V-Learning—A Simple, Efficient, Decentralized Algorithm for Multiagent RL

Chi Jin, Qinghua Liu & Yuanhao Wang
Princeton University
Princeton, NJ 08544, USA
{chij,qinghual,yuanhao}@princeton.edu

Tiancheng Yu
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
yutc@mit.edu

Abstract

A major challenge of multiagent reinforcement learning (MARL) is the curse of multiagents, where the size of the joint action space scales exponentially with the number of agents. This remains to be a bottleneck for designing efficient MARL algorithms even in a basic scenario with finitely many states and actions. This paper resolves this challenge for the model of episodic Markov games. We design a new class of fully decentralized algorithms—V-learning, which provably learns Nash equilibria (in the two-player zero-sum setting), correlated equilibria and coarse correlated equilibria (in the multiplayer general-sum setting) in a number of samples that only scales with \( \max_{i \in [m]} A_i \), where \( A_i \) is the number of actions for the \( i \)th player. This is in sharp contrast to the size of the joint action space which is \( \prod_{i=1}^m A_i \).

V-learning (in its basic form) is a new class of single-agent RL algorithms that convert any adversarial bandit algorithm with suitable regret guarantees into an RL algorithm. Similar to the classical Q-learning algorithm, it performs incremental updates to the value functions. Different from Q-learning, it only maintains the estimates of V-values instead of Q-values. This key difference allows V-learning to achieve the claimed guarantees in the MARL setting by simply letting all agents run V-learning independently.

1 Introduction

A wide range of modern artificial intelligence challenges can be cast as multi-agent reinforcement learning (MARL) problems, in which agents learn to make a sequence of decisions in the presence of other agents whose decisions will influence the outcome and can adapt to the strategies of the agents. Modern MARL systems have achieved significant success recently on a rich set of traditionally challenging tasks, including the game of GO (Silver et al., 2016, 2017), Poker (Brown & Sandholm, 2019), real-time strategy games (Vinyals et al., 2019; OpenAI, 2018), decentralized controls or multiagent robotics systems (Brambilla et al., 2013), autonomous driving (Shalev-Shwartz et al., 2016), as well as complex social scenarios such as hide-and-seek (Baker et al., 2020). While single-agent RL has been the focus of recent intense theoretical study, MARL has been comparatively underexplored, which leaves several fundamental questions open even in the basic model of Markov games [Shapley, 1953] with finitely many states and actions.

One such unique challenge of MARL is the curse of multiagents—let \( A_i \) be the number of actions for the \( i \)th player, then the number of possible joint actions (as well as the number of parameters to specify a Markov game) scales with \( \prod_{i=1}^m A_i \), which grows exponentially with the number of agents \( m \). This remains to be a bottleneck even for the best existing algorithms for learning Markov games. In fact, a majority of these algorithms adapt the classical single-agent algorithms, such as value iteration or Q-learning, into the multiagent setting [Bai et al., 2020; Liu et al., 2021], whose sample complexity scales at least linearly with respect to \( \prod_{i=1}^m A_i \). This is prohibitively large in practice even for fairly small multiagent applications, say only ten agents are involved with ten actions available for each agent.

1 A subset of the results in this paper was presented in the conference version [Bai et al., 2020] at NeurIPS 2020.
Table 1: A summary of sample complexities for V-learning under different settings. Here $H$ is the length of each episode, $S$ is the number of states, $A = \max_{i \in [m]} A_i$ is the largest number of actions for each agent, $\epsilon$ is the error tolerance. An information theoretical lower bound for all objectives is $\Omega(H^3SA/\epsilon^2)$, Jin et al. (2018); Bai & Jin (2020).

<table>
<thead>
<tr>
<th>Objective</th>
<th>Multiplayer General-sum</th>
<th>Two-player Zero-sum</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nash Equilibria</td>
<td>$\tilde{O}(H^5SA/\epsilon^2)$</td>
<td>PPAD-complete</td>
<td>-</td>
</tr>
<tr>
<td>Coarse Correlated Equilibria</td>
<td>$\tilde{O}(H^5SA/\epsilon^2)$</td>
<td>$\tilde{O}(H^5SA/\epsilon^2)$</td>
<td>-</td>
</tr>
<tr>
<td>Correlated Equilibria</td>
<td>$\tilde{O}(H^5SA^2/\epsilon^2)$</td>
<td>$\tilde{O}(H^5SA^2/\epsilon^2)$</td>
<td>-</td>
</tr>
</tbody>
</table>

Another challenge of the MARL is to design decentralized algorithms. While a centralized algorithm requires the existence of a centralized controller which gathers all information and jointly optimizes the policies of all agents, a decentralized algorithm allows each agent to only observe her own actions and rewards while optimizing her own policy independently. Decentralized algorithms are often preferred over centralized algorithms in practice since (1) decentralized algorithms are typically cleaner, easier to implement; (2) decentralized algorithms are more versatile as the individual learners are indifferent to the interaction and the number of other agents; and (3) they are also faster due to less communication required. While several provable decentralized MARL algorithms have been developed (see, e.g., Zhang et al., 2018; Sayin et al., 2021; Daskalakis et al., 2021), they either have only asymptotic guarantees or work only under certain reachability assumptions (see Section [A]). The existing provably efficient algorithms for general Markov games (without further assumptions) are exclusively centralized algorithms (Bai & Jin, 2020; Xie et al., 2020; Liu et al., 2021).

This motivates us to ask the following open question:

**Can we design decentralized MARL algorithms that break the curse of multiagents?**

This paper addresses both challenges mentioned above, and provide the first positive answer to this question in the basic setting of tabular episodic Markov games. We propose a new class of single-agent RL algorithms—V-learning, which converts any adversarial bandit algorithm with suitable regret guarantees into a RL algorithm. Similar to the classical Q-learning algorithm, V-learning also performs incremental updates to the values. Different from Q-learning, V-learning only maintains the V-value functions instead of the Q-value functions. We remark that the number of parameters of Q-value functions in MARL is $O(S \prod_{i=1}^m A_i)$, where $S$ is the number of states, while the number of parameters of V-value functions is only $O(S)$. This key difference allows V-learning to be readily extended to the MARL setting by simply letting all agents run V-learning independently, which gives a fully decentralized algorithm.

We consider the standard learning objectives in the game theory—Nash equilibrium (NE), correlated equilibrium (CE) and coarse correlated equilibrium (CCE). Except for the task of finding a NE in multiplayer general-sum games which is PPAD-complete even for matrix games, we prove that V-learning finds a NE for two-player zero-sum Markov games within $\tilde{O}(H^5SA/\epsilon^2)$ episodes, where $H$ is the length of each episode, $A = \max_{i \in [m]} A_i$ is the maximum number of actions of all agents, and $\epsilon$ is the error tolerance for the objective. We further prove that for multiplayer general-sum Markov games, V-learning coupled with suitable adversarial bandit algorithms is capable of finding a CCE within $\tilde{O}(H^5SA/\epsilon^2)$ episodes, and a CE within $\tilde{O}(H^5SA^2/\epsilon^2)$ episodes. All sample complexities mentioned above do not grow with the number of the agents, which thus break the curse of multiagents (See Table 1 for a summary and Appendix [A] for a more comprehensive discussion on related literature.).
2 Preliminaries

We consider the model of Markov Games (MG) [Shapley, 1953] (also known as stochastic games in the literature) in its most generic—multiplayer general-sum form. Formally, we denote an tabular episodic MG with \( m \) players by a tuple \( \text{MG}(H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathcal{P}, \{r_i\}_{i=1}^m) \), where \( H, \mathcal{S} \) denote the length of each episode and the state space with \( |\mathcal{S}| = |H| \). \( \mathcal{A}_i \) denotes the action space for the \( i \)\textsuperscript{th} player and \( |\mathcal{A}_i| = A_i \). We let \( \alpha := (a_1, \cdots, a_m) \) denote the (tuple of) joint actions by all \( m \) players, and \( \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m \). \( \mathcal{P} = \{P_h\}_{h \in [H]} \) is a collection of transition matrices, so that \( P_h(s, a) \) gives the distribution of the next state if actions \( a \) are taken at state \( s \) at step \( h \), and \( r = \{r_{i,h}\}_{h \in [H]} \) is a collection of reward functions for the \( i \)\textsuperscript{th} player, so that \( r_{i,h}(s, a) \in [0,1] \) gives the deterministic reward received by the \( i \)\textsuperscript{th} player if actions \( a \) are taken at state \( s \) at step \( h \).\footnote{Our results directly generalize to random reward functions, since learning transitions is more difficult than learning rewards.}

We remark that since the relation among the rewards of different agents can be arbitrary, this model of MGs incorporates both cooperation and competition.

In each episode, we start with a fixed initial state \( s_1 \).\footnote{While we assume a fixed initial state for notational simplicity, our results readily extend to the setting where the initial state is sampled from a fixed initial distribution.} At each step \( h \in [H] \), each player \( i \) observes state \( s_h \in \mathcal{S} \), picks action \( a_{i,h} \in \mathcal{A}_i \) simultaneously, observes the actions played by other players\footnote{We assume the knowledge of other players’ actions for the convenience of later definitions.} and receives her own reward \( r_{i,h}(s_h, a_h) \). Then the environment transitions to the next state \( s_{h+1} \sim P_h(\cdot | s_h, a_h) \). The episode ends when \( s_{H+1} \) is reached.

**Policy, value function** A (random) policy \( \pi_i \) of the \( i \)\textsuperscript{th} player is a set of \( H \) maps \( \pi_i := \{\pi_{i,h} : \Omega \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \mathcal{A}_i\}_{h \in [H]} \), where \( \pi_{i,h} \) maps a random sample \( \omega \) from a probability space \( \Omega \) and a history of length \( h \)—say \( \tau_h := (s_1, a_1, \cdots, s_h) \), to an action in \( \mathcal{A}_i \). To execute policy \( \pi_i \), we first draw a random sample \( \omega \) at the beginning of the episode. Then, at each step \( h \), the \( i \)\textsuperscript{th} player simply takes action \( \pi_{i,h}(\omega, \tau_h) \). We note here \( \omega \) is shared among all steps \( h \in [H] \), \( \omega \) encodes both the correlation among steps and the individual randomness of each step. We further say a policy \( \pi_i \) is deterministic if \( \pi_{i,h}(\omega, \tau_h) = \pi_{i,h}(\tau_h) \) which is independent of the choice of \( \omega \).

An important subclass of policy is Markov policy, which can be defined as \( \pi_i := \{\pi_{i,h} : \Omega \times \mathcal{S} \to \mathcal{A}_i\}_{h \in [H]} \). Instead of depending on the entire history, a Markov policy takes actions only based on the current state. Furthermore, the randomness in each step of Markov policy is independent. Therefore, when it is clear from the context, we write Markov policy as \( \pi_i := \{\pi_{i,h} : \mathcal{S} \to \Delta_{\mathcal{A}_i}\}_{h \in [H]} \), where \( \Delta_{\mathcal{A}_i} \) denotes the simplex over \( \mathcal{A}_i \). We also use notation \( \pi_{i,h}(a | s) \) to denote the probability of the \( i \)\textsuperscript{th} agent taking action \( a \) at state \( s \) at step \( h \).

A joint (potentially correlated) policy is a set of policies \( \{\pi_i\}_{i=1}^m \), where the same random sample \( \omega \) is shared among all agents, which we denote as \( \pi = \pi_1 \odot \pi_2 \odot \cdots \odot \pi_m \). We also denote \( \pi_{-i} = \pi_1 \odot \cdots \pi_i \odot \pi_{i+1} \odot \cdots \odot \pi_m \), to be the joint policy excluding the \( i \)\textsuperscript{th} player. A special case of joint policy is the product policy where the random sample has special form \( \omega = (\omega_1, \cdots, \omega_m) \), and for any \( i \in [m] \), \( \pi_i \) only uses the randomness in \( \omega_i \), which is independent of remaining \( \{\omega_j\}_{j \neq i} \), which we denote as \( \pi = \pi_1 \times \pi_2 \times \cdots \times \pi_m \).

We define the value function \( V_{i,1}^{\pi} (s_1) \) as the expected cumulative reward that the \( i \)\textsuperscript{th} player will receive if the game starts at initial state \( s_1 \) at the 1\textsuperscript{st} step and all players follow joint policy \( \pi \):

\[
V_{i,1}^{\pi}(s_1) := \mathbb{E}_\pi \left[ \sum_{h=1}^{H} r_{i,h}(s_h, a_h) \big| s_1 \right].
\] (1)

where the expectation is taken over the randomness in transition and the random sample \( \omega \) in policy \( \pi \).

**Best response and strategy modification** For any strategy \( \pi_{-i} \), the best response of the \( i \)\textsuperscript{th} player is defined as a policy of the \( i \)\textsuperscript{th} player which is independent of the randomness in \( \pi_{-i} \), and achieves the highest value for herself conditioned on all other players deploying \( \pi_{-i} \). In symbol, the best
response is the maximizer of $\max_{\pi_i} V^\pi_{i,1}(s_1)$ whose value we also denote as $V^\pi_{i,1}(s_1)$ for simplicity. By its definition, we know the best response can always be achieved at deterministic policies.

A strategy modification $\phi_i$ for the $i$th player is a set of maps $\phi_i := \{(S \times A)^{h-1} \times S \times A_i \rightarrow A_i\}$ where $\phi_{i,h}$ can depend on the history $\tau_h$ and maps actions in $A_i$ to different actions in $A_i$. For any policy of the $i$th player $\pi_i$, the modified policy (denoted as $\phi_i \circ \pi_i$) changes the action $\pi_{i,h}(\omega, \tau_h)$ under random sample $\omega$ and history $\tau_h$ to $\phi_i(\tau_h, \pi_{i,h}(\omega, \tau_h))$. For any joint policy $\pi$, we define the best strategy modification of the $i$th player as the maximizer of $\max_{\phi_i} V^\phi_{i,1}(\phi_i \circ \pi_{-i})(s_1)$.

Different from the best response, which is completely independent of the randomness in $\pi_{-i}$, the best strategy modification changes the policy of the $i$th player while still utilizing the shared randomness among $\pi_i$ and $\pi_{-i}$. Therefore, the best strategy modification is more powerful than the best response: formally one can show that $\max_{\phi_i} V^\phi_{i,1}(\phi_i \circ \pi_{-i})(s_1) \geq \max_{\pi_i} V^\pi_{i,1}(\pi_{-i})(s_1)$ for any policy $\pi$.

### 2.1 Learning Objectives

A special case of Markov game is Markov Decision Process (MDP). One can show there always exists an optimal policy $\pi^* = \arg\max_\pi V^\pi(1)$, Denote the value of the optimal policy as $V^*$. The objective of learning MDPs is to find an $\epsilon$-optimal policy $\pi$, which satisfies $V^*_1(s_1) - V^\pi_1(s_1) \leq \epsilon$.

For Markov games, there are three common learning objectives in the game theory literature—Nash Equilibrium, Correlated Equilibrium (CE) and Coarse Correlated Equilibrium (CCE).

First, a Nash equilibrium is defined as a product policy where no player can increase her value by changing only her own policy. Formally,

**Definition 1 (Nash Equilibrium).** A product policy $\pi$ is a **Nash equilibrium** if $\max_{i \in [m]} (V^\pi_{i,1} - V^\pi_{i,1})(s_1) = 0$. A product policy $\pi$ is an $\epsilon$-approximate Nash equilibrium if $\max_{i \in [m]} (V^\pi_{i,1} - V^\pi_{i,1})(s_1) \leq \epsilon$.

We remark that, except for the special case of two-player zero-sum Markov games where reward $r_{2,h} = -r_{1,h}$ for any $h \in [H]$, the Nash equilibrium in general has been proved PPAD-hard to compute [Daskalakis, 2013]. Therefore, we only present results for finding Nash equilibria in two-player zero-sum MCGs in this paper.

Second, a coarse correlated equilibrium is defined as a joint (potentially correlated) policy where no player can increase her value by playing a different independent strategy. In symbol,

**Definition 2 (Coarse Correlated Equilibrium).** A joint policy $\pi$ is a **CCE** if $\max_{i \in [m]} (V^\pi_{i,1} - V^\pi_{i,1})(s_1) = 0$. A joint policy $\pi$ is an $\epsilon$-approximate CCE if $\max_{i \in [m]} (V^\pi_{i,1} - V^\pi_{i,1})(s_1) \leq \epsilon$.

The only difference between Definition 1 and Definition 2 is that Nash equilibrium requires the policy $\pi$ to be a product policy while CCE does not. Thus, it is clear that CCE is a relaxed notion of Nash equilibrium, and a Nash equilibrium is always a CCE.

Finally, a correlated equilibrium is defined as a joint (potentially correlated) policy where no player can increase her value by using a strategy modification. In symbol,

**Definition 3 (Correlated Equilibrium).** A joint policy $\pi$ is a **CE** if $\max_{i \in [m]} \max_{\phi_i} (V^\phi_{i,1} - V^\phi_{i,1})(s_1) = 0$. A joint policy $\pi$ is an $\epsilon$-approximate CE if $\max_{i \in [m]} \max_{\phi_i} (V^\phi_{i,1} - V^\phi_{i,1})(s_1) \leq \epsilon$.

In Markov games, we also have that a Nash equilibrium is a CE, and a CE is a CCE (see Proposition 3 in Appendix C for more details).

---

3Here, we only introduce the deterministic strategy modification for simplicity of notation, which is sufficient for discussion in the context of this paper. The random strategy modification can also be defined by introducing randomness in $\phi_i$ which is independent of randomness in $\pi_i$ and $\pi_{-i}$. It can be shown that the best strategy modification can always be deterministic.

4Technically, to ensure $r_{2,h} \in [0,1]$, we choose $r_{2,h} = 1 - r_{1,h}$. We note that adding a constant to the reward function has no effect on the equilibrium, which is our learning objective.
Algorithm 1 V-LEARNING

1: Initialize: for any \( (s, a, h), V_h(s) \leftarrow H + 1 - h, N_h(s) \leftarrow 0, \pi_h(a|s) \leftarrow 1/A. \)
2: for episode \( k = 1, \ldots, K \) do
3: \hspace{1em} receive \( s_1 \).
4: \hspace{2em} for step \( h = 1, \ldots, H \) do
5: \hspace{3em} take action \( a_h \sim \pi_h(\cdot|s_h) \), observe reward \( r_h \) and next state \( s_{h+1} \).
6: \hspace{3em} \( t = N_h(s_h) \leftarrow N_h(s_h) + 1. \)
7: \hspace{3em} \( \tilde{V}_h(s_h) \leftarrow (1 - \alpha_t)\tilde{V}_h(s_h) + \alpha_t (r_h + V_{h+1}(s_{h+1}) + \beta_t). \)
8: \hspace{3em} \( V_h(s_h) \leftarrow \min\{H + 1 - h, \tilde{V}_h(s_h)\}. \)
9: \hspace{3em} \( \pi_h(\cdot|s_h) \leftarrow \text{ADV_BANDIT_UPDATE}(a_h, \frac{H - r_h - V_{h+1}(s_{h+1})}{H}) \) on \( (s_h, h) \)th adversarial bandit.

3 V-LEARNING ALGORITHM

In this section, we introduce V-learning algorithm as a new class of single-agent RL algorithms, which converts any adversarial bandit algorithm with suitable regret guarantees into a RL algorithm. We also present its theoretical guarantees for finding a nearly optimal policy in the single-agent setting.

Training algorithm To begin with, we describe the V-learning algorithm (Algorithm 1). It maintains a value \( \tilde{V}_h(s) \), a counter \( N_h(s) \), and a policy \( \pi_h(\cdot|s) \) for each state \( s \) and step \( h \), and initializes them to be the max value, 0, and uniform distribution respectively. V-learning also instantiates \( S \times H \) different adversarial bandit algorithms—one for each \( (s, h) \) pair. At each step \( h \) in each episode \( k \), the algorithm performs three major steps:

- Policy execution (Line 5-6): the algorithm takes action \( a_h \) according to the maintained \( \pi_h \), then observes the reward \( r_h \) and the next state \( s_{h+1} \), and increases the counter \( N_h(s_h) \) by 1.
- V-value update (Line 7-8): the algorithm performs incremental update to the value function:
\[
\tilde{V}_h(s_h) \leftarrow (1 - \alpha_t)\tilde{V}_h(s_h) + \alpha_t (r_h + V_{h+1}(s_{h+1}) + \beta_t) \tag{2}
\]
where \( \alpha_t \) is the learning rate, and \( \beta_t \) is the bonus to promote optimism (and exploration). The choices of both quantities will be specified later. Next, we simply update \( V_h \) as a truncated version of \( \tilde{V}_h \).
- Policy update (Line 9): the algorithm feeds the action \( a_h \) and its “loss” \( \frac{H - r_h + V_{h+1}(s_{h+1})}{H} \) to the \( (s_h, h) \)th adversarial bandit algorithm, and receives the updated policy \( \pi_h(\cdot|s_h) \).

Throughout this paper, we will always use the following learning rate \( \alpha_t \). We also define an auxiliary sequence \( \{\alpha_t^j\}_{i=1}^j \) based on the learning rate, which will be frequently used across the paper.
\[
\alpha_t = \frac{H + 1}{H + t}, \quad \alpha_t^0 = \prod_{j=1}^{t} (1 - \alpha_j), \quad \alpha_t^j = \alpha_t \prod_{j=i+1}^{t} (1 - \alpha_j). \tag{3}
\]

We remark that our incremental update equation (2) bears significant similarity to Q-learning, and our choice of learning rate is precisely the same as the choice in Q-learning (Jin et al., 2018). However, a key difference is that the V-learning algorithm maintains V-value functions instead of Q-value functions. This is crucial when extending V-learning to the multiplayer setting where the number of parameters of Q-value functions becomes \( O(HS \prod_{i=1}^{m} A_i) \) while the number of parameters of V-value functions is only \( O(HS) \). Since V-learning does not use action-value functions, it resorts to adversarial bandit algorithms to update its policy.

ADV_BANDIT_UPDATE subroutine: Consider a multi-arm bandit problem with adversarial loss, where we denote the action set by \( B \) with |\( B \)\| = \( B \). At round \( t \), the learner picks a strategy (distribution over actions) \( \theta_t \in \Delta_B \), and the adversary chooses a loss vector \( \ell_t \in [0, 1]^B \). Then the learner takes an action \( b_t \) that is sampled from distribution \( \theta_t \), and receives a noisy bandit-feedback
One can show that, if omitting the optimistic bonus, \( \hat{\pi} \) is a stochastic estimate of the value of policy \( \hat{\pi} \) (Algorithm 3). Let \( \{\hat{\pi}_h(\cdot|s_h)\}_{i=1}^{H} \) be a set of maps \( \hat{\pi}_h(\cdot|s_h) \) \( \in \pi_h \). The choice of action at each step \( h \) depends on a joint randomness \( \omega \in \Omega \) which is shared among all steps, and the history of past states \( (s_1, \ldots, s_h) \). We refer readers to Appendix H for more details.

### Algorithm 3 Executing Output Policy \( \hat{\pi} \) of V-learning

1: \( \text{Sample } k \leftarrow \text{Uniform}(\{K_i\}). \)
2: \( \text{For } h = 1, \ldots, H \) do
3: \( \text{Observe } s_h, \text{and set } t \leftarrow N_h^k(s_h). \)
4: \( \text{Set } k \leftarrow k_h^i(s_h), \text{where } i \in [t] \) is sampled with probability \( \alpha_t^i \).
5: \( \text{Take action } a_h \sim \pi_h^k(\cdot|s_h). \)

\( \tilde{\ell}_t(b_i) \in [0,1] \) where \( \mathbb{E}[\tilde{\ell}_t(b_i)|\ell_t, b_i] = \ell_t(b_i) \). Then, the adversarial bandit algorithm performs updates based on \( b_t \) and \( \tilde{\ell}_t(b_t) \), and outputs the strategy for next round \( \theta_{t+1} \), which we abstract as \( \theta_{t+1} \leftarrow \text{ADV_BANDIT_UPDATE}(b_t, \tilde{\ell}_t(b_t)) \) (see Protocol 2).

### Output policy

We define the final output policy \( \hat{\pi} \) of V-learning by how to execute this policy (see Algorithm 3). Let \( V^k, N^k, \pi^k \) be the value, counter and policy maintained by V-learning algorithm at the beginning of episode \( k \). The output policy maintains a scalar \( k \), which is initially uniformly sampled from \( [K] \). At each step \( h \), after observing \( s_h \), \( \hat{\pi} \) plays a mixture of policy \( \{\pi_h^k(\cdot|s_h)\}_{i=1}^{H} \) with corresponding probability \( \{\alpha_t^i\}_{i=1}^{H} \) defined in equation 3. Here \( t = N_h^k(s_h) \) is the number of times \( s_h \) is visited at step \( h \) at the beginning of episode \( k \), and \( k^i \) is short for \( k_h^i(s_h) \) which is the index of the episode when \( s_h \) is visited at step \( h \) for the \( i \)th time. After that, \( \hat{\pi} \) sets \( k \) to be the index \( k_h^i(s_h) \) whose policy is just played within the mixture, and continue the same process for the next step. This mixture form of output policy \( \hat{\pi} \) is mainly due to the incremental updates of V-learning. One can show that, if omitting the optimistic bonus, \( V^k_1(s_1) \) computed in the V-learning algorithm is a stochastic estimate of the value of policy \( \hat{\pi} \).

We remark that \( \hat{\pi} \) is not a Markov policy, but a general random policy (see Definition in Section 2), which can be written as a set of maps \( \pi_h : \Omega \times S_t \rightarrow A_t \). The choice of action at each step \( h \) depends on a joint randomness \( \omega \in \Omega \) which is shared among all steps, and the history of past states \( (s_1, \ldots, s_h) \). In Section 5 we will further introduce a simple monotone technique that allows V-learning to output a Markov policy in both the single-agent and the two-player zero-sum setting.

### Single-agent guarantees

We first state our requirement for the adversarial bandit algorithm used in V-learning, which is to have a high probability weighted external regret guarantee as follows. The weights \( \{\alpha_t^i\}_{i=1}^{H} \) are defined in equation 3.

**Assumption 1.** For any \( t \in \mathbb{N} \) and any \( \delta \in (0,1) \), with probability at least \( 1 - \delta \), we have

\[
\max_{\theta \in \Delta_S} \sum_{i=1}^{t} \alpha_t^i \langle \theta, \ell_i \rangle - \langle \theta, \ell_t \rangle \leq \xi(B, t, \log(1/\delta)). \tag{4}
\]

We further assume the existence of an upper bound \( \mathbb{E}(B, t, \log(1/\delta)) \geq \sum_{i=1}^{t} \xi(B, t, \log(1/\delta)) \) where (i) \( \xi(B, t, \log(1/\delta)) \) is non-decreasing in \( B \) for any \( t, \delta \); (ii) \( \mathbb{E}(B, t, \log(1/\delta)) \) is concave in \( t \) for any \( B, \delta \).

Assumption 1 can be satisfied by modifying many existing algorithms with unweighted external regret to the weighted setting. In particular, we prove that the Follow-the-Regularized-Leader (FTRL) algorithm (Algorithm 5) satisfies the Assumption 1 with bounds \( \xi(B, t, \log(1/\delta)) \leq O(\sqrt{H B t \log(B/\delta)/t}) \) and \( \mathbb{E}(B, t, \log(1/\delta)) \leq O(\sqrt{H B t \log(B/\delta)}) \). The \( H \) factor comes into the bounds because our choice of weights \( \{\alpha_t^i\} \) in equation 3 involves \( H \). We refer readers to Appendix H for more details.
We are now ready to introduce the theoretical guarantees of V-learning for finding near-optimal policies in the single-agent setting.

**Theorem 4.** Suppose subroutine \( \text{ADV\_BANDIT\_UPDATE} \) satisfies Assumption 7. For any \( \delta \in (0, 1) \) and \( K \in \mathbb{N} \), let \( \iota = \log(\text{HSAK}/\delta) \). Choose learning rate \( \alpha_t \) according to equation 3 and bonus \( \{\beta_t\}_{t=1}^{K} \) so that \( \sum_{t=1}^{T} \alpha_t \beta_t = \Theta(H^{5} \xi(A, t, \iota) + \sqrt{H^{5} \iota / t}) \) for any \( t \in [K] \). Then, with probability at least \( 1 - \delta \), after running Algorithm 7 for \( K \) episodes, we have the output policy \( \hat{\pi} \) by Algorithm 3 satisfies

\[
V_{1}^{\hat{\pi}}(s_1) - V_{1}^{\pi}(s_1) \leq O((H^{2}S/K) \cdot \Xi(A, K/S, \iota) + \sqrt{H^{5}S/K}).
\]

In particular, when instantiating subroutine \( \text{ADV\_BANDIT\_UPDATE} \) by FTRL (Algorithm 5), we can choose \( \beta_t = c \cdot \sqrt{H^{3} \xi / t} \) for some absolute constant \( c \), where \( V_{1}^{\hat{\pi}}(s_1) - V_{1}^{\pi}(s_1) \leq O(\sqrt{H^{5}SA/K}) \).

The special cases of Theorem 4 and Theorem 5 (when the subroutine is instantiated by FTRL) were firstly presented in the conference version of this paper (Bai et al. 2020), with an additional \( \sqrt{H} \) factor in the error.

Theorem 4 characterizes how fast the suboptimality of \( \hat{\pi} \) decreases with respect to the total number of episode \( K \). In particular, to obtain an \( \epsilon \)-optimal output policy \( \hat{\pi} \), we only need to use a number of episodes \( K = \tilde{O}(H^{5}SA/c^2) \). This is \( H^2 \) factor larger than the information-theoretic lower bound \( \Omega(H^{3}SA/c^2) \) in this setting (Jin et al., 2018). We remark that one extra \( H \) factor is due to the incremental update and the use of learning rate in equation 3 which is exactly the same for Q-learning algorithm (Jin et al., 2018). The other \( H \) factor can be potentially improved by using refined first-order regret bound (counterparts of Bernstein concentration) in V-learning.

While V-learning seems to be no better than classical value iteration or Q-learning in the single-agent setting, its true power starts to show up in the multiagent setting: Value iteration and Q-learning require highly nontrivial efforts to adapt them to the multiagent setting, and by design they suffer from the curse of multiagents (Bai et al., 2020; Liu et al., 2021). In the following sections, we will show that V-learning can be directly extended to the multiagent setting by simply letting all agents run V-learning independently. Furthermore, V-learning breaks the curse of multiagents.

### 4 Two-player Zero-sum Markov Games

In this section, we provide the sample efficiency guarantee for V-learning to find Nash equilibria in two-player zero-sum Markov games.

In the two-player zero-sum setting, we have two agents whose rewards satisfy \( r_{1,h} = -r_{2,h} \) for any \( h \in [H] \). Our algorithm is simply that both agents run V-learning (Algorithm 1) independently with learning rate \( \alpha_t \) as specified in equation 3. Each player will use her own set of bonus \( \{\beta_{j,t}\} \) that depends on the number of her actions and will be specified later. To execute the output policy, both agents simply execute Algorithm 3 independently using their own intermediate policies computed by V-learning.

We have the following theorem for V-learning. For clean presentation, we denote \( A = \max_{j \in [2]} A_j \).

**Theorem 5.** Suppose subroutine \( \text{ADV\_BANDIT\_UPDATE} \) satisfies Assumption 7. For any \( \delta \in (0, 1) \) and \( K \in \mathbb{N} \), let \( \iota = \log(\text{HSAK}/\delta) \). Choose learning rate \( \alpha_t \) according to equation 3 and bonus \( \{\beta_{j,t}\}_{t=1}^{K} \) of the \( j \)-th player so that \( \sum_{t=1}^{T} \alpha_t \beta_t = \Theta(H^{5} \xi(A_j, t, \iota) + \sqrt{H^{5} \iota / t}) \) for any \( t \in [K] \). Then, with probability at least \( 1 - \delta \), after running Algorithm 7 for \( K \) episodes, let \( \hat{\pi}_1, \hat{\pi}_2 \) be the output policies by Algorithm 5 for each player, then we have the product policy \( \hat{\pi} = \hat{\pi}_1 \times \hat{\pi}_2 \) satisfies

\[
\max_{j \in [2]} |V_{j,1}^{\hat{\pi}_1} - V_{j,1}^{\pi_1}(s_1) - V_{j,2}^{\hat{\pi}_2} + V_{j,2}^{\pi_2}(s_1)| \leq O((H^{2}S/K) \cdot \Xi(A, K/S, \iota) + \sqrt{H^{5}S/K}).
\]

When instantiating \( \text{ADV\_BANDIT\_UPDATE} \) by FTRL (Algorithm 5), we can choose \( \beta_{j,t} = c \cdot \sqrt{H^{3} \xi / t} \) for some absolute constant \( c \), where leads to max \( j \in [2] |V_{j,1}^{\hat{\pi}_1} - V_{j,2}^{\pi_2}(s_1)| \leq O(\sqrt{H^{5}SA/K}) \).

Theorem 5 claims that, to find an \( \epsilon \)-approximate Nash equilibrium, we only need to use a number of episodes \( K = \tilde{O}(H^{5}SA/c^2) \), where \( A = \max_{j \in [2]} A_j \). In contrast, value iteration or Q-learning
based algorithms require at least $\Omega(H^3SA_1A_2/\epsilon^2)$ episodes to find Nash equilibria [Bai et al., 2020; Liu et al., [2021]. Furthermore, V-learning is a fully decentralized algorithm. To our best knowledge, V-learning is the only algorithm up to today that achieves sample complexity linear in $A$ for finding Nash equilibrium in two-player zero-sum Markov games.

We remark that V-learning only performs $O(1)$ operations and calls subroutine $\text{ADV\_BANDIT\_UPDATE}$ once every time a new sample is observed. As long as the adversarial bandit algorithm used in V-learning is computationally efficient (which is the case for FTRL), V-learning itself is also computationally efficient.

5 Monotonic V-Learning

In the previous sections, we present the V-learning algorithm whose output policy (Algorithm 3) is a nested mixture of Markov policies. Storing such a output policy requires $O(HSA_1K)$ space for the $j$th player. In Section 3 we argue this approach has a significant advantage over directly storing a general correlated policy when the number of agents is large. Nevertheless, this space complexity can be undesirable when the number of agents is small.

In this section, we introduce a simple monotonic techique to V-learning, which allows each agent to output a Markov policy when finding Nash equilibria in the two-player zero-sum setting. Storing a Markov policy only takes $O(HSA_j)$ space for the $j$th player. A similar result for the single-agent setting can be immediately obtained by setting the second player in the Markov game to be a dummy player with only a single action to choose from.

Monotonic update Monotonic V-learning is almost the same as V-learning with only the Line 8 in Algorithm 1 changed to

$$V_h(s_h) \leftarrow \min\{H + 1 - h, \tilde{V}_h(s_h), V_h(s_h)\},$$

(5)

This step guarantees $V_h(s_h)$ to monotonically decrease at each step. This is helpful because in two-player zero-sum Markov games, all Nash equilibria share a unique value which we denote as $V^*$. By design, we can prove that the $V$-values maintained in V-learning are high probability upper bounds of $V^*$ (Lemma 18). This monotonic update allows our V-value estimates to always get closer to $V^*$ after each update, which improves the accuracy of our V-value estimates.

Markov output policy For an arbitrary fixed $(s, h) \in S \times [H]$, let $t_3$ be the last episode when the value $V_{1,h}(s)$ is updated (i.e., strictly decreases), and let $t_2$ be the last episode when the value $V_{2,h}(s)$ is updated. Then the output policy for this $(s, h)$ has the following form.

$$\tilde{\pi}_{1,h}(\cdot|s) := \sum_{i=1}^{t_2} \alpha_i^2 \pi^i_{1,h}(\cdot|s), \quad \tilde{\pi}_{2,h}(\cdot|s) := \sum_{i=1}^{t_1} \alpha_i^1 \pi^i_{2,h}(\cdot|s),$$

(6)

where $k^i$ denotes the index of episode when state $s$ is visited at step $h$ is visited for the $i$th time. Recall that $\pi^i_{j,h}(\cdot|s)$ is the policy maintained by the $j$th player at the beginning of the $k^i$th episode when she runs V-learning. That is, the new output policy is simply the weighted average of policies computed in the V-learning at each $(s, h)$ pair. Clearly, the policies $\tilde{\pi}_1$ and $\tilde{\pi}_2$ defined by equation 6 are Markov policies.

We remark that although the execution of $\tilde{\pi}_1$ and $\tilde{\pi}_2$ can be fully decentralized, in equation 6 the computation of $\tilde{\pi}_{1,h}(\cdot|s)$ depends on $t_3$ while the computation of $\tilde{\pi}_{2,h}(\cdot|s)$ depends on $t_1$. That is, two players need to communicate at the end the indexes of the most recent episodes when their $V$-values are updated. As a result, monotonic V-learning is not fully decentralized.

Theorem 6. Monotonic V-learning with output policy $\tilde{\pi} = \tilde{\pi}_1 \times \tilde{\pi}_2$ as specified by equation 6 has the same theoretical guarantees as Theorem 5 with the same choices of hyperparameters.

Theorem 6 asserts that V-learning can be modified to output Markov policies when finding Nash equilibria of two-player zero-sum Markov games. As a special case, the same technique and results directly apply to the single-agent setting.
REFERENCES


Presented at the Gamification and Multiagent Solutions Workshop (ICLR 2022)


Tor Lattimore and Csaba Szepesvári. Bandit algorithms. 2018.


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


A RELATED WORK

In this section, we focus our attention on theoretical results for the tabular setting, where the numbers of states and actions are finite. We acknowledge that there has been much recent work in RL for continuous state spaces (see, e.g., Jiang et al., 2017; Jin et al., 2020; Zanette et al., 2020; Lin et al., 2021a; Xie et al., 2020; Jin et al., 2021b), but this setting is beyond our scope.

Markov games. Markov Game (MG), also known as stochastic game (Shapley, 1953), is a popular model in multi-agent RL (Littman, 1994). Early works have mainly focused on finding Nash equilibria of MGs under strong assumptions, such as known transition and reward (Littman, 2001; Hu & Wellman, 2003; Hansen et al., 2013; Wei et al., 2020), or certain reachability conditions (Wei et al., 2017, 2021) (e.g., having access to simulators (Jia et al., 2019; Sidford et al., 2020; Zhang et al., 2020) that alleviate the challenge in exploration.

A recent line of works provide non-asymptotic guarantees for learning two-player zero-sum tabular MGs without further structural assumptions. Bai & Jin (2020) and Xie et al. (2020) develop the first provably-efficient learning algorithms in MGs based on optimistic value iteration. Liu et al. (2021) improves upon these works and achieve best-known sample complexity for finding an ε-Nash equilibrium—O(H^3SA_1A_2/ε^2) episodes.

For multiplayer general-sum tabular MGs, Liu et al. (2021) is the only existing work that provides non-asymptotic guarantees in the exploration setting. It proposes centralized model-based algorithms based on value-iteration, and shows that Nash equilibria (although computationally inefficient), CCE and CE can be all learned within O(H^4SA_1S^2A_2/ε^2) episodes. Note this result suffers from the curse of multiagents.

V-learning—initially coupled with the FTRL algorithm as adversarial bandit subroutine—is firstly proposed in the conference version of this paper Bai et al. (2020), for finding Nash equilibria in the two-player zero-sum setting. During the preparation of this draft, we note two very recent independent works Song et al. (2021), Mao & Başar (2021), whose results partially overlap with the results of this paper in the multiplayer general-sum setting. In particular, Mao & Başar (2021) use V-learning with stabilized online mirror descent as adversarial bandit subroutine, and learn ε-CCE in O(H^6SA_1/ε^2) episodes, where A = max_{j∈[m]} A_j. This is one H factor larger than what is required in Theorem 7 of this paper. Song et al. (2021) considers similar V-learning style algorithms for learning both ε-CCE and ε-CE. For the latter objective, they require O(H^6SA_1^2/ε^2) episodes which is again one H factor larger than what is required in Theorem 8 of this paper. Song et al. (2021) also considers learning Markov potential games by algorithms that are not V-learning, which is beyond the scope of this paper. We remark that both parallel works have not presented V-learning as a generic class of algorithms which can be coupled with any adversarial bandit algorithms with suitable regret guarantees in a black-box fashion.

Strategic games. Strategic game is one of the most basic game forms studied in the game theory literature Osborne & Rubinstein (1994). It can be viewed as Markov games without state and transition. The fully decentralized algorithm that breaks the curse of multiagents is known in the setting of strategic games. By independently running no-regret (or no-swap-regret) algorithm for all agents, one can find Nash Equilibria (in the two-player zero-sum setting), correlated equilibria and coarse correlated equilibria (in the multiplayer general-sum setting) in a number of samples that only scales with max_{j∈[m]} A_j [Cesa-Bianchi & Lugosi, 2006; Hart & Mas-Colell, 2000; Blum & Mansour, 2007]. However, such successes do not directly extend to the Markov games due to the additional temporal structures involving both states and transition. In particular, there is no computationally efficient no-regret algorithm for Markov games [Radanovic et al., 2019; Bai et al., 2020].

Extensive-form games. There is another long line of research on MARL based on the model of extensive-form games (EFG) (see, e.g., Koller & Megiddo, 1992; Gilpin & Sandholm, 2006; Zinkevich et al., 2007; Brown & Sandholm, 2018, 2019; Celli et al., 2020). EFGs can be viewed as

---

1We remark that, while Wei et al. (2017) studies the infinite-horizon average-reward setting, a recent refined analysis tailored to the episodic setting shows that the algorithms proposed in Wei et al. (2017) can learn Nash equilibria in two-player zero-sum MGs without additional reachability assumptions in O(H^4S^2A_1A_2/ε^2) (Wei, 2021).
special cases of Markov games where any state at the $h$th step can be reached from only one state at the $(h - 1)$th step (due to tree structure of the game). Therefore, results on learning EFGs do not directly imply results for learning MGs.

Decentralized MARL. There is a long line of empirical works on decentralized MARL (see, e.g., Lowe et al. [2017] [Iqbal & Sha [2019] [Sunahag et al. [2017] [Rashid et al. [2018] [Son et al. [2019]. A majority of these works focus on the cooperative setting. They additionally attack the challenge where each agent can only observe a part of the underlying state, which is beyond the scope of this paper. For theoretical results, Zhang et al. [2018] consider the cooperative setting while Sayin et al. [2021] study the two-player zero-sum Markov games. Both develop decentralized MARL algorithms but provide only asymptotic guarantees. Daskalakis et al. [2021] analyze the convergence rate of independent policy gradient method in episodic two-player zero-sum MGs. Their result requires the additional reachability assumptions (concentrability) which alleviates the difficulty of exploration.

Single-agent RL. There is a rich literature on reinforcement learning in MDPs (see e.g. Jaksch et al. [2010] Osband et al. [2014] Azar et al. [2017] Dann et al. [2017] Strehl et al. [2006] Jin et al. [2018] [2021a]). MDPs are special cases of Markov games, where only a single agent interacts with a stochastic environment. For the tabular episodic setting with nonstationary dynamics and no simulators, the best sample complexity achieved by existing model-based and model-free algorithms are $\tilde{O}(H^3SA/\epsilon^2)$ (achieved by value iteration Azar et al. [2017]) and $\tilde{O}(H^4SA/\epsilon^2)$ (achieved by Q-learning Jin et al. [2018]), respectively, where $S$ is the number of states, $A$ is the number of actions, $H$ is the length of each episode. Both of them (nearly) match the lower bound $\Omega(H^4SA/\epsilon^2)$ [Jaksch et al. [2010] Osband & Van Roy [2016] Jin et al. [2018].

B Multiplayer General-sum Markov Games

In multiplayer general-sum games, finding Nash equilibria is computationally hard in general (which is technically PPAD-complete Daskalakis [2013]). In this section, we focus on finding two commonly-used alternative notions of equilibria in the game theory—coarse correlated equilibria (CCE), and correlated equilibria (CE). Both are relaxed notions of Nash equilibra.

B.1 Finding Coarse Correlated Equilibria

The algorithm for finding CCE is again running V-learning (Algorithm 1) independently for each agent $j$ with learning rate $\alpha_t$ (as specified in equation 3) and bonus $\{\beta_{j,t}\}$ (to be specified later). The major difference from the case of finding Nash equilibria is that CCE and CE require the output policy to be joint correlated policy. We achieve this correlation by feeding the same random seed to all agents at the very beginning when they execute the output policy according to Algorithm 3. That is, while training can be done in the fully decentralized fashion, we require one round of communication at the beginning of the execution to broadcast the shared random seed. After that, each agent can simply execute her own output policy independently. During the execution, since the states visited are shared among all agents, shared random seed allows the same index $i$ to be sampled across all agents in the Step 4 of Algorithm 3 at every step. We denote this correlated joint output policy as $\hat{\pi} = \hat{\pi}_1 \otimes \ldots \otimes \hat{\pi}_m$.

We remark that to specify a correlated policy in general, we need to specify the probability for taking all action combinations $\{a_1, \ldots, a_m\}$ for each $(s, h)$. This requires at least $\Omega(HS\prod_{j=1}^m A_j)$ space, which grows exponentially with the number of agents $m$. The way V-learning specifies the joint policy only requires agents to store their own intermediate counters and policies computed during training. This only takes a total of $O(HSK(\sum_{j=1}^m A_j))$ space, which scales only linearly with the number of agents. Our approach dramatically improve over the former approach in space complexity when the number of agents is large.

We now present the guarantees for V-learning to learn a CCE as follows. Let $A = \max_{j \in [m]} A_j$.

Theorem 7. Suppose subroutine $\text{ADV.BANDIT.UPDATE}$ satisfies Assumption 2. For any $\delta \in (0, 1)$ and $K \in \mathbb{N}$, let $t = \log(HSAK/\delta)$. Choose learning rate $\alpha_t$ according to equation 3 and bonus $\{\beta_{j,t}\}_{t=1}^K$ of the $j$th player so that $\sum_{t=1}^{\bar{t}} \alpha_t \beta_{j,t} = \Theta(HS(A_j, t, t) + \sqrt{H^3t/\epsilon})$ for any $t \in [K]$. Then,
with probability at least $1 - \delta$, after all the players running Algorithm 2 for $K$ episodes, let $\hat{\pi}_j$ be the output policy by Algorithm 3 for the $j$th player, then we have the joint policy $\hat{\pi} = \hat{\pi}_1 \odot \ldots \odot \hat{\pi}_m$ satisfies

$$\max_{j \in [m]} \left[ V^\dagger_{j,1} \hat{\pi}^{-j}(s_1) - V^\dagger_{j,1}(s_1) \right] \leq O((H^2S/K) \cdot \Xi(A, K/S, \epsilon) + \sqrt{H^5St/K}).$$

When instantiating Adv_Bandit_Update by FTRL (Algorithm 5), we can choose $\beta_{j,t} = c \cdot \sqrt{H^3A_j/\epsilon}$ for some absolute constant $c$, which leads to $\max_{j \in [m]} \left[ V^\dagger_{j,1} \hat{\pi}^{-j}(s_1) - V^\dagger_{j,1}(s_1) \right] \leq O((H^3SA_t/K)).$

Theorem 7 claims that, to find an $\epsilon$-approximate CCE, V-learning only needs to use a number of episodes $K = O(H^3SA_t/\epsilon^2)$, where $A = \max_{j \in [m]} A_j$. This is in sharp contrast to the prior results for multiplayer general-sum Markov games, which use value iteration based algorithms, and require at least $\Omega(H^4S^2 (\prod_{j=1}^m A_j)/\epsilon^2)$ episodes [Liu et al. 2021]. As a result, V-learning is the first algorithm that breaks the curse of multiagents for finding CCE in Markov games.

### B.2 Finding Correlated Equilibria

The algorithm for finding CE is almost the same as the algorithm for finding CCE except that we now require a different Adv_Bandit_Update subroutine, which has the following high probability weighted swap regret guarantee.

**Assumption 2.** For any $t \in \mathbb{N}$ and any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\max_{\psi \in \Psi} \sum_{i=1}^t \alpha_i^j [\langle \theta_i, l_i \rangle - \langle \psi \circ \theta_i, l_i \rangle] \leq \xi_m(B, t, \log(1/\delta)). \quad (7)$$

We assume the existence of an upper bound $\Xi_m(B, t, \log(1/\delta)) \geq \sum_{t'=1}^{t'} \xi_m(B, t', \log(1/\delta))$ where (i) $\xi_m(B, t, \log(1/\delta))$ is non-decreasing in $B$ for any $t, \delta$; (ii) $\Xi_m(B, t, \log(1/\delta))$ is concave in $t$ for any $B, \delta$.

Here $\Psi$ denotes the set $\{ \psi : B \to B \}$ which consists of all maps from actions in $B$ to actions in $B$. Meanwhile, for any $\theta \in \Delta_B$, the term $\psi \circ \theta \in \Delta_B$ denotes the distribution over actions where $\psi \circ \theta(b) = \sum_{b' : \psi(b') = \theta} \theta(b')$. We note that bounded swap regret is a stronger requirement compared to bounded external regret as in equation 4 since by maximizing over a subset of functions in $\Psi$ which map all actions in $B$ to one single action, we recover the external regret by equation 7.

Assumption 2 can be satisfied by modifying many existing algorithms with external regret to the swap regret setting. In particular, we prove that the Follow-the-Regularized-Leader for swap regret (FTRL_swap) algorithm (Algorithm 6) satisfies Assumption 2 with bounds $\xi_m(B, t, \log(1/\delta)) \leq O(B\sqrt{H\log(B/\delta)/t})$ and $\Xi_m(B, t, \log(1/\delta)) \leq O(B\sqrt{Ht \log(B/\delta)})$. Both bounds have one extra $\sqrt{B}$ factor comparing to the counterparts in external regret. We refer readers to Appendix A for more details.

We now present the guarantees for V-learning to learn a CCE as follows. Let $A = \max_{j \in [m]} A_j$.

**Theorem 8.** Suppose subroutine Adv_Bandit_Update satisfies Assumption 2. For any $\delta \in (0, 1)$ and $K \in \mathbb{N}$, let $t = \log(mHSAK/\delta)$. Choose learning rate $\alpha_i$ according to equation 3 and bonus $\{\beta_{j,t}\}_{i=1}^K$ of the $j$th player so that $\sum_{i=1}^t \alpha_i^j \beta_{j,i} = \Theta(H\xi_m(A_j, t, \epsilon) + \sqrt{H^3t/\epsilon})$ for any $t \in [K]$. Then, with probability at least $1 - \delta$, after all the players running Algorithm 2 for $K$ episodes, let $\hat{\pi}_j$ be the output policy by algorithm 3 for the $j$th player, then we have the joint policy $\hat{\pi} = \hat{\pi}_1 \odot \ldots \odot \hat{\pi}_m$ satisfies

$$\max_{j \in [m]} \max_{\phi_j} [V^{\phi_j \odot \hat{\pi}}_{j,1}(s_1) - V^{\hat{\pi}}_{j,1}(s_1)] \leq O((H^2S/K) \cdot \Xi_m(A, K/S, \epsilon) + \sqrt{H^5St/K}).$$

When instantiating Adv_Bandit_Update by FTRL_swap (Algorithm 5), we can choose $\beta_{j,t} = c \cdot A_j \sqrt{H^3t/\epsilon}$ for some absolute constant $c$, which leads to $\max_{j \in [m]} \max_{\phi_j} [V^{\phi_j \odot \hat{\pi}}_{j,1}(s_1) - V^{\hat{\pi}}_{j,1}(s_1)] \leq O(A \sqrt{H^3St/K}).$
Finally, for simplicity of notation, we define two operators \( P \in A \) \( \subset \) Nash Proposition 9 games. We first present a proposition which clarify the relations among the three different kind of equilibria.

This allows us to define the corresponding best response of \( \pi \) at \( h \) step as \( \phi_h := \{ \phi_{i,h'} : (S \times A_i)^{h'-h} \times S \rightarrow A_i \} \) for strategies starting from the \( h \) step. We also introduce a new policy \( \hat{\pi}^k \) that can be viewed as a part of the output policy in Algorithm 3. The definition of \( \hat{\pi}^k \) is very similar to \( \hat{\pi} \) except two differences: (1) \( \hat{\pi}^k \) is a policy for step \( h, \ldots, H \) while \( \hat{\pi} \) is a policy for step 1, \ldots, \( H \); (2) in \( \hat{\pi} \) the initial value of \( k \) is sampled uniformly at random from \( [K] \) at the very beginning while in \( \hat{\pi}^k \) the initial value of \( k \) is given.

We remark that \( \hat{\pi}^k \) is a non-Markov policy that does not depends on history before the \( h \) step. In particular, we show that, similar to strategic games, we also have Nash \( \subset \) CE in Markov games.

Theorem [8] claims that, to find an \( \epsilon \)–approximate CE, V-learning only needs to use a number of episodes \( K = \mathcal{O}(H^2SA^2/\epsilon^2) \), where \( A = \max_{j \in [m]} A_j \). It has an extra \( A \) multiplicative factor comparing to the sample complexity of finding CCE, since CE is a subset of CCE thus finding CE is expected to be more difficult. Nevertheless, the sample complexity presented here is far better than value iteration based algorithm, which requires at least \( \Omega(H^2SA^2(\prod_{i=1}^m A_i)/\epsilon^2) \) episodes for finding CE [Liu et al. 2021]. V-learning is also the first algorithm that breaks the curse of multiagents for finding CE in Markov games.

C Notations and Basic Lemmas

C.1 Notations

In this subsection, we introduce some notations that will be frequently used in appendixes. Recall that we use \( V^k, N^k, \pi^k \) to denote the value, counter and policy maintained by V-learning algorithm at the beginning of the episode \( k \).

We also introduce a new policy \( \hat{\pi}^k \) for a single agent (defined by its execution in Algorithm 4), which can be viewed as a part of the output policy in Algorithm [3]. The definition of \( \hat{\pi}^k \) is very similar to \( \hat{\pi} \) except two differences: (1) \( \hat{\pi}^k \) is a policy for step \( h, \ldots, H \) while \( \hat{\pi} \) is a policy for step 1, \ldots, \( H \); (2) in \( \hat{\pi} \) the initial value of \( k \) is sampled uniformly at random from \( [K] \) at the very beginning while in \( \hat{\pi}^k \) the initial value of \( k \) is given.

We remark that \( \hat{\pi}^k \) is a non-Markov policy that does not depends on history before the \( h \) step. In symbol, we can express this class of policy as \( \pi_j := \{ \pi_j,h' : \Omega \times (S \times A)^{h'-h} \times S \rightarrow A_j \} \) for strategies starting from the \( h \) step, and denote it as \( \Pi_h \). Similar to Section 2 we can also define joint policy \( \pi = \pi_1 \otimes \ldots \otimes \pi_m \) and product policy \( \pi = \pi_1 \times \ldots \times \pi_m \) for policies in \( \Pi_h \). We can also define value \( V^k(\pi) \) for joint policy \( \pi \in \Pi_h \) as

\[
V^k_\pi(s) := \mathbb{E}_\pi \left[ \sum_{h'=h}^{H} r_{i,h'}(s_{h'}, a_{h'}) \Big| s_h = s \right].
\]

This allows us to define the corresponding best response of \( \pi_{-i} \) as the maximizer of \( \max_{\pi_i} \mathbb{E}_{\pi_{-i}} \left[ V^k_{\pi_i \times \pi_{-i}}(s) \right] \). We also denote this maximum value as \( V^{i,h}_{\pi_{-i}}(s) \). We define the strategy modification for policies starting from the \( h \) step as \( \phi_h := \{ \phi_{i,h'} : (S \times A_i)^{h'-h} \times S \rightarrow A_i \} \) for policies in \( \Pi_h \) and denote the set of such strategy modification as \( \Phi_h \).

Finally, for simplicity of notation, we define two operators \( \mathbb{P} \) and \( \mathbb{D} \) as follows:

\[
\begin{align*}
\mathbb{P}_h[V](s, a) &= \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot | s, a)}[V(s')], \\
\mathbb{D}_\pi(Q)(s) &= \mathbb{E}_{a \sim \pi(\cdot | s)}[Q(s, a)],
\end{align*}
\]

for any value function \( V, Q \) and any one-step Markov policy \( \pi \).

C.2 Basic Lemmas

We first present a proposition which clarify the relations among the three different kind of equilibria. In particular, we show that, similar to strategic games, we also have Nash \( \subset \) CE \( \subset \) CCE in Markov games.

**Proposition 9** (Nash \( \subset \) CE \( \subset \) CCE). In Markov games, any \( \epsilon \)–approximate Nash equilibrium is an \( \epsilon \)–approximate CE, and any \( \epsilon \)–approximate CE is an \( \epsilon \)–approximate CCE.
Finally, step (d) follows from \( \pi \). We can consider a subset of strategy modification \( \phi \). Step (c) is because by definition of strategy modification \( \phi \) is equivalent to maximizing over a new independent policy. Step (b) directly follows from \( \pi \) being an \( \epsilon \)-approximate Nash equilibrium. By definition, this proves that \( \pi \) is also an \( \epsilon \)-approximate CE.

For CE \( \subset \) CCE, let \( \pi = \pi_1 \otimes \pi_2 \otimes \cdots \otimes \pi_m \) be an \( \epsilon \)-approximate CE, then we have

\[
\max_{\phi_i} V_{i,1}^{(\phi_i, \pi_i)}(s_1) \leq \max_{\phi_i} V_{i,1}^{\pi_i}(s_1) + \epsilon,
\]

Step (c) is because by definition of strategy modification \( \phi_i := \{ \phi_{i,h} : (S \times A)^{h-1} \times S \times A_i \rightarrow A_i \} \), we can consider a subset of strategy modification \( \phi'_i := \{ \phi'_{i,h} : (S \times A)^{h-1} \times S \rightarrow A_i \} \) which modifies the policy ignoring whatever the action \( \pi_i \) takes. It is not hard to see that maximizing over the strategy modification in this subset is equivalent to maximizing over a new independent policy \( \pi'_i \). Therefore, maximizing over all strategy modification is greater or equal to maximizing over \( \pi'_i \). Finally, step (d) follows from \( \pi \) being an \( \epsilon \)-approximate CE. By definition, this proves that \( \pi \) is also an \( \epsilon \)-approximate CCE.

Proof. We prove two claims separately.

For Nash \( \subset \) CE, let \( \pi = \pi_1 \times \pi_2 \times \cdots \times \pi_m \) be an \( \epsilon \)-approximate Nash equilibrium, then

\[
\max_{\phi_i} V_{i,1}^{(\phi_i, \pi_i)}(s_1) \leq \max_{\phi_i} V_{i,1}^{\pi_i}(s_1) + \epsilon,
\]

Step (a) is because that \( \pi \) is a product policy, where the randomness of different agents are completely independent. In this case, maximizing over strategy modification \( \phi_i \) is equivalent to maximizing over a new independent policy. Step (b) directly follows from \( \pi \) being an \( \epsilon \)-approximate Nash equilibrium. By definition, this proves that \( \pi \) is also an \( \epsilon \)-approximate CE.

Next, we present some basic lemmas that will be used in the proofs of different theorems. We start by introducing some useful properties of sequence \( \{\alpha_t^i\} \) defined in equation (9).

**Lemma 10.** (\cite{Jin:2018:RML} Lemma 4.1), \cite{tian2021} Lemma 2) The following properties hold for \( \alpha_t^i \):

1. \( \frac{1}{\sqrt{t}} \leq \sum_{t=1}^{T} \alpha_t^i \leq \frac{2}{\sqrt{t}} \) and \( \frac{1}{t} \leq \sum_{t=1}^{T} \alpha_t^i \leq \frac{2}{t} \) for every \( t \geq 1 \).
2. \( \max_{t \in [T]} \alpha_t^i \leq \frac{2H}{t} \) and \( \sum_{t=1}^{T} (\alpha_t^i)^2 \leq \frac{2H^2}{t} \) for every \( t \geq 1 \).
3. \( \sum_{t=1}^{\infty} \alpha_t^i = 1 + \frac{1}{H} \) for every \( i \geq 1 \).

Finally, we have the following lemma which express the \( \hat{V} \) maintained in V-learning in the form of weighted sum of earlier updates.

**Lemma 11.** Consider an arbitrary fixed \((s, h, k)\) tuple. Let \( t = N_h^k(s) \) denote the number of times \( s \) is visited at step \( h \) at the beginning of episode \( k \), and suppose \( s \) was previously visited at episodes \( k^1, \ldots, k^t < k \) at the \( h \)-th step. Then the two V-values \( \hat{V} \) and \( V \) in Algorithm 7 satisfy the following equation:

\[
\hat{V}_{j,h}^k(s) = \alpha_t^0(H-h+1) + \sum_{i=1}^{t} \alpha_t^i \left[ r_{j,h}(s, a_h^k) + V_{j,h+1}(s_{h+1}^k) + \beta_{j,i} \right], \quad j \in [m]. \tag{9}
\]

Proof. The proof follows directly from the update rule in Line 7 Algorithm 1. Note that \( \alpha_t^0 \) is equal to zero for any \( t > 1 \) and equal to one for \( t = 0 \).

**D PROOFS FOR COMPUTING CCE IN GENERAL-SUM MGs**

In this section, we give complete proof of Theorem 7. To avoid repeatedly state the condition of Theorem 7 in each lemma, we will use

- Condition of the adversarial bandit sub-procedure (Assumption 1) and
- Set the bonus \( \sum_{i=1}^{t} \alpha_t^i \beta_{j,i} = \Theta(H \xi(A, t, i) + \sqrt{H^3 t/i}) \) for any \( t \in [K] \).
throughout the whole section.

The following Lemma is a direct consequence of Assumption 1 which will play an important role in our later analysis.

**Lemma 12.** Under Assumption 1 the following event is true with probability at least $1 - \delta$: for any $(s, h, k) \in \mathcal{S} \times [H] \times [K]$, let $t = N^k_h(s)$ and suppose $s$ was previously visited at episodes $k^1, \ldots, k^t < k$ at the $h$-th step, then for all $j \in [m]$

$$
\max_{\mu} \sum_{i=1}^{t} \alpha_{i, \mu \times \pi_{j,h}}^{k_i} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{k_i} \right) (s) - \sum_{i=1}^{t} \alpha_{i, \mu \times \pi_{j,h}}^{k_i} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{k_i} \right) (s) \leq H \xi(A_j, t, \epsilon),
$$

where $\epsilon = \log(mHSAK/\delta)$.

**Proof.** By Assumption 1 and the adversarial bandit update step in Algorithm 1 we have that with probability at least $1 - \delta$, for any $(s, h, k) \in \mathcal{S} \times [H] \times [K]$,

$$
\max_{\mu} \sum_{i=1}^{t} \alpha_{i, \mu \times \pi_{j,h}}^{k_i} \left( \frac{H - r_{j,h} - \mathbb{P}_{h} V_{j,h+1}^{k_i}}{H} \right) (s) - \sum_{i=1}^{t} \alpha_{i, \mu \times \pi_{j,h}}^{k_i} \left( \frac{H - r_{j,h} - \mathbb{P}_{h} V_{j,h+1}^{k_i}}{H} \right) (s) \leq \xi(A_j, t, \epsilon),
$$

which implies the desired result by simple algebraic transformation. \qed

Then we show $V$ is actually an optimistic estimation of the value function of player $j$’th best response to the output policy.

**Lemma 13 (Optimism).** For any $\alpha \in (0, 1]$, with probability at least $1 - \delta$, for any $(s, h, k, j) \in \mathcal{S} \times [H] \times [K] \times [m]$, $V_{j,h}^k(s) \geq V_{j,h}^{1, \hat{\pi}_{j,h}}(s)$.

**Proof of Lemma 13** We prove by backward induction. The claim is trivially satisfied for $h = H + 1$.

Suppose it is true for $h + 1$, consider a fixed state $s$. It suffices to show $V_{j,h}^k(s) \geq V_{j,h}^{1, \hat{\pi}_{j,h}}(s)$ because $V_{j,h}^k(s) = \min V_{j,h}^k(s), H - h + 1$. Let $t = N^k_h(s)$ and suppose $s$ was previously visited at episodes $k^1, \ldots, k^t < k$ at the $h$-th step. Then using Lemma 11

$$
\hat{V}_{j,h}(s) = \alpha_{t}^0 H + \sum_{i=1}^{t} \alpha_{i} \left( r_{j,h}(s, \alpha^k_{h,i}) + V_{j,h+1}(s_{h+1}) + \beta_{j,i} \right)
$$

\[\geq \sum_{i=1}^{t} \alpha_{i} \mathbb{E} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{k_i} \right) (s) + \sum_{i=1}^{t} \alpha_{i} \beta_{j,i} - \mathcal{O} \left( \sqrt{\frac{H \delta}{t}} \right) \]

\[\geq \max_{\mu} \sum_{i=1}^{t} \alpha_{i} \mathbb{E} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{k_i} \right) (s) + \sum_{i=1}^{t} \alpha_{i} \beta_{j,i} - \mathcal{O} \left( \sqrt{\frac{H \delta}{t}} \right) - H \xi(A_j, t, \epsilon) \]

\[\geq \max_{\mu} \sum_{i=1}^{t} \alpha_{i} \mathbb{E} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{k_i} \right) (s) \]

\[\geq \max_{\mu} \sum_{i=1}^{t} \alpha_{i} \mathbb{E} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{1, \hat{\pi}_{j,h}} \right) (s) \]

\[\geq \max_{\mu} \sum_{i=1}^{t} \alpha_{i} \mathbb{E} \left( r_{j,h} + \mathbb{P}_{h} V_{j,h+1}^{1, \hat{\pi}_{j,h}} \right) (s) \]

where (i) is by martingale concentration and Lemma 2 (ii) is by Lemma 12 (iii) is by the definition of $\beta_{j,i}$, and (iv) is by induction hypothesis.

Finally, we remark that (v) is not directly from Bellman equation since $\hat{\pi}_{j,h}$ is non-Markov policy and the best response of a non-Markov policy is not necessary a Markov policy. We prove (v) as
follows. Recalls definitions for policies in $\Pi_h$ as in Appendix [C] by the definition, we have

\[
V_{j,h}^{\hat{\pi}_h} (s) = \max_{\mu \in \Pi_h} V_{j,h}^{\mu \times \hat{\pi}_h} (s) = \max_{\mu \in \Pi_h} \max_{(h+1) : H} \sum_{t=1}^{H-h} \alpha_t \mathbb{E}_{a \sim \mu_h \times \pi_{j,h}^k} \left( r_{j,h} (s, a) + \mathbb{E}_{s'} V_{j,h}^{\mu_{(h+1) : H} \times \hat{\pi}_{j,h+1}^k} (s, a, s') \right)
\]

\[
\leq \max_{\mu_h} \sum_{i=1}^{t} \alpha_t \mathbb{E}_{a \sim \mu_h \times \pi_{j,h}^k} \left( r_{j,h} (s, a) + \mathbb{E}_{s'} \max_{\mu_{(h+1) : H}} V_{j,h+1}^{\mu_{(h+1) : H} \times \hat{\pi}_{j,h+1}^k} (s, a, s') \right)
\]

\[
= \max_{\mu} \sum_{i=1}^{t} \alpha_t \mathbb{D}_{\mu \times \pi_{j,h}^k} \left( r_{j,h} + \mathbb{P}_h V_{j,h+1}^{\hat{\pi}_{j,h+1}^k} \right) (s)
\]

where $V_{j,h}^{\pi_h} (s, a, s')$ for policy $\pi \in \Pi_h$ is defined as:

\[
V_{j,h+1}^{\pi_h} (s, a, s') := \mathbb{E}_\pi \left[ \sum_{h'=h+1}^{H} r_{h'} (s_{h'}, a_{h'}) \right] \quad s_h = s, a_h = a, s_{h+1} = s'.
\]

Step (a) uses the relation between $\hat{\pi}_{j,h}^k$ and $\{\hat{\pi}_{j,h+1}^k\}_i$. Step (b) pushes max inside summation and expectation. Step (c) is because the Markov nature of Markov game and that $\{\hat{\pi}_{j,h+1}^k\}_i$ are policies that do not depend on history at step $h$, we know the maximization over $\mu_{(h+1) : H}$ is achieved at policies in $\Pi_{h+1}$. This finishes the proof. \(\square\)

To proceed with the analysis, we need to introduce two pessimistic V-estimations $\hat{V}$ and $\tilde{V}$ that are defined similarly as $\hat{V}$ and $\tilde{V}$. Formally, let $t = N_h^k (s)$ denote the number of times $s$ is visited at step $h$ at the beginning of episode $k$, and suppose $s$ was previously visited at episodes $k^1, \ldots, k^t < k$ at the $h$-th step. Then

\[
V_{j,h}^k (s) = \sum_{i=1}^{t} \alpha_t \left[ r_{j,h} (s, a_h^k) + \tilde{V}_{j,h+1}^{k} (s_{h+1}) - \beta_{j,s} \right], \quad (10)
\]

\[
V_{j,h}^k (s) = \max \{0, V_{j,h}^k (s)\}, \quad (11)
\]

for any player $j \in [m]$ and $k \in [K]$. We emphasize that $\hat{V}$ and $\tilde{V}$ are defined only for the purpose of analysis. Neither do they influence the decision made by each agent, nor do the agents need to maintain these quantities when running V-learning.

Equipped with the lower estimations, we are ready to lower bound $V_{j,h}^{\hat{\pi}_h}$. \(\square\)

**Lemma 14 (Pessimism).** For any $\delta \in (0, 1]$, with probability at least $1 - \delta$, the following holds for any $(s, h, k, j) \in S \times [H] \times [K] \times [m]$ and any player $j$, $V_{j,h}^k (s) \leq V_{j,h}^{\hat{\pi}_h} (s)$.

**Proof of Lemma 14** We prove by backward induction. The claim is trivially satisfied for $h = H+1$. Suppose it is true for $h + 1$, consider a fixed state $s$. It suffices to show $V_{j,h}^k (s) \leq V_{j,h}^{\hat{\pi}_h} (s)$ because $V_{j,h}^k (s) = \max \{ V_{j,h}^k (s), 0 \}$. 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. Then by Equation 10
\[ V_{j,h}^k(s) = \sum_{i=1}^{t} \alpha_i [r_{j,h}(s, a^k_{i}) + V_{j,h+1}^{k}(s^k_{h+1}) - \beta_{j,i}] \]
\[ \leq \sum_{i=1}^{t} \alpha_i \mathbb{P}_{i,k} \left( r_{h} + \mathbb{P}_{h} V_{j,h+1}^{k} \right)(s) - \sum_{i=1}^{t} \alpha_i \beta_{j,i} + O \left( \frac{H^3}{t} \right) \]
\[ \sum \alpha_i \beta_{j,i} + O \left( \frac{H^3}{t} \right) \]
where $(i)$ is by martingale concentration, $(ii)$ is by the definition of $\beta_{j,i}$, and $(iii)$ is by induction hypothesis.

To prove Theorem 7, it remains to bound the gap $\sum_{k=1}^{K} \max_j (V_{1,j}^k - V_{1,j}^k)(s_1)$.

**Proof of Theorem 7** Consider player $j$, we define $\delta_j^k := V_{j,h}^k(s^k_j) - V_{j,h}^k(s^k_j) \geq 0$. The non-negativity here is a simple consequence of the update rule and induction. We want to bound $\delta_j^k := \max_j \delta_j^k$. Let $n^k_j = N^k_j(s^k_j)$ and suppose $s^k_j$ was previously visited at episodes $k^1, \ldots, k^t_j < k$ at the $h$-th step. Now by the update rule of $V_{j,h}^k(s^k_j)$ and $V_{j,h}^k(s^k_j)$,
\[ \delta_j^k = V_{j,h}^k(s^k_j) - V_{j,h}^k(s^k_j) \]
\[ \leq \alpha_0 \frac{H}{H^3/n^k_j} + \sum \alpha_i \frac{H}{H^3/n^k_j} \left( \left(V_{j,h+1}^{k} - V_{j,h+1}^{k}\right)(s^k_{h+1}) + 2 \beta_{j,i} \right) \]
\[ \leq \alpha_0 \frac{H}{H^3/n^k_j} + \sum \alpha_i \frac{H}{H^3/n^k_j} \delta_{j,h+1} + O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]
where in the last step we have used $\sum_{i=1}^{t} \alpha_i \beta_{j,i} = \Theta(H \xi(A, t, t) + \sqrt{H^3/t})$.

Now by taking maximum w.r.t. $j$ on both sides and notice $\xi(B, t, t)$ is non-decreasing in $B$, we have
\[ \delta_j^k \leq \alpha_0 \frac{H}{H^3/n^k_j} + \sum \alpha_i \frac{H}{H^3/n^k_j} \delta_{j,h+1} + O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]

Summing the first two terms w.r.t. $k$,
\[ \sum_{k=1}^{K} \alpha_0 \frac{H}{H^3/n^k_j} = \sum_{k=1}^{K} \frac{H}{H^3/n^k_j} \{ n^k_j = 0 \} \leq SH, \]
\[ \sum_{k=1}^{K} \alpha_i \frac{H}{H^3/n^k_j} \delta_{j,h+1} \leq \sum_{k=1}^{K} \delta_{j,h+1} \sum_{i=n^k_j}^{\infty} \alpha_i \delta_{j,h+1} \leq \left( 1 + \frac{1}{H} \right) \sum_{k=1}^{K} \delta_{j,h+1}. \]
where $(i)$ is by changing the order of summation and $(ii)$ is by Lemma 10. Putting them together,
\[ \sum_{k=1}^{K} \delta_j^k \leq \sum_{k=1}^{K} \alpha_0 \frac{H}{H^3/n^k_j} + \sum_{k=1}^{K} \alpha_i \frac{H}{H^3/n^k_j} \delta_{j,h+1} + \sum_{k=1}^{K} O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]
\[ \leq HS + \left( 1 + \frac{1}{H} \right) \sum_{k=1}^{K} \delta_{j,h+1} + \sum_{k=1}^{K} O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]

\[ \sum_{k=1}^{K} \delta_{j,h+1} + \sum_{k=1}^{K} O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]

\[ \sum_{k=1}^{K} O(H \xi(A, n^k_j, t) + \sqrt{H^3/n^k_j}) \]
Reversing this argument for \( h \in [H] \) gives
\[
\sum_{k=1}^{K} \delta^k_{i} \leq cSH^2 + e \sum_{h=1}^{H} \sum_{k=1}^{K} O(H\xi(A,n^k_h,\iota) + \sqrt{H^3/n^k_h})
\]

By pigeonhole argument,
\[
\sum_{k=1}^{K} (H\xi(A,n^k_h,\iota) + \sqrt{H^3/n^k_h}) = O(1) \sum_s \sum_{n=1}^{N^K_h(s)} \left( H\xi(A,n,\iota) + \sqrt{H^3/n} \right)
\]
\[
\leq O(1) \sum_s \left( H\Xi(A,N^K_h(s),\iota) + \sqrt{H^3N^K_h(s)\iota} \right)
\]
\[
\leq O \left( HS\Xi(A,K/S,\iota) + \sqrt{H^3SK\iota} \right),
\]
where in the last step we have used concavity.

Finally take the sum w.r.t. \( h \in [H] \) we have
\[
\sum_{k=1}^{K} \max_j [V^k_{1,j} - V^k_{j,1}] (s_1) \leq O \left( H^2S\Xi(A,K/S,\iota) + \sqrt{H^3SK\iota} \right),
\]
which implies
\[
\max_{j \in [m]} [V^1_{j,1} - V^1_{j,1}] (s_1) \leq O(\frac{H^2S}{K}) \cdot \Xi(A,K/S,\iota) + \sqrt{H^3S\iota/K}).
\]

## E Proofs for Computing CE in General-sum MGs

In this section, we give complete proof of Theorem 8. To avoid repeatedly state the condition of Theorem 8 in each lemma, we will use

- Condition of the adversarial bandit sub-procedure (Assumption 2) and
- Set the bonus \( \{\beta_{j,t}\}_{t=1}^{K} \) of the \( j \)th player so that \( \sum_{i=1}^{t} \alpha_i \beta_{j,t} = \Theta(H\xi_{sw}(A_j,t,\iota) + \sqrt{H^3t/\iota}) \) for any \( t \in [K] \).

throughout the whole section.

We begin with a swap regret version of Lemma 12.

**Lemma 15.** The following event is true with probability at least 1 - \( \delta \): for any \( (s,h,k) \in S \times [H] \times [K] \), let \( t = N^K_h(s) \) and suppose \( s \) was previously visited at episodes \( k_1, \ldots, k_l < k \) at the \( h \)-th step, then for all \( j \in [m] \)
\[
\max_{j} \sum_{i=1}^{t} \alpha_i \mathcal{D}_{\phi_j \circ \pi^k_{i,j,h},n^k_{i,j,h}} \left[ r_{j,h} + \mathbb{P}_h V^{k_i}_{j,h+1} \right] (s) - \sum_{i=1}^{t} \alpha_i \mathcal{D}_{\pi^k_{i,j,h}} \left( r_{j,h} + \mathbb{P}_h V^{k_i}_{j,h+1} \right) (s) \leq H\xi_{sw}(A_j,t,\iota),
\]
where \( \iota = \log(KHS/\delta) \).

**Proof.** By Assumption 2 and the adversarial bandit update step in Algorithm 1 we have that with probability at least 1 - \( \delta \), for any \( (s,h,k) \in S \times [H] \times [K] \),
\[
\max_{j} \sum_{i=1}^{t} \alpha_i \mathcal{D}_{\phi_j \circ \pi^k_{i,j,h},n^k_{i,j,h}} \left( \frac{H - r_{j,h} + \mathbb{P}_h V^{k_i}_{j,h+1}}{H} \right) (s) - \sum_{i=1}^{t} \alpha_i \mathcal{D}_{\pi^k_{i,j,h}} \left( \frac{H - r_{j,h} + \mathbb{P}_h V^{k_i}_{j,h+1}}{H} \right) (s) \leq \xi_{sw}(A_j,t,\iota).
\]

We begin with proving \( V \) is actually an optimistic estimation of the value function under best response.
Lemma 16 (Optimism). For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for any $(s, h, k, j) \in S \times [H] \times [K] \times [m]$, $V_{j,h}^k(s) \geq \max_{\phi_j} V_{j,h}^{(\phi_j \circ \pi_{j,h}^k) \circ \#_{j,h}^k}(s)$.

Proof of Lemma 16 We prove by backward induction. The claim is trivially satisfied for $h = H + 1$. Suppose it is true for $h + 1$, consider a fixed state $s$. It suffices to show $\tilde{V}_{j,h}^k(s) \geq \max_{\phi_j} V_{j,h}^{(\phi_j \circ \pi_{j,h}^k) \circ \#_{j,h}^k}(s)$ because $V_{j,h}^k(s) = \min \{V_{j,h}^k(s), H + 1\}$. Let $t = N_h^k(s)$ and suppose $s$ was previously visited at episodes $k^i, \ldots, k^t < k$ at the $h$-th step. Then using Lemma 11

$$V_{j,h}^k(s) = \alpha_t^0(H - h + 1) + \sum_{i=1}^t \alpha_t^i \left[ r_{j,h}(s, a_h^{k^i}) + V_{j,h+1}^{k^i}(s_{h+1}) + \beta_{j,i} \right]$$

$$\geq \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\pi_{j,h}^{k^i}} \left[ r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right](s) + \sum_{i=1}^t \alpha_t^i \beta_{j,i} - \mathcal{O} \left( \sqrt{\frac{H^3 t}{t}} \right)$$

$$\geq \max_{\phi_j} \sum_{i=1}^t \alpha_t^i \mathbb{E}_{(\phi_j \circ \pi_{j,h}^{k^i}) \circ \#_{j,h}^k} \left[ r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right](s) + \sum_{i=1}^t \alpha_t^i \beta_{j,i} - \mathcal{O} \left( \sqrt{\frac{H^3 t}{t}} \right) - H \xi(A_j, t, t)$$

where (i) is by martingale concentration and Lemma 2. (ii) is by Lemma 15. (iii) is by the definition of $\beta_{j,i}$, and (iv) is by induction hypothesis. Finally, (v) follows from a similar reasoning as in the proof of Lemma 13 which we omit here. 

We still need to lower bound $\tilde{V}_{j,h}^{\#_h}$. To do this, we estimate $\tilde{V}$ and $\tilde{F}$ defined by Equation 10 and Equation 11. These quantities are indeed the lower bounds we need.

Lemma 17 (Pessimism). For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for any $(s, h, k, j) \in S \times [H] \times [K] \times [m]$, $\tilde{V}_{j,h}^k(s) \leq V_{j,h}^{\#_h}(s)$.

Proof of Lemma 17 We prove by backward induction. The claim is trivially satisfied for $h = H + 1$. Suppose it is true for $h + 1$, consider a fixed state $s$. It suffices to show $\tilde{V}_{j,h}^k(s) \leq V_{j,h}^{\#_h}(s)$ because $\tilde{V}_{j,h}^k(s) = \max \{V_{j,h}^k(s), 0\}$. Let $t = N_h^k(s)$ and suppose $s$ was previously visited at episodes $k^i, \ldots, k^t < k$ at the $h$-th step. Then by equation 10

$$V_{j,h}^k(s) = \sum_{i=1}^t \alpha_t^i \left[ r_{j,h}(s, a_h^{k^i}) + V_{j,h+1}^{k^i}(s_{h+1}) - \beta_{j,i} \right]$$

$$\leq \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\pi_{j,h}^{k^i}} \left[ r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right](s) - \sum_{i=1}^t \alpha_t^i \beta_{j,i} + \mathcal{O} \left( \sqrt{\frac{H^3 t}{t}} \right)$$

$$\leq \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\pi_{j,h}^{k^i}} \left[ r_{j,h} + \mathbb{P}_h V_{j,h+1}^{k^i} \right](s)$$

$$\leq \sum_{i=1}^t \alpha_t^i \mathbb{E}_{\pi_{j,h}^{k^i}} \left[ r_{j,h} + \mathbb{P}_h V_{j,h+1}^{\#_h^{k^i}} \right](s)$$
where (i) is by martingale concentration, (ii) is by the definition of $\beta_{j,i}$, and (iii) is by induction hypothesis.

To prove Theorem 8, it remains to bound the gap $\sum_{k=1}^K \max_j (V_{i,j}^k - V_{i,j}^k(s_1))$.

**Proof of Theorem 8.** Consider player $j$, we define $\delta_{j,h}^k := V_{j,h}^k(s_h^k) - V_{j,h}^k(s_h^k) \geq 0$. The non-negativity here is a simple consequence of the update rule and induction. We want to bound $\delta_{h}^k := \max_j \delta_{j,h}^k$. Let $n_h^k = N_h^k(s_h^k)$ and suppose $s_h^k$ was previously visited at episodes $k^1, \ldots, k^m < k$ at the $h$-th step. Now by the update rule of $V_{j,h}^k(s_h^k)$ and $V_{j,h}^k(s_h^k)$,

$$\delta_{j,h}^k = V_{j,h}^k(s_h^k) - V_{j,h}^k(s_h^k) \leq \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \left[ (V_{j,h+1}^k(s_h^k) - V_{j,h+1}^k(s_h^k)) + 2\beta_{j,i} \right]$$

$$= \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{j,h+1}^k + O(H\xi(A_j, n_h^k, t) + \sqrt{H^3 t/n_h^k})$$

where in the last step we have used $\sum_{i=1}^t \alpha_{j,i}^i \leq \Theta(H\xi(A_j, t, t) + \sqrt{H^3 t/n_h^k})$.

Now by taking maximum w.r.t. $j$ on both sides and notice $\xi(A, B, t)$ is non-decreasing in $B$, we have

$$\delta_h^k \leq \alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^k + O(H\xi(A, n_h^k, t) + \sqrt{H^3 t/n_h^k})$$

Summing the first two terms w.r.t. $k$,

$$\sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^k \leq \sum_{k=1}^K H \{ n_h^k = 0 \} \leq SH,$$

$$\sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_0}^i \delta_{h+1}^k \leq \left( 1 + \frac{1}{H} \right) \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^k.$$

where (i) is by changing the order of summation and (ii) is by Lemma 10. Putting them together,

$$\sum_{k=1}^K \delta_h^k = \sum_{k=1}^K \alpha_{n_h^k}^0 H + \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^k + \sum_{k=1}^K O(H\xi(A, n_h^k, t) + \sqrt{H^3 t/n_h^k})$$

$$\leq HS + \left( 1 + \frac{1}{H} \right) \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^k + \sum_{k=1}^K O(H\xi(A, n_h^k, t) + \sqrt{H^3 t/n_h^k})$$

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

$$\sum_{k=1}^K \delta_h^k \leq eSH^2 + \sum_{h=1}^H \sum_{k=1}^K O(H\xi(A, n_h^k, t) + \sqrt{H^3 t/n_h^k})$$

By pigeonhole argument,

$$\sum_{k=1}^K (H\xi(A, n_h^k, t) + \sqrt{H^3 t/n_h^k}) = O(1) \sum_{s} \sum_{n=1}^{N_h^k(s)} (H\xi(A, n, t) + \sqrt{H^3 t/n})$$

$$\leq O(1) \sum_{s} (H\xi(A, N_h^k(s), t) + \sqrt{H^3 N_h^k(s) t})$$

$$\leq O(HS\xi(A, K/S, t) + \sqrt{H^3 SK t})$$

23
where in the last step we have used concavity.

Finally take the sum w.r.t. \( h \in [H] \) we have
\[
\sum_{k=1}^{K} \max_j [V_{1,j}^k - V_{1,j}^k](s_1) \leq O \left( H^2 S \Xi_{sw}(A, K/S, \epsilon) + \sqrt{H^5 S K \epsilon} \right),
\]
which implies
\[
\max_{j \in [m]} [V_{j,\pi_j}(s) - V_{j,\pi_j}(s)] \leq O((H^2 S/K) \cdot \Xi_{sw}(A, K/S, \epsilon) + \sqrt{H^5 S \epsilon/K}).
\]

\[\square\]

## F Proofs for MDPs and Two-Player Zero-Sum MGs

In this section, we prove the main theorems for V-learning in the setting of single-agent (MDPs) and two-player zero-sum MGs.

**Proof of Theorem 4** To begin with, we notice an equivalent definition of two-player zero-sum MGs is that the reward function satisfies \( r_{1,h} = 1 - r_{2,h} \) for all \( h \in [H] \). The reason we use this definition instead of the common version \( r_{1,h} = -r_{2,h} \) is we want to make it consistent with our assumption that the reward function takes value in \([0, 1]\) for any player. Although this definition does not satisfy the zero-sum condition, its Nash equilibria are the same as those of the zero-sum version because adding a constant to the reward function of player 2 per step will not change the dynamics of the game.

In order to show \( \hat{\pi} = \hat{\pi}_1 \times \hat{\pi}_2 \) is an approximate Nash policy, it suffices to control
\[
\max_{\pi_1} V_{1,1}^{\pi_1 \cdot \pi_2}(s) - \min_{\pi_2} V_{1,1}^{\pi_1 \cdot \pi_2}(s).
\]
Since \( r_{1,h} = 1 - r_{2,h} \) for all \( h \in [H] \), with probability at least \( 1 - \delta \)
\[
\max_{\pi_1} V_{1,1}^{\pi_1 \cdot \pi_2}(s) - \min_{\pi_2} V_{1,1}^{\pi_1 \cdot \pi_2}(s)
\]
\[
= \max_{\pi_1} V_{1,1}^{\pi_1 \cdot \pi_2}(s) - (H - \max_{\pi_2} V_{2,1}^{\pi_1 \cdot \pi_2}(s))
\]
\[
= \left( \max_{\pi_1} V_{1,1}^{\pi_1 \cdot \pi_2}(s) - V_{1,1}^{\pi_1 \cdot \pi_2}(s) \right) + \left( \max_{\pi_2} V_{2,1}^{\pi_1 \cdot \pi_2}(s) - V_{2,1}^{\pi_1 \cdot \pi_2}(s) \right)
\]
\[
\leq O((H^2 S/K) \cdot \Xi(A, K/S, \epsilon) + \sqrt{H^5 S \epsilon/K}),
\]
where the last inequality follows from Theorem 7. The reason we can use Theorem 7 here is the precondition of Theorem 5 is a special case of the precondition of Theorem 7.

**Proof of Theorem 7** Since MDPs is a subclass of two-player zero-sum MGs by simply choosing the action set of the second player to be a singleton, it suffices to only prove Theorem 5 from which the single-agent guarantee, Theorem 4 trivially follows.

## G Proofs for Monotonic V-Learning

In this section, we prove Theorem 6. The algorithm is V-learning with monotonic update, and the setting we consider is two-player zero-sum Markov games. As before, we assume \( r_{1,h}(s, a) = 1 - r_{2,h}(s, a) \) for all \( s, a, h \). The reason for assuming \( r_{1,h}(s, a) = 1 - r_{2,h}(s, a) \) instead of \( r_{1,h}(s, a) = -r_{2,h}(s, a) \) can be found in Appendix F.

For two player zero-sum MGs, we can define its minimax value function (Nash value function) by the following Bellman equations
\[
\begin{align*}
V_j^*(s) &= \max_{\pi_j} \min_{\pi_{-j} \cdot \pi_j} D_{\pi_j, h} \cdot \Xi_{\pi_{-j} \cdot \pi_j, h}(Q_j^*, h)(s), \\
Q_j^*(s, a) &= r_j^h(s, a) + \mathbb{P}_h[V_{j, h+1}^*(s, a)], \\
V_{j, h+1}^*(s) &= Q_j^*(s, a) = 0.
\end{align*}
\]  

(12)
Lemma 18 (Optimism of V-estimates). With probability at least $1 - \delta$, for any $(s, h, k, j) \in S \times [H] \times [K] \times [2]$,}

$$
\hat{V}_{j,h}^k(s) \geq V_{j,h}^k(s) \geq V_{j,h}^{\pi - \delta}(s) \geq V_{j,h}^*(s),
$$

where $V_{j,h}^*$ is the minimax (Nash) value function defined above.

Proof of Lemma 18. Note that $\hat{V}_{j,h}^k(s) \geq V_{j,h}^k(s)$ is straightforward by the update rule of V-learning, and $V_{j,h}^{\pi - \delta}(s) \geq V_{j,h}^*(s)$ directly follows from the definition of minimax value function. Therefore, we only need to prove the second inequality. We do this by backward induction.

The claim is true for $h = H + 1$. Assume for any $s$ and $k$, $V_{j,h+1}^k(s) \geq V_{j,h+1}^{\pi - \delta}(s)$. For a fixed $(s, h, k) \in S \times [H] \times [K]$, let $t = N_h(s)$ and suppose $s$ was previously visited in episode $k^1, \ldots, k^t < k$ at the $h$-th step. By Bellman equation,}

$$
\hat{V}_{j,h}^{\pi - \delta}(s) \leq \alpha(t) (H - h + 1) + \max_q \sum_{i=1}^t \alpha(i) \mathbb{L}_{\mu \times \pi_{j,h}^{t-1}} \left( r_{j,h} + \mathbb{P}_h V_{j,h+1}^{\pi - \delta} \right)(s)
$$

$$
\leq \alpha(t) (H - h + 1) + \max_q \sum_{i=1}^t \alpha(i) \mathbb{L}_{\pi_{j,h}^{t-1}} \left( r_{j,h} + \mathbb{P}_h V_{j,h+1}^k \right)(s) + H \xi(A_j, t, \epsilon)
$$

where the second inequality follows from our induction hypothesis and the monotonicity of $V_k$, the third inequality follows from Lemma 12, and the last one follows from martingale concentration as well as Lemma 10. By Lemma 11 and the precondition of Theorem 6, we know the RHS is no larger than $\hat{V}_{j,h}^k(s)$. Note that $V_k$ can be equivalently defined as

$$
V_{j,h}^k(s) = \min_k \{ \min_{t \in [k]} \hat{V}_{j,h}^t(s), H - h + 1 \},
$$

we conclude $V_{j,h}^k(s) \geq V_{j,h}^{\pi - \delta}(s)$ for any $k \in [K]$.

Now we are ready to prove Theorem 6.

Proof of Theorem 6. By the monotonicity of $V$ and Lemma 18

$$
\hat{V}_{1,1}^{\pi_2}(s_1) - \min_{\pi_2} V_{1,1}^{\pi_1 \times \pi_2}(s_1)
$$

$$
= \hat{V}_{1,1}^{\pi_2}(s_1) - \left( H - \hat{V}_{1,1}^{\pi_1}(s_1) \right)
$$

$$
\leq V_{1,1}^K(s_1) + V_{2,1}^K(s_1) - H
$$

$$
\leq \frac{1}{K} \sum_{k=1}^K \left( V_{1,1}^k(s_1) + V_{2,1}^k(s_1) - H \right)
$$

$$
\leq \frac{1}{K} \sum_{k=1}^K \left( \hat{V}_{1,1}^k(s_1) + \hat{V}_{2,1}^k(s_1) - H \right),
$$

where the first equality follows from the definition of two-player zero-sum game, i.e., $r_{1,h} = 1 - r_{2,h}$. Now we can mimic the proof of Theorem 7. Define $\delta_h^k := \hat{V}_{1,h}^k(s_h^k) + \hat{V}_{2,h}^k(s_h^k) - (H - h + 1)$. The non-negativity here follows from Lemma 18 as below

$$
\hat{V}_{1,h}^k(s_h^k) + \hat{V}_{2,h}^k(s_h^k) - (H - h + 1) \geq V_{1,h}^*(s_h^k) + V_{2,h}^*(s_h^k) - (H - h + 1) = (H - h + 1) - (H - h + 1) = 0.
$$
For any step. By Lemma 11 and the fact that \( r \) of weights \( \sqrt{i} \) is the number of actions, and our normalization condition requires loss \( \tilde{I}_i(b_i) \), we have
\[
\delta_h^k = \tilde{V}_{1,h}^k(s_h^k) + \tilde{V}_{2,h}^k(s_h^k) - (H - h + 1)
\]
\[
= 2\alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \left( \tilde{V}_{1,h+1}^{k_i} - \tilde{V}_{2,h+1}^{k_i} \right) \left( s_{h+1}^k \right) - (H - h) + \beta_{1,i} + \beta_{2,i}
\]
\[
= 2\alpha_{n_h^k}^0 H + \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k_i} + O(H_{\xi}(A, n_h^k, t) + \sqrt{H^3t/n_h^k})
\]
where in the last step we used \( \sum_{t=1}^{t} \alpha_{n_h^k}^i \beta_{j,i} = \Theta(H_{\xi}(A, t, \xi) + \sqrt{H_{\xi}t}) \).

The remaining steps follow exactly the same as the proof of Theorem 17. As a result, we obtain
\[
V^{t,\tilde{T}_2}_{1,1}(s_1) - \min_{\pi_2} V^{\pi_2}_{1,1}(s_1) \leq \frac{1}{K} \sum_{k=1}^{K} \left( \tilde{V}_{1,1}^k(s_1) + \tilde{V}_{2,1}^k(s_1) - H \right)
\]
\[
\leq O \left( \frac{H^2S}{K} \cdot \xi(A, K/S, \xi) + \frac{H^2S\epsilon}{K} \right),
\]
which completes the proof.

\[ \square \]

## H ADVERSARIAL BANDIT WITH WEIGHTED EXTERNAL REGRET

In this section, we present a Follow-the-Regularized-Leader (FTRL) style algorithm that achieves low weighted (external) regret for the adversarial bandit problem. Although FTRL is a classical algorithm in the adversarial bandit literature, we did not find a good reference of FTRL with changing step size, weighted regret and high probability bound. For completeness of this work, we provide detailed derivations here.

We present the FTRL algorithm in Algorithm 5. In Corollary 19, we prove that FTRL satisfies the Assumption 14 with good regret bounds. Recall that \( B \) is the number of actions, and our normalization condition requires loss \( \tilde{I}_i \in [0, 1]^B \) for any \( t \).

**Corollary 19.** By choosing hyperparameter \( w_t = \alpha_t \left( \prod_{i=2}^{t} (1 - \alpha_i) \right)^{-1} \) and \( \eta_t = \gamma_t = \sqrt{\frac{H_{\xi_B}B}{t}} \), FTRL (Algorithm 5) satisfies Assumption 14 with
\[
\xi(B, t, \log(1/\delta)) = 10\sqrt{HB \log(B/\delta)/t}, \quad \Xi(B, t, \log(1/\delta)) = 20\sqrt{HB t \log(B/\delta)}
\]

To prove Corollary 19, we show a more general weighted regret guarantee which works for any set of weights \( \{w_i\}_{i=1}^{\infty} \) in addition to \( \{\alpha_t\}_{t=1}^{T} \). In particular, a general weighted regret is defined as
\[
\mathcal{R}(t) = \max_{\theta^*} \sum_{i=1}^{t} w_i (\theta_i - \theta^*, \tilde{I}_i)
\]
(14)

**Theorem 20.** For any \( t \leq K \), following Algorithm 5 if \( \eta_t \leq 2\gamma_t \) and \( \eta_t \) is non-increasing for all \( i \leq t \), let \( \bar{\eta} = \log(B/\delta) \), then with probability \( 1 - 3\delta \), we have
\[
\mathcal{R}(t) \leq \frac{w_t \log B}{\eta_t} + \frac{B}{2} t \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i t + B \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2t \sum_{i=1}^{t} w_i^2 + \max_{i \leq t} w_i t / \gamma_t}.
\]
We postpone the proof of Theorem 20 to the end of this section. We first show how to obtain Corollary 19 from Theorem 20.

**Proof of Corollary 19.** The weights \( \{w_t\}_{t=1}^{K} \) we choose satisfy a nice property: for any \( t \) we have

\[
\frac{w_i}{w_j} = \frac{\alpha_i}{\alpha_j} \prod_{k=i+1}^{j} (1 - \alpha_k),
\]

We prove this for \( i \leq j \) and the other case is similar. By definition,

\[
\frac{w_i}{w_j} = \alpha_i \prod_{k=i+1}^{j} (1 - \alpha_k),
\]

and

\[
\frac{\alpha_i}{\alpha_j} = \prod_{k=i+1}^{j} (1 - \alpha_k).
\]

We can easily verify that the RHS are the same.

Define \( \tilde{R}(t) := \max_{\theta \in \Delta} \sum_{i=1}^{t} \alpha_i \left[ \langle \theta, \ell_i \rangle - \langle \theta, \ell_i \rangle \right] \). By plugging \( w_t = \alpha_t \left( \prod_{i=2}^{t} (1 - \alpha_i) \right)^{-1} \) into Theorem 20 and using the property above, we have the regret guarantee

\[
\tilde{R}(t) \leq \frac{\alpha_t \log B}{\eta_t} + B \sum_{i=1}^{t} \eta_i \alpha_i + \frac{1}{2} \alpha_t + B \sum_{i=1}^{t} \gamma_i \alpha_i + \sqrt{2t \sum_{i=1}^{t} (\alpha_i)^2 + \alpha_t / \gamma_t},
\]

By choosing \( \eta_t = \gamma_t = \sqrt{H \log B / t} \) and using Lemma 10 we can further upper bound the regret by

\[
\tilde{R}(t) \leq \frac{(H + 1) \log B}{H / t} + \frac{Bt}{H \log B} + \frac{3}{2} \sqrt{H B \log B} \sum_{i=1}^{t} \alpha_i^2 \sqrt{t} + \frac{(H + 1) t}{2 (H / t)} + \sqrt{2t \sum_{i=1}^{t} (\alpha_i)^2} + \frac{(H + 1) t}{(H / t)} \sqrt{\frac{Bt}{H \log B}}
\]

\[
\leq 2 \sqrt{\frac{H B \log B}{t}} + 3 \sqrt{\frac{H B \log B}{t}} + \frac{H}{t} + 2 \sqrt{\frac{H t}{t} + 2 \sqrt{\frac{H B \log B}{t}}}
\]

\[
\leq 9 \sqrt{H B t / t + H / t}.
\]

To further simplify the above upper bound, consider two cases:

- If \( H / t \leq 1, \sqrt{H / t} \geq H / t \) and thus \( \tilde{R}(t) \leq 10 \sqrt{H B t / t} \).
- If \( H / t \geq 1, \sqrt{H B t / t} \geq 1 \geq \tilde{R}(t) \) where the last step is by the definition of \( \tilde{R}(t) \). Therefore we have \( \tilde{R}(t) \leq \sqrt{H B t / t} \).

Combining the two cases above gives \( \tilde{R}(t) \leq 10 \sqrt{H B t / t} \).

Finally, we pick \( \xi(B, t, \log(1/\delta)) := 10 \sqrt{H B t / t} \), which is non-decreasing in \( B \). Since \( \sum_{t=1}^{t} \xi(B, t, \log(1/\delta)) \leq 20 \sqrt{H B / t} \), we choose \( \Xi(B, t, \log(1/\delta)) = 20 \sqrt{H B t / t} \), which is concave in \( t \). □
To prove Theorem 20, we first note that the weighted regret equation 14 can be decomposed into three terms

\[
\sum_{i=1}^{t} w_i \langle \theta_i - \theta^*, l_i \rangle = \sum_{i=1}^{t} w_i \langle \theta_i - \theta^*, l_i \rangle \\
= \sum_{i=1}^{t} w_i \langle \theta_i - \theta^*, \hat{l}_i \rangle + \sum_{i=1}^{t} w_i \langle \theta_i, l_i - \hat{l}_i \rangle + \sum_{i=1}^{t} w_i \langle \theta^*, \hat{l}_i - l_i \rangle \tag{15}
\]

The rest of this section is devoted to bounding three terms above. We begin with the following useful lemma adapted from Lemma 1 in Neu (2015), which is crucial in achieving high probability guarantees.

**Lemma 21.** For any sequence of coefficients \( c_1, c_2, \ldots, c_t \) s.t. \( c_i \in [0, 2\gamma_i]^B \) is \( \mathcal{F}_t \)-measurable, we have with probability \( 1 - \delta \),

\[
\sum_{i=1}^{t} w_i \langle c_i, \hat{l}_i - l_i \rangle \leq \max_{i \leq t} w_i c_i.
\]

**Proof.** Define \( w = \max_{i \leq t} w_i \). By definition,

\[
w_i \hat{l}_i (b) = \frac{w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{\theta_i (b) + \gamma_i} \leq \frac{w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{\theta_i (b) + \frac{w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{w} \gamma_i} = \frac{w}{2\gamma_i} \frac{2\gamma_i w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{\gamma_i w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}} \tag{i}
\]

where \( i \) follows from \( \frac{z}{1+z} \leq \log (1+z) \) for all \( z \geq 0 \).

Defining the sum

\[
\hat{S}_i = \frac{w_i}{w} \langle c_i, \hat{l}_i \rangle, \quad S_i = \frac{w_i}{w} \langle c_i, l_i \rangle.
\]

we have

\[
\mathbb{E}_t \left[ \exp \left( \hat{S}_i \right) \right] \leq \mathbb{E}_t \left[ \exp \left( \sum_{b} c_i (b) \log \left( 1 + \frac{2\gamma_i w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{w \theta_i (b)} \right) \right) \right] \tag{i}
\]

\[
\leq \mathbb{E}_t \left[ \prod_{b} \left( 1 + \frac{c_i (b) w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{w \theta_i (b)} \right) \right] \tag{i}
\]

\[
= \mathbb{E}_t \left[ 1 + \sum_{b} \frac{c_i (b) w_i \hat{l}_i (b) \mathbb{1} \{ b_i = b \}}{w \theta_i (b)} \right] = 1 + S_i \leq \exp \left( S_i \right)
\]

where \( i \) follows from \( z_1 \log (1 + z_2) \leq \log (1 + z_1 z_2) \) for any \( 0 \leq z_1 \geq 1 \) and \( z_2 \geq -1 \). Note that here we are using the condition \( c_i (b) \leq 2\gamma_i \) for all \( b \in [B] \).
Equipped with the above bound, we can now prove the concentration result.

\[ P \left[ \sum_{i=1}^{t} \left( \hat{S}_i - S_i \right) \geq \epsilon \right] = P \left[ \exp \left[ \sum_{i=1}^{t} \left( \hat{S}_i - S_i \right) \right] \geq \frac{B}{\delta} \right] \]

\[ \leq \frac{\delta}{B} \mathbb{E}_t \left[ \exp \left[ \sum_{i=1}^{t} \left( \hat{S}_i - S_i \right) \right] \right] \]

\[ \leq \frac{\delta}{B} \mathbb{E}_{t-1} \left[ \exp \left[ \sum_{i=1}^{t-1} \left( \hat{S}_i - S_i \right) \right] \right] E_t \left[ \exp \left( \hat{S}_t - S_t \right) \right] \]

\[ \leq \cdots \leq \frac{\delta}{B} \cdot \delta \]

We conclude the proof by taking a union bound.

With Lemma 21, we can bound the three terms \((A), (B)\) and \((C)\) in equation 15 separately as below.

**Lemma 22.** For any \(t \in [K]\), suppose \(\eta_i \leq 2\gamma_i\) for all \(i \leq t\). Then with probability at least \(1 - \delta\), for any \(\theta^* \in \Delta^B\),

\[
\sum_{i=1}^{t} w_i \left( \theta_i - \theta^*, \hat{l}_i \right) \leq \frac{w_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i t .
\]

**Proof.** We use the standard analysis of FTRL with changing step size, see for example Exercise 28.13 in Lattimore & Szepesvári (2018). Notice the essential step size is \(\eta_t / w_t\).

\[
\sum_{i=1}^{t} w_i \left( \theta_i - \theta^*, \hat{l}_i \right) \leq \frac{w_t \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i \left( \hat{l}_i \right) + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i (b) \]

\[
\leq \frac{w_t \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i \left( \hat{l}_i \right) + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i (b) + \frac{1}{2} \max_{i \leq t} w_i t \]

\[
\leq \frac{w_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i t .
\]

where \((i)\) is by using Lemma 21 with \(c_i(b) = \eta_i\) for any \(b\). The any-time guarantee is justified by taking union bound.

**Lemma 23.** For any \(t \in [K]\), with probability \(1 - \delta\),

\[
\sum_{i=1}^{t} w_i \left( \theta_i, l_i - \hat{l}_i \right) \leq B \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2t \sum_{i=1}^{t} w_i^2} .
\]

**Proof.** We further decompose it into

\[
\sum_{i=1}^{t} w_i \left( \theta_i, l_i - \hat{l}_i \right) = \sum_{i=1}^{t} w_i \left( \theta_i, l_i - \mathbb{E} \hat{l}_i \right) + \sum_{i=1}^{t} w_i \left( \theta_i, \mathbb{E} \hat{l}_i - \hat{l}_i \right) .
\]
The first term is bounded by
\[
\sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - E_i \hat{l}_i \right\rangle = \sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \frac{\theta_i}{\theta_i + \gamma_i} l_i \right\rangle \\
= \sum_{i=1}^{t} w_i \left\langle \theta_i, \frac{\gamma_i}{\theta_i + \gamma_i} l_i \right\rangle \leq B \sum_{i=1}^{t} \gamma_i w_i.
\]

To bound the second term, notice
\[
\left\langle \theta_i, \hat{l}_i \right\rangle \leq \sum_{b \in B} \theta_i(b) \frac{I\{b_i = b\}}{\theta_i(b)} \leq \sum_{b \in B} I\{b_i = b\} = 1,
\]
thus \(\{w_i \left\langle \theta_i, E_i \hat{l}_i - \hat{l}_i \right\rangle\}_{i=1}^{t}\) is a bounded martingale difference sequence w.r.t. the filtration \(\{F_i\}_{i=1}^{t}\). By Azuma-Hoeffding,
\[
\sum_{i=1}^{t} \left\langle \theta_i, E_i \hat{l}_i - \hat{l}_i \right\rangle \leq \sqrt{2t \sum_{i=1}^{t} w_i^2}.
\]

**Lemma 24.** For any \(t \in [K]\), with probability \(1 - \delta\), for any \(\theta^* \in \Delta^B\), if \(\gamma_i\) is non-increasing in \(i\),
\[
\sum_{i=1}^{t} w_i \left\langle \theta^*, \hat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \gamma_i / \gamma_i.
\]

**Proof.** Define a basis \(\{e_j\}_{j=1}^{B}\) of \(\mathbb{R}^{B}\) by
\[
e_j(b) = \begin{cases} 1 & \text{if } a = j \\ 0 & \text{otherwise} \end{cases}
\]
Then for all the \(j \in [B]\), we can apply Lemma 21 with \(c_i = \gamma_i e_j\). Since \(c_i(b) \leq \gamma_i \leq \gamma_i\), the condition in Lemma 21 is satisfied. As a result,
\[
\sum_{i=1}^{t} w_i \left\langle e_j, \hat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \gamma_i / \gamma_i.
\]
Since any \(\theta^*\) is a convex combination of \(\{e_j\}_{j=1}^{B}\), by taking the union bound over \(j \in [B]\), we have
\[
\sum_{i=1}^{t} w_i \left\langle \theta^*, \hat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \gamma_i / \gamma_i.
\]

Finally we are ready to prove Theorem 20.

**Proof of Theorem 20.** Note the conditions in Lemma 22 and Lemma 24 are satisfied by assumptions. Recall the regret decomposition equation 15. By bounding \((A)\) in Lemma 22, \((B)\) in Lemma 23 and \((C)\) in Lemma 24, with probability \(1 - 3\delta\), we have that
\[
\mathcal{R}(t) \leq \frac{w_t \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \gamma_i + \frac{B}{2} \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2t \sum_{i=1}^{t} w_i^2 + \max_{i \leq t} w_i \gamma_i}.
\]
Algorithm 6 FTRL for Weighted Swap Regret (FTRL\textsubscript{swap})

1: **Initialize:** for any \( b \in B, \theta_t(b) \leftarrow 1/B. \\
2: **for** episode \( t = 1, \ldots, K \) **do**
3: Take action \( b_t \sim \theta_t(\cdot) \), and observe loss \( \hat{l}_t(b_t) \).
4: **for** each action \( b \in B \) **do**
5: \( \hat{l}_t(b) \leftarrow \theta_t(b)\hat{l}_t(b) / \{b_t = \cdot\}(\theta_t(\cdot) + \gamma_t) \).
6: \( \theta_{t+1}(\cdot) \propto \exp\left[-(\eta_t/w_t) \cdot \sum_{i=1}^t w_i \hat{l}_i(\cdot | b)\right] \)
7: Set \( \theta_{t+1}(\cdot) \) such that \( \theta_{t+1}(\cdot) = \sum_a \theta_{t+1}(b)\theta_{t+1}(\cdot | b) \).

I Adversarial Bandit with Weighted Swap Regret

In this section, we adapt Follow-the-Regularized-Leader (FTRL) algorithm that achieves low weighted swap regret for the adversarial bandit problem. We follow a similar technique presented in [Blum & Mansour (2007)] which adapts external regret algorithms to swap regret algorithms for the unweighted case.

We present the FTRL\textsubscript{swap} algorithm in Algorithm 6. Different from FTRL (Algorithm 5), FTRL\textsubscript{swap} maintains an additional \( B \times B \) matrix \( \theta_t(\cdot) \), and uses its eigenvector when taking actions. The matrix will be updated similarly to FTRL, with a subtle difference that the loss estimator here \( \hat{l}_t(b) \) is \( \theta_t(b) \) times the loss estimator \( \hat{l}_t(\cdot) \) in the FTRL algorithm (Line 4 in Algorithm 5).

In Corollary 25 we prove that FTRL\textsubscript{swap} satisfies the Assumption 2 with good swap regret bounds. Recall that \( B \) is the number of actions, and our normalization condition requires loss \( l_t \in [0, 1]^B \) for any \( t \).

**Corollary 25.** By choosing hyperparameter \( w_t = \alpha_t \left( \prod_{i=2}^t (1 - \alpha_i) \right)^{-1} \) and \( \eta_t = \gamma_t = \sqrt{\frac{H \log B}{t}}, \) FTRL\textsubscript{swap} (Algorithm 6) satisfies Assumption 2 with

\[
\xi_{sw}(B, t, \log(1/\delta)) = 10B \sqrt{H \log(B^2/\delta)/t}, \quad \Xi_{sw}(B, t, \log(1/\delta)) = 20B \sqrt{H \log(B^2/\delta)}
\]

Again, we prove Corollary 25 by showing a more general weighted swap regret guarantee which works for any set of weights \( \{w_t\}_{t=1}^\infty \) in addition to \( \{\alpha_t\}_{t=1}^T \). A general weighted swap regret is defined as

\[
\mathcal{R}_{sw}(t) := \min_{\psi \in \Psi} \sum_{i=1}^t w_i[\langle \theta_i, l_i \rangle - \langle \psi \circ \theta_i, l_i \rangle].
\]

**Theorem 26.** For any \( t \leq K \), following Algorithm 6, if \( \eta_t \leq 2\gamma_t \) and \( \eta_t \) is non-increasing for all \( i \leq t \), let \( \eta = \log(B^2/\delta) \), then with probability \( 1 - 3\delta \), we have

\[
\mathcal{R}_{sw}(t) \leq \frac{w_B B \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^t \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \eta + B \sum_{i=1}^t \gamma_i w_i + B \left[ 2t \max_{i \leq t} w_i^2 + B \max_{i \leq t} w_i \gamma_t \right]
\]

We postpone the proof of Theorem 26 to the end of this section. We show first how Theorem 26 directly implies Corollary 25.

**Proof of Corollary 25.** As shown in the proof of Corollary 19 the weights \( \{w_t\}_{t=1}^K \) we choose satisfies a nice property: for any \( t \) we have

\[
\frac{w_i}{w_j} = \frac{\alpha^t_i}{\alpha^t_j}
\]
Define $\tilde{R}_{\text{swap}}(t) := \max_{\psi \in \Psi} \sum_{i=1}^{t} \alpha_i[\langle \theta_i, l_i \rangle - \langle \psi \diamond \theta_i, l_i \rangle]$. Plugging our choice of $w_i = \alpha_t \left( \prod_{i=2}^{t} (1 - \alpha_i) \right)^{-1}$ into Theorem 26, we have

$$\tilde{R}_{\text{swap}}(t) \leq \frac{\alpha_t B \log B}{\eta_t} + B \sum_{i=1}^{t} \gamma_i \alpha_i^t + B \sum_{i=1}^{t} (\alpha_i^t)^2 + B \alpha_t \eta_t.$$ 

By choosing $\eta_t = \gamma_t = \sqrt{\frac{H \log B}{t}}$ and using Lemma 10, we can further upper bound the swap regret by

$$\tilde{R}_{\text{swap}}(t) \leq \frac{(H + 1) B \log B}{H + t} \sqrt{\frac{t}{H \log B}} + \frac{3B}{2} \sqrt{\frac{H \log B}{t}} \sum_{i=1}^{t} \alpha_i^t$$

$$+ B \sum_{i=1}^{t} \gamma_i \alpha_i^t + B \sqrt{\frac{t}{H \log B}} \sum_{i=1}^{t} (\alpha_i^t)^2 + B \alpha_t \eta_t.$$

To further simplify the above upper bound, consider two cases:

- If $H_t / t \leq 1$, $\sqrt{H_t / t} \geq H_t / t$ and thus $\tilde{R}_{\text{swap}}(t) \leq 10B \sqrt{H_t / t}$.

- If $H_t / t \geq 1$, $B \sqrt{H_t / t} \geq 1 \geq \tilde{R}_{\text{swap}}(t)$ where the last step is by the definition of $\tilde{R}_{\text{swap}}(t)$. Therefore we have $\tilde{R}_{\text{swap}}(t) \leq B \sqrt{H_t / t}$.

Combine the above two cases, $\tilde{R}_{\text{swap}}(t) \leq 10B \sqrt{H_t / t}$.

Finally, we pick $\xi_{\text{sw}}(B, t, \log(1/\delta)) := 10B \sqrt{H_t / t}$, which is non-decreasing in $B$. On the other hand, since $\sum_{t=1}^{t} \xi_{\text{sw}}(B, t, \log(1/\delta)) \leq 20B \sqrt{H_t t_t}$, we choose $\Xi_{\text{sw}}(B, t, \log(1/\delta)) = 20B \sqrt{H_t t_t}$, which is concave in $t$.

To prove Theorem 26, we again first decompose the swap regret. We first note that by Line 7 of Algorithm 6, we have:

$$w_i \langle \theta_i, l_i \rangle = \sum_{b \in B} w_i \langle \hat{\theta}_i(b), \theta_i(b) l_i \rangle.$$

On the other hand, by the definition of strategy modification $\Psi$, we have

$$\min_{\psi \in \Psi} \sum_{i=1}^{t} w_i \langle \psi \circ \theta_i, l_i \rangle = \sum_{b \in B} \min_{\theta \in \theta^*} \sum_{i=1}^{t} w_i \theta_i(b) \cdot <\theta^*(b), l_i \rangle.$$
Therefore, we have the following decomposition of the swap regret

\[ R_{\text{swap}}(t) := \min_{\psi \in \mathbb{W}} \sum_{i=1}^{t} w_i [\langle \theta_i, l_i \rangle - \langle \psi \circ \theta_i, l_i \rangle] = \sum_{b \in B} \sum_{i=1}^{t} w_i [\langle \hat{\theta}_i(b) - \theta^*(b), \theta_i(b) l_i \rangle] \]

\[ = \sum_{b \in B} \sum_{i=1}^{t} w_i \left( \hat{\theta}_i(b) - \theta^*(b), \hat{\theta}_i(b) \right) + \sum_{b \in B} \sum_{i=1}^{t} w_i \left( \hat{\theta}_i(b), \theta_i(b) l_i(b) - \hat{\theta}_i(b) \right) \]

\[ + \sum_{b \in B} \sum_{i=1}^{t} w_i \left( \theta^*(b), \hat{\theta}_i(b) - \theta_i(b) l_i(b) \right) \]

For the remaining proof, we bound term \((A), (B), (C)\) separately in Lemma 27, Lemma 28.

**Lemma 27.** For any \( t \in [K] \), suppose \( \eta_i \leq 2\gamma_i \) for all \( i \leq t \). The with probability \( 1 - \delta \), for any \( \theta^* \),

\[ \sum_{b \in B} \sum_{i=1}^{t} w_i \langle \hat{\theta}_i(b) - \theta^*(b), \hat{\theta}_i(b) \rangle \leq \frac{w_t B \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \eta_i. \]

**Proof.** Similar to Lemma 22, we have,

\[ \sum_{i=1}^{t} w_i \langle \hat{\theta}_i(b) - \theta^*(b), \hat{\theta}_i(b) \rangle \leq \frac{w_t \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \eta_i w_i \langle \hat{\theta}_i(b), \hat{\hat{\theta}}_i^2(b) \rangle \]

\[ = \frac{w_t \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{b' \in B} \eta_i w_i \hat{\theta}_i(b') \frac{\theta_i^2(b') \hat{\theta}_i(b') \hat{\theta}_i(b) - \theta_i^2(b) \hat{\theta}_i(b') \hat{\theta}_i(b)}{\theta_i(b') + \gamma_i} \]

\[ \leq \frac{w_t \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{b' \in B} \eta_i w_i \hat{\theta}_i(b') \frac{\hat{\theta}_i(b') \theta_i(b) \hat{\theta}_i(b)}{\theta_i(b')} \]

Summing over \( b \) and using the fact that \( \sum_{b \in B} \hat{\theta}_i(b') \theta_i(b) = \theta_i(b') \),

\[ \sum_{b \in B} \sum_{i=1}^{t} w_i \langle \hat{\theta}_i(b) - \theta^*(b), \hat{\theta}_i(b) \rangle \leq \frac{w_t B \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{b' \in B} \eta_i w_i \frac{\hat{\theta}_i(b') \theta_i(b)}{\theta_i(b')} \]

\[ \leq \frac{w_t B \log B}{\eta_t} + \frac{1}{2} \sum_{i=1}^{t} \sum_{b' \in B} \eta_i w_i \frac{\hat{\theta}_i(b') \theta_i(b)}{\theta_i(b')} + \frac{1}{2} \max_{i \leq t} w_i \eta_i \]

\[ \leq \frac{w_t B \log B}{\eta_t} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \eta_i \]

where \((i)\) is by using Lemma 21 with \( c_i(b) = \eta_i \). Notice the quantity \( \frac{\hat{\theta}_i(b') \theta_i(b)}{\theta_i(b')} \) actually doesn’t depend on \( b \), so it is well-defined even after we take the summation with respect to \( b \). The any-time guarantee is justified by taking union bound.

**Lemma 28.** For any \( t \in [K] \), with probability \( 1 - \delta \),

\[ \sum_{b \in B} \sum_{i=1}^{t} w_i \langle \hat{\theta}_i(b), \theta_i(b) l_i(b) - \hat{\theta}_i(b) \rangle \leq B \sum_{i=1}^{t} \gamma_i w_i + B \sqrt{2t \sum_{i=1}^{t} w_i^2}. \]
Proof. We further decompose it into
\[
\sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot) - \hat{i}_i(\cdot|b) \right) = \sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot) - \mathbb{E}_i \hat{i}_i(\cdot|b) \right) + \sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \mathbb{E}_i \hat{i}_i(\cdot|b) - \hat{i}_i(\cdot|b) \right).
\]

The first term is bounded by
\[
\sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot) - \mathbb{E}_i \hat{i}_i(\cdot|b) \right) = \sum_{i=1}^{t} w_i \theta_i(b) \left( \hat{\theta}_i(\cdot|b), (1 - \frac{\theta_i(\cdot)}{\theta_i(\cdot) + \gamma_i})l_i(\cdot) \right) = \sum_{i=1}^{t} w_i \theta_i(b) \left( \hat{\theta}_i(\cdot|b), \frac{\gamma_i}{\theta_i(\cdot) + \gamma_i} l_i(\cdot) \right).
\]

So by taking the sum with respect to \( b \), we have
\[
\sum_{b \in B} \sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \theta_i(b)l_i(\cdot) - \mathbb{E}_i \hat{i}_i(\cdot|b) \right) \leq \sum_{b \in B} \sum_{i=1}^{t} w_i \theta_i(b) \left( \hat{\theta}_i(\cdot|b), \frac{\gamma_i}{\theta_i(\cdot) + \gamma_i} l_i(\cdot) \right) \leq \sum_{b \in B} \sum_{i=1}^{t} w_i \gamma_i l_i(b') \leq B \sum_{i=1}^{t} \gamma_i w_i.
\]

To bound the second term, notice \( \hat{\theta}_i(b'|b) \theta_i(b) \leq \theta_i(b') \) for any \( b, b' \in B \),
\[
\left( \hat{\theta}_i(\cdot|b), \hat{i}_i(\cdot|b) \right) \leq \sum_{b' \in B} \hat{\theta}_i(b'|b) \theta_i(b) \sum_{b' \in B} \|b_i = b'\| \leq \sum_{b' \in B} \sum_{b \in B} \|b_i = b'\| = 1,
\]
thus \( \{ w_i \left( \hat{\theta}_i(\cdot|b), \mathbb{E}_i \hat{i}_i(\cdot|b) - \hat{i}_i(\cdot|b) \right) \}_{i=1}^{t} \) is a bounded martingale difference sequence w.r.t. the filtration \( \{ \mathcal{F}_t \}_{t=1}^{K} \). By Azuma-Hoeffding,
\[
\sum_{i=1}^{t} w_i \left( \hat{\theta}_i(\cdot|b), \mathbb{E}_i \hat{i}_i(\cdot|b) - \hat{i}_i(\cdot|b) \right) \leq \sqrt{2t \sum_{i=1}^{t} w_i^2}.
\]

The proof is completed by taking the summation with respect to \( b \) and a union bound.

Lemma 29. For any \( t \in [K] \), suppose \( \gamma_i \) is non-increasing in \( i \), then with probability \( 1 - \delta \), and any \( \theta^* \),
\[
\sum_{b \in B} \sum_{i=1}^{t} w_i \left( \theta^*(\cdot|b), \hat{i}_i(\cdot|b) - \theta_i(b)l_i(\cdot) \right) \leq B \max_{i \leq t} w_i / \gamma_i.
\]

Proof. The proof follows from Lemma 24 and taking the summation with respect to \( b \).

Finally, we are ready to prove Theorem 26.

Proof of Theorem 26 Recall the decomposition of swap regret equation 17. We bound (A) in Lemma 27 in Lemma 28 and (C) in Lemma 29. Putting everything together, we have
\[
\mathcal{R}_{\text{swap}}(t) \leq \frac{w_i B \log B}{\eta_i} + \frac{B}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i + B \sum_{i=1}^{t} \gamma_i w_i + B \sqrt{2t \sum_{i=1}^{t} w_i^2 + B \max_{i \leq t} w_i / \gamma_i}.
\]

\[\square\]