota^3}{\varepsilon} \right),$$

episodes ($\iota = \log(m \max_{i \in [m]} A_i HSK/(p\varepsilon))$ is a log factor). Then with probability at least $1 - p$, the certified correlated policy $\widehat{\pi}$ defined in Algorithm 2 is an ε-approximate CE i.e. $\max_{i \in [m]} \sup_{\phi \in \Phi_i} V_{h,i}^{\phi \diamond \pi}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1) \leq \varepsilon$.

Discussions To the best of our knowledge, Theorem 6 presents the first result for learning CCE that scales polynomially with $\max_{i\in[m]} A_i$, which is significantly better than the best known existing algorithm of Multi-Nash-VI with CE subroutine (Liu et al., 2021) whose sample complexity scales with $\prod_{i\in[m]} A_i$. Similar as in Theorem 3, this follows as our CE-V-LEARNING performs uncoupled learning for each player, whereas Liu et al. (2021) uses model-based centralized learning.

We also observe that our sample complexity for learning CE is higher than for learning CCE by a factor of $\widetilde{\mathcal{O}}(H \max_{i\in[m]} A_i)$, which is expected as CE is a strictly harder notion of equilibrium and required a more sophisticated mixed-expert FTRL technique in the inner loop of the algorithm.

Finally, combining Theorem 3 & 6 with the exponential lower bound for learning Nash (Section 2.1), we obtain a full characterization of which equilibria can be learned with $\mathrm{poly}(m)$ sample complexity in general-sum Markov games: This is possible for CCE and CE, but not Nash.

5 LEARNING NASH EQUILIBRIA IN MARKOV POTENTIAL GAMES

In this section, we consider learning Nash equilibria in Markov potential games (MPGs), an important subclass of general-sum MGs. Despite the curse of number of players of learning Nash in general-sum MGs, recent work shows that learning Nash in MPGs does not require sample size exponential in m, by using stochastic policy gradient based algorithms (Leonardos et al., 2021; Zhang et al., 2021). In this section, we provide an alternative algorithm NASH-CA that also achieves a mild dependence on m and an improved dependence on ε by a simple reduction to single-agent learning.

5.1 MARKOV POTENTIAL GAMES

We first present the definition of MPGs. Our definition is the finite-horizon variant[2] of the definitions of Macua et al. (2018); Leonardos et al. (2021); Zhang et al. (2021) and is slightly more general as we only require (4) on the total return. Throughout this section, π denotes a Markov product policy.

Definition 7. *(Markov potential games) A general-sum Markov game is a Markov potential game if there exists a potential function $\Phi : (\Delta_{\mathcal{A}_1})^H \times \cdots \times (\Delta_{\mathcal{A}_m})^H \to [0, \Phi_{\max}]$, such that for any $i \in [m]$, any two policies π_i, π_i' of the ith player, and any policy π_{-i} of other players, the difference of the value functions of the ith player with policies (π_i, π_{-i}) and (π_i', π_{-i}) is equals the difference of the potential function on the same policies, i.e.,*

$$V_{1,i}^{\pi_i,\pi_{-i}}(s_1) - V_{1,i}^{\pi_i',\pi_{-i}}(s_1) = \Phi(\pi_i, \pi_{-i}) - \Phi(\pi_i', \pi_{-i}). \tag{4}$$

Note that the range of the potential function $\Phi_{\max}$ admits a trivial upper bound $\Phi_{\max} \le mH$ (this can be seen by varying π_i for one i at a time). An important example of MPGs is Markov Cooperative Games (MCGs) where all players share the same reward $r_i \equiv r$.

5.2 ALGORITHM AND THEORETICAL GUARANTEE

We present a simple algorithm NASH-CA (Nash Coordinate Ascent) for learning an ε-approximate Nash equilibrium in MPGs. As its name suggests, the algorithm operates by solving single-agent Markov Decision Processes (MDPs) one player at a time, and intrinsically performing coordinate ascent on the potential function of the Markov game. Due to the potential structure of MPGs and the boundedness of the potential function, the local improvements of players across the steps can have an accumulative effect on the potential function, and the algorithm is guaranteed to stop after a bounded number of steps. We give the full description of the NASH-CA in Algorithm 3. We remark in passing that NASH-CA is additionally guaranteed to output a *pure-strategy* Nash equilibrium (cf. Appendix E for definition).

Theorem 8 (Sample complexity for NASH-CA)**.** *For Markov potential games, with probability at least $1-p$, Algorithm 3 terminates within $4\Phi_{max}/\varepsilon$ steps of the while loop, and outputs an ε-approximate (pure-strategy) Nash equilibrium. The total episodes of play is at most*

$$K = \mathcal{O}\left(\frac{\Phi_{\max} H^3 S \sum_{i=1}^m A_i \iota}{\varepsilon^3} + \frac{\Phi_{max} H^3 S^2 \sum_{i=1}^m A_i \iota^2}{\varepsilon^2}\right),$$

[2]Our results can easily adapted to the discounted infinite time horizon setup.

Algorithm 3 NASH-CA for Markov potential games

Require: Error tolerance ε

Initialize: $\pi = \{\pi_i\}_{i\in[m]}$, where $\pi_i = \{\pi_{h,i}\}_{(h,i)\in[H]\times[m]}$ for some deterministic policy $\pi_{h,i}$.
while true **do**
  Execute policy π for $N = \Theta(\frac{H^2\iota}{\varepsilon^2})$ episodes and obtain $\widehat{V}_{1,i}(\pi)$ which is the empirical average of the total return under policy π.
  for player $i = 1, \ldots, m$ **do**
    Fix π_{-i}, and let the ith player run UCBVI-UPLOW (Algorithm 7) for $K_i = \Theta(\frac{H^3SA_i\iota}{\varepsilon^2} + \frac{H^3S^2A_i\iota^2}{\varepsilon})$ episodes and get a new deterministic policy $\widehat{\pi}_i$.
    Execute policy $(\widehat{\pi}_i, \pi_{-i})$ for $N = \Theta(\frac{H^2\iota}{\varepsilon^2})$ episodes and obtain $\widehat{V}_{1,i}(\widehat{\pi}_i, \pi_{-i})$ which is the empirical average of the total return under policy $(\widehat{\pi}_i, \pi_{-i})$.
    Set $\Delta_i \leftarrow \widehat{V}_{1,i}(\widehat{\pi}_i, \pi_{-i}) - \widehat{V}_{1,i}(\pi)$.
  if $\max_{i\in[m]} \Delta_i > \varepsilon/2$ **then**
    Update $\pi_j \leftarrow \widehat{\pi}_j$ where $j = \arg\max_{i\in[m]} \Delta_i$.
  else
    return π

where $\iota = \log(\frac{mHSK \max_{1\le i\le m} A_i}{\varepsilon p})$ *is a log factor.*

Discussions For small enough ε, the sample complexity for the NASH-CA algorithm in the above Theorem is $\widetilde{\mathcal{O}}(\Phi_{\max} H^3 S \sum_{i\le m} A_i/\varepsilon^3)$. As $\Phi_{\max} \le mH$, this at most scales with the number of players as $m\sum_{i\le m} A_i$, which is much better than the exponential in m sample complexity for general-sum MGs without additional structures. Compared with recent results on learning Nash via policy gradients Leonardos et al. (2021); Zhang et al. (2021), the NASH-CA algorithm also achieves $\mathrm{poly}(m, \max_{i\le m} A_i)$ dependence, and significantly improves on the ε dependence from their ε^{-6} to ε^{-3}. In addition, our algorithm does require assumptions such bounded distribution mismatch coefficient as they do, due to the exploration nature of our single-agent MDP subroutine.

Also, compared with the sample complexity bound $\widetilde{\mathcal{O}}(H^3 S \prod_{i=1}^m A_i/\varepsilon^2)$ of the Nash-VI algorithm (Liu et al., 2021) for general-sum MGs (not restricted to MPGs), our NASH-CA algorithm doesn't suffer from the exponential dependence on m thanks to the MPG structure. We do achieve a looser in the dependence on ε, yet overall our sample complexity is still better unless $\varepsilon < (\sum_{i=1}^m A_i)/(\prod_{i=1}^m A_i)$ is exponentially small.

A lower bound To accompany Theorem 8, we establish a sample complexity lower bound of $\Omega(H^2 \sum_{i=1}^m A_i/\varepsilon^2)$ for learning pure-strategy Nash in MCGs and hence MPGs (Theorem F.1 in Appendix F). This lower bound improves in the A_i dependence over the naive reduction to single-player MDPs (Domingues et al., 2021), which gives $\Omega(H^3 S \max_{i\in[m]} A_i/\varepsilon^2)$, though is loose on the S, H dependence. The improved A_i dependence is achieved by constructing a novel class of hard instances of on one-step games (Lemma F.2), which may be of further technical interest. However, there is still a large gap between these lower bounds and the best current upper bound of either our $\widetilde{\mathcal{O}}(\sum_{i=1}^m A_i/\varepsilon^3)$ or the $\widetilde{\mathcal{O}}(\prod_{i=1}^m A_i/\varepsilon^2)$ of Liu et al. (2021), which we leave as future work.

6 CONCLUSION

This paper investigates the question of when can we solve general-sum Markov games (MGs) sample-efficiently with a mild dependence on the number of players. Our results show that this is possible for learning approximate (Coarse) Correlated Equilibria in general-sum MGs, as well as learning approximate Nash equilibrium in Markov potential games. In both cases, our sample complexity bounds improve over existing results in many aspects. Our work opens up many interesting directions for future work, such as sharper algorithms for both problems, sample complexity lower bounds, or how to perform sample-efficient learning in general-sum MGs with function approximations. In addition to Markov potential games, it would also be interesting to explore alternative structural assumptions that permit sample-efficient learning.

REFERENCES

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

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272. PMLR, 2017.

Yakov Babichenko. Query complexity of approximate nash equilibria. *Journal of the ACM (JACM)*, 63(4):1–24, 2016.

Yakov Babichenko and Siddharth Barman. Query complexity of correlated equilibrium. *ACM Transactions on Economics and Computation (TEAC)*, 3(4):1–9, 2015.

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

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

Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *arXiv preprint arXiv:2102.11494*, 2021.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

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

Andrea Celli, Alberto Marchesi, Gabriele Farina, and Nicola Gatti. No-regret learning dynamics for extensive-form correlated equilibrium. *arXiv preprint arXiv:2004.00603*, 2020.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Xi Chen and Binghui Peng. Hedging in games: Faster convergence of external and swap regrets. *arXiv preprint arXiv:2006.04953*, 2020.

Xi Chen, Yu Cheng, and Bo Tang. Well-supported versus approximate nash equilibria: Query complexity of large games. *arXiv preprint arXiv:1511.00785*, 2015.

Zixiang Chen, Dongruo Zhou, and Quanquan Gu. Almost optimal algorithms for two-player markov games with linear function approximation. *arXiv preprint arXiv:2102.07404*, 2021.

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

Michael B Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 410–419, 2017b.

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. *arXiv preprint arXiv:1510.08906*, 2015.

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

Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *arXiv preprint arXiv:2108.06924*, 2021.

Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pp. 578–598. PMLR, 2021.

John Fearnley and Rahul Savani. Finding approximate nash equilibria of bimatrix games via payoff queries. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):1–19, 2016.

John Fearnley, Martin Gairing, Paul W Goldberg, and Rahul Savani. Learning equilibria of games via payoff queries. *J. Mach. Learn. Res.*, 16:1305–1344, 2015.

Paul W Goldberg and Aaron Roth. Bounds for the query complexity of approximate equilibria. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):1–25, 2016.

Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

Sergiu Hart. Adaptive heuristics. *Econometrica*, 73(5):1401–1430, 2005.

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

Sergiu Hart and Noam Nisan. The query complexity of correlated equilibria. *Games and Economic Behavior*, 108:401–410, 2018.

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

Baihe Huang, Jason D Lee, Zhaoran Wang, and Zhuoran Yang. Towards general function approximation in zero-sum markov games. *arXiv preprint arXiv:2107.14702*, 2021.

Shinji Ito. A tight lower bound and efficient reduction for swap regret. *Advances in Neural Information Processing Systems*, 33, 2020.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4868–4878, 2018.

Chi Jin, Qinghua Liu, and Tiancheng Yu. The power of exploiter: Provable multi-agent rl in large state spaces. *arXiv preprint arXiv:2106.03352*, 2021.

Robert Kleinberg, Georgios Piliouras, and Éva Tardos. 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.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in markov potential games. *arXiv preprint arXiv:2106.01969*, 2021.

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

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

Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pp. 7001–7010. PMLR, 2021.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo. Learning parametric closed-loop policies for markov potential games. In *International Conference on Learning Representations*, 2018.

John Nash. Non-cooperative games. *Annals of mathematics*, pp. 286–295, 1951.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *arXiv preprint arXiv:1506.03271*, 2015.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

Gerasimos Palaiopanos, Ioannis Panageas, and Georgios Piliouras. Multiplicative weights update with constant step-size in congestion games: Convergence, limit cycles and chaos. *arXiv preprint arXiv:1703.01138*, 2017.

Julien Pérolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning nash equilibrium for general-sum markov games from batch data. In *Artificial Intelligence and Statistics*, pp. 232–241. PMLR, 2017.

Aviad Rubinstein. 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.

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419): 1140–1144, 2018.

Gilles Stoltz. *Incomplete information and internal regret in prediction of individual sequences*. PhD thesis, Université Paris Sud-Paris XI, 2005.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *arXiv preprint arXiv:1507.00407*, 2015.

Yi Tian, Yuanhao Wang, Tiancheng Yu, and Suvrit Sra. Online learning in unknown markov games. In *International Conference on Machine Learning*, pp. 10279–10288. PMLR, 2021.

Alexander Trott, Sunil Srinivasa, Douwe van der Wal, Sebastien Haneuse, and Stephan Zheng. Building a foundation for data-driven, interpretable, and robust policy design using the ai economist. *arXiv preprint arXiv:2108.02904*, 2021.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Jun Wang, Weinan Zhang, and Shuai Yuan. Display advertising with real-time bidding (rtb) and behavioural targeting. *arXiv preprint arXiv:1610.03013*, 2016.

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

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *arXiv preprint arXiv:2106.04895*, 2021.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5571–5580. PMLR, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Kaiqing Zhang, Sham M Kakade, Tamer Başar, and Lin F Yang. Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461*, 2020.

Runyu Zhang, Zhaolin Ren, and Na Li. Gradient play in multi-agent markov stochastic games: Stationary points and convergence. *arXiv preprint arXiv:2106.00198*, 2021.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.

A Exponential in m Lower Bound for Learning Nash in General-sum MGs

In this section, we give a sample complexity lower bound for computing approximate Nash equilibrium in one-step binary-action general-sum MGs ($H = 1$, $S = 1$ and $A_i = 2$) which has an exponential dependence in m, the number of players. The result is built on the lower bound of query complexity in Rubinstein (2016).

We use $\mathcal{G}$ to denote the one-step Markov game ($H = 1$ and $S = 1$), in which there are m players and $A = 2$ actions for each player. We index the players by $[m] = \{1, \ldots, m\}$ and denote the actions space of each player by $[A] = \{1, 2\}$. Since we restricted attention to binary-action games (i.e. $A = 2$), the total number of joint actions is 2^m.

We define a (exact) *query* as the procedure where the algorithm queries a joint action $\boldsymbol{a} \in [A]^m$ and observes the (deterministic) reward $r_i(\boldsymbol{a}) \in [0, 1]$. We define the *query complexity* (Chen et al., 2015) for learning ε-approximate Nash equilibrium (ANE) as the following.

Definition A.1 (Query complexity)**.** *The query complexity $QC_p(\mathrm{ANE}(m, \varepsilon))$ for learning ε-*ANE *is defined as the smallest n such that there exists a randomized oracle algorithm $\mathcal{A}$ satisfying the following: for any binary-action, m-player game $\mathcal{G}$, the algorithm $\mathcal{A}$ can use no more than n sequential queries of the reward to output an ε-*ANE *with probability at least $1 - p$.*

In one-step MGs with deterministic reward, the query complexity is equivalent to the sample complexity, since each query obtains a reward entry. The following result in Rubinstein (2016) gives a $2^{\Omega(m)}$ query complexity lower bound for learning ε_0-ANE in m-player binary action games.

Proposition A.2 (Corollary 4.5, (Rubinstein, 2016))**.** *There exists absolute constants $\varepsilon_0 > 0$ and $c > 0$, such that for all m,*

$$QC_p(\mathrm{ANE}(m, \varepsilon_0)) = 2^{\Omega(m)}, \text{ where } p = 2^{-cm}.$$

This result shows that it is impossible for any algorithm to learn an ε_0-ANE for every binary action game with probability at least $(1 - p)$ using $\mathrm{poly}(m, \log(1/p))$ samples: such an algorithm with $p = 2^{-cm}$ would only use $\mathrm{poly}(m, \log(2^{cm})) = \mathrm{poly}(m)$ samples, yet the sample complexity lower bound in Proposition A.2 requires at least $2^{\Omega(m)} = \exp(\Omega(m))$ samples. Since Proposition A.2 allows $\varepsilon_0 = \Theta(1)$, this also rules out the possibility of learning ε-ANE with $\mathrm{poly}(m, \log(1/p), 1/\varepsilon)$ samples for all small ε.

B Q function and Bellman equations

In general-sum Markov games, recall that we have defined the value function $V^{\pi}_{h,i}(s) : \mathcal{S} \to \mathbb{R}$ of a Markov product policy π as the expected cumulative reward for the ith player when policy π is taken

starting from state s and step h:

$$V_{h,i}^{\pi}(s) := \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'})\middle| s_h = s\right].$$

We can similarly define the Q-value function $Q_h^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as the expected cumulative rewards received by the ith player starting from state-action pair $(s, \boldsymbol{a})$ and step h:

$$Q_{h,i}^{\pi}(s, \boldsymbol{a}) := \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'})\middle| s_h = s, \boldsymbol{a}_h = \boldsymbol{a}\right].$$

For convenience of notation, we define operators $\mathbb{P}_h$ and $\mathbb{D}_{\pi}$: For any value function V and any action-value function Q,

$$[\mathbb{P}_h V](s, \boldsymbol{a}) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,\boldsymbol{a})} V(s'), \quad [\mathbb{D}_{\pi} Q](s) := \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|s)} Q(s, \boldsymbol{a}).$$

From these definitions, we have the Bellman equations

$$Q_{h,i}^{\pi}(s, \boldsymbol{a}) = (r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\pi})(s, \boldsymbol{a}), \quad V_{h,i}^{\pi}(s) = (\mathbb{D}_{\pi_h} Q_{h,i}^{\pi})(s)$$

for all $(i, s, \boldsymbol{a}, h) \in [m] \times \mathcal{S} \times \mathcal{A} \times [H]$ (where we have set $V_{H+1,i}^{\pi}(s) = 0$ for all $h, i \in [H] \times [m]$).

C Proofs for Section 3

C.1 Algorithm for learning CCE in general-sum Markov games

Our algorithm used to learn CCE in general-sum MGs is a combination of Algorithm 4 and Algorithm 2. In particular, Algorithm 4 computes a set of policies and plays these policies in each episode. Algorithm 2 used the full history in Algorithm 4 to produce a certified, general correlated policy which we will show to be a CCE (we will also use the same Algorithm 2 to produce the certified policy in the algorithm of learning CE). During the execution of Algorithm 2, if the index t is 0 at some step h, the certified policy can choose any action at and after step h.

In Algorithm 4, we choose the hyper-parameters as follows:

$$\alpha_t = \frac{H+1}{H+t}, \quad \eta_t = \sqrt{\frac{H\iota}{A_i t}}, \quad \overline{\beta}_t = c\sqrt{\frac{H^3 A_i \iota}{t}} + 2c\frac{H^2\iota}{t},$$

where $c > 0$ is some absolute constant, and $\iota = \log(\frac{m \max_{i\in[m]} A_i HSK}{p\varepsilon})$ is a log factor. The choice of η_t follows the V-OL algorithm in Tian et al. (2021) which helps to shave off an H factor in the sample complexity compared with the original Nash V-Learning algorithm in Bai et al. (2020).

Here, we have a short comment on the log factor ι. In fact, we need ι to be $C\log(\frac{m \max_{i\in[m]} A_i HSK}{p\varepsilon})$ for some absolute constant C. For the cleanness of the results, in this paper, we ignore this difference since this would not harm the correctness of all the results we present.

C.2 Proof of Theorem 3

We begin with an auxiliary lemma on α_t^j (its definition is in (3)).

Lemma C.1 (Lemma 4.1 in Jin et al. (2018))**.** *The following properties hold for α_t^j:*

1. $\frac{1}{\sqrt{t}} \le \sum_{j=1}^{t} \frac{\alpha_t^j}{\sqrt{j}} \le \frac{2}{\sqrt{t}}$ *for every* $t \ge 1$.
2. $\max_{j\in[t]} \alpha_t^j \le \frac{2H}{t}$ *and* $\sum_{j=1}^{t} \left(\alpha_t^j\right)^2 \le \frac{2H}{t}$ *for every* $t \ge 1$.
3. $\sum_{t=j}^{\infty} \alpha_t^j = 1 + \frac{1}{H}$ *for every* $j \ge 1$.
4. $\sum_{j=1}^{t} \frac{\alpha_t^j}{j} \ge \frac{1}{2t}$ *for every* $t \ge 1$.

Algorithm 4 CCE-V-Learning for General-sum MGs (i-th player's version)

Initialize: For any (s,a,h), set $\overline{V}_{h,i}(s) \leftarrow H$, $\underline{V}_{h,i}(s) \leftarrow 0$, $\overline{L}_{h,i} \leftarrow 0$, $\mu_h(a|s) \leftarrow 1/A_i$, $N_h(s) \leftarrow 0$.
for episode $k = 1, \ldots, K$ **do**
  Receive s_1.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h \sim \mu_h(\cdot|s_h)$, observe the action $\boldsymbol{a}_{h,-i}$ from the other players.
    Observe reward $r_{h,i} = r_{h,i}(s_h, a_h, \boldsymbol{a}_{h,-i})$ and next state s_{h+1} from the environment.
    Update accumulators: $t := N_h(s_h) \leftarrow N_h(s_h) + 1$.
    $\overline{V}_{h,i}(s_h) \leftarrow (1-\alpha_t)\overline{V}_{h,i}(s_h) + \alpha_t\left(r_{h,i}(s_h, a_h, \boldsymbol{a}_{h,-i}) + \overline{V}_{h+1,i}(s_{h+1}) + \overline{\beta}_t\right)$
    $\underline{V}_{h,i}(s_h) \leftarrow (1-\alpha_t)\underline{V}_{h,i}(s_h) + \alpha_t\left(r_{h,i}(s_h, a_h, \boldsymbol{a}_{h,-i}) + \underline{V}_{h+1,i}(s_{h+1}) - \overline{\beta}_t\right)$
    for all $a \in \mathcal{A}_i$ **do**
      $\overline{\ell}_{h,i}(s_h, a) \leftarrow \frac{1}{H}[H-h+1-r_{h,i}-\min\{\overline{V}_{h+1,i}(s_{h+1}), H-h\}]\mathbf{1}\{a_h = a\}/[\mu_h(a_h|s_h)+\overline{\eta}_t]$.
      $\overline{L}_{h,i}(s_h, a) \leftarrow (1-\alpha_t)\overline{L}_{h,i}(s_h, a) + \alpha_t\overline{L}_{h,i}(s_h, a)$
    Set $\mu_h(\cdot|s_h) \propto \exp[-(\overline{\eta}_t/\alpha_t)\overline{L}_{h,i}(s_h, \cdot)]$.

Property 4 above does not appear in (Jin et al., 2018), for which we provide a quick proof here:

$$\sum_{j=1}^{t} \frac{\alpha_t^j}{j} \geq \sum_{j=[t/2]}^{t} \frac{\alpha_t^j}{j} \geq 1/t \sum_{j=[t/2]}^{t} \alpha_t^j \overset{(i)}{\geq} 1/(2t) \sum_{j=1}^{t} \alpha_t^j = 1/(2t).$$

Here, (i) uses α_t^j is increasing in j for fixed t. □

We will use the following notations several times later (in this and the next section): suppose state s was visited at episodes $k^1, k^2, \ldots$ at the h-th step. Since the definition of k^j depends on the state s and step h, we will write k^j as $k_h^j(s)$ when there is a possibility of confusion. We also define $N_h^s(k)$ as the number of times the state s has been visited at h-th step before the start of k-th episode. We denote $\overline{V}^k$, $\underline{V}^k$ and μ^k for values and policies at the beginning of k-th episode; And $\boldsymbol{a}_h^k$ is the taken action at step h and episode k. We also use $\mathbb{D}_{\mu_h}[r + \mathbb{P}_h V](s)$ to denote

$$\mathbb{E}_{\boldsymbol{a}_h \sim \mu_h}[(r(s, \boldsymbol{a}_h) + \mathbb{E}_{s_{h+1} \sim \mathbb{P}_h(\cdot|s, \boldsymbol{a}_h)} V(s_{h+1})].$$

for all reward function r and value function V.

Towards proving Theorem 3, we begin with a simple consequence of the update rule in Algorithm 4, which will be used several times later.

Lemma C.2. *Let $t = N_h^k(s)$ and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the h-th step. The update rules in Algorithm 4 is equivalent to the following equations.*

$$\overline{V}_{h,i}^k(s) = \alpha_t^0 H + \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \overline{V}_{h+1,i}^{k^j}\left(s_{h+1}^{k^j}\right) + \overline{\beta}_j\right],$$

$$\underline{V}_{h,i}^k(s) = \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \underline{V}_{h+1,i}^{k^j}\left(s_{h+1}^{k^j}\right) - \overline{\beta}_j\right].$$

We next prove the following lemma which helps to explain why our choice of the bonus term is $\overline{\beta}_t$. The c in $\overline{\beta}_t$ is actually the same with the c in this lemma.

Lemma C.3. *Let $t = N_h^k(s)$ and suppose s was previously visited at episodes $k^1, \ldots, k^t < k$ at the h-th step. With probability at least $1 - \frac{p}{2}$, for any $(i, s, h, t) \in [m] \times \mathcal{S} \times [H] \times [K]$, there exist a constant c s.t.*

$$\max_{\mu} \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu \times \mu_{h,-i}^{k^j}} \left(r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H-h\}\right)(s)$$

$$-\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\right]\leq c\sqrt{H^3A_i\iota/t}+cH^2\iota/t.$$

Proof of Lemma C.3 First, we decompose

$$\max_{\mu}\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s)$$
$$-\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\right]$$

into $R^\star(i,s,h,t)+U(i,s,h,t)$ where

$$R^\star(i,s,h,t):=\max_{\mu}\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s)$$
$$-\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu_{h,i}^{k^j}\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s),$$

and

$$U(i,s,h,t):=\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu_{h,i}^{k^j}\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s)$$
$$-\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\right].$$

We first bound $U(i,s,h,t)$. Define $\mathcal{F}_j$ as the σ-algebra generated by all the random variables up to the end of the j -th episode. Then k^j are increasing stopping times w.r.t. $(\mathcal{F}_j)$. Define $\mathcal{G}_j$ as the stopped filtration $\mathcal{F}_{k^j}$. Under $\mathcal{G}_{j-1}$, by the definition of operator $\mathbb{D}$ and $\mathbb{P}$, we have

$$\mathbb{E}\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\middle|\mathcal{G}_{j-1}\right]$$
$$=\mathbb{D}_{\mu_{h,i}^{k^j}\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s)$$

So we can apply Azuma-Hoeffding inequality. Note that $\sum_{j=1}^{t}(\alpha_t^j)^2\leq 2H/t$ by Lemma C.1. Using Azuma-Hoeffding inequality, we have with probability at least $1-\frac{p}{4mHSK}$

$$U(i,s,h,t)=\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu_{h,i}^{k^j}\times\mu_{h,-i}^{k^j}}\left(r_{h,i}+\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}\right)(s)$$
$$-\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\right]$$
$$\leq\sqrt{2H^2\log(4mHSK/p)\sum_{j=1}^{t}(\alpha_t^j)^2}\leq 2\sqrt{H^3\iota/t}.$$

After taking a union bound, we have the following statement is true with probability at least $1-p/4$,

$$U(i,s,h,t)\leq 2\sqrt{H^3\iota/t}\quad\text{for all }(i,s,h,t)\in[m]\times\mathcal{S}\times[H]\times[K].$$

Then we bound $R^\star(i,s,h,t)$. For fixed (i,s,h), if we define the loss function

$$\ell_j(a)=\frac{1}{H}\mathbb{E}_{a_i=a,\boldsymbol{a}_{-i}\sim\mu_{h,-i}^{k^j}}[H-h+1-r_{h,i}(s,\boldsymbol{a})-\mathbb{P}_h\min\{\overline{V}_{h+1,i}^{k^j},H-h\}(s)]\in[0,1].$$

Algorithm 5 Correlated policy $\widehat{\pi}_h^k$ for general-sum Markov games

1: **for** step $h' = h, \ldots, H$ **do**
2: observe $s_{h'}$, and set $t \leftarrow N_{h'}^k(s_{h'})$.
3: sample $l \in [t]$ with $\mathbb{P}(l = j) = \alpha_t^j$.
4: $k \leftarrow k_t^l(s_{h'})$.
5: jointly take action $(a_{h',1}, a_{h',2}, \ldots, a_{h',m}) \sim \prod_{i=1}^m \mu_{h',i}^k(\cdot|s_{h'})$.

Then $R^\star(i,s,h,t) = H \max_\mu \sum_{j=1}^t \alpha_t^j \left\langle \mu - \mu_{h,i}^{k^j}, \ell_j \right\rangle$ becomes the weighted regret with weight α_t^j. Note that the update rule for $\mu_{h,i}^{k^j}(\cdot|s)$ is essentially performing follow the regularized leader (FTRL) algorithm with changing step size for each state s and each step h to solve an adversarial bandit problem. Lemma 17 in Bai et al. (2020)[3] bounds the weight regret with high probability. Using that lemma, we have with probability at least $1 - \frac{p}{4mHS}$,

$$R^\star(i,s,h,t) \le \frac{H\alpha_t^t \log A_i}{\eta_t} + \frac{3HA_i}{2}\sum_{j=1}^t \eta_j \alpha_t^j + H\sqrt{2\iota \sum_{j=1}^t (\alpha_t^j)^2} + \frac{H}{2}\max_{j\le t}\alpha_t^j \iota + H \max_{j \le t} \alpha_t^j \iota/\eta_t$$

simultaneously for all $t \in [K]$. By Lemma C.1 and $\eta_t = \sqrt{\frac{H\iota}{A_i t}}$, we have with probability at least $1 - \frac{p}{4mHS}$

$$\begin{aligned} R^\star(i,s,h,t) &\le 2\sqrt{H^3 A_i \iota/t} + 3\sqrt{H^3 A_i \iota/t} + 2\sqrt{H^3 \iota/t} + H^2\iota/t + 2\sqrt{H^3 A_i \iota/t} \\ &\le 10\sqrt{H^3 A_i \iota/t} + 10H^2\iota/t \quad \text{for all } t \in [K]. \end{aligned}$$

Again, taking a union bound in all $(i,s,h) \in [m] \times \mathcal{S} \times [H]$, we have with probability at least $1 - p/4$,

$$R^\star(i,s,h,t) \le 10\sqrt{H^3 A_i \iota/t} + 10H^2\iota/t \quad \text{for all } (i,s,h,t) \in [m] \times \mathcal{S} \times [H] \times [K].$$

Finally, we concluded that with probability at least $1 - p/2$, we have

$$U(i,s,h,t) + R^\star(i,s,h,t) \le c\sqrt{H^3 A_i \iota/t} + cH^2\iota/t \quad \text{for all } (i,s,h,t) \in [m] \times \mathcal{S} \times [H] \times [K].$$

for some absolute constant c. □

Suppose the output certified correlated policy in Algorithm 2 is $\widehat{\pi}$. To go on with the proof, we need to capture the characteristic of the certified correlated policy $\widehat{\pi}$. Algorithm 5 constructs a series of new policies. These $\widehat{\pi}_h^k$ is independent of the history before step h. From the construction, there's a relationship between $\widehat{\pi}$ and $\widehat{\pi}_h^k$: when players jointly play policy $\widehat{\pi}$, they first sample k from Uniform($[K]$), then they use policy $\widehat{\pi}_1^k$ according to the same sampled k together. As a result, we have the following relationship:

$$V_{1,i}^{\widehat{\pi}}(s_1) = \frac{1}{K}\sum_{k=1}^K V_{1,i}^{\widehat{\pi}_1^k}(s_1) \tag{5}$$

Lemma C.4. *We have*

$$\overline{V}_{h,i}^k(s) \ge V_{h,i}^{\dagger,\widehat{\pi}_{h,-i}^k}(s), \quad \underline{V}_{h,i}^k(s) \le V_{h,i}^{\widehat{\pi}_h^k}(s)$$

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

Proof of Lemma C.4

[3] A very similar result is Lemma G.4 in our paper. But we need Lemma 17 in Bai et al. (2020) to get the optimal H dependency.

We proof this lemma by backward induction. The claim is true for $h = H + 1$. Suppose the claim is true for $h + 1$. We begin with $V_{h,i}^{\dagger,\widehat{\pi}_{h,-i}^k}(s)$, by the definition of certified policy, letting $t = N_h^k(s)$, because $\widehat{\pi}_h^k$ which can be decomposed as a series of $\widehat{\pi}_{h+1}^{k^j}$, we have for any policy μ_i,

$$V_{h,i}^{\mu_i,\widehat{\pi}_{h,-i}^k}(s) = \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_{h,i}\times\mu_{h,-i}^{k^j}}[r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\mu_i,\widehat{\pi}_{h+1,-i}^{k^j}}](s).$$

By taking supermum, we have

$$V_{h,i}^{\dagger,\widehat{\pi}_{h,-i}^k}(s) \leq \sup_{\mu_i} \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_i\times\mu_{h,-i}^{k^j}}[r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\dagger,\widehat{\pi}_{h+1,-i}^{k^j}}](s).$$

If the inequality in Lemma C.3 is true, then we use the induction hypothesis and the inequality in Lemma C.3 to obtain

$$\begin{aligned}
V_{h,i}^{\dagger,\widehat{\pi}_{h,-i}^k}(s) &\leq \sup_{\mu_i} \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_i\times\mu_{h,-i}^{k^j}}[r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H-h\}](s) \\
&\leq \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H-h\}\right] + c\sqrt{H^3 A_i \iota/t} + cH^2\iota/t \\
&\overset{(i)}{\leq} \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H-h\} + \overline{\beta}_j\right] \\
&\leq \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \overline{V}_{h+1,i}^{k^j}\left(s_{h+1}^{k^j}\right) + \overline{\beta}_j\right] \\
&= \overline{V}_{h,i}^{k}(s).
\end{aligned}$$

Here, (i) uses our choice of $\overline{\beta}_j = c\sqrt{\frac{H^3 A_i \iota}{j}} + 2c\frac{H^2\iota}{j}$ and $\frac{1}{\sqrt{t}} \leq \sum_{j=1}^{t} \frac{\alpha_t^j}{\sqrt{j}}$, $\frac{1}{t} \leq \sum_{j=1}^{t} \frac{2\alpha_t^j}{j}$.

Meanwhile, for $\underline{V}_{h,i}^k(s)$, by the definition of certified policy and induction hypothesis,

$$\begin{aligned}
V_{h,i}^{\widehat{\pi}_h^k}(s) &= \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\widehat{\pi}^{k^j}}](s) \\
&\geq \sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \max\left\{\underline{V}_{h+1,i}^{k^j}, 0\right\}](s).
\end{aligned}$$

Then we note that $\left\{\mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \max\left\{\underline{V}_{h+1,i}^{k^j}, 0\right\}](s) - \left[r_{h,i}\left(s, a_h^{k_h^j}, \boldsymbol{a}_{h,-i}^{k_h^j}\right) + \max\{\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}), 0\}\right]\right\}_{j\geq 1}$ is a martingale difference w.r.t. the filtration $\mathcal{G}_j$, which is defined in the proof of Lemma C.3. So by Azuma-Hoeffding inequality, with probability at least $1 - \frac{p}{2mSKH}$

$$\begin{aligned}
&\sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \max\{\underline{V}_{h+1,i}^{k^j}, 0\}](s) \\
\geq &\sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k_h^j}, \boldsymbol{a}_{h,-i}^{k_h^j}\right) + \max\{\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}), 0\}\right] - 2\sqrt{\frac{H^3\iota}{t}}.
\end{aligned} \tag{6}$$

On this event, we have

$$\sum_{j=1}^{t} \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \max\left\{\underline{V}_{h+1,i}^{k^j}, 0\right\}](s)$$

$$\overset{(i)}{\geq} \sum_{j=1}^{t} \alpha_t^j \left[r_{h,i}\left(s, a_h^{k_h^j}, \boldsymbol{a}_{h,-i}^{k_h^j}\right) + \max\left\{ \underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}), 0 \right\} - \overline{\beta}_j \right]$$
$$= \underline{V}_{h,i}^k(s).$$

Here, (i) uses $\sum_{j=1}^t \alpha_t^j \beta_j \geq 2\sum_{j=1}^t \alpha_t^j \sqrt{H^3\iota/j} \geq 2\sqrt{H^3\iota/t}$.

As a result, the backward induction would work well for all h as long as the inequalities in Lemma C.3 and (6) hold for all $(i,s,h,k) \in [m] \times \mathcal{S} \times [H] \times [K]$. Taking a union bound in all $(i,s,h,k) \in [m] \times \mathcal{S} \times [H] \times [K]$, we have with probability at least $1-p/2$, the inequality in (6) is true simultaneously for all $(i,s,h,k) \in [m] \times \mathcal{S} \times [H] \times [K]$. Therefore the inequalities in Lemma C.3 and (6) hold simultaneously for all $(i,s,h,k) \in [m] \times \mathcal{S} \times [H] \times [K]$ with probability at least $1-p$. This finishes the proof of this lemma. □

Equipped with these lemmas, we are ready to prove Theorem 3.

Proof of Theorem 3 Suppose the inequalities in Lemma C.4 hold (this happens with probability at least $1-p$):

$$\overline{V}_{h,i}^k(s) \geq V_{h,i}^{\dagger,\widehat{\pi}_{h,-i}^k}(s), \quad \underline{V}_{h,i}^k(s) \leq V_{h,i}^{\widehat{\pi}_h^k}(s)$$

for all $(i,s,h,k) \in [m] \times \mathcal{S} \times [H] \times [K]$. Then, choosing $h=1$ and $s=s_1$, we have

$$V_{1,i}^{\dagger,\widehat{\pi}_{1,-i}^k}(s_1) - V_{1,i}^{\widehat{\pi}_1^k}(s_1) \leq \overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1).$$

Moreover, by (5), value function of certified policy can be decomposed as

$$V_{1,i}^{\widehat{\pi}}(s) = \frac{1}{K}\sum_{k=1}^K V_{1,i}^{\widehat{\pi}_1^k}(s), \quad V_{1,i}^{\mu_1,\widehat{\pi}_{-i}}(s_1) = \frac{1}{K}\sum_{k=1}^K V_{1,i}^{\mu_1,\widehat{\pi}_{1,-i}^k}(s_1)$$

where the decomposition is due to the first line in the Algorithm 2: sample $k \leftarrow$Uniform($[K]$).

So we have

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

To prove $\widehat{\pi}$ is an approximate CCE, we only need to bound $\sum_{k=1}^K \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \right)$. Letting $\delta_{h,i}^k := \overline{V}_{h,i}^k(s_h^k) - \underline{V}_{h,i}^k(s_h^k)$ and $t = N_h^k(s)$. Suppose s was previously visited at episodes $k^1, \ldots, k^t$ at the h-th step. By the update rule,

$$\begin{aligned} \delta_{h,i}^k &= \overline{V}_{h,i}^k(s_h^k) - \underline{V}_{h,i}^k(s_h^k) \\ &= \alpha_t^0 H + \sum_{j=1}^t \alpha_t^j [\overline{V}_{h+1,i}^k(s_{h+1}^{k^j}) - \underline{V}_{h+1,i}^k(s_{h+1}^{k^j}) + 2\overline{\beta_j}] \\ &= \alpha_t^0 H + \sum_{j=1}^t \alpha_t^j \delta_{h+1,i}^{k^j} + \sum_{j=1}^t 2\alpha_t^j \overline{\beta}_j \\ &= \alpha_t^0 H + \sum_{j=1}^t \alpha_t^j \delta_{h+1,i}^{k^j} + 2c\sum_{j=1}^t \alpha_t^j \sqrt{\frac{A_i H^3 \iota}{j}} + 4c\sum_{j=1}^t \alpha_t^j \frac{H^2\iota}{j}. \end{aligned}$$

We can use Lemma C.1 which gives $\sum_{j=1}^t \alpha_t^j/\sqrt{j} \leq 2/t$ and $\max_{j\leq t} \alpha_t^j \leq 2H/t$ to get

$$\delta_{h,i}^k \leq \alpha_t^0 H + \sum_{j=1}^t \alpha_t^j \delta_{h+1,i}^{k^j} + 4c\sqrt{H^3 A_i \iota/t} + 8cH^3\iota(1+\log t)/t,$$

where we also uses $\sum_{j=1}^{t} 1/j \leq 1 + \log t$.

Taking the summation w.r.t. k, we begin by the first two terms;

$$\sum_{k=1}^{K} \alpha_t^0 H = \sum_{k=1}^{K} H \mathbf{1}\{t=0\} = \sum_{k=1}^{K} H \mathbf{1}\left\{N_h^k(s_h^k)=0\right\} \leq SH.$$

$$\sum_{k=1}^{K} \sum_{j=1}^{N_h^k(s_h^k)} \alpha^j_{N_h^k(s_h^k)} \delta_{h+1,i}^{k_h^j(s_h^k)} \overset{(i)}{\leq} \sum_{k'=1}^{K} \delta_{h+1,i}^{k'} \sum_{j=N_h^{k'}(s_h^{k'})+1}^{\infty} \alpha_j^{N_h^{k'}(s_h^{k'})} \overset{(ii)}{\leq} (1+\frac{1}{H}) \sum_{k'=1}^{K} \delta_{h+1,i}^{k'},$$

where (i) is by changing the order of summation and (ii) is by Lemma C.1.

So

$$\sum_{k=1}^{K} \delta_{h,i}^k \leq SH + \left(1+\frac{1}{H}\right) \sum_{k=1}^{K} \delta_{h+1,i}^k + 4c\sqrt{H^3 A_i \iota} \sum_{k=1}^{K} \frac{1}{\sqrt{N_h^k(s)}} + 8cH^3\iota \sum_{k=1}^{K} \frac{1+\log N_h^k(s)}{N_h^k(s)}.$$

By pigeonhole argument,

$$\sum_{k=1}^{K} \frac{1}{\sqrt{N_h^k(s)}} = \sum_{s\in\mathcal{S}} \sum_{n=1}^{N_h^K(s)} \frac{1}{\sqrt{n}} \leq \mathcal{O}(1)\sqrt{SK}.$$

Similarly,

$$\sum_{k=1}^{K} \frac{1+\log N_h^k(s)}{N_h^k(s)} \leq \mathcal{O}(1)(1+\log(K)S(1+\log(K/S))) \leq \mathcal{O}(1)S(\iota+\iota^2) \leq \mathcal{O}(1)S\iota^2,$$

where we assume $\iota \geq 1$. So we have

$$\sum_{k=1}^{K} \delta_{h,i}^k \leq SH + \left(1+\frac{1}{H}\right) \sum_{k=1}^{K} \delta_{h+1,i}^k + \mathcal{O}(1)\sqrt{H^3 A_i SK\iota} + \mathcal{O}(1)SH^3\iota^3.$$

Recursing this argument for $h \in [H]$ gives

$$\sum_{k=1}^{K} \delta_{1,i}^k \leq eH^2 S + \mathcal{O}(1)\sqrt{H^5 A_i SK\iota} + \mathcal{O}(1)SH^4\iota^3.$$

To conclude,

$$\begin{aligned} V_{1,i}^{\dagger,\widehat{\pi}_{-i}}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1) \leq & \frac{1}{K}\sum_{k=1}^{K}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) = \frac{1}{K}\sum_{k=1}^{K}\delta_{1,i}^k \\ \leq & \mathcal{O}(1)SH^4\iota^3/K + \mathcal{O}\left(\sqrt{H^5 S \max_{i\in[m]} A_i \iota/K}\right). \end{aligned}$$

Therefore, $K \geq \mathcal{O}(\frac{H^5 S \max_{i\in[m]} A_i\iota}{\varepsilon^2} + \frac{SH^4\iota^3}{\varepsilon})$ guarantees that we have $V_{1,i}^{\dagger,\widehat{\pi}_{-i}}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1) \leq \varepsilon$ for all $i \in [m]$. This complete the proof of Theorem 3. □

D Proofs for Section 4

Before proving Theorem 6, we note that through the execution of Algorithm 1, we can record $\underline{V}_{h,i}^k(s)$ similar to Algorithm 4 which has update rule:

$$\underline{V}_{h,i}(s_h) \leftarrow (1-\alpha_t)\underline{V}_{h,i}(s_h) + \alpha_t\left(r_{h,i}(s_h,a_h,\boldsymbol{a}_{h,-i}) + \underline{V}_{h+1,i}(s_{h+1}) - \overline{\beta}_t\right). \tag{7}$$

Auxiliary information $\underline{V}_{h,i}(s_h)$ is necessary for our proof.

We restate and use several notations we introduced in the last section: suppose state s was visited at episodes $k^1, k^2, \dots$ at the h-th step. Since the definition of k^j depends on the state s and step h, we will write k^j as $k_h^j(s)$ when there is a possibility of confusion. We also define $N_h^s(k)$ as the number of times the state s has been visited at h-th step before the start of k-th episode. We denote $\overline{V}^k$, $\underline{V}^k$ and μ^k for values and policies at the beginning of k-th episode; And $\boldsymbol{a}_h^k$ is the taken action at step h and episode k. We also use $\mathbb{D}_{\mu_h}[r + \mathbb{P}_h V](s)$ to denote

$$\mathbb{E}_{\boldsymbol{a}_h \sim \mu_h}[(r(s, \boldsymbol{a}_h) + \mathbb{E}_{s_{h+1} \sim \mathbb{P}_h(\cdot|s, \boldsymbol{a}_h)} V(s_{h+1})].$$

The following lemma is the same as Lemma C.2 in the CCE case.

Lemma D.1. *Let $t = N_h^k(s)$ and suppose s was previously visited at episodes $k^1, \dots, k^t < k$ at the h-th step. The update rule for $\overline{V}_{h,i}(s_h)$ and $\underline{V}_{h,i}(s_h)$ in Algorithm 1 and (7)is equivalent to the following equations.*

$$\overline{V}_{h,i}^k(s) = \alpha_t^0 H + \sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \overline{V}_{h+1,i}^{k^j} + \overline{\beta}_j \right],$$

$$\underline{V}_{h,i}^k(s) = \sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \underline{V}_{h+1,i}^{k^j}\left(s_{h+1}^{k^j}\right) - \overline{\beta}_j \right].$$

We next prove the following lemma which helps explain our choice of the bonus term $\overline{\beta}_t$. The c in $\overline{\beta}_t$ is the same with the c in this lemma.

Lemma D.2. *Let $t = N_h^k(s)$ and suppose s was previously visited at episodes $k^1, \dots, k^t < k$ at the h-th step. With probability $1 - \frac{p}{2}$, for any $(i, s, h, t) \in [m] \times \mathcal{S} \times [H] \times [K]$, there exist a constant c s.t.*

$$\sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi_{h,s} \diamond \mu_h^{k^j}} \left(r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H - h\} \right)(s)$$
$$- \sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H - h\} \right] \le cH^2 A_i \sqrt{2\iota/t} + cH^2 A_i \iota/t.$$

Proof of Lemma D.2 First, like the prove in Lemma C.3, we decompose

$$\sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi_{h,s} \diamond \mu_h^{k^j}} \left(r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H - h\} \right)(s)$$
$$- \sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H - h\} \right]$$

into $R^\star(i, s, h, t) + U(i, s, h, t)$ where

$$R^\star(i, s, h, t) := \sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi_{h,s} \diamond \mu_h^{k^j}} \left(r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H - h\} \right)(s)$$
$$- \sum_{j=1}^t \alpha_t^j \mathbb{E}_{\boldsymbol{a}_{-i} \sim \mu_{h,-i}^{k^j}} \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{-i}\right) + \mathbb{E}_{s_{h+1} \sim \mathbb{P}_h(\cdot|s, a_h^{k^j}, \boldsymbol{a}_{-i})} \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}), H - h\} \right],$$

and

$$U(i, s, h, t) := \sum_{j=1}^t \alpha_t^j \mathbb{E}_{\boldsymbol{a}_{-i} \sim \mu_{h,-i}^{k^j}} \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{-i}\right) + \mathbb{E}_{s_{h+1} \sim \mathbb{P}_h(\cdot|s, a_h^{k^j}, \boldsymbol{a}_{-i})} \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}), H - h\} \right]$$

$$-\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k^j},\boldsymbol{a}_{h,-i}^{k^j}\right)+\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}\right].$$

We first bound $U(i,s,h,t)$. By the same reason in proof of Lemma C.3, we can apply Azuma-Hoeffding inequality. Note that $\sum_{j=1}^{t}(\alpha_t^j)^2\le 2H/t$ by Lemma C.1. Using Azuma-Hoeffding inequality, we have with probability at least $1-\frac{p}{4mHSK}$,

$$U(i,s,h,t)\le\sqrt{2H^2\log(4mHSK/p)\sum_{j=1}^{t}(\alpha_t^j)^2}\le 2\sqrt{H^3\iota/t}.$$

After taking a union bound, the following statement is true with probability at least $1-p/4$,

$$U(i,s,h,t)\le 2\sqrt{H^3\iota/t}\quad\text{for all }(i,s,h,t)\in[m]\times\mathcal{S}\times[H]\times[K].$$

Then we bound $R^\star(i,s,h,t)$. For fixed (i,s,h), we define loss function

$$\ell_j(a)=\frac{1}{H}\mathbb{E}_{a_i=a,\boldsymbol{a}_{-i}\sim\mu_{h,-i}^{k^j}}[H-h+1-r_{h,i}(s,\boldsymbol{a})-\mathbb{E}_{s_{h+1}\sim\mathbb{P}_h(\cdot|s,a,\boldsymbol{a}_{-i})}\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}),H-h\}].$$

Then we have $\ell_j(a)\in[0,1]$ for all a and the realized loss function:

$$\tilde{\ell}_j(a_h^{k^j})=\frac{1}{H}[H-h+1-r_{h,i}(s,a_h^{k^j},\boldsymbol{a}_{-i}^{k^j})-\min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}),H-h\}]\in[0,1].$$

is an unbiased estimator of $\ell_j(a_h^{k^j})$. Then $R^\star(i,s,h,t)$ can be written as

$$R^\star(i,s,h,t)=H\sup_{\phi\in\Phi_i}\sum_{j=1}^{t}\alpha_t^j[\ell_j(a_{h,i}^{k^j})-\langle\phi\diamond\mu_{h,i}^{k^j},\ell_j\rangle].$$

For fixed step h, state s, suppose s is visited at episode $k_h^1(s),\dots,k_h^t(s)$. So $q_h^b(a|s)$ and $N_h^b(s)$ is only updated at episode $k_h^1(s),\dots,k_h^t(s)$. Focusing on how $q_h^b(a|s)$ and $N_h^b(s)$ are updated when s is visited for the j-th time at step h, we can formulate the update algorithm as Algorithm 8. So we can bound $R^\star(i,s,h,t)$ using Lemma G.2.

Then by Lemma G.2, if we choose the log term as $\iota=4\log\frac{10mSAHK}{p}$, we have

$$H\sup_{\phi\in\Phi_i}\sum_{j=1}^{t}\alpha_t^j[\ell_j(a_{h,i}^{k^j})-\langle\phi_{h,s}\diamond\mu_{h,i}^{k^j},\ell_j\rangle]\le 40H^2A_i\sqrt{\iota/t}+40H^2A_i\iota/t\quad\text{for all }t\in[K]$$

with probability at least $1-p/(4mSH)$. Taking a union bound over all $(i,s,h)\in[m]\times\mathcal{S}\times[H]$, we have with probability at least $1-p/4$,

$$R^\star(i,s,h,t)\le 40H^2A_i\sqrt{\iota/t}+40H^2A_i\iota/t\quad\text{for all }(i,s,h,t)\in[m]\times\mathcal{S}\times[H]\times[K].$$

Finally, we concluded that with probability at least $1-p/2$, we have

$$U(i,s,h,t)+R^\star(i,s,h,t)\le cH^2A_i\sqrt{\iota/t}+cH^2A_i\iota/t\quad\text{for all }(i,s,h,t)\in[m]\times\mathcal{S}\times[H]\times[K].$$

for some absolute constant c. □

Suppose the output certified correlated policy in Algorithm 2 is $\widehat{\pi}$. To go on with the proof, we need to capture the characteristic of the certified correlated policy $\widehat{\pi}$. Algorithm 6 constructs a series of new policies (same as Algorithm 5). These $\widehat{\pi}_h^k$ is independent of the history before step h. From the construction, there's a relationship between $\widehat{\pi}$ and $\widehat{\pi}_h^k$: when players jointly play policy $\widehat{\pi}$, they first sample k from Uniform($[K]$), then they use policy $\widehat{\pi}_1^k$ according to the same sampled k together. As a result, we have the following relationship:

$$V_{1,i}^{\widehat{\pi}}(s)=\frac{1}{K}\sum_{k=1}^{K}V_{1,i}^{\widehat{\pi}_1^k}(s)\tag{8}$$

Algorithm 6 Correlated policy $\widehat{\pi}_h^k$ for general-sum Markov games

1: **for** step $h' = h, \ldots, H$ **do**
2: observe $s_{h'}$, and set $t \leftarrow N_{h'}^k(s_h)$.
3: sample $l \in [t]$ with $\mathbb{P}(l = j) = \alpha_t^j$.
4: $k \leftarrow k_t^l(s_{h'})$.
5: jointly take action $(a_{h',1}, a_{h',2}, \ldots, a_{h',m}) \sim \prod_{i=1}^m \mu_{h',i}^k(\cdot|s_{h'})$.

Lemma D.3. *We have*

$$\overline{V}_{h,i}^k(s) \geq \sup_{\phi \in \Phi_i} V_{h,i}^{\phi \diamond \widehat{\pi}_h^k}(s), \quad \underline{V}_{h,i}^k(s) \leq V_{h,i}^{\widehat{\pi}_h^k}(s)$$

for all $(i, k, h, s) \in m \times [K] \times [H] \times \mathcal{S}$ *with probability at least* $1 - p/2$.

Proof of Lemma D.3 We proof this lemma by backward induction. The claim is true for $h = H+1$. Suppose the claim is true for $h+1$. We begin with $\sup_{\phi \in \Phi_i} V_{h,i}^{\phi \diamond \widehat{\pi}_h^k}(s)$, by the definition of certified policy. Letting $t = N_h^k(s)$, because $\widehat{\pi}_h^k$ which can be decomposed as a series of policies, $\widehat{\pi}_{h+1}^{k^j}$, for any $\phi \in \Phi_i$, we have

$$V_{h,i}^{\phi \diamond \widehat{\pi}_h^k}(s) = \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi \diamond \mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\phi \diamond \widehat{\pi}_{h+1}^{k^j}}](s).$$

Taking supermum, we have

$$\sup_{\phi \in \Phi_i} V_{h,i}^{\phi \diamond \widehat{\pi}_h^k}(s) \leq \sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi \diamond \mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \sup_{\phi \in \Phi_i} V_{h+1,i}^{\phi \diamond \widehat{\pi}_{h+1}^{k^j}}](s)$$

If the inequality in Lemma D.2 is true, then we can use the induction hypothesis and the inequality in Lemma D.2 to obtain

$$\begin{aligned}
&\sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi \diamond \mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \sup_{\phi \in \Phi_i} V_{h+1,i}^{\phi \diamond \widehat{\pi}_{h+1}^{k^j}}](s) \\
\leq &\sup_{\phi \in \Phi_i} \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\phi \diamond \mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \min\{\overline{V}_{h+1,i}^{k^j}, H-h\}](s) \\
\leq &\sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H-h\} \right] + cH^2 A_i \sqrt{\iota/t} + cH^2 A_i \iota/t \\
\overset{(i)}{\leq} &\sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \min\{\overline{V}_{h+1,i}^{k^j}(s_{h+1}^{k^j}), H-h\} + \overline{\beta}_j \right] \\
\leq &\sum_{j=1}^t \alpha_t^j \left[r_{h,i}\left(s, a_h^{k^j}, \boldsymbol{a}_{h,-i}^{k^j}\right) + \overline{V}_{h+1,i}^{k^j} + \overline{\beta}_j \right] \\
= &\overline{V}_{h,i}^k(s).
\end{aligned}$$

Here, (i) uses our choice of $\overline{\beta}_j = cH^2 A_i \sqrt{\frac{\iota}{j}} + 2c\frac{H^2 A_i \iota}{j}$ and $\frac{1}{\sqrt{j}} \leq \sum_{j=1}^t \frac{\alpha_t^j}{\sqrt{j}}$, $\frac{1}{t} \leq \sum_{j=1}^t \frac{2\alpha_t^j}{j}$.

Meanwhile, for $\underline{V}_{h,i}^k(s)$, by the definition of certified policy and induction hypothesis,

$$\begin{aligned}
V_{h,i}^{\widehat{\pi}_h^k}(s) &= \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\widehat{\pi}_{h+1}^{k^j}}](s) \\
&\geq \sum_{j=1}^t \alpha_t^j \mathbb{D}_{\mu_h^{k^j}}[r_{h,i} + \mathbb{P}_h \max\{\underline{V}_{h+1,i}^{k^j}, 0\}](s).
\end{aligned}$$

Note that $\left\{\mathbb{D}_{\mu_h^{k^j}}[r_{h,i}+\mathbb{P}_h \max\left\{\underline{V}_{h+1,i}^{k^j},0\right\}](s)-\left[r_{h,i}\left(s,a_h^{k_h^j},\boldsymbol{a}_{h,-i}^{k_h^j}\right)+\max\{\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}),0\}\right]\right\}_{j\geq 1}$ is a martingale difference sequence w.r.t. the filtration $\mathcal{G}_j$, which is defined in the proof of Lemma C.3. So by Azuma-Hoeffding inequality, with probability at least $1-\frac{p}{2mSKH}$,

$$\begin{aligned}&\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu_h^{k^j}}[r_{h,i}+\mathbb{P}_h\max\{\underline{V}_{h+1,i}^{k^j},0\}](s)\\ \geq&\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k_h^j},\boldsymbol{a}_{h,-i}^{k_h^j}\right)+\max\{\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}),0\}\right]-2\sqrt{\frac{H^3\iota}{t}}.\end{aligned}\tag{9}$$

On this event, we have

$$\begin{aligned}&\sum_{j=1}^{t}\alpha_t^j\mathbb{D}_{\mu_h^{k^j}}[r_{h,i}+\mathbb{P}_h\max\left\{\underline{V}_{h+1,i}^{k^j},0\right\}](s)\\ \overset{(i)}{\geq}&\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k_h^j},\boldsymbol{a}_{h,-i}^{k_h^j}\right)+\max\left\{\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j}),0\right\}-\overline{\beta}_j\right]\\ \geq&\sum_{j=1}^{t}\alpha_t^j\left[r_{h,i}\left(s,a_h^{k_h^j},\boldsymbol{a}_{h,-i}^{k_h^j}\right)+\underline{V}_{h+1,i}^{k_h^j}(s_{h+1}^{k_h^j})-\overline{\beta}_j\right]\\ =&\underline{V}_{h,i}^{k}(s).\end{aligned}$$

Here, (i) uses $\sum_{j=1}^t\alpha_t^j\beta_j\geq 2\sum_{j=1}^t\alpha_t^j\sqrt{H^3\iota/j}\geq 2\sqrt{H^3\iota/t}$.

As a result, the backward induction would work well for all h as long as the inequalities in Lemma D.2 and (9) hold for all $(i,s,h,k)\in[m]\times\mathcal{S}\times[H]\times[K]$. Taking a union bound in all $(i,s,h,k)\in[m]\times\mathcal{S}\times[H]\times[K]$, we have with probability at least $1-p/2$, the inequality in (9) is true simultaneously for all $(i,s,h,k)\in[m]\times\mathcal{S}\times[H]\times[K]$. Therefore the inequalities in Lemma C.3 and (9) hold simultaneously for all $(i,s,h,k)\in[m]\times\mathcal{S}\times[H]\times[K]$ with probability at least $1-p$. This finishes the proof of this lemma.

□

Equipped with these lemmas, we are ready to prove Theorem 6.

Proof of Theorem 6 Suppose the inequalities in Lemma D.3 hold (this happens with probability at least $1-p$):

$$\overline{V}_{h,i}^{k}(s)\geq\sup_{\phi\in\Phi_i}V_{h,i}^{\phi\diamond\widehat{\pi}_h^k}(s),\quad \underline{V}_{h,i}^{k}(s)\leq V_{h,i}^{\widehat{\pi}_h^k}(s)$$

for all $(i,s,h,k)\in[m]\times\mathcal{S}\times[H]\times[K]$. Then, choosing $h=1$ and $s=s_1$, we have

$$\max_{\phi\in\Phi_i}V_{1,i}^{\phi\diamond\widehat{\pi}_1^k}(s_1)-V_{1,i}^{\widehat{\pi}_1^k}(s_1)\leq\overline{V}_{1,i}^{k}(s_1)-\underline{V}_{1,i}^{k}(s_1).$$

Moreover, by (8), value function of certified policy can be decomposed as

$$V_{1,i}^{\widehat{\pi}}(s_1)=\frac{1}{K}\sum_{k=1}^{K}V_{1,i}^{\widehat{\pi}_1^k}(s_1),\quad V_{1,i}^{\phi\diamond\widehat{\pi}}(s_1)=\frac{1}{K}\sum_{k=1}^{K}V_{1,i}^{\phi\diamond\widehat{\pi}_1^k}(s_1),$$

where the decomposition is due to the first line in the Algorithm 2: sample $k\leftarrow$Uniform($[K]$). So we have

$$\max_{\phi\in\Phi_i}V_{1,i}^{\phi\diamond\widehat{\pi}}(s_1)=\sup_{\phi\in\Phi_i}\frac{1}{K}\sum_{k=1}^{K}V_{1,i}^{\phi\diamond\widehat{\pi}_1^k}(s_1)$$

$$\leq \frac{1}{K}\sum_{k=1}^{K}\sup_{\phi\in\Phi_i} V_{1,i}^{\phi\diamond\widehat{\pi}_1^k}(s_1).$$

The we bound $\max_{\phi\in\Phi_i} V_{1,i}^{\phi\diamond\widehat{\pi}}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1)$,

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

By Lemma D.3 Letting $\delta_{h,i}^k := \overline{V}_{h,i}^k(s_h^k) - \underline{V}_{h,i}^k(s_h^k)$ and $t = N_h^k(s)$. By the update rule, we have

$$\begin{aligned}\delta_{h,i}^k &= \overline{V}_{h,i}^k(s_h^k) - \underline{V}_{h,i}^k(s_h^k)\\ &= \alpha_t^0 H + \sum_{j=1}^{t}\alpha_t^j[\overline{V}_{h+1,i}^k(s_{h+1}^{k^j}) - \underline{V}_{h+1,i}^k(s_{h+1}^{k^j}) + 2\overline{\beta_j}]\\ &= \alpha_t^0 H + \sum_{j=1}^{t}\alpha_t^j\delta_{h+1,i}^{k^j} + \sum_{j=1}^{t}2\alpha_t^j\overline{\beta}_j\\ &= \alpha_t^0 H + \sum_{j=1}^{t}\alpha_t^j\delta_{h+1,i}^{k^j} + 2cA_iH^2\sum_{j=1}^{t}\alpha_t^j\sqrt{\frac{\iota}{j}} + 4c\sum_{j=1}^{t}\alpha_t^j\frac{H^2A_i\iota}{j}.\end{aligned}$$

Taking the summation w.r.t. k, by the same argument in the proof of Theorem 3, we can get

$$\begin{aligned}\max_{\phi\in\Phi} V_{1,i}^{\phi\diamond\widehat{\pi}}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1) =& \frac{1}{K}\sum_{k=1}^{K}\left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) = \frac{1}{K}\sum_{k=1}^{K}\delta_{1,i}^k\\ \leq& \mathcal{O}(1)SH^4\max_{i\in[m]}A_i\iota^3/K + \mathcal{O}\left(\sqrt{H^6S\max_{i\in[m]}A_i^2\iota/K}\right).\end{aligned}$$

If $K \geq \mathcal{O}(\frac{H^6S\max_{i\in[m]}A_i^2\iota}{\varepsilon^2} + \frac{H^4S\max_{i\in[m]}A_i\iota^3}{\varepsilon})$, we have $\max_{\phi\in\Phi} V_{1,i}^{\phi\diamond\widehat{\pi}}(s_1) - V_{1,i}^{\widehat{\pi}}(s_1) \leq \varepsilon$ holds for all $i \in [m]$, which means $\widehat{\pi}$ is an ε-approximate CE. This completes the proof of Theorem 3. □

E Proofs for Section 5

Here, we first define pure-strategy Nash equilibrium. We say policy π is a pure strategy (deterministic policy) if and only if for any $(h, i, s) \in [H] \times [m] \times \mathcal{S}$, $\pi_{h,i}(a_i|s) = \mathbb{I}_{a_i = a'_{h,i}(s)}$ for some $a'_{h,i}(s)$. We say π a *pure-strategy Nash equilibrium* if π is a pure-strategy and is a Nash equilibrium. Similarly, we say π is a pure strategy ε-approximate Nash equilibrium if π is a pure strategy and NE-gap$(\pi) \leq \varepsilon$. Pure-strategy Nash equilibrium does not always exist in general-sum MGs, but is guaranteed to exist (as we will see) in Markov Potential Games.

E.1 Existence of pure-strategy Nash equilibria in Markov potential games

A particular property of MPGs is that, there always exists a pure-strategy Nash equilibrium. Such a property does not hold for every general-sum MG. Pure-strategy Nash equilibria are preferred in many scenarios since each player can take deterministic actions. In this section, we will focus on learning a pure-strategy Nash equilibrium of MPGs.

Proposition E.1. *For any Markov potential games, there exists a pure-strategy Nash equilibrium.*

See Theorem 3.1 in Leonardos et al. (2021) or Proposition 1 in Zhang et al. (2021) for a proof of Proposition E.1.

Algorithm 7 UCB-VI with Upper and Lower Confidence Bounds (UCBVI-UPLOW)

1: **Initialize:** For any (s, a, h, s'): $\overline{Q}_h(s,a) \leftarrow H$, $\underline{Q}_h(s,a) \leftarrow 0$, $N_h(s) = N_h(s,a) = N_h(s,a,s') \leftarrow 0$.
2: **for** episode $k = 1, \ldots, K$ **do**
3: **for** step $h = H, \ldots, 1$ **do**
4: **for** $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
5: Set $t \leftarrow N_h(s,a)$.
6: **if** $t > 0$ **then**
7: $\beta \leftarrow \textsc{Bonus}(t, \widehat{\mathbb{V}}_h[(\overline{V}_{h+1} + \underline{V}_{h+1})/2](s,a))$ (c.f. Eq. (10))
8: $\gamma \leftarrow (c/H) \cdot \widehat{\mathbb{P}}_h(\overline{V}_{h+1} - \underline{V}_{h+1})(s,a)$.
9: $\overline{Q}_h(s,a) \leftarrow \min\Big\{(r_h + \widehat{\mathbb{P}}_h \overline{V}_{h+1})(s,a) + \gamma + \beta, H\Big\}$.
10: $\underline{Q}_h(s,a) \leftarrow \max\Big\{(r_h + \widehat{\mathbb{P}}_h \underline{V}_{h+1})(s,a) - \gamma - \beta, 0\Big\}$
11: **for** $s \in \mathcal{S}$ **do**
12: $\pi_h(s) \leftarrow \arg\max_{a \in \mathcal{A}} \overline{Q}_h(s,a)$.
13: $\overline{V}_h(s) \leftarrow \overline{Q}_h(s, \pi_h(s))$; $\underline{V}_h(s) \leftarrow \underline{Q}_h(s, \pi_h(s))$.
14: Receive the initial state s_1 from the MDP.
15: **for** step $h = 1, \ldots, H$ **do**
16: Take action $a_h = \pi_h(s_h)$, observe reward r_h and next state s_{h+1}.
17: Increment $N_h(s_h)$, $N_h(s_h, a_h)$, and $N_h(s_h, a_h, s_{h+1})$ by 1.
18: $\widehat{\mathbb{P}}_h(\cdot|s_h, a_h) \leftarrow N_h(s_h, a_h, \cdot)/N_h(s_h, a_h)$.
19: Let $(\overline{V}_h^k, \underline{V}_h^k, \pi^k)$ denote the value estimates and policy at the beginning of episode k.
20: **return** Policy $\pi^{k_\star}$ where $k_\star = \arg\min_{k \in [K]} (\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1))$.

E.2 Single agent reinforcement learning and the UCBVI-UPLOW sub-routine

We first remark that although in Algorithm 3 we propose to use the UCBVI-UPLOW algorithm to search for a near optimal policy, many alternative algorithms can be used to find the near optimal policy (e.g., UCBVI (Azar et al., 2017) or Q-learning (Jin et al., 2018)). Here we choose the UCBVI-UPLOW algorithm because 1) it has a tight sample complexity bound; 2) it outputs a deterministic policy which can be used to find a *pure-strategy* approximate Nash equilibrium.

In this subsection we consider the problem of learning a near optimal policy in the fixed horizon stochastic reward RL problem $\mathrm{MDP}(H, \mathcal{S}, \mathcal{A}, \mathbb{P}, r)$. The setting is standard (c.f. Jin et al. (2018)) and is a special case of Markov games (c.f. Section 2) by setting the number of agents $m = 1$. We will use the same notations including policies and value functions as that of the Markov games as introduced in Section 2, except that we will omit the sub-scripts i's since here we just have a single agent.

We consider the UCBVI-UPLOW algorithm (Algorithm 7), which is adapted from (Xie et al., 2021; Liu et al., 2021), for learning approximate optimal policy in reinforcement learning problems. Such an algorithm is used as a sub-routine in Algorithm 3 to learn approximate pure-strategy Nash equilibria in MPGs.

In the description of Algorithm 7, the $\widehat{\mathbb{P}}_h$ quantity appeared in lines 8,9,10, and 18 can be viewed either as a set of empirical probability distributions or as an operator: for any fixed (s_h, a_h), we can view $\widehat{\mathbb{P}}_h(\cdot|s_h, a_h)$ as a probability distribution over $\mathcal{S}$; for any given function $V : \mathcal{S} \to \mathbb{R}$, we can view $\widehat{\mathbb{P}}_h$ as an operator by defining $(\widehat{\mathbb{P}}_h V)(s,a) := \mathbb{E}_{s' \sim \widehat{\mathbb{P}}_h(\cdot|s,a)}[V(s')]$. The $\widehat{\mathbb{V}}_h$ operator in line 7 is the empirical variance operator defined as $\widehat{\mathbb{V}}_h V = \widehat{\mathbb{P}}_h V^2 - (\widehat{\mathbb{P}}_h V)^2$. The BONUS function in the algorithm is chosen to be the Bernstein type bonus function

$$\textsc{Bonus}(t, \widehat{\sigma}^2) = c(\sqrt{\widehat{\sigma}^2 \iota / t} + H^2 S \iota / t). \tag{10}$$

We have the following sample complexity guarantee for the UCBVI-UPLOW algorithm returning an ε-approximate optimal policy.

Lemma E.2. *The* UCBVI-UPLOW *algorithm always returns a deterministic policy* $\pi^{k_\star}$*. Moreover, for any* $p \in (0,1]$*, letting* $\iota = \log(SAHK/p)$ *and taking the number of episodes*

$$K \geq \mathcal{O}(H^3SA\iota/\varepsilon^2 + H^3S^2A\iota^2/\varepsilon),$$

then with probability at least $1-p$*, the returned policy* $\pi^{k_\star}$ *is* ε*-approximate optimal, i.e.,* $\sup_\mu V_1^\mu(s_1) - V_1^{\pi^{k_\star}}(s_1) \leq \varepsilon$*.*

Proof of Lemma E.2 First, $\pi^{k_\star}$ is obviously deterministic from line 12 of Algorithm 7.

The sample complexity guarantee of the UCBVI-UPLOW algorithm is a consequence of the sample complexity guarantee of the Nash-VI algorithm for learning Nash in zero-sum Markov games as proved in Liu et al. (2021).

More specifically, we denote the rewards and transition matrices by $\{r_h(s,a)\}_{h\in[H]}$ and $\mathbb{P} = \{P_h(s_{h+1}|s_h,a_h)\}_{h\in[H]}$ for the $\mathrm{MDP}(H,\mathcal{S},\mathcal{A},\mathbb{P},r)$. Such a MDP can be viewed as a zero-sum Markov game $\mathrm{MG}(H,\mathcal{S},\mathcal{A},\mathcal{B},\overline{\mathbb{P}},\bar{r})$, in which $(H,\mathcal{S},\mathcal{A})$ are the same as that of the MDP, the action space for the min-player is a singleton $\mathcal{B}=1$ so that the rewards $\bar{r}_h(s,a,b) \equiv r_h(s,a)$ do not depend on the action of the min-player and the transition matrices $\overline{\mathbb{P}}_h(\cdot|s,a,b) = \mathbb{P}_h(\cdot|s,a_1)$ do not depend on the action of the min-player either. Then, for any ε-approximate Nash equilibrium (μ,ν) of the associated zero-sum Markov game, the max-player's policy μ must be ε-approximate optimal for the original MDP.

By this correspondence, the UCBVI-UPLOW algorithm is actually a specific version of the Nash-VI algorithm in Liu et al. (2021), and Line 12 in UCBVI-UPLOW is actually a specific version of line 12 in Nash-VI in Liu et al. (2021): this is because in this specific Markov game, $\overline{Q}(s,a,b)$ and $\underline{Q}(s,a,b)$ only depend on s and a, so that $\pi_h(s) = \arg\max_{a\in\mathcal{A}} \overline{Q}_h(s,a)$ is actually in the CCE set $\mathrm{CCE}(\overline{Q}_h(s,a,b), \underline{Q}_h(s,a,b))$.

By this reduction and by Theorem 4 in Liu et al. (2021), this lemma is proved. □

E.3 Proof of Theorem 8

In this subsection, we give a formal proof of Theorem 8.

Proof of Theorem 8 We use superscript t to represent variables at the t-th step (before π is updated) of the while loop.

Because we choose the log factor as $\iota = \log(\frac{mHSK\max_{i\in[m]}A_i}{\varepsilon p})$, for each execution of UCBVI-UPLOW, by Lemma E.2, it return a $\varepsilon/4$-optimal deterministic policy with probability at least $1-\frac{p\varepsilon}{8m^2H}$. Taking a union bound, we have

$$\max_{\mu_i} V_{1,i}(\mu_i,\pi^t_{-i}) - V_{1,i}(\widehat{\pi}^t_i,\pi^t_i) \leq \varepsilon/4 \tag{11}$$

simultaneously for all $i\in[m]$ and $t \leq 4mH/\varepsilon$ with probability at least $1-p/2$. For the empirical estimator $\widehat{V}^t_{1,i}$, it's bounded in $[0,H]$. Thus by Hoeffding's inequality, for fixed $i\in[m]$ and t

$$\mathbb{P}(|\widehat{V}^t_{1,i} - V^t_{1,i}| \geq \varepsilon/8) \leq 2\exp\left(-\frac{N\varepsilon^2}{32H^2}\right)$$

Choosing $N = CH^2\iota/\varepsilon^2$ for some large constant C, we have

$$\mathbb{P}(|\widehat{V}^t_{1,i} - V^t_{1,i}| \geq \varepsilon/8) \leq \frac{\varepsilon p}{16m^2H}.$$

Apply this inequality to $\widehat{V}^t_{1,i}(\widehat{\pi}^t_i,\pi^t_{-i})$ and $\widehat{V}_{1,i}(\pi^t)$ and taking a union bound, we have

$$|\widehat{V}^t_{1,i}(\widehat{\pi}^t_i,\pi^t_{-i}) - V_{1,i}(\widehat{\pi}^t_i,\pi^t_{-i})| \leq \varepsilon/8 \quad |\widehat{V}^t_{1,i}(\pi^t) - V_{1,i}(\pi^t)| \leq \varepsilon/8 \tag{12}$$

simultaneously for all $i \in [m]$ and $t \leq \frac{4mH}{\varepsilon}$ with probability at least $1 - p/2$. As a result, by (11) and (12), we have

$$\begin{aligned} \max_{\mu_i} V_{1,i}(\mu_i, \pi^t_{-i}) - V_{1,i}(\widehat{\pi}^t_i, \pi^t_i) &\leq \varepsilon/4 \\ |\widehat{V}^t_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) - V_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i})| &\leq \varepsilon/8 \\ |\widehat{V}^t_{1,i}(\pi^t) - V_{1,i}(\pi^t)| &\leq \varepsilon/8 \end{aligned} \tag{13}$$

simultaneously for all $i \in [m]$ and $t \leq 4mH/\varepsilon$ with probability at least $1 - p$. On this event,

$$\begin{aligned} \Delta^t_i &= \widehat{V}^t_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) - \widehat{V}^t_{1,i}(\pi^t) \\ &\leq V_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) - V_{1,i}(\pi^t) + \varepsilon/4. \end{aligned}$$

If the while loop doesn't end after the t-th iteration and $t \leq 4mH/\varepsilon$, there exists j^t s.t. $\Delta^t_{j^t} \geq \varepsilon/2$, so we have

$$\begin{aligned} \Phi(\pi^{t+1}) - \Phi(\pi^t) &= \Phi(\widehat{\pi}^t_{j^t}, \pi^t_{-j^t}) - \Phi(\pi^t) \\ &\overset{(i)}{=} V_{1,j^t}(\widehat{\pi}^t_{j^t}, \pi^t_{-j^t}) - V_{1,j^t}(\pi^t) \\ &\geq \Delta^t_{j^t} - \varepsilon/4 = \varepsilon/4. \end{aligned}$$

Here, (i) follows the definition of potential function. Because Φ is bounded, the while loop ends within $4\Phi_{\max}/\varepsilon \leq 4mH/\varepsilon$ steps. Therefore, (13) holds simultaneously for all $i \in [m]$ and t before the end of while loop with probability at least $1 - p$. Again, on this event, if the while loop stops at the end of t-th step, we have $\max_{i \in [m]} \Delta^t_i \leq \varepsilon/2$, then

$$\begin{aligned} \max_{\mu_i} V_{1,i}(\mu_i, \pi^t_{-i}) - V_{1,i}(\pi^t) &= \max_{\mu_i} V_{1,i}(\mu_i, \pi^t_{-i}) - V_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) + V_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) - V_{1,i}(\pi^t) \\ &\leq \varepsilon/4 + \widehat{V}^t_{1,i}(\widehat{\pi}^t_i, \pi^t_{-i}) - \widehat{V}^t_{1,i}(\pi^t) + 2\varepsilon/8 \\ &= \varepsilon/2 + \Delta^t_i \\ &\leq \varepsilon. \end{aligned}$$

So the returned policy π^t is a ε-approximate Nash equilibrium. Moreover, since UCBVI-UPLOWoutputs a pure-strategy policy and our initial policy is also a pure-strategy policy, we can conclude that with probability at least $1 - p$, within $4\Phi_{\max}/\varepsilon$ steps of the while loop, Algorithm 3 outputs an ε-approximate (pure-strategy) Nash equilibrium.

Finally, the number of episodes within each step of the while loop is

$$\begin{aligned} N + \sum_{i=1}^m (K_i + N) &= \mathcal{O}\left(\frac{H^3 S \sum_{i=1}^m A_i \iota}{\varepsilon^2} + \frac{H^3 S^2 \sum_{i=1}^m A_i \iota^2}{\varepsilon} + \frac{H^2 m \iota}{\varepsilon^2}\right) \\ &= \mathcal{O}\left(\frac{H^3 S \sum_{i=1}^m A_i \iota}{\varepsilon^2} + \frac{H^3 S^2 \sum_{i=1}^m A_i \iota^2}{\varepsilon}\right). \end{aligned}$$

So the total sample complexity (episodes) is at most

$$K = \mathcal{O}\left(\frac{\Phi_{\max} H^3 S \sum_{i=1}^m A_i \iota}{\varepsilon^3} + \frac{\Phi_{\max} H^3 S^2 \sum_{i=1}^m A_i \iota^2}{\varepsilon^2}\right).$$

□

F Sample Complexity Lower Bound of Finding Approximate Pure-Strategy Nash Equilibrium

In this section, we present an result on the sample complexity's lower bound for learning a pure-strategy Nash equilibrium in MPGs. We concentrate on Markov cooperative games (MCGs), which are also MPGs. Note that for MCGs, the potential function Φ is bounded in $[0, H]$, so by Theorem 8, the sample complexity of Nash-CA (Algorithm 3) is $\widetilde{\mathcal{O}}(\sum_{i=1}^m A_i/\varepsilon^3)$ highlighting the dependency on ε, m and $A_i,\ i = 1, \ldots, m$. We would show this $\sum_{i=1}^m A_i$ dependency is inevitable for learning ε-approximate pure-strategy Nash equilibrium in MCGs by proving an $\Omega(\sum_{i=1}^m A_i/\varepsilon^2)$ lower bound.

F.1 LOWER BOUND FOR FINDING APPROXIMATE PURE-STRATEGY NASH EQUILIBRIUM IN MCGS

We first present our main theorem:

Theorem F.1. *Suppose $A_i = 2k$, $i = 1, \ldots, m$, $H \geq 2$, $S \geq 3$ and $m \geq 4$. Then, there exists an absolute constant c_0 such that for any $\varepsilon \leq 0.4$ and any online finetuning algorithm $\mathcal{M}$ that outputs a pure-strategy policy $\widehat{\pi} = (\widehat{\pi}_1, \ldots, \widehat{\pi}_m)$, if the number of episodes*

$$K \leq c_0 \frac{H^2 \sum_{i=1}^m A_i}{\varepsilon^2} = c_0 \frac{2kmH^2}{\varepsilon^2},$$

then there exists general-sum Markov cooperative game MG on which the algorithm $\mathcal{M}$ suffers from $\varepsilon/4$-suboptimality,i.e.

$$\mathbb{E}_M \textit{NE-gap}(\widehat{\pi}) \geq \varepsilon/4,$$

where the expectation $\mathbb{E}_M$ is w.r.t. the randomness during the algorithm's execution within Markov game MG.

This theorem can be viewed as a corollary of the following lemma by a simple reduction. We would prove this theorem in the next subsection. One-step (general-sum) game is a game with only one state and one step. In a one-step game, all players choose an action simultaneously and then receive it's own reward. The Nash equilibrium and NE-gap can be defined similarly in one-step games.

Lemma F.2. *Suppose $A_i = 2k$, $i = 1, \ldots, m$ and $m \geq 4$. Then, there exists an absolute constant c_0 such that for any $\varepsilon \leq 0.4$ and any online finetuning algorithm that outputs a **pure** strategy $\widehat{\pi} = (\widehat{\pi}_1, \ldots, \widehat{\pi}_m)$, if the number of samples*

$$n \leq c_0 \frac{\sum_{i=1}^m A_i}{\varepsilon^2} = c_0 \frac{2km}{\varepsilon^2},$$

then there exists a one-step game M with stochastic reward*, on which the algorithm suffers from $\varepsilon/4$-suboptimality,i.e.*

$$\mathbb{E}_M \textit{NE-Gap}(\widehat{\pi}) = \mathbb{E}_M \max_i \left[\max_{\pi_i} u_i(\pi_i, \widehat{\pi}_{-i}) - u_i(\widehat{\pi}) \right] \geq \varepsilon/4,$$

where the expectation $\mathbb{E}_M$ is w.r.t. the randomness during the algorithm's execution within game M. $u_i(\pi)$ is the expected reward of the ith player when strategy π are taken for each player.

The proof of this lemma is also in the next subsection. In the proof, we first construct a class of one-step games which reward is Bernoulli($\frac{1}{2}$) or Bernoulli($\frac{1}{2}+\varepsilon$) depending on the taken joint-action. The proportion of joint-actions with reward Bernoulli($\frac{1}{2}+\varepsilon$) is relatively small. Most importantly, every pure-strategy ε-approximate Nash equilibrium has reward Bernoulli($\frac{1}{2}+\varepsilon$). So in order to find an ε-approximate pure-strategy Nash equilibrium, we must explore sufficient joint-actions. The number of the joint-actions with reward Bernoulli($\frac{1}{2}+\varepsilon$) can be bounded by the covering number of $[2k]^m$ under Hamming distance. Then we use KL divergence decomposition (Lemma F.5) to argue rigorously that we need to explore how many joint-actions to get an ε-approximate pure-strategy Nash equilibrium.

There's $\sum_{i=1}^m A_i$ dependency[4] in the lower bound of sample complexity for finding pure-strategy ε-approximate Nash equilibrium in MCGs. This bound is novel and improves the existing result. The existing proof in sample complexity's lower bound of Markov games (Bai & Jin (2020)) relies on an reduction from Markov games to single-agent MDPs, so the existing lower bound's dependency on $A_i,\ i = 1, \ldots, m$ is $\max_{i\in[m]} A_i$.

Here, we don't include S factor in our lower bound. The difficulty is that the NE-gap only depends on the player with the most suboptimality. For a single-agent MDP, if the player can change the policy at each state to improve the expected cumulative reward by ε, then the player can change policy at all state to improve the expected cumulative reward to the utmost extent. In general-sum Markov games, at different state, maybe different players can change the policy for this state to improve his expected cumulative reward by ε. However, the definition NE-gap only allows one player to change the policy. This difference in nature makes S and $\sum_{i=1}^m A_i$ incompatible in the lower bound.

[4]We only prove this lower bound when A_i all equal to $2k$. This case is representative.

If we consider another notion of suboptimality, i.e., changing maximum to summation:

$$\text{NE-gap}'(\pi) := \sum_{i\in[m]} \left[\sup_{\mu_i} V_{1,i}^{\mu_i,\pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1)\right].$$

This definition of NE-gap is different from the previous definition. With NE-gap$'$, if each player can change his policy to improve his expected cumulative reward by ε, the NE-gap$'$ would be at least $m\varepsilon$. Then we can define ε-approximate Nash equilibrium similarly as the policy pi such that NE-gap$'(\pi) \leq \varepsilon$. We simply point out that with this new definition of NE-gap$'$ and ε-approximate Nash equilibrium, mimicking the proof of Theorem 2 in Dann & Brunskill (2015), we can prove the sample complexity's lower bound for learning a pure-strategy ε-approximate Nash equilibrium in Markov (cooperative) games is $\Omega(H^2 S \sum_{i=1}^m A_i/\varepsilon^2)$.

F.2 Proof in lower bound

In this subsection, we complete the proof of results we proposed in last subsection. For convenience, we call the joint-action (in one-step game) which is Nash equilibrium Nash strategy. We begin with a special case of Lemma F.2, i.e. the case when $A_i = 2$ for all $i \in [m]$.

Lemma F.3. *Suppose $A_i = 2$, $i = 1, \ldots, m$ and $m \geq 4$. Then, there exists an absolute constant c_0 such that for any $\varepsilon \leq 0.4$ and any algorithm that outputs a pure strategy $\widehat{\pi} = (\widehat{\pi}_1, \ldots, \widehat{\pi}_m)$, if the number of samples*

$$n \leq c_0 \frac{2m}{\varepsilon^2},$$

then there exists a one step game M with stochastic reward on which the algorithm suffers from $\varepsilon/4$-suboptimality,i.e.,

$$\mathbb{E}_M \textit{NE-Gap}(\widehat{\pi}) = \mathbb{E}_M \max_i \left[\max_{\pi_i} u_i(\pi_i, \widehat{\pi}_{-i}) - u_i(\widehat{\pi})\right] \geq \varepsilon/4,$$

where the expectation $\mathbb{E}_M$ is w.r.t. the randomness during the algorithm's execution within game M. $u_i(\pi)$ is the expected reward of the ith player when strategy π are taken for each player.

The proof of this lemma further rely on the following lemma.

Lemma F.4. *There exists a one-step game for m players where each player has two actions. The deterministic reward is 0 or 1 and the number of joint actions that have 1 is at most $\frac{2^{m+1}}{m}$. Moreover, the only pure-strategy Nash equilibria are these joint actions which have reward 1.*

Proof of Lemma F.4 We use $r(\boldsymbol{a})$ to denote the reward of (joint) actions $\boldsymbol{a} \in \{1,2\}^m$ and define $d(\boldsymbol{a}, \boldsymbol{a}') = \#\{i : a_i \neq a_i'\}$. To ensure pure-strategy Nash equilibria must have reward 1. We need to ensure that for a $\boldsymbol{a} \in \{1,2\}^m$, there exists one $\boldsymbol{a}' \in \{1,2\}^m$ such that

$$r(\boldsymbol{a}') = 1, \quad d(\boldsymbol{a}, \boldsymbol{a}') \leq 1.$$

In other words, the set $\{\boldsymbol{a} : r(\boldsymbol{a}) = 1\}$ is a 1-net of $\{1,2\}^m$ under the distance $d(\cdot,\cdot)$. By the definition of covering number, we only need to prove

$$\mathcal{N}(\{1,2\}^m, d, 1) \leq \frac{2^{m+1}}{m}.$$

Define $K(m,1) = \mathcal{N}(\{1,2\}^m, d, 1)$. By hamming code (Hamming (1950)), we know that for any integer $k \geq 1$,

$$K(2^k - 1, 1) = 2^{2^k - k - 1}.$$

Moreover, we also have $K(n,1) \leq 2K(n-1,1)$ by adding 0 and 1 behind the 1-net of $\{0,1\}^{n-1}$. Taking largest k such that $2^k - 1 \leq n$ and iterating this construction on the Hamming code we get $K(m,1) \leq 2^{m-[\log_2(m+1)]} \leq \frac{2^{m+1}}{m}$. This ends the proof. □

The next lemma is the KL divergence lemma (Lemma 15.1 of (Lattimore & Szepesvári, 2020)]), we restate it in one-step games.

Lemma F.5 (KL divergence decomposition in one-step games)**.** *For any one-step games with stochastic reward and any algorithm. Let $X = (\boldsymbol{a}_1, r_1, \boldsymbol{a}_2, r_2, \ldots, \boldsymbol{a}_n, r_n)$, where $\boldsymbol{a}_k$ is the action (adaptively) chosen by the algorithm at the k^{th}round and r_k is the reward received at the k^{th}round after $\boldsymbol{a}_k$ is taken. $\mathbb{P}$ and $\mathbb{Q}$ are two probability measure for the stochastic reward. Let $N(\boldsymbol{a})$ be the total number of actions $\boldsymbol{a}$ in the first n rounds. Then*

$$\mathrm{KL}(X|_{\mathbb{P}} \| X|_{\mathbb{Q}}) = \sum_{\boldsymbol{a}} \mathbb{E}_{\mathbb{P}}[N(\boldsymbol{a})] \mathrm{KL}(\mathbb{P}(\cdot|\boldsymbol{a}) \| \mathbb{Q}(\cdot|\boldsymbol{a})).$$

Then we return to the proof of Lemma F.3. Suppose a game M satisfies the condition in Lemma F.4 is M, by permuting the actions of M, we get 2^m games. They all satisfy the condition in Lemma F.4. Suppose the reward of the i-th game is $r^{(i)}(\cdot)$.

We consider the following family of one-step games with stochastic reward: Let $\boldsymbol{a} = (a_1, a_2, \ldots, a_m)$.

$$\mathfrak{M}(\varepsilon) = \{\mathcal{M}^{(i)} \in \mathbb{R}^{\{1,2\}^m} \text{ with } \mathcal{M}^{(i)}(\boldsymbol{a}) = \frac{1}{2} + r^{(i)}(\boldsymbol{a})\varepsilon,\ i = 1, 2, \ldots, 2^m\},$$

where in one-step game $\mathcal{M}^{(i)}$, the reward is sampled from Bernoulli($\mathcal{M}^{(i)}(\boldsymbol{a})$) if the i-th player picks a_i and $\boldsymbol{a} = (a_1, a_2, \ldots, a_m)$. Moreover, we define $\mathcal{M}^{(0)}$ as a game whose reward is sampled from Bernoulli($\frac{1}{2}$) independent of the action.

We further let ν denote the uniform distribution on $\{1, 2, \ldots, 2^m\}$.

Proof of Lemma F.3 Fix a one-step game $\mathcal{M}^{(i)} \in \mathfrak{M}(\varepsilon)$, by Lemma F.4, it's clear that pure-strategy Nash equilibria form a set $D^{(i)} := \boldsymbol{a} \in \{\boldsymbol{a} : r^{(i)}(\boldsymbol{a}) = 1\}$. We have $\#D^{(i)} \leq 2^{m+1}/m$. For any online finetuning algorithm $\mathcal{A}$ that outputs a pure strategy $\widehat{\pi}$. Suppose $\widehat{\pi}$ means the i-th player's action is $\widehat{a}_i$ and let $\widehat{\boldsymbol{a}} = (\widehat{a}_1, \widehat{a}_2, \cdots, \widehat{a}_m)$. From the structure of the $\mathfrak{M}(\varepsilon)$, we know

$$\text{NE-Gap}(\widehat{\pi}) = \varepsilon \mathbb{P}_i(\widehat{\boldsymbol{a}} \notin D^{(i)}),$$

where $\mathbb{P}_i$ is w.r.t. the randomness of the algorithm and game $\mathcal{M}^{(i)}$. So we only need to use n, the number of samples to give a lower bound of $\mathbb{P}_i(\widehat{\boldsymbol{a}} \notin D^{(i)})$.

Since $X = \{\boldsymbol{a}_1, r_1, \cdots, \boldsymbol{a}_n, r_n\}$ is a sufficient statistics for the posterior distribution $D^{(i)}$, so we have

$$\begin{aligned}
\mathbb{E}_{i\sim\nu}\mathbb{P}_i(\widehat{\boldsymbol{a}} \notin D^{(i)}) &\geq \inf_g \mathbb{E}_{i\sim\nu}\mathbb{P}_i(g(X) \notin D^{(i)}) \\
&\overset{(i)}{\geq} \inf_g \mathbb{E}_{i\sim\nu}\mathbb{P}_0(g(X) \notin D^{(i)}) - \mathbb{E}_{i\sim\nu}\mathrm{TV}(X|_{\mathbb{P}_0}, X|_{\mathbb{P}_i}) \\
&\overset{(ii)}{\geq} \frac{1}{2} - \mathbb{E}_{i\sim\nu}\mathrm{TV}(X|_{\mathbb{P}_0}, X|_{\mathbb{P}_i}) \\
&\overset{(iii)}{\geq} \frac{1}{2} - \mathbb{E}_{i\sim\nu}\sqrt{\frac{1}{2}\mathrm{KL}(X|_{\mathbb{P}_0} \| X|_{\mathbb{P}_i})} \\
&\overset{(iv)}{=} \frac{1}{2} - \mathbb{E}_{i\sim\nu}\sqrt{\frac{1}{2}\sum_{\boldsymbol{a}} \mathbb{E}_{\mathbb{P}_0}[N(\boldsymbol{a})]\mathrm{KL}(\mathbb{P}_0(\cdot|\boldsymbol{a}) \| \mathbb{P}_i(\cdot|\boldsymbol{a}))}.
\end{aligned}$$

Above, (i) follows from the definition of total variation distance; (ii) uses the fact that under $\mathbb{P}_0$, we can't get any information about $\boldsymbol{a}^\star$, so $\mathbb{E}_{i\sim\nu}\mathbb{P}_0(g(X) \notin D^{(i)}) = 1 - \frac{|D^{(1)}|}{2^m} \geq \frac{1}{2}$; (iii) uses Pinsker's inequality; (iv) uses KL divergence decomposition (Lemma F.5).

For $\boldsymbol{a} \in D^{(i)}$, we have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_0(\cdot|\boldsymbol{a}) \| \mathbb{P}_i(\cdot|\boldsymbol{a})) =& \mathrm{KL}\left(\left(\frac{1}{2}, \frac{1}{2}\right) \| \left(\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon\right)\right) \\
=& \frac{1}{2}\log\frac{1}{1 - 4\varepsilon^2} \\
\leq& 4\varepsilon^2
\end{aligned}$$

if $\varepsilon \in (0, 0.4]$. For $\boldsymbol{a} \notin D^{(i)}$, $\mathrm{KL}(\mathbb{P}_0(\cdot|\boldsymbol{a})\|\mathbb{P}_i(\cdot|\boldsymbol{a})) = 0$. So

$$\begin{aligned}
\mathbb{E}_{i\sim\nu}\mathbb{P}_i(\widehat{\boldsymbol{a}} \notin D^{(i)}) &\geq \frac{1}{2} - \mathbb{E}_{i\sim\nu}\sqrt{2\sum_{\boldsymbol{a}\in D^{(i)}}\mathbb{E}_{\mathbb{P}_0}[N(\boldsymbol{a})]\varepsilon^2} \\
&\overset{(i)}{\geq} \frac{1}{2} - \sqrt{2\mathbb{E}_{i\sim\nu}\sum_{\boldsymbol{a}\in D^{(i)}}\mathbb{E}_{\mathbb{P}_0}[N(\boldsymbol{a})]\varepsilon^2} \\
&\overset{(ii)}{=} \frac{1}{2} - \sqrt{2\frac{n|D^{(i)}|}{2^m}\varepsilon^2} \\
&\geq \frac{1}{2} - \sqrt{4\frac{n}{m}\varepsilon^2}.
\end{aligned}$$

Above, (i) uses Jensen's inequality, (ii) uses the fact that $\mathbb{E}_{\mathbb{P}_0}[N(a)]$ is independent of i which gives

$$\begin{aligned}
\mathbb{E}_{i\sim\nu}\sum_{\boldsymbol{a}\in D^{(i)}}\mathbb{E}_{\mathbb{P}_0}[N(\boldsymbol{a})] &= \frac{1}{2^m}\sum_{\boldsymbol{a}}\mathbb{E}_{\mathbb{P}_0}[N(\boldsymbol{a})]\sum_{i:\boldsymbol{a}\in D^{(i)}}1 \\
&= \frac{1}{2^m}n|D^{(1)}|,
\end{aligned}$$

where $\sum_{i:\boldsymbol{a}\in D^{(i)}} 1 = |D^{(1)}|$ is because of the permutation. Finally, we choose $c_0 = \frac{1}{32}$, if $n \leq \frac{1}{32}\frac{2m}{\varepsilon^2}$, then

$$\mathbb{E}_{i\sim\nu}\mathbb{P}_i(\widehat{\boldsymbol{a}} \notin D^{(i)}) \geq \frac{1}{4}.$$

So there's a game instance $\mathcal{M}^{(i)}$ on which the algorithm suffer from $\varepsilon/4$ sub-optimality. □

The difference between Lemma F.3 and F.2 is the size of action space. To prove the $A_i = 2k$ case, we need to generalized F.4 to the case each player have $2k$ actions.

Lemma F.6. *For all positive integers m and k. There exists a one-step game for m players where each player has $2k$ actions. The deterministic reward is 0 or 1 and the number of joint actions that have 1 is at most $\frac{2\cdot(2k)^m}{km}$. Moreover, the only pure-strategy Nash equilibria are these joint actions which has reward 1.*

Proof of Lemma F.6 We would prove this lemma by induction. Without loss of generality, we suppose the action for each player is $1, 2, \ldots, 2k$. First, we define the hamming distance between two vectors.

$$d(\boldsymbol{x}, \boldsymbol{y}) := \#\{i : \boldsymbol{x}(i) \neq \boldsymbol{y}(i)\}.$$

The joint action space is $[2k]^m = \{1, 2, \ldots, 2k\}^m$. Suppose we have an 1-net of $[2k]^m$, D, which means for every $y \in [2k]^m$, we can find a $\boldsymbol{x} \in D$, s.t. $d(\boldsymbol{x}, \boldsymbol{y}) \leq 1$. If the reward of a game satisfy:

$$r(\boldsymbol{a}) = 1 \text{ for all } \boldsymbol{a} \in D, \quad \text{and} \quad r(\boldsymbol{a}) = 0 \text{ for all } \boldsymbol{a} \notin D.$$

Then for every pure strategy $\boldsymbol{a}$ which has reward 0, we can find another pure strategy $\boldsymbol{a}'$ s.t. $r(\boldsymbol{a}') = 1$ and $d(\boldsymbol{a}, \boldsymbol{a}') = 1$. This means that one player can change to obtain higher reward. So $\boldsymbol{a}$ is not a pure-strategy Nash equilibrium.

As a result, we can construct a game based on the 1-net. The only thing left to be verified is that the number of joint actions that have reward 1 is at most $\frac{2\cdot(2k)^m}{km}$. Note that the number of joint actions that have reward 1 is actually $|D|$, so we need to prove

$$\mathcal{N}([2k]^m, d, 1) \leq \frac{2\cdot(2k)^m}{km}, \tag{14}$$

where $\mathcal{N}(\cdot, d, \varepsilon)$ denotes the covering number.

For $k = 1$, from the proof of Lemma F.4, (14) is true. For $k > 1$, we first decompose $[2k]^m$ into k^m smaller blocks, i.e.

$$\Omega(l_1, l_2, \ldots, l_m) = \{2l_1 - 1, 2l_1\} \times \{2l_2 - 1, 2l_2\} \times \cdots, \times\{2l_m - 1, 2l_m\},$$

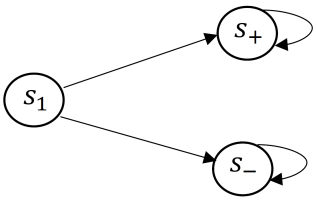

Figure 1: A class of hard-to-learn MDPs where s_+ and s_- are absorbing state.

where $\{l_1, l_2, \ldots, l_m\} \in [k]^m$. Among these k^m blocks, we choose the blocks which satisfy $k|l_1 + l_2 + \cdots + l_m$. Apparently, there are k^{m-1} blocks satisfying $m|l_1 + l_2 + \cdots + l_m$. Because (14) is true for $k = 1$, we can pick a 1-net of $\Omega(1, 1, \ldots, 1)$ with at most $\frac{2 \cdot 2^m}{m}$ elements. By a translation, (all elements add a constant vector), we can pick a 1-net of each block with at most $\frac{2 \cdot 2^m}{m}$ elements. Totally there are at most $\frac{2 \cdot 2^m k^{m-1}}{m} = \frac{2 \cdot (2k)^m}{km}$ elements. These elements form a set P. We would prove P is a 1-net of $[2k]^m$.

In fact, for any $\boldsymbol{x} \in [2k]^m$, suppose $\boldsymbol{x}$ is in $\Omega(l_1, l_2, \ldots, l_m)$. After translating this block to $\Omega(1, 1, \ldots, 1)$, $\boldsymbol{x}$ is coincided with $\boldsymbol{x_0}$.

If $\boldsymbol{x_0} \in P$, change l_1 to l_1' such that $k|l_1' + l_2 + \cdots + l_m$. Suppose $\boldsymbol{x}'$ is the vector in block $\Omega(l_1', l_2, \ldots, l_m)$ that corresponds to $\boldsymbol{x}_0$ in $\Omega(1, 1, \ldots, 1)$. This gives $\boldsymbol{x}' \in P$. Moreover, $d(\boldsymbol{x}, \boldsymbol{x}') \leq 1$ because the translation from $\Omega(l_1, l_2, \ldots, l_m)$ to $\Omega(l_1', l_2, \ldots, l_m)$ only changes the first component.

If $\boldsymbol{x}_0 \notin P$, we can find $\boldsymbol{x}_0'$ in $P \cap \Omega(1, 1, \ldots, 1)$ such that $d(\boldsymbol{x}_0, \boldsymbol{x}_0') = 1$ because $P|_{\Omega(1,1,\ldots,1)}$ is a 1-net. Suppose $\boldsymbol{x_0}$ and $\boldsymbol{x_0'}$ differs at the j-th component. We can change l_j to l_j' s.t. $k|l_1 + l_2 + \cdots + l_{j-1} + l_j' + l_{j+1} + \cdots + l_m$. Suppose $\boldsymbol{x_0'}$ corresponds to $\boldsymbol{x}'$ in $\Omega(l_1, \ldots, l_m)$ and $\boldsymbol{x}''$ in $\Omega(l_1, \ldots, l_{j-1}, l_j', l_{j+1}, \ldots, l_m)$. By the definition of P, we know $\boldsymbol{x}'' \in P$. Moreover, $\boldsymbol{x}$ and $\boldsymbol{x}'$ only differ at the j-th component because of translation and the fact that $\boldsymbol{x_0}$ and $\boldsymbol{x_0'}$ only differ at the j-th component. $\boldsymbol{x}'$ and $\boldsymbol{x}''$ also only differ at the j-th component because of translation. So we have $\boldsymbol{x}$ and $\boldsymbol{x''}$ may only differ at the j-th component, i.e. $d(\boldsymbol{x}, \boldsymbol{x}'') \leq 1$.

Recall that $\boldsymbol{x}'' \in P$. To conclude, we can always find another vector $\boldsymbol{y} \in P$ such that $d(\boldsymbol{x}, \boldsymbol{y}) \leq 1$, this means P is a 1-net of $[2k]^m$. So (14) is proved. □

Using this fundamental lemma, by the argument in proof of Lemma F.3, we can easily prove Lemma F.2. So here we omit the proof.

Finally, we give a proof of Theorem F.1 based on Lemma F.2.

Proof of Theorem F.1 Because $S \geq 3$, we construct a class of general-sum Markov games as follows (see figure 1): the set of all joint-actions $\mathcal{A}$ can be divided into two sets $\mathcal{A}^+$ an $\mathcal{A}^-$.

Transition Starting from s_1, for $\boldsymbol{a} \in \mathcal{A}^+$, $\mathbb{P}_1(s_+|s_1, \boldsymbol{a}) = \frac{1}{2} + \frac{\varepsilon}{2(H-1)}$ and $\mathbb{P}_1(s_-|s_1, \boldsymbol{a}) = \frac{1}{2} - \frac{\varepsilon}{2(H-1)}$. For $\boldsymbol{a} \in \mathcal{A}^-$, $\mathbb{P}_1(s_+|s_1, \boldsymbol{a}) = \mathbb{P}_1(s_-|s_1, \boldsymbol{a}) = \frac{1}{2}$. For all $h \geq 1$ and $\boldsymbol{a} \in \mathcal{A}$, $\mathbb{P}_h(s_+|s_+, \boldsymbol{a}) = 1$ and $\mathbb{P}_h(s_-|s_-, \boldsymbol{a}) = 1$.

Rewards For any $\boldsymbol{a}$ and $i \in [m]$, $r_{1,i}(s, \boldsymbol{a}) = 0$ $r_{h,i}(s_+, \boldsymbol{a}) = 1$ and $r_{h,i}(s_-, \boldsymbol{a}) = 0$, where $h > 1$. This also implies the Markov game is cooperative.

If we choose $\mathcal{A}^+$ as the joint action with reward Bernoulli$(\frac{1}{2} + \varepsilon)$ in $\mathcal{M}^{(i)}$. By the construction, for any pure-strategy policy π, if the joint actions taken at step 1 is $\boldsymbol{a} \notin \mathcal{A}^+$, we have NE-gap$(\pi) \geq \varepsilon$. The only useful information in each episode is the transition at step 1. Transition from s_1 to s_+ can be viewed as getting a 1 reward and transition from s_1 to s_- can be viewed as getting a 0 reward. So learning pure-strategy Nash equilibrium of this class of Markov games is equivalent to learning pure-strategy Nash equilibrium of a game for m players with stochastic reward. As a result, by

Algorithm 8 Mixed-expert FTRL for weighted adversarial bandits

1: **Input**: Sequence $\left\{\alpha_t^i\right\}_{1\le i\le t<\infty}$.
2: **Initialize**: Accumulators $t_{b'} \leftarrow 0$ for all $b' \in [A]$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Compute weight $u_t \leftarrow \alpha_t^t / \alpha_t^1$.
5: Compute action distributions for all sub-experts $b' \in [A]$ by FTRL:

$$q^{b'}(a) \propto_a \exp\left(-(\eta_{t_{b'}}/u_t) \cdot \sum_{\tau=1}^{t_{b'}} w_\tau(b') \widehat{\ell}_\tau^{b'}(a)\right).$$

6: Compute $p^t \in \Delta_{[A]}$ by solving the linear system $p^t(\cdot) = \sum_{b'=1}^{A} p^t(b') q^{b'}(\cdot)$.
7: Sample sub-expert $b \sim p^t$.
8: Sample action $a^t \sim q^b$ from sub-expert b.
9: Play the action a^t, and observe the (bandit-feedback) loss $\tilde{\ell}_t(a^t)$.
10: Update accumulator for sub-expert b: $t_b \leftarrow t_b + 1$.
11: Compute the loss estimate $\widehat{\ell}_{t_b}^b(\cdot) \in \mathbb{R}_{\ge 0}^A$ as

$$\widehat{\ell}_{t_b}^b(a) \leftarrow \frac{\tilde{\ell}_{t_b}(a) \mathbf{1}\{a = a^t\}}{q^b(a) + \gamma_{t_b}}.$$

12: Set $w_{t_b}(b) \leftarrow u_t$.

Lemma F.2, if the number of episodes

$$K \le c_0 \frac{\sum_{i=1}^m A_i}{\varepsilon^2/(H-1)^2} \le 4c_0 \frac{H^2 \sum_{i=1}^m A_i}{\varepsilon^2},$$

then there exists general-sum Markov cooperative game MG on which the algorithm suffers from $\varepsilon/4$-suboptimality,

$$\mathbb{E}_M \text{NE-gap}(\widehat{\pi}) \ge \varepsilon/4.$$

This finish the proof of Theorem F.1. □

G Weighted Adversarial Bandit with Low Swap Regret

This section is for Algorithm 8. We suppose the action space is $[A]$. An strategy modification is a function $F : [A] \to [A]$ and for any action distribution μ, $F \diamond \mu$ gives the swap distribution which takes action $F(a)$ with probability $\mu(a)$.

Throughout this section, we assume that the realized loss values $\tilde{\ell}_t \in [0,1]^A$, and let $\ell_t(a) := \mathbb{E}_t[\tilde{\ell}_t(a)]$ where $\mathbb{E}_t$ denotes the expectation w.r.t. the randomness in sampling a^t(Line 7 and Line 8 in Algorithm 8) at the t-th episode.

The swap regret (Blum & Mansour, 2007; Ito, 2020) measures the difference between the cumulative realized loss for the algorithm and that for swapped action sequences generated by an arbitrary strategy modification F. Here, we consider the weighted swap regret which attaches different weights at different episodes. It can be defined as

$$R_{\text{swap}}(t) := \max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(a^i))]. \tag{15}$$

Besides swap regret, we also define $R(t)$ as

$$R(t) := \max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \langle F(p^i), \ell_i \rangle], \tag{16}$$

where p^t is the distribution from which we sample the sub-expert at the t-th episode. We need to bound $R(t)$ in the proof of Lemma D.2. The following of this section is organized as follows: we first start with some properties of Algorithm 8. Then we bound (15) and (16) in Lemma G.1 and Lemma G.2 respectively. Finally, we end with auxiliary lemmas for weighted adversarial bandit with predictable weights.

Solution of the linear system In line 6, we need to compute $p^t \in \Delta_{[A]}$ by solving the linear system $p^t(\cdot) = \sum_{b'=1}^{A} p^t(b') q^{b'}(\cdot)$. Such $p^t(\cdot)$ can be interpreted as the stationary distribution of a Markov chain over $[A]$ with transition probability $\mathbb{P}(b|a) = q^a(b)$. This guarantee the existence of $p^t \in \Delta_{[A]}$ such that $p^t(\cdot) = \sum_{b'=1}^{A} p^t(b') q^{b'}(\cdot)$. Moreover, $p^t(\cdot)$ can be computed efficiently (see e.g. Cohen et al. (2017b)).

FTRL performance of the sub-expert We can focus on the action distribution q^b of sub-expert b. When b is sampled in line 7 at episode t. We have accumulator t_b and weight $w_{t_b}(b) = u_t = \alpha_t^t / \alpha_t^1$ at the end of episode t. Action a^t is chosen from distribution

$$q^b(a) \propto_a \exp\left(-(\eta_{t_b} / w_{t_b}) \cdot \sum_{\tau=1}^{t_b - 1} w_\tau(b') \widehat{\ell}_\tau^b(a) \right)$$

after $w_{t_b}(b) = u_t = \alpha_t^t / \alpha_t^1$ is observed. The loss estimate $\widehat{\ell}_{t_b}$ is

$$\widehat{\ell}_{t_b}^b(a) \leftarrow \frac{\tilde{\ell}_{t_b}(a) \mathbf{1}\{a = a^t\}}{q^b(a) + \gamma_{t_b}}.$$

So from the sub-expert b's perspective, he is performing FTRL (follow the regularized leader) with changing step size and random weight (We summarize this in Algorithm 9). Suppose the sub-expert b is chosen at episode $t = k_1, k_2, \ldots, k_{t_b}$, then the weighted regret for sub-expert b becomes

$$R_t(b) := \sum_{\tau=1}^{t_b} w_\tau(b) (\ell_{k_\tau}(a^{k_\tau}) - \langle \theta^*, \ell_{k_\tau} \rangle). \tag{17}$$

Bounded weight $w_i(b)$ This kind of weight $w_{t_b}(b) = \frac{\alpha_t^t}{\alpha_t^1}$ is designed delicately to ensure the desired α_t^i weight in the swap regret:

$$R_{\text{swap}}(t) = \max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(a^i))].$$

We would explain this in detail in the proof of Lemma G.1. Moreover, because

$$\frac{\alpha_t^t}{\alpha_t^1} \leq \frac{1}{\alpha_t^1} = \frac{1}{\alpha_1(1-\alpha_2)\cdots(1-\alpha_t)} = \frac{(H+2)(H+3)\cdots(H+t)}{1 \cdot 2 \cdots (t-1)} \leq (H+t)^{H+1}.$$

So

$$\alpha_t^t / \alpha_t^1 \leq (H+T)^{2H} \equiv W. \tag{18}$$

The weight $w_{n_b}(b) = \alpha_t^t / \alpha_t^1$ have a (non-random) upper bound W.

Choice of hyperparameter We choose the parameter

$$\eta_t = \gamma_t = \sqrt{\frac{\iota}{At}},$$

where the log factor

$$\iota = 4 \log \frac{8HAT}{p}. \tag{19}$$

G.1 MAIN RESULT FOR MINIMIZING WEIGHTED SWAP REGRET

In this subsection, we would state and prove the main result for bounding (15) and (16) with high probability.

Lemma G.1 (Bound on weighted swap regret)**.** *If we execute Algorithm 8 for T rounds, then with probability at least $1-p/2$, we have the following bound on the swap regret simultaneously for all $t \in [T]$:*

$$\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(a^i))] \le 35(HA\sqrt{\iota/t} + HA\iota/t).$$

Proof of Lemma G.1 We use b^i to denote the sampled sub-expert b at the i-th episode. Define $\mathcal{G}_i$ as the σ-algebra generated by all the random variables observed up to the end of the i-th episode.

First observe that

$$\begin{aligned}
&\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(a^i))] \\
\le &\underbrace{\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(b^i))]}_{\mathrm{I}} + \underbrace{\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(F(b^i)) - \ell_i(F(a^i))]}_{\mathrm{II}}.
\end{aligned}$$

We first bound term II by concentration. Note that at the i-th episode, as p^i obtained in Line 6 solves the equation

$$p^i(a) = \sum_{b'=1}^{A} p^i(b') q^{b'}(a'),$$

we have $b^i \sim p^i(\cdot)$ and $a^i \sim q^{b^i}(\cdot)$ has the same marginal distribution conditioned on $\mathcal{G}_{i-1}$, where $\mathcal{G}_{i-1}$ denote the σ-algebra generated by ℓ_i and all the random variables in the first $i-1$ episodes. Therefore, fixing any strategy modification $F : [A] \to [A]$, we have $\left\{\alpha_t^i[\ell_i(F(b^i)) - \ell_i(F(a^i))]\right\}_{1\le i\le t}$ is a bounded martingale difference sequence w.r.t. the filtration $\{\mathcal{G}_i\}$, so by Azuma-Hoeffding inequality and $\sum_{i=1}^{t}(\alpha_t^j)^2 \le 2H/t$,

$$\mathbb{P}\left(\sum_{i=1}^{t} \alpha_t^i [\ell_i(F(b^i)) - \ell_i(F(a^i))] \ge 2\sqrt{\frac{H\log 1/p}{t}}\right) \le p.$$

As there is at most A^A strategy modifications, we can substitute p with $p/(4A^A T)$, and take a union bound to get

$$\mathrm{II} = \max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i [\ell_i(F(b^i)) - \ell_i(F(a^i))] \le 2\sqrt{HA\log(4AT/p)/t} \le 2\sqrt{HA\iota/t} \tag{20}$$

simultaneously for all $t \in [T]$ with probability at least $1 - p/4$.

We next bound term I. Define $\mathcal{U}_b := \left\{i \in [t] : b^i = b\right\}$ and let n_b^t be the value of t_b at the end of the t-th episode, i.e. $n_b^t = \#\left\{i : b^i = b\right\}$. We also suppose the sub-expert b was chosen at episode $t = k_1^b, k_2^b, \ldots, k_{n_b^t}^b$ up to episode t. Then we have

$$\begin{aligned}
\sum_{i=1}^{t} \alpha_t^i [\ell_i(a^i) - \ell_i(F(b^i))] &= \sum_{b\in\mathcal{A}} \sum_{i\in\mathcal{U}_b} \alpha_t^i [\ell_i(a^i) - \ell_i(F(b))] \\
&= \sum_{b\in\mathcal{A}} \sum_{\tau=1}^{n_b^t} \alpha_t^1 \frac{\alpha_t^{k_\tau^b}}{\alpha_t^1} [\ell_{k_\tau^b}(a^{k_\tau^b}) - \ell_{k_\tau^b}(F(b))]
\end{aligned}$$

$$= \sum_{b\in\mathcal{A}} \sum_{\tau=1}^{n_b^t} \alpha_t^1 w_\tau(b)[\ell_{k_\tau^b}(a^{k_\tau^b}) - \ell_{k_\tau^b}(F(b))].$$

Here the last equation is because our choice of $w_\tau(a) = \alpha_{k_\tau^b}^{k_\tau^b}/\alpha_{k_\tau^b}^1 = \alpha_t^{k_\tau^b}/\alpha_t^1$ which is a simple corollary from the definition of α_t^i in Eq. (3).

As we have seen in (17), the term

$$\sum_{\tau=1}^{n_b^t} w_\tau(b)[\ell_{k_\tau^b}(a^{k_\tau^b}) - \ell_{k_\tau^b}(F(b))],$$

is exactly the weighted regret $R_t(b)$ for sub-expert b. Because

$$R_t(b) := \sum_{\tau=1}^{n_b^t} w_\tau(b)[\ell_{k_\tau^b}(a^{k_\tau^b}) - \ell_{k_\tau^b}(F(b))] \le \max_{\theta^*} \sum_{\tau=1}^{n_b^t} w_\tau(b)[\ell_{k_\tau^b}(a^{k_\tau^b}) - \langle \ell_{k_\tau^b}, \theta^* \rangle],$$

we can use our result on weighted adversarial bandits with predictable weights (Lemma G.3) to bound this term (the upper bound W of the weight $w_\tau(b)$ can be taken as $(H+T)^{2H}$ by the calculation of (18)). Recall that our choice of log term is $\iota = 4\log\frac{4HAT}{p} = \log\frac{AT\lceil\log_2 W\rceil}{p'}$ where $p' \le p/(4A)$. Thus by Lemma G.3, with probability at least $1 - p/(4A)$,

$$R_t(b) \le 15 \max_{\tau\le n_b^t} w_\tau(b)\left[\sqrt{An_b^t\iota} + \iota\right] \quad \text{for all } t \in [T].$$

Taking a union bound, we get

$$R_t(b) \le 15 \max_{\tau\le n_b^t} w_\tau(b)\left[\sqrt{An_b^t\iota} + \iota\right] \quad \text{for all } (t,b) \in [T]\times[A] \tag{21}$$

with probability at least $1 - p/4$.

On this event, for all $F : [A] \to [A]$, we have

$$\begin{aligned}
\sum_{i=1}^t \alpha_t^i[\ell_i(a^i) - \ell_i(F(b^i))] &= \sum_{b\in\mathcal{A}} \sum_{\tau=1}^{n_b^t} \alpha_t^1 w_\tau(b)[\ell_{k_\tau^b}(a^{k_\tau^b}) - \ell_{k_\tau^b}(F(b))] \\
&= \sum_{b\in\mathcal{A}} \alpha_t^1 R_t(b) \\
&\overset{(i)}{\le} \sum_{b\in\mathcal{A}} \alpha_t^1 \max_{\tau\le n_b^t} w_\tau(a)(15\sqrt{An_b^t\iota} + 15\iota) \\
&\overset{(ii)}{=} \sum_{b\in\mathcal{A}} \alpha_t^1 \frac{\alpha_t^{k_{n_b^t}^b}}{\alpha_t^1}(15\sqrt{An_b^t\iota} + 15\iota) \\
&\overset{(iii)}{\le} \sum_{a\in\mathcal{A}} 2H/t(15\sqrt{An_b^t\iota} + 15\iota).
\end{aligned}$$

Here, (i) uses (21), (ii) uses the $\alpha_t^{t'}$ is increasing w.r.t. t' and (iii) uses $\max_{0\le t'\le t}\alpha_t^{t'} \le 2H/t$. Finally, because $\sum_{b\in\mathcal{A}} n_b^t = t$, by the concavity of $x \mapsto \sqrt{x}$, we have with probability at least $1 - p/4$,

$$\begin{aligned}
\mathrm{I} = \max_{F:[A]\to[A]} \sum_{i=1}^t \alpha_t^i[\ell_i(a^i) - \ell_i(F(b^i))] &\le 30H/t\left(\sum_{b\in\mathcal{A}}(\sqrt{An_b^t\iota} + \iota)\right) \\
&\le 30H/t(A\sqrt{t\iota} + A\iota) \\
&= 30HA\sqrt{\iota/t} + 30HA\iota/t.
\end{aligned}$$

Combining this with (20), we finish the proof. □

Lemma G.2. *If we execute Algorithm 8 for T rounds, then with probability at least $1-p$, we have the following bound simultaneously for all $t \in [T]$:*

$$\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(a^i) - \langle F \diamond p^i, \ell_i\rangle] \le 40(HA\sqrt{\iota/t} + HA\iota/t).$$

Proof of Lemma G.2 Similar to the proof of Lemma G.1, we define $\mathcal{G}_{i-1}$ as the σ-algebra generated by ℓ_i and all the random variables in the first $i-1$ episodes. Note that for any fixed strategy modification $F : [A] \to [A]$, we have that $\left\{\alpha_t^i[\ell_i(F(a^i)) - \left\langle F \diamond p^i, \ell_i\right\rangle]\right\}_{1\le t\le T}$ is a bounded martingale difference sequence w.r.t. the filtration $\{\mathcal{G}_i\}$. So by Azuma-Hoeffding and $\sum_{i=1}^{t}(\alpha_t^j)^2 \le 2H/t$, we have

$$\mathbb{P}\left(\sum_{i=1}^{t} \alpha_t^i[\ell_i(F(a^i)) - \left\langle F \diamond p^i, \ell_i\right\rangle] \ge 2\sqrt{\frac{H\log 1/p}{t}}\right) \le p.$$

The number of strategy modifications is A^A, so we can substitute p with $p/(4A^AT)$, and take a union bound to get, with probability at least $1-p/2$,

$$\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(F(a^i)) - \left\langle F \diamond p^i, \ell_i\right\rangle] \le 2\sqrt{HA\log(4AT/p)/t} \le 2\sqrt{HA\iota/t} \quad \text{for all } t \in [T].$$

By Lemma G.1, with probability at least $1-p/2$, we have

$$\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(a^i) - \ell_i(F(a^i))] \le 35(HA\sqrt{\iota/t} + HA\iota/t) \quad \text{for all } t \in [T].$$

So we conclude that with probability at least $1-p$, for all $t \in [T]$, we have

$$\begin{aligned}
&\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(a^i) - \left\langle F \diamond p^i, \ell_i\right\rangle] \\
\le\ &\max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(a^i) - \ell_i(F(a^i))] + \max_{F:[A]\to[A]} \sum_{i=1}^{t} \alpha_t^i[\ell_i(F(a^i)) - \left\langle F \diamond p^i, \ell_i\right\rangle] \\
\le\ &40(HA\sqrt{\iota/t} + HA\iota/t).
\end{aligned}$$

This proves the lemma. □

G.2 Auxiliary lemmas for weighted adversarial bandit with predictable weights

In this subsection, we consider the Follow the Regularized Leader (FTRL) algorithm with

1. changing step size,
2. weighted regret with $\mathcal{F}_{t-1}$-measurable weights and loss distributions,
3. high probability regret bound.

We present these results because the predictable weights we would use are potentially unbounded from above; if weights are predictable and also bounded, then there may be an easier analysis.

Interaction protocol We first describe the interaction protocol between the environment and the player for this problem. At each episode t, the environment adversarially choose a weight w_t which takes values in $\mathbb{R}_{>0}$, and distributions $(\mathcal{L}_{at})_{a\in[A]}$, where each $\mathcal{L}_{at}$ is a distribution that takes values in $\mathcal{P}([0,1])$ (the space of distributions supported on $[0,1]$). Then the player receives the weight w_t, chooses an action a_t, and observes a loss $\tilde{\ell}_t(a_t) \sim \mathcal{L}_{a_t,t}$. The action chosen a_t can be based

Algorithm 9 FTRL for Weighted Regret with Changing Step Size and Predictable Weights

1: **for** episode $t = 1, \ldots, T$ **do**
2: Observe $w_t > 0$.
3: $\theta_t(a) \propto \exp(-\eta_t / w_t \sum_{i=1}^{t-1} w_i \widehat{\ell}_i(a))$.
4: Take action $a_t \sim \theta_t(\cdot)$, and observe loss $\tilde{\ell}_t(a_t)$.
5: $\widehat{\ell}_t(a) \leftarrow \tilde{\ell}_t(a) \mathbf{1}\{a_t = a\} / (\theta_t(a) + \gamma_t)$.

on all the information in $\mathcal{D}_t \equiv \{(a_i, w_i, \tilde{\ell}_i(a_i))_{i \le t-1}, w_t\}$. Then, the environment can observe the player's action a_t and the incurred loss realization $\tilde{\ell}_t(a_t)$, and use these information and some external randomness z_t to choose the weight and distributions $(w_{t+1}, (\mathcal{L}_{a,t+1})_{a \in [A]})$ of the next episode. We will consider the following variant of FTRL algorithm for the player.

We denote $\ell_t \equiv (\mathbb{E}_{\tilde{\ell} \sim \mathcal{L}_{at}}[\tilde{\ell}])_{a \in [A]} \in [0,1]^A$ which is a random vector but is $\sigma((\mathcal{L}_{at})_{a \in [A]})$-measurable. For some $\theta^* \in \mathbb{R}^A$, the regret of the player up to episode t is defined as

$$R_t \equiv \sum_{i=1}^{t} w_i (\ell_i(a_i) - \langle \theta^*, \ell_i \rangle).$$

Let $(z_t)_{t \ge 1}$ be a sequence of external random variables which are identically and independently distributed as Unif$([0,1])$ and are independent of all other random variables. We define $\mathcal{F}_t = \sigma(\{a_i, \tilde{\ell}_i, z_i\}_{i \le t})$ to be the sigma algebra generated by the random chosen action a_i, the random loss $\tilde{\ell}_i(a_i)$, and the external random variables z_i by episode t. We assume that the random weights and the random loss distributions $(w_t, (\mathcal{L}_{at})_{a \in [A]})_{t \ge 1}$ are a $(\mathcal{F}_t)_{t \ge 1}$-*predictable sequence*, in the sense that $(w_t, (\mathcal{L}_{at})_{a \in [A]})$ is $\mathcal{F}_{t-1}$-measurable. Then $\mathcal{F}_t$ contains all information (random chosen action, random observed loss, random weight, and random loss distributions) before action a_{t+1} is taken.

We assume the predictable sequence $(w_t)_{1 \le t \le T}$ have a global (non-random) upper bound W. Then we define the log term $\iota = \log(AT \lceil \log_2 W \rceil / p)$. We set

$$\eta_t = \gamma_t = \sqrt{\frac{\iota}{At}}.$$

G.2.1 Regret bound

In the following, we consider to give a high probability weighted regret bound for Algorithm 9.

Lemma G.3. *Let $(w_t, (\mathcal{L}_{at})_{a \in [A]})_{t \ge 1}$ be any $\mathcal{F}_t$-predictable sequence satisfying $1 \le \min_{i \le t} w_i \le \max_{i \le t} w_i \le W$ for some constant (non-random) $W > 0$ almost surely. Then, following Algorithm 9, with probability at least $1 - 4p$, for any $\theta^* \in \Delta^A$ and $t \le T$ we have*

$$\sum_{i=1}^{t} w_i (\ell_i(a^i) - \langle \theta^*, \ell_i \rangle) \le 15 \max_{i \le t} w_i \cdot \left[\sqrt{At\iota} + \iota \right],$$

where $\iota = \log(AT \lceil \log_2 W \rceil / p)$.

Proof. This lemma follows from the bound in Lemma G.4 and a concentration step that we establish below.

Define $M = \lceil \log_2 W \rceil$, and $w_i(k) = w_i \mathbf{1}\left\{w_i \le 2^k\right\}$. Then $w_i(k)$ is also $\mathcal{F}_{i-1}$ measurable. We Consider sequence $\left\{w_i(k)(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle)\right\}_{i \ge 1}$. Since w_i is $\mathcal{F}_{i-1}$-measurable, $\mathbb{E}[w_i(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) | \mathcal{F}_{i-1}] = 0$, which means $\left\{w_i(k)(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle)\right\}_{i \ge 1}$ is a martingale difference sequence w.r.t. filtration $(\mathcal{F}_i)$. So by Azuma-Hoeffding inequality,

$$\mathbb{P}\left(\sum_{i=1}^{t} w_i(k)(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \le \sqrt{2 \cdot \iota t (2^k)^2} \right) \ge 1 - \frac{p}{TM}.$$

Taking a union bound, we have

$$\mathbb{P}\left(\forall k \in [M], \sum_{i=1}^{t} w_i(k)(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2^k \sqrt{2t\iota}\right) \geq 1 - p/T. \tag{22}$$

Denote $k' = \lceil \log_2 \max_{i \leq t} w_i \rceil$, then we have

$$\begin{aligned}
&\left\{\forall k \in [M], \sum_{i=1}^{t} w_i(k)(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2^k \sqrt{2t\iota}\right\} \\
\subseteq &\left\{\sum_{i=1}^{t} w_i(k')(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2^k \sqrt{2t\iota}\right\} \subseteq \left\{\sum_{i=1}^{t} w_i(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2 \max_{i \leq t} w_i \sqrt{2t\iota}\right\}.
\end{aligned}$$

Then probability bound gives

$$\mathbb{P}\left(\sum_{i=1}^{t} w_i(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2 \max_{i \leq t} w_i \sqrt{2t\iota}\right) \geq 1 - p/T.$$

Taking a union bound, we get

$$\sum_{i=1}^{t} w_i(\ell_i(a^i) - \langle \theta_i, \ell_i \rangle) \leq 2 \max_{i \leq t} w_i \sqrt{2t\iota} \quad \text{for all } t \in [T]$$

with probability at least $1 - p$. Summing the above and the regret bound shown in Lemma G.4, we finish the proof. □

Lemma G.4. *Let $(w_t, (\mathcal{L}_{at})_{a \in [A]})_{t \geq 1}$ be any $\mathcal{F}_t$-predictable sequence satisfying $1 \leq \min_{i \leq t} w_i \leq \max_{i \leq t} w_i \leq W$ for some constant (non-random) $W > 0$ almost surely. Then, following Algorithm 9, with probability at least $1 - 3p$, for any $\theta^* \in \Delta^A$ and $t \leq T$ we have*

$$\sum_{i=1}^{t} w_i \langle \theta_i - \theta^*, \ell_i \rangle \leq 10 \max_{i \leq t} w_i \cdot \left[\sqrt{At\iota} + \iota\right],$$

where $\iota = \log(AT\lceil \log_2 W \rceil / p)$.

Proof. The regret $R_t(\theta^*)$ can be decomposed into three terms

$$\begin{aligned}
R_t(\theta^*) = &\sum_{i=1}^{t} w_i \langle \theta_i - \theta^*, \ell_i \rangle \\
= &\underbrace{\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^*, \widehat{\ell}_i \right\rangle}_{(A)} + \underbrace{\sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \widehat{\ell}_i \right\rangle}_{(B)} + \underbrace{\sum_{i=1}^{t} w_i \left\langle \theta^*, \widehat{\ell}_i - \ell_i \right\rangle}_{(C)}
\end{aligned}$$

and we bound (A) in Lemma G.6, (B) in Lemma G.7 and (C) in Lemma G.8.

Setting $\eta_t = \gamma_t = \sqrt{\frac{\iota}{At}}$, the conditions in Lemma G.6 and Lemma G.8 are satisfied. Putting them together and take union bound, we have with probability $1 - 3p$

$$\begin{aligned}
R_t(\theta^*) \leq &\frac{w_t \log A}{\eta_t} + \frac{A}{2} \sum_{i=1}^{t} \eta_i w_i + \max_{i \leq t} w_i \iota + A \sum_{i=1}^{t} \gamma_i w_i + 2 \max_{i \leq t} w_i \sqrt{2t\iota} + 2 \max_{i \leq t} w_i \iota / \gamma_t \\
\leq &\max_{i \leq t} w_i \left[\sqrt{At\iota} + \frac{3\sqrt{A\iota}}{2} \sum_{i=1}^{t} \frac{1}{\sqrt{t}} + \iota + 2\sqrt{2t\iota} + 2\sqrt{At\iota}\right] \\
\leq &10 \max_{i \leq t} w_i \left[\sqrt{At\iota} + \iota\right] \quad \text{for all } t \in [T]
\end{aligned}$$

This proves the lemma. □

The rest of this section is devoted to the proofs of the Lemmas used in the proofs of Lemma G.4. We begin the following useful lemma adapted from Lemma 1 in Neu (2015), which is crucial in constructing high probability guarantees.

Lemma G.5. *For any predictable sequence of coefficients $c_1, c_2, \ldots, c_t$ s.t. $c_i \in [0, 2\gamma_i]^A$ w.r.t. $(\mathcal{F}_i)_{i\geq 1}$ and fixing t, we have with probability at least $1 - p/(AT)$,*

$$\sum_{i=1}^{t} w_i \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \leq 2 \max_{i\leq t} w_i \iota$$

Proof. Define $M = \lceil \log_2 W \rceil$, and $w_i(k) = w_i \mathbf{1}\left\{w_i \leq 2^k\right\}$. By definition,

$$\begin{aligned} w_i(k)\widehat{\ell}_i(a) =& \frac{w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{\theta_i(a) + \gamma_i} \leq \frac{w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{\theta_i(a) + \frac{w_i(k)\tilde{\ell}_i(a)\mathbf{1}\{a_i=a\}}{2^k}\gamma_i} \\ =& \frac{2^k}{2\gamma_i} \frac{\frac{2\gamma_i w_i(k)\tilde{\ell}_i(a)\mathbf{1}\{a_i=a\}}{2^k\theta_i(a)}}{1 + \frac{\gamma_i w_i(k)\tilde{\ell}_i(a)\mathbf{1}\{a_i=a\}}{2^k\theta_i(a)}} \overset{(i)}{\leq} \frac{2^k}{2\gamma_i} \log\left(1 + \frac{2\gamma_i w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{2^k\theta_i(a)}\right) \end{aligned}$$

where (i) is because $\frac{z}{1+z/2} \leq \log(1+z)$ for all $z \geq 0$.

Defining the sum

$$\widehat{S}_i = \frac{w_i(k)}{2^k}\left\langle c_i, \widehat{\ell}_i \right\rangle, \quad S_i = \frac{w_i(k)}{2^k}\left\langle c_i, \ell_i \right\rangle,$$

Then S_i is $\mathcal{F}_{i-1}$-measurable since w_i, c_i, ℓ_i are $\mathcal{F}_{i-1}$-measurable. Using $\mathbb{E}_i[\cdot]$ to denote the conditional expectation $\mathbb{E}[\cdot|\mathcal{F}_i]$, we have

$$\begin{aligned} \mathbb{E}_{i-1}\left[\exp\left(\widehat{S}_i\right)\right] &\leq \mathbb{E}_{i-1}\left[\exp\left(\sum_a \frac{c_i(a)}{2\gamma_i}\log\left(1 + \frac{2\gamma_i w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{2^k\theta_i(a)}\right)\right)\right] \\ &\overset{(i)}{\leq} \mathbb{E}_{i-1}\left[\prod_a \left(1 + \frac{c_i(a)\,w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{2^k\theta_i(a)}\right)\right] \\ &= \mathbb{E}_{i-1}\left[1 + \sum_a \frac{c_i(a)\,w_i(k)\tilde{\ell}_i(a)\,\mathbf{1}\{a_i = a\}}{2^k\theta_i(a)}\right] \\ &= 1 + S_i \leq \exp(S_i) \end{aligned}$$

where (i) is because $z_1 \log(1 + z_2) \leq \log(1 + z_1 z_2)$ for any $0 \leq z_1 \leq 1$ and $z_2 \geq -1$. Here we are using the condition $c_i(a) \leq 2\gamma_i$ to guarantee the condition is satisfied.

Equipped with the above bound, we can now prove the concentration result.

$$\begin{aligned} \mathbb{P}\left[\sum_{i=1}^{t}\left(\widehat{S}_i - S_i\right) \geq \iota\right] &\leq \mathbb{P}\left[\exp\left[\sum_{i=1}^{t}\left(\widehat{S}_i - S_i\right)\right] \geq \frac{ATM}{p}\right] \\ &\leq \frac{p}{ATM}\mathbb{E}_{t-1}\left[\exp\left[\sum_{i=1}^{t}\left(\widehat{S}_i - S_i\right)\right]\right] \\ &\leq \frac{p}{ATM}\mathbb{E}_{t-2}\left[\exp\left[\sum_{i=1}^{t-1}\left(\widehat{S}_i - S_i\right)\right] E_{t-1}\left[\exp\left(\widehat{S}_t - S_t\right)\right]\right] \\ &\leq \frac{p}{ATM}\mathbb{E}_{t-2}\left[\exp\left[\sum_{i=1}^{t-1}\left(\widehat{S}_i - S_i\right)\right]\right] \\ &\leq \cdots \leq \frac{p}{ATM}. \end{aligned}$$

So we have

$$\mathbb{P}\Big(\sum_{t=1}^{t} w_i(k)\left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \leq 2^k\iota\Big) \geq 1 - p/(ATM).$$

Taking a union bound,

$$\mathbb{P}\Big(\forall k \in [M], \sum_{t=1}^{t} w_i(k) \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \le 2^k \iota\Big) \ge 1 - p/(AT). \tag{23}$$

Denote $k' = \lceil \log_2 \max_{i \le t} w_i \rceil$, and note that

$$\begin{aligned}
&\Big\{\forall k \in [M], \sum_{t=1}^{t} w_i(k) \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \le 2^k \iota\Big\} \\
\subseteq &\Big\{\sum_{t=1}^{t} w_i(k') \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \le 2^{k'} \iota\Big\} \subseteq \Big\{\sum_{t=1}^{t} w_i \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \le 2 \max_{i \le t} w_i \iota\Big\}.
\end{aligned}$$

Therefore, we have

$$\mathbb{P}\left(\sum_{i=1}^{t} w_i \left\langle c_i, \widehat{\ell}_i - \ell_i \right\rangle \le 2 \max_{i \le t} w_i \iota\right) \ge 1 - p/(AT).$$

This proves the lemma. □

Using Lemma G.5, we can bound the $(A)(B)(C)$ separately as below.

Lemma G.6. *If $\eta_i \le 2\gamma_i$ for all $i \le t$, with probability $1 - p$, for any $t \in [T]$ and $\theta^* \in \Delta^A$,*

$$\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^*, \widehat{\ell}_i \right\rangle \le \frac{w_t \log A}{\eta_t} + \frac{A}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \le t} w_i \iota$$

Proof. We use the standard analysis of FTRL with changing step size, see for example Exercise 28.13 in Lattimore & Szepesvári (2020). Notice the essential step size is η_t / w_t,

$$\begin{aligned}
\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^*, \widehat{\ell}_i \right\rangle &\le \frac{w_t \log A}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i \left\langle \theta_i, \widehat{l}_i^2 \right\rangle \\
&\le \frac{w_t \log A}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{a \in \mathcal{A}} \eta_i w_i \widehat{\ell}_i(a) \\
&\overset{(i)}{\le} \frac{w_t \log A}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{a \in \mathcal{A}} \eta_i w_i \ell_i(a) + \max_{i \le t} w_i \iota \\
&\le \frac{w_t \log A}{\eta_t} + \frac{A}{2} \sum_{i=1}^{t} \eta_i w_i + \max_{i \le t} w_i \iota
\end{aligned}$$

where (i) is by using Lemma G.5 with $c_i(a) = \eta_i$. The any-time guarantee is justified by taking union bound. □

Lemma G.7. *With probability $1 - p$, for any $t \in [T]$,*

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \widehat{\ell}_i \right\rangle \le A \sum_{i=1}^{t} \gamma_i w_i + 2 \max_{i \le t} w_i \sqrt{2t\iota}$$

Proof. We further decompose the left side into

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \widehat{\ell}_i \right\rangle = \sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \mathbb{E}_{i-1}[\widehat{\ell}_i] \right\rangle + \sum_{i=1}^{t} w_i \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle.$$

The first term is bounded by

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \mathbb{E}_{i-1}[\widehat{\ell}_i] \right\rangle = \sum_{i=1}^{t} w_i \left\langle \theta_i, \ell_i - \frac{\theta_i}{\theta_i + \gamma_i} \ell_i \right\rangle$$

$$= \sum_{i=1}^{t} w_i \left\langle \theta_i, \frac{\gamma_i}{\theta_i + \gamma_i} \ell_i \right\rangle \leq A \sum_{i=1}^{t} \gamma_i w_i$$

To bound the second term, we use similar argument in the proof of Lemma G.5 , we define $w = \max_{i \leq t} w_i$, $M = \lceil \log_2 W \rceil$. and $w_i(k) = w_i \mathbf{1}\left\{w_i \leq 2^k\right\}$, notice

$$\left\langle \theta_i, \widehat{\ell}_i \right\rangle \leq \sum_{a \in \mathcal{A}} \theta_i(a) \frac{\mathbf{1}\{a_t = a\}}{\theta_i(a) + \gamma_i} \leq \sum_{a \in \mathcal{A}} \mathbf{1}\{a_i = a\} = 1,$$

thus $\{w_i(k) \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle\}_{i=1}^{t}$ is a bounded martingale difference sequence w.r.t. the filtration $\{\mathcal{F}_i\}_{i=1}^{t}$. By Azuma-Hoeffding inequality,

$$\sum_{i=1}^{t} w_i(k) \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle \leq \sqrt{2\iota \sum_{i=1}^{t} w_i(k)^2} \leq \sqrt{2\iota \cdot t (2^k)^2}$$

with probability at least $1 - p/TM$. Taking a union bound, we get

$$\sum_{i=1}^{t} w_i(k) \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle \leq \sqrt{2\iota \cdot t 2^{2k}}, \quad \text{for all } (t,k) \in [T] \times [M]$$

with probability at least $1 - p$. On this event, choosing $k' = \lceil \log_2 w \rceil$, we have

$$\begin{aligned} \sum_{i=1}^{t} w_i \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle \leq & \sum_{i=1}^{t} w_i(k') \left\langle \theta_i, \mathbb{E}_{i-1}[\widehat{\ell}_i] - \widehat{\ell}_i \right\rangle \\ \leq & 2^{k'} \sqrt{2\iota \cdot t} \leq 2 \max_i w_i \sqrt{2\iota \cdot t}. \end{aligned}$$

This ends the proof. □

Lemma G.8. *With probability $1 - p$, for any $t \in [T]$ and any $\theta^* \in \Delta^A$, if γ_i is non-increasing in i,*

$$\sum_{i=1}^{t} w_i \left\langle \theta^*, \widehat{\ell}_i - \ell_i \right\rangle \leq 2 \max_{i \leq t} w_i \iota / \gamma_t$$

Proof. Define a basis $\{e_j\}_{j=1}^{A}$ of $\mathbb{R}^A$ by

$$e_j(a) = \begin{cases} 1 \text{ if } a = j \\ 0 \text{ otherwise} \end{cases}$$

Then for all the $j \in [A]$, apply Lemma G.5 with $c_i = \gamma_t e_j$. Since now $c_i(a) \leq \gamma_t \leq \gamma_i$, the condition in Lemma G.5 is satisfied. As a result, for any $t \in [T]$ and $j \in [A]$, we have with probability at least $1 - p/(TA)$ that

$$\sum_{i=1}^{t} w_i \left\langle e_j, \widehat{\ell}_i - \ell_i \right\rangle \leq 2 \max_{i \leq t} w_i \iota / \gamma_t$$

Taking a union bound, we have with probability at least $1 - p$,

$$\sum_{i=1}^{t} w_i \left\langle e_j, \widehat{\ell}_i - \ell_i \right\rangle \leq 2 \max_{i \leq t} w_i \iota / \gamma_t \quad \text{for all } (j,t) \in [m] \times [T].$$

Since any θ^* is a convex combination of $\{e_j\}_{j=1}^{A}$, on this event, we also have

$$\sum_{i=1}^{t} w_i \left\langle \theta^*, \widehat{\ell}_i - \ell_i \right\rangle \leq 2 \max_{i \leq t} w_i \iota / \gamma_t \quad \text{for all } t \in [T].$$

This finishes the proof. □