Policy Optimization for Markov Games: Unified Framework and Faster Convergence

Runyu Zhang*
Harvard University
runyuzhang@fas.harvard.edu

Qinghua Liu*
Princeton University
qinghual@princeton.edu

Huan Wang
Salesforce Research
huan.wang@salesforce.com

Caiming Xiong
Salesforce Research
cxiong@salesforce.com

Na Li
Harvard University
nali@seas.harvard.edu

Yu Bai
Salesforce Research
yu.bai@salesforce.com

Abstract

This paper studies policy optimization algorithms for multi-agent reinforcement learning. We begin by proposing an algorithm framework for two-player zero-sum Markov Games in the full-information setting, where each iteration consists of a policy update step at each state using a certain matrix game algorithm, and a value update step with a certain learning rate. This framework unifies many existing and new policy optimization algorithms. We show that the *state-wise average policy* of this algorithm converges to an approximate Nash equilibrium (NE) of the game, as long as the matrix game algorithms achieve low weighted regret at each state, with respect to weights determined by the speed of the value updates. Next, we show that this framework instantiated with the Optimistic Follow-The-Regularized-Leader (OFTRL) algorithm at each state (and smooth value updates) can find an $\widetilde{\mathcal{O}}(T^{-5/6})$ approximate NE in T iterations, and a similar algorithm with slightly modified value update rule achieves a faster $\widetilde{\mathcal{O}}(T^{-1})$ convergence rate. These improve over the current best $\widetilde{\mathcal{O}}(T^{-1/2})$ rate of symmetric policy optimization type algorithms. We also extend this algorithm to multi-player general-sum Markov Games and show an $\widetilde{\mathcal{O}}(T^{-3/4})$ convergence rate to Coarse Correlated Equilibria (CCE). Finally, we provide a numerical example to verify our theory and investigate the importance of smooth value updates, and find that using "eager" value updates instead (equivalent to the independent natural policy gradient algorithm) may significantly slow down the convergence, even on a simple game with $H = 2$ layers.

1 Introduction

Policy optimization, i.e. algorithms that learn to make sequential decisions by local search on the agent's policy directly, is a widely used class of algorithms in reinforcement learning [40, 44, 45]. Policy optimization algorithms are particularly advantageous in the multi-agent reinforcement learning (MARL) setting (e.g. compared with value-based counterparts), due to their typically lower representational cost and better scalability in both training and execution. A variety of policy optimization algorithms such as Independent PPO [14], MAPPO [56], QMix [42] have been proposed to solve real-world MARL problems [4, 39, 43]. These algorithms share a same high-level structure with iterative *value updates* (for certain value estimates) and *policy updates* (often independently with each agent) using information from the value estimates and/or true rewards.

*The two authors contributed equally to this work.

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

While policy optimization for MARL has been studied theoretically in a growing body of work, there are still gaps between algorithms used in practice and provably-efficient algorithms studied in theory—Algorithms in practice generally follow two natural design principles: *symmetric updates* among all agents, and *simultaneous learning* of values and policies [59, 56]. By contrast, policy optimization algorithms studied in theory often diverge from these principles and incorporate some tweaks, such as (i) asymmetric updates, where one agent takes a much smaller learning rate than the others (two time-scale) [11] or waits until the other agents learn an approximate best response [62]; and (ii) batch-like learning, where policies are optimized to sufficient precision with respect to the current value estimate before the next value update [8]. There is so far a lacking of systematic studies on the performance of the more vanilla policy optimization algorithms following the above two principles, even under the setting where full-information feedback from the game is available.

Towards bridging these gaps, this paper studies policy optimization algorithms for Markov games, with a focus on algorithms with symmetric updates and simultaneous learning of values and policies. Our contributions can be summarized as follows:

- We propose an algorithm framework for two-player zero-sum Markov games in the full-information setting (Section 3). This framework unifies many existing and new policy optimization algorithms such as Nash V-Learning, Gradient Descent/Ascent, as well as seemingly disparate algorithms such as Nash Q-Learning (Section 3.2). We prove that the *state-wise average policy* outputted by the above algorithm is an approximate Nash Equilibrium (NE), so long as suitable per-state weighted regrets are bounded (Section 3.1). This generic result can be instantiated in a modular fashion to derive convergence guarantees for the many examples above.
- We instantiate our framework to show that a new algorithm based on Optimistic Follow-The-Regularized-Leader (OFTRL) and smooth value updates finds an $\widetilde{\mathcal{O}}(T^{-5/6})$ approximate NE in T iterations (Section 4). This improves over the current best rate of $\widetilde{\mathcal{O}}(T^{-1/2})$ achieved by symmetric policy optimization type algorithms. In addition, we also propose a slightly modified OFTRL algorithm that further improves the rate to $\widetilde{\mathcal{O}}(T^{-1})$, which matches with the known best rate for all policy optimization type algorithm.
- We additionally extend the above OFTRL algorithm to multi-player general-sum Markov games and show an $\widetilde{\mathcal{O}}(T^{-3/4})$ convergence rate to Coarse Correlated Equilibria (CCE), which is also the first rate faster than $\widetilde{\mathcal{O}}(T^{-1/2})$ for policy optimization in general-sum Markov games (Section 4.1).
- We perform simulations on a carefully constructed zero-sum Markov game with $H = 2$ layers to verify our convergence guarantees. The numerical tests further suggest the importance of *smooth value updates*: the Independent Natural Policy Gradient algorithm (as one instantiation of our algorithm framework with "eager" value updates) appears to converge much slower (Section 5).

1.1 Related work

Two-player zero-sum MGs Markov games (MGs) [30] (also known as Stochastic Games [47]) is a widely studied model for multi-agent reinforcement learning. In the most basic setting of two-player zero-sum MGs, algorithms for computing the NE have been extensively studied in both the full-information setting [31, 20, 19] and the sample-based/online setting [6, 52, 22, 48, 58, 2, 55, 3, 33, 25, 21, 57, 10, 34]. Our algorithm framework incorporates (the full-information version of) several algorithms in this line of work.

Policy optimization for zero-sum MGs Policy optimization for single-agent Markov Decision Processes has been extensively in a recent line of work, e.g. [1, 5, 32, 46, 37, 35, 7, 15, 54] and the many references therein. For two-player zero-sum MGs, the Nash V-Learning algorithm of Bai et al. [3] (originally proposed for the sample-based online setting) can be viewed as an independent policy optimization algorithm, and can be adapted to the full-information setting with $\widetilde{\mathcal{O}}(T^{-1/2})$ convergence rate. Daskalakis et al. [11] prove that the independent policy gradient algorithm with an asymmetric two time-scale learning rate can learn the NE (for one player only) with polynomial iteration/sample complexity. Zhao et al. [62] show that another asymmetric algorithm that simulates a policy gradient/best response dynamics converges to NE with $\widetilde{\mathcal{O}}(1/T)$ rate. Cen et al. [8] use a symmetric optimistic (extragradient) subroutine for matrix games to learn zero-sum MGs in a *layer-wise* fashion (the matrix games at each state are learned to sufficient precision before the

backup, more like a Value Iteration type algorithm), and also derive an $\widetilde{\mathcal{O}}(1/T)$ convergence rate. The closest to our work is Wei et al. [53] which proves that Optimistic Gradient Descent/Ascent (OGDA), combined with smooth value updates at all layers simultaneously, converges to an NE with $\widetilde{\mathcal{O}}(T^{-1/2})$ rate for both the average duality gap and the last iterate. The $\widetilde{\mathcal{O}}(T^{-5/6})$ rate of our OFTRL algorithm improves over [53] and is the first such faster rate for symmetric, policy optimization type algorithms. The $\widetilde{\mathcal{O}}(1/T)$ rate of our modified OFTRL algorithm matches with rates in [62, 7], while still maintaining symmetric update and simultaneous learning of values and policies.

Multi-player general-sum MGs A recent line of work shows that a generalization of the V-learning algorithm to the multi-player general-sum setting can learn Coarse Correlated Equilibria (CCE) [49, 24, 36] and Correlated Equilibria (CE) [49, 24]. The algorithmic designs in these works are specially tailored to the sample-based setting, where the best possible rate is $\widetilde{\mathcal{O}}(T^{-1/2})$.[2] In contrast, this paper considers the full-information setting and proposes new algorithms achieving the faster $\widetilde{\mathcal{O}}(T^{-3/4})$ for learning CCE in general-sum MGs. The complexity of learning a structured CCE (such as Markovian or stationary Markovian) has been studed in [13, 26]. Another recent line of work considers learning NE in Markov Potential Games [60, 29, 49, 16, 61], which can be seen as a cooperative-type subclass of general-sum MGs.

Optimistic algorithms in normal-form games Technically, our accelerated rates build on the recent line of work on faster rates for optimistic no-regret algorithms in normal-form games [50, 41, 9, 12]. Specifically, our $\tilde{\mathcal{O}}(T^{-5/6})$ rate for two-player zero-sum MGs builds upon a first-order smoothness analysis of [9], our improved $\widetilde{\mathcal{O}}(T^{-1})$ rate in the same setting (achieved by the modified OFTRL algorithm) leverages analysis of [41, 50] on bounding the summed regret over the two players, and our $\tilde{\mathcal{O}}(T^{-3/4})$ rate for multi-player general-sum MGs follows from the RVU-property [50, Definition 3]. Our incorporation of these techniques involves non-trivial new components such as *weighted* first-order smoothness bounds and handling changing game rewards.

2 Preliminaries

We consider the tabular episodic (finite-horizon) two-player-zero-sum Markov games (MGs), which can be denoted as $\mathcal{M}(H, \mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$, where H is the horizon length; $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$; $\mathcal{A}, \mathcal{B}$ are the action space of the *max-player* and *min-player* respectively, with $|\mathcal{A}| = A, |\mathcal{B}| = B$; $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is the transition probabilities, where each $\mathbb{P}_h(s'|s,a,b)$ gives the probability of transition to state s' from state-action (s,a,b); $r = \{r_h\}_{h=1}^H$ are the reward functions, such that $r_h(s,a,b)$ is reward[3] of the max-player and $-r_h(s,a,b)$ is the reward of the min-player at time step h and state-action (s,a,b). In each episode, the MG starts with a deterministic initial state s_1. Then at each time step $1 \le h \le H$, both players observes the state s_h, the max-player takes an action $a_h \in \mathcal{A}$, and the min-player takes an action $b_h \in \mathcal{B}$. Then, both players receive their rewards $r_h(s_h, a_h, b_h)$ and $-r_h(s_h, a_h, b_h)$, respectively, and the system transits to the next state $s_{t+1} \sim \mathbb{P}_h(\cdot|s_h, a_h, b_h)$.

Policies & value functions A (Markov) policy μ of the max-player is a collection of policies $\mu = \{\mu_h : \mathcal{S} \to \Delta_{\mathcal{A}}\}_{h=1}^H$, where each $\mu_h(\cdot|s_h) \in \Delta_{\mathcal{A}}$ specifies the probability of taking action a_h at (h, s_h). Similarly, a (Markov) policy ν of the min-player is defined as $\nu = \{\nu_h : \mathcal{S} \to \Delta_{\mathcal{B}}\}$. For any policy (μ, ν) (not necessarily Markov), we use $V_h^{\mu,\nu} : \mathcal{S} \to \mathbb{R}$ and $Q_h^{\mu,\nu} : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ to denote the value function and Q-function at time step h, respectively, i.e.

$$V_h^{\mu,\nu}(s) := \mathbb{E}_{\mu,\nu}\Big[\textstyle\sum_{h=h'}^H r_{h'}(s_{h'}, a_{h'}, b_{h'}) \mid s_h = s\Big], \tag{1}$$

$$Q_h^{\mu,\nu}(s,a,b) := \mathbb{E}_{\mu,\nu}\Big[\textstyle\sum_{h=h'}^H r_{h'}(s_{h'}, a_{h'}, b_{h'}) \mid s_h = s, a_h = a, b_h = b\Big]. \tag{2}$$

For notational simplicity, we use the following abbreviation: $[\mathbb{P}_h V](s,a,b) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a,b)} V(s')$ for any value function V. By definition of the value functions and Q-functions, we have the following Bellman equations

$$Q_h^{\mu,\nu}(s,a,b) = \big(r_h + \mathbb{P}_h V_{h+1}^{\mu,\nu}\big)(s,a,b),$$

[2] Even if we specialize their algorithms to the full-information setting, the attained rates are no better than $\widetilde{\mathcal{O}}(T^{-1/2})$ because the bandit subroutines they deployed in V-learning converge no faster than $\Omega(T^{-1/2})$ even in the simplest setting of full-information matrix games.

[3] This assumes deterministic rewards; our results can be generalized directly to the case of stochastic rewards.

$$V_h^{\mu,\nu}(s,a,b) = \mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}[Q_h^{\mu,\nu}(s,a,b)] = \langle Q_h^{\mu,\nu}(s,\cdot,\cdot), \mu(\cdot|s)\times\nu(\cdot|s)\rangle\,.$$

The goal for the max-player is to maximize the value function, whereas the goal for the min-player is to minimize the value function.

Best response & Nash equilibrium For any Markov policy μ of a max-player, there exists a best response for the min-player, which can be taken as a Markov policy $\nu^\dagger(\mu)$ such that $V_h^{\mu,\nu^\dagger(\mu)}(s) = \inf_\nu V_h^{\mu,\nu}(s)$ for all $(s,h)\in\mathcal{S}\times[H]$. For simplicity we define $V_h^{\mu,\dagger} := V_h^{\mu,\nu^\dagger(\mu)}$. By symmetry, we can also define $\mu^\dagger(\nu)$ and $V_h^{\dagger,\nu}$. It is known (e.g. [17]) that there exist Markov policies $(\mu^\star,\nu^\star)$ that perform optimally against best responses. These policies are also equivalent to Nash Equilibria (NEs) of the game, where no player can gain by switching to a different policy unilaterally. It can also be shown that any NE $(\mu^\star,\nu^\star)$ satisfies the following minimax equation

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

Thus, while the NE policy $(\mu^\star,\nu^\star)$ may not be unique, all of them share the same value functions, which we denote as $V_h^\star := V_h^{\mu^\star,\nu^\star}$. The Q-function $Q_h^\star$ can be defined similarly. In this paper, our main goal is to find an approximate NE, which is formally defined below.

Definition 1 (ε-approximate Nash Equilibrium)**.** *For any $\varepsilon\ge 0$, a policy (μ,ν) is an ε-approximate Nash Equilibrium (ε-NE) if* $\mathrm{NEGap}(\mu,\nu) := V_1^{\dagger,\nu}(s_1) - V_1^{\mu,\dagger}(s_1) \le \varepsilon$.

Full-information setting We consider finding an approximate NE under the full-information setting, where we can query the exact value of $r_h + \mathbb{P}_h V_{h+1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}\times\mathcal{B}}$ for any layer $h\in[H]$ and V function $V_{h+1}\in\mathbb{R}^{\mathcal{S}}$. Further, the majority of algorithms are policy optimization algorithms that only query $(r_h+\mathbb{P}_hV_{h+1})\nu_h \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $(r_h+\mathbb{P}_hV_{h+1})^\top\mu_h \in \mathbb{R}^{\mathcal{S}\times\mathcal{B}}$ for policies μ_h,ν_h. (See Appendix I.1 for additional discussions.)

Additional notation For any $(h,s)\in[H]\times\mathcal{S}$ and Q function $Q_h:\mathcal{S}\times\mathcal{A}\times\mathcal{B}\to\mathbb{R}$, we define shorthand $[(\mu_h)^\top Q_h\nu_h](s) := \langle Q_h(s,\cdot,\cdot), \mu_h(\cdot|s)\times\nu_h(\cdot|s)\rangle$ for any policy (μ,ν). Similarly, we let $[Q_h\nu_h](s,\cdot) := \mathbb{E}_{b\sim\nu_h(\cdot|s)}[Q_h(s,\cdot,b)]\in\mathbb{R}^A$, and $[Q_h^\top\mu_h](s,\cdot) := \mathbb{E}_{a\sim\mu_h(\cdot|s)}[Q_h(s,a,\cdot)]\in\mathbb{R}^B$. We use $A\vee B := \max\{A,B\}$.

3 An algorithm framework for zero-sum Markov games

We begin by presenting an algorithm framework that unifies many existing and new algorithms for two-player zero-sum Markov Games, and its performance guarantee that could be specialized to yield concrete convergence results for many specific algorithms.

Our algorithm framework, described in Algorithm 1, consists of two main components: the *policy update* step computing policies (μ^t,ν^t), and the *value update* step computing the Q estimate Q_h^t's.

Policy update via matrix game algorithms In the policy update step (Line 5), for each (h,s), the two players update policies $(\mu_h^t(\cdot|s),\nu_h^t(\cdot|s))$ at (h,s) using some matrix game algorithm $\mathsf{MatrixGameAlg}$ which takes as input all past Q matrices and all past policies of both players. The $\mathsf{MatrixGameAlg}$ offers a flexible interface that allows many choices such as the matrix NE subroutine over the most recent Q matrix $\mathsf{MatrixNE}(Q_h^{t-1}(s,\cdot,\cdot))$, or any independent no-regret algorithm (for both players), such as Follow-The-Regularized-Leader (FTRL) (10) or projected Gradient Descent-Ascent (11) considered in the examples later.

Value update with learning rate $\{\beta_t\}$ For any (h,s,a,b), the value update step (Line 6) updates $Q_h^t(s,a,b)$ by the newest value function $r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t\nu_{h+1}^t]$ propagated from layer $h+1$, using a sequence of learning rates $\{\beta_t\}_{t\ge1}$ which we assume to be within $[0,1]$ (with $\beta_1 := 1$). $\{\beta_t\}$ controls the speed of the value update, with two important special cases:

(1) *Eager* value updates, where we set $\beta_t = 1$ so that Q_h^t performs *policy evaluation* of the current policy (μ^t,ν^t), that is, $Q_h^t = Q_h^{\mu^t,\nu^t}$.

(2) *Smooth* (incremental) value updates, where we choose $\beta_t\to 0$ as $t\to\infty$. In this case, the Q_h^t moves slower (resembling a *critic* in Actor-Critic like algorithms), and becomes a weighted average of all past updates. A standard choice that is frequently used is from [23] (and many subsequent work),

$$\beta_t = \alpha_t := (H+1)/(H+t). \tag{3}$$

Algorithm 1 Algorithm framework for two-player zero-sum Markov Games

Require: Learning rate $\{\beta_t\}_{t\geq 1} \subset [0,1]$ (with $\beta_1 = 1$); Algorithm MatrixGameAlg.

Initialize: $Q_h^0(s,a,b) \leftarrow H - h + 1$ for all $(h,s,a,b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$.

for $t = 1, \ldots, T$ **do**

  for $h = H, \ldots, 1$ **do**

    Policy update: Update policies for all $s \in \mathcal{S}$:

$$(\mu_h^t(\cdot|s), \nu_h^t(\cdot|s)) \leftarrow \mathsf{MatrixGameAlg}\Big(\big\{Q_h^i(s,\cdot,\cdot)\big\}_{i=1}^{t-1}, \big\{\mu_h^i(\cdot|s)\big\}_{i=1}^{t-1}, \big\{\nu_h^i(\cdot|s)\big\}_{i=1}^{t-1}\Big).$$

    Value update: Update Q-value for all $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$:

$$Q_h^t(s,a,b) \leftarrow (1-\beta_t) Q_h^{t-1}(s,a,b) + \beta_t\big(r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t \nu_{h+1}^t]\big)(s,a,b). \quad (4)$$

Output: State-wise average policy $(\widehat{\mu}^T, \widehat{\nu}^T)$, with β_T^t defined in (6):

$$\widehat{\mu}_h^T(\cdot|s) \leftarrow \textstyle\sum_{t=1}^T \beta_T^t \mu_h^t(\cdot|s), \quad \widehat{\nu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \beta_T^t \nu_h^t(\cdot|s) \quad \text{for all } (h,s) \in [H] \times \mathcal{S}. \quad (5)$$

For any $\{\beta_t\}$, the update (4) implies that

$$Q_h^t(s,a,b) = \textstyle\sum_{i=1}^t \beta_t^i \Big(\Big[r_h + \mathbb{P}_h[\big(\mu_{h+1}^i\big)^\top Q_{h+1}^i \nu_{h+1}^i]\Big](s,a,b)\Big),$$

where β_t^i's are a group of weights summing to one ($\sum_{i=1}^t \beta_t^i = 1$) defined as

$$\beta_t^t = \beta_t; \qquad \beta_t^i = \textstyle\prod_{j=i+1}^t (1-\beta_j)\beta_i, \ \text{ for } \ i \in [t-1]. \quad (6)$$

Note that with smooth value updates ($\beta_t < 1$), Q_h^t is not necessarily the Q-function of any policy. Upon finishing, the algorithm outputs the *state-wise average policy* $(\widehat{\mu}^T, \widehat{\nu}^T)$ defined in (5), where each $\widehat{\mu}_h^T(\cdot|s)$ is the weighted average of $\mu_h^t(\cdot|s)$ using weights $\{\beta_T^t\}_{t=1}^T$ (and similarly for $\widehat{\nu}^T$), which we remark can be implemented efficiently using moving averages (cf. Appendix C.1).

Symmetric & simultaneous learning, (de)centralization We remark that Algorithm 1 by definition performs simultaneous learning (of policies and values) at all layers, and also yields symmetric (policy) updates if MatrixGameAlg is a symmetric algorithm with respect to μ and ν. Also, although Algorithm 1 appears to be a *centralized* algorithm as it maintains Q values in (4), this does not preclude possibilities that the algorithm can be executed in a *decentralized* fashion. This can happen e.g. when the Q-update (4) can be rewritten as an equivalent V-update (cf. Example 1 & 2).

3.1 Theoretical guarantee

We are now ready to state the main theoretical guarantee of Algorithm 1, which states that the $(\widehat{\mu}^T, \widehat{\nu}^T)$ is an approximate NE, as long as the algorithm achieves low *per-state weighted regrets* w.r.t. weights $\{\beta_t^i\}_{i=1}^t$, defined as

$$\begin{aligned}
&\mathrm{reg}_{h,\mu}^t(s) := \max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \textstyle\sum_{i=1}^t \beta_t^i \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \left[Q_h^i \nu_h^i\right](s,\cdot)\right\rangle, \\
&\mathrm{reg}_{h,\nu}^t(s) := \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \textstyle\sum_{i=1}^t \beta_t^i \left\langle \nu_h^i(\cdot|s) - \nu^\dagger, \left[(Q_h^i)^\top \mu_h^i\right](s,\cdot)\right\rangle, \\
&\mathrm{reg}_h^t := \max_{s \in \mathcal{S}} \max\{\mathrm{reg}_{h,\mu}^t(s), \mathrm{reg}_{h,\nu}^t(s)\}.
\end{aligned} \quad (7)$$

Theorem 2 (Main guarantee of Algorithm 1)**.** *Suppose that the per-state regrets can be upper-bounded as* $\mathrm{reg}_h^t \leq \overline{\mathrm{reg}}_h^t$ *for all* $(h,t) \in [H] \times [T]$*, where* $\overline{\mathrm{reg}}_h^t$ *is non-increasing in* t*:* $\overline{\mathrm{reg}}_h^t \geq \overline{\mathrm{reg}}_h^{t+1}$ *for all* $t \geq 1$*. Then, the output policy* $(\widehat{\mu}^T, \widehat{\nu}^T)$ *of Algorithm 1 satisfies*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq C\left[H \max_{h \in [H]} \overline{\mathrm{reg}}_h^T + H^2 c_\beta^H \log T \cdot \frac{1}{T} \sum_{t=1}^T \max_{h \in [H]} \overline{\mathrm{reg}}_h^t\right] \quad (8)$$

for all $T \geq 2$ *and some absolute constant* $C > 0$*, where* c_β *is a constant depending on* $\{\beta_t\}_{t\geq 1}$*:*

$$c_\beta := \sup_{j \geq 1} \textstyle\sum_{t=j}^\infty \beta_t^j \geq 1. \quad (9)$$

Specifically, $c_\beta = \left(1 + \frac{1}{H}\right)$ *if* $\beta_t = \alpha_t = \frac{H+1}{H+t}$*, and* $c_\beta = 1$ *if* $\beta_t = 1$*.*

Bound (8) is typically dominated by the second term on the right hand side, suggesting that the NEGap can be bounded by the average weighted regret $\widetilde{\mathcal{O}}\Big(\frac{1}{T}\sum_{t=1}^{T}\max_h \overline{\mathrm{reg}}_h^t\Big)$, if $c_\beta^H = O(1)$. Theorem 2 serves as a modular tool for analyzing a broad class of algorithms: As long as this average regret is sublinear in T (including—but not limited to—choosing MatrixGameAlg as uncoupled no-regret algorithms), the output policy will be an approximate NE. We emphasize though that this result is not yet end-to-end, as each reg_h^t is a weighted regret w.r.t. *the particular set of weights* $\{\beta_t^i\}_{i=1}^t$, minimizing which may require careful algorithm designs and/or case-by-case analyses. We provide some concrete examples in Section 3.2 to demonstrate the usefulness of Theorem 2.

We remark that the state-wise average policy considered in Theorem 2 is an average policy that is also *Markovian* by definition, which is different from existing work which considers either the (Markovian) last iterate [53] or non-Markovian average policies (e.g. [3]). However, this guarantee relies on full-information feedback (so that per-state regret bounds are available), and it remains an open question how such guarantees could be generalized to sample-based settings.

Proof overview The proof of Theorem 2 follows by (1) bounding $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T)$ in terms of per-state regrets w.r.t. the *Nash value functions* $Q_h^\star$'s by performance difference arguments (Lemma C.1); (2) recursively bounding the value estimation error $\delta_h^t := \|Q_h^t - Q_h^\star\|_\infty$ (Lemma C.2) which yields the constant c_β; and (3) combining the above to translate the regret from $Q_h^\star$'s to Q_h^t's (which we assume to be bounded by $\overline{\mathrm{reg}}_h^t$) and obtain the theorem. The full proof can be found in Appendix C.

3.2 Examples

We now demonstrate the generality of Algorithm 1 and Theorem 2 by showing that they subsume many existing algorithms (and yield new algorithms) for two-player-zero-sum Markov games, and provide new guarantees with the particular output policy (5).

Example 1 (Nash V-Learning [3], full-information version)**:** The full algorithm (Algorithm 5) can be found in Appendix D.1. The algorithm is a special case of Algorithm 1 with $\beta_t = \alpha_t = (H+1)/(H+t)$, and MatrixGameAlg chosen as the weighted FTRL algorithm

$$\mu_h^t(a|s) \propto_a \exp\left(\frac{\eta}{w_{t-1}}\sum_{i=1}^{t-1} w_i\big[Q_h^i \nu_h^i\big](s,a)\right),\ \nu_h^t(b|s) \propto_b \exp\left(-\frac{\eta}{w_{t-1}}\sum_{i=1}^{t-1} w_i\Big[\big(Q_h^i\big)^\top \mu_h^i\Big](s,b)\right), \tag{10}$$

where $w_t := \alpha_t^t/\alpha_t^1$. Combining Theorem 2 with the standard regret bound of weighted FTRL, this algorithm achieves $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le \widetilde{\mathcal{O}}(H^{7/2}/\sqrt{T})$ choosing $\eta \asymp 1/\sqrt{T}$ (Proposition D.2).

Additionally, although the original Nash V-learning algorithm [3] updates the V values (which makes the algorithm implementable in a decentralized fashion) instead of the Q values used in Algorithm 1, these two forms are actually equivalent in the full-information setting (Proposition D.1). $\diamondsuit$

Compared with the $\widetilde{\mathcal{O}}(\sqrt{H^5 S \max\{A,B\}/T})$ guarantee of (the non-Markovian output policy of) Nash V-Learning in the sample-based online setting [3, 51, 24], our rate achieves better (logarithmic) S, A, B dependence due to our full-information setting, and worse H dependence which happens as our output policy is the (Markovian) state-wise average policies, whose guarantee (Theorem 2) follows from a different analysis.

Example 2 (GDA-Critic)**:** This algorithm is a special case of Algorithm 1 with $\beta_t = \alpha_t = (H+1)/(H+t)$, and MatrixGameAlg as projected gradient descent/ascent (GDA), i.e.,

$$\mu_h^t(\cdot|s) \leftarrow \mathcal{P}_{\Delta_\mathcal{A}}\big(\mu_h^{t-1}(\cdot|s) + \eta\big[Q_h^{t-1}\nu_h^{t-1}\big](s)\big),\ \ \nu_h^t(\cdot|s) \leftarrow \mathcal{P}_{\Delta_\mathcal{B}}\big(\nu_h^{t-1}(\cdot|s) - \eta\big(\big[Q_h^{t-1}\big)^\top \mu_h^{t-1}\big](s)\big). \tag{11}$$

Similar as Nash V-Learning, GDA-Critic also admits an equivalent form with V value updates (full description in Algorithm 6). As GDA achieves weighted regret bounds with any monotone weights including $\{\alpha_t^i\}_{i=1}^t$ (Lemma B.1), we can invoke Theorem 2 to show that this algorithm achieves $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le \widetilde{\mathcal{O}}(H^{7/2}(A \vee B)^{1/2}/\sqrt{T})$ if we choose $\eta \asymp 1/\sqrt{T}$ (Proposition D.4).

The GDA-critic algorithm is also similar to the OGDA-MG algorithm of Wei et al. [53], except that we use the (non-optimistic) vanilla version of GDA. To our best knowledge, the above algorithm and guarantee are not known. We remark that even ignoring difference between GDA and OGDA, the

above guarantee cannot be obtained by direct adaptation of the results of [53] which focus on either the average duality gap and/or last-iterate convergence. $\diamondsuit$

Besides the above examples, Algorithm 1 also incorporates the following algorithms which are typically not categorized as policy optimization algorithms (see Appendix I.2 for a discussion of the categorization).

Example 3 (Nash Q-Learning [20, 3], full-information version)**:** This algorithm is a special case of Algorithm 1 with $\beta_t = \alpha_t = (H+1)/(H+t)$ and MatrixGameAlg as the matrix Nash subroutine

$$(\mu_h^t(\cdot|s), \nu_h^t(\cdot|s)) \leftarrow \mathsf{MatrixNE}(Q_h^{t-1}(s,\cdot,\cdot)) := \arg\left(\min_{\mu\in\Delta_{\mathcal{A}}} \max_{\nu\in\Delta_{\mathcal{B}}} \mu^\top Q_h^{t-1}(s,\cdot,\cdot)\nu\right).$$

(Full description in Algorithm 7.) Although $\mathsf{MatrixNE}(Q_h^{t-1}(s,\cdot,\cdot))$ is not by default a no-regret algorithm, using the fact that $\left\|Q_h^t - Q_h^{t-1}\right\|_\infty$ is small (due to the small α_t) we can show that it is close to a (hypothetical) "Be-The-Leader" style algorithm that computes the matrix NE of the current Q matrix Q_h^t which achieves ≤ 0 regret (Lemma D.3). Combining this with Theorem 2 shows that this algorithm achieves $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq \widetilde{\mathcal{O}}(H^4/T)$ (Proposition D.5). $\diamondsuit$

Example 4 (Nash Policy Iteration (Nash-PI))**:** This classical algorithm (Algorithm 8) performs iterative policy evaluation and policy improvement (also similar to Nash Value Iteration [47, 2, 33]):

$$(\mu_h^{t+1}(\cdot|s), \nu_h^{t+1}(\cdot|s)) \leftarrow \mathsf{MatrixNE}(Q_h^{\mu^t,\nu^t}(s,\cdot,\cdot)). \tag{12}$$

This is also a special case of Algorithm 1 with $\beta_t = 1$ and MatrixGameAlg set as MatrixNE. It is a standard result that this algorithm converges exactly (achieving zero NE gap) in H steps, and this fact can be obtained using our framework as well (Proposition D.6). $\diamondsuit$

4 Fast convergence of optimistic FTRL

In this section, we instantiate Algorithm 1 by choosing MatrixGameAlg as the Optimistic Follow-The-Regularized-Leader (OFTRL) algorithm. OFTRL is also an uncoupled no-regret algorithm that is known to enjoy faster convergence than standard FTRL under additional loss smoothness assumptions [41, 50, 9, 12]. We show that, using OFTRL, Algorithm 1 enjoys faster convergence than the $\widetilde{\mathcal{O}}(1/\sqrt{T})$ rate of using FTRL or GDA (cf. Example 1 & 2).

Concretely, we use the following weighted OFTRL algorithm at each (h,s,t):

$$\begin{aligned}
&\mu_h^t(a|s) \propto_a \exp\left((\eta/w_t)\cdot\left[\textstyle\sum_{i=1}^{t-1} w_i(Q_h^i\nu_h^i)(s,a) + w_{t-1}(Q_h^{t-1}\nu_h^{t-1})(s,a)\right]\right),\\
&\nu_h^t(b|s) \propto_b \exp\left(-(\eta/w_t)\cdot\left[\textstyle\sum_{i=1}^{t-1} w_i((Q_h^i)^\top\mu_h^i)(s,b) + w_{t-1}((Q_h^{t-1})^\top\mu_h^{t-1})(s,b)\right]\right),
\end{aligned} \tag{13}$$

where w_t is the same weights as defined in Example 1, and we choose $\beta_t = \alpha_t = (H+1)/(H+t)$.

Theorem 3 (Fast convergence of OFTRL in zero-sum Markov Games)**.** *Suppose Algorithm 1 is instantiated with $\beta_t = \alpha_t = (H+1)/(H+t)$ and* MatrixGameAlg *to be the OFTRL algorithm* (13) *with any $\eta \leq 1/H$ (full description in Algorithm 9). Then the per-state regret can be bounded as follows for some absolute constant $C > 0$:*

$$\mathrm{reg}_h^t \leq \overline{\mathrm{reg}}_h^t := C\left[\frac{H^2\log(A\vee B)}{\eta t} + \eta^5 H^6\right] \quad \textit{for all } (h,t)\in[H]\times[T]. \tag{14}$$

Further, choosing $\eta = \mathrm{poly}(H, \log(A\vee B), \log T)\cdot T^{-1/6}$, the output (state-wise average) policy $(\widehat{\mu}^T, \widehat{\nu}^T)$ achieves approximate NE guarantee

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq \mathcal{O}\big(\mathrm{poly}(H, \log(A\vee B), \log T)\cdot T^{-5/6}\big). \tag{15}$$

To our best knowledge, the $\widetilde{\mathcal{O}}(T^{-5/6})$ rate asserted in Theorem 3 is the first rate faster than the standard $\widetilde{\mathcal{O}}(1/\sqrt{T})$ for symmetric, policy optimization type algorithms in two-player zero-sum Markov games. The closest existing result to this is of Wei et al. [53], who analyze the OGDA algorithm with smooth value updates and show a $\widetilde{\mathcal{O}}(1/\sqrt{T})$ convergence of both the average NEGap

and the NEGap of the last-iterate. However, these only imply at most a $\widetilde{\mathcal{O}}(1/\sqrt{T})$ rate for the average policies, and not our faster rate[4]. Cen et al. [8], Zhao et al. [62] show $\widetilde{\mathcal{O}}(1/T)$ convergence of algorithms with optimistic gradient-based policy updates, which are however very different styles of algorithms that either performs *layer-wise learning* (the matrix games at each state are learned to sufficient precision before the backup) similar as Value Iteration (the matrix games at each state are learned to sufficient precision before the backup) [8], or uses strongly *asymmetric* updates that simulate a policy gradient-best response dynamics [62]. By contrast, our Algorithm 9 (as well as its modified version in Algorithm 10 with $\widetilde{\mathcal{O}}(T^{-1})$ rate) runs symmetric no-regret dynamics for both players, simultaneously at all layers.

Proof overview The proof of Theorem 3 (deferred to Appendix E) builds upon the recent line of work on fast convergence of optimistic algorithms [41, 50, 9], in particular the work of Chen and Peng [9] which shows an $\widetilde{\mathcal{O}}(T^{-5/6})$ convergence rate of OFTRL for two-player normal-form games. Our regret bound (14) generalizes this result non-trivially by additionally handling (1) The *weighted* regret, which requires bounding the weighted stability of the OFTRL iterates by a new analysis of the potential functions (Lemma B.4), and (2) The errors induced by *changing game matrices*, as $Q_h^t(s,\cdot,\cdot)$ changes over t. Plugging (14) into Theorem 2 yields the policy guarantee (15).

Modified OFTRL algorithm with $\widetilde{\mathcal{O}}(T^{-1})$ rate We further slightly modify Algorithm 9 to design a new OFTRL style algorithm with $\widetilde{\mathcal{O}}(T^{-1})$ convergence rate (Algorithm 10 and Theorem F.1), which improves over the $\widetilde{\mathcal{O}}(T^{-5/6})$ of Theorem 3 and matches the known best convergence rate for policy optimization type algorithms in two-player zero-sum Markov games. Algorithm 10 still uses OFTRL in its policy update step, and the main difference from Algorithm 9 is in its value update step: Rather than maintaining a single Q_h^t, the two players now each maintain their own value estimate $\overline{Q}_h^t, \underline{Q}_h^t$ which are still updated in an incremental fashion similar to (though not strictly speaking an instantiation of) the update rule (4) in our main algorithm framework. Details of the algorithm as well as the proofs are deferred to Appendix F.

4.1 Extension to multi-player general-sum Markov games

Our fast convergence result can be extended to the more general setting of *multi-player general-sum* Markov games. Concretely, we consider general-sum Markov games with $m \geq 2$ players, S states, H steps, where the i-th player has action space $\mathcal{A}_i$ with $A_{\max} := \max_{i\in[m]} |\mathcal{A}_i|$ and her own reward function. The goal is to find a correlated policy over all players that is an approximate Coarse Correlated Equilibrium (CCE) of the game (see Appendix G.1 for the detailed setup).

We show that the OFTRL algorithm works for general-sum Markov games as well, with a fast $\widetilde{\mathcal{O}}(T^{-3/4})$ convergence to CCE. The formal statement and proof is in Theorem G.1 & Appendix G.4.

Theorem 4 (Fast convergence of OFTRL in general-sum Markov Games; Informal version of Theorem G.1)**.** *For m-player general-sum Markov Games, running the OFTRL algorithm (Algorithm 12) for T rounds, the output (correlated) policy $\widehat{\pi}$ is an ε-approximate CCE, where*

$$\varepsilon \leq \mathcal{O}\big(\mathrm{poly}(H, \log A_{\max}, \log T) \cdot (m-1)^{1/2} \cdot T^{-3/4}\big).$$

A baseline result for this problem would be $\widetilde{\mathcal{O}}(T^{-1/2})$, which may be obtained directly by adapting existing proofs of the V-Learning algorithm [49, 24] to the full-information setting. Our Theorem 4 shows that a faster $\widetilde{\mathcal{O}}(T^{-3/4})$ rate is available by using the OFTRL algorithm, which to our best knowledge is the first such result for policy optimization in general-sum Markov games. We also remark that the output policy $\widehat{\pi}$ above is not a state-wise average policy as in the zero-sum setting, but rather a mixture policy that is in general non-Markov (cf. Algorithm 13), which is similar as (and slightly simpler than) the "certified policies" used in existing work [3, 49, 24]. The proof of Theorem 4 builds upon the RVU property of OFTRL [50] and additionally handles changing game rewards, similar as in Theorem 3. A proof sketch and comparison with the $\widetilde{\mathcal{O}}(T^{-5/6})$ analysis of the zero-sum case can be found in Appendix G.2.

[4] See also [18] for another example where last-iterates are *provably* slower than averages.

5 Simulations

We perform numerical studies on the various policy optimization algorithms. Our goal is two-fold: (1) Verify the convergence guarantees in our theorems and examples; (2) Test some other important special cases of Algorithm 1 that may not yet admit a provable guarantee.

To this end, we consider three algorithms covered by the framework in Algorithm 1:

1. **FTRL** (Nash V-Learning) with smooth value updates $\beta_t = \alpha_t$ (Example 1 & Algorithm 5). Here the output policy $(\widehat{\mu}^T, \widehat{\nu}^T)$ are the state-wise averages with weights $\{\alpha_T^i\}_{i=1}^T$, and achieves $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \lesssim T^{-1/2}$ if we choose $\eta \asymp T^{-1/2}$ (Proposition D.2).
2. **OFTRL** with smooth value updates $\beta_t = \alpha_t$ (Algorithm 9). Here the output policy $(\widehat{\mu}^T, \widehat{\nu}^T)$ are the state-wise averages with weights $\{\alpha_T^i\}_{i=1}^T$, and achieves $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \lesssim T^{-5/6}$ if we choose $\eta \asymp T^{-1/6}$ (Theorem 3). We also consider the more aggressive choice $\eta = 1$.
3. **INPG** (Independent Natural Policy Gradients). This algorithm is an instantiation of Algorithm 1 (cf. Appendix H.3 for formal justifications) with *eager* value updates ($\beta_t = 1$), and MatrixGameAlg chosen as standard unweighted FTRL (a.k.a. Hedge) for all (h, s, t):

$$\mu_h^t(a|s) \propto_a \mu_h^{t-1}(a|s) \exp\big(\eta\big[Q_h^{t-1}\nu_h^{t-1}\big](s)\big), \;\; \nu_h^t(b|s) \propto_b \nu_h^{t-1}(b|s) \exp\Big(-\eta\Big[\big(Q_h^{t-1}\big)^\top \mu_h^{t-1}\Big](s)\Big).$$

   For this algorithm, we choose two standard learning rates: $\eta = 1$, and $\eta = T^{-1/2}$, and use the *vanilla (state-wise) average* as the output policies (since the last-iterate is known to be cyclic):

$$\widehat{\mu}_h^T(\cdot|s) = \tfrac{1}{T}\textstyle\sum_{t=1}^T \mu_h^t(\cdot|s), \;\; \widehat{\nu}_h^T(\cdot|s) = \tfrac{1}{T}\textstyle\sum_{t=1}^T \nu_h^t(\cdot|s) \quad \text{for all } (h,s) \in [H] \times \mathcal{S}.$$

The main motivation for considering INPG is that it is a natural generalization of both the widely-studied NPG algorithm for single-agent RL, and the standard Hedge algorithm for zero-sum matrix games. In both cases the algorithm admits favorable convergence guarantees: NPG converges with rate $\mathcal{O}(T^{-1})$ [1, 27, 38, 7] (in both last iterate and averaging) using $\eta = O(1)$; Hedge converges with rate $\mathcal{O}(T^{-1/2})$ in zero-sum matrix games (e.g. [41]) using $\eta \asymp T^{-1/2}$. However, to our best knowledge, the convergence of INPG for zero-sum Markov games is unclear, and it is commented by Wei et al. [53, Section 5] that eager value updates ($\beta_t = 1$) could cause the value function of the $(h+1)$th layer to oscillate, which make learning unstable or even biased within the h-th layer.

A two-layer numerical example We design a simple zero-sum Markov game with two layers and small state/action spaces ($H = 2$, $S = 4$, $A = 2$; see Appendix H.1 for the detailed description). The main feature of this game is that the reward in the first layer is much lower magnitude than that of the second layer (the scale is roughly $|r_1(s,\cdot,\cdot)| \approx 0.1|r_2(s,\cdot,\cdot)|$), which may exaggerate the aforementioned unstable effect. We also choose a careful initialization (μ^1, ν^1) which is non-uniform (and modify the FTRL / OFTRL algorithms to start at this initialization, cf. Appendix H.1) but with all entries bounded in $[0.15, 0.85]$. We test all three algorithms above on this game, with this initialization, $T \in \{10^3, 3\times 10^3, 10^4, \ldots, 10^7\}$, and η chosen correspondingly as described above.

Results Figure 1a plots the NEGap of the final output policies, one for each {algorithm, (T, η)}. Observe that FTRL converges with rate roughly $T^{-.570} \lesssim T^{-1/2}$, and OFTRL with $\eta = T^{-1/6}$ converges with rate $T^{-.835} \approx T^{-5/6}$, both corroborating our theory. Further, OFTRL with $\eta = 1$ appears to converge with rate T^{-1}; showing this may be an interesting open theoretical question.

On the other hand, the INPG algorithm appears to be much slower: The $\eta = 1$ version does not seem to converge, whereas the convergence of $\eta = T^{-1/2}$ version is not clear but at least substantially slower than $T^{-1/2}$ ($T^{-.308}$ given by the linear fit) .

To further understand the behavior of INPG, we visualize its *layer-wise* NEGap's for $h \in \{1, 2\}$ (on our example), defined as the NEGap of the h-th layer's policies with respect to $Q_h^\star$:

$$\text{NEGap-Layer-}h(\mu,\nu) := \max_s \Big(\max_{\mu_h^\dagger}\Big[(\mu_h^\dagger)^\top Q_h^\star \nu_h\Big](s) - \min_{\nu_h^\dagger}\Big[\mu_h^\top Q_h^\star \nu_h^\dagger\Big](s)\Big), \; h = 1, 2.$$

Note that NEGap-Layer-1 is a lower bound of $\mathrm{NEGap}(\mu, \nu)$ (cf. Appendix H.3) and thus needs to be minimized by any convergent algorithm. By contrast, on our example, NEGap-Layer-2 is concerned with the last layer only, and can be minimized by any algorithm that works on matrix games.

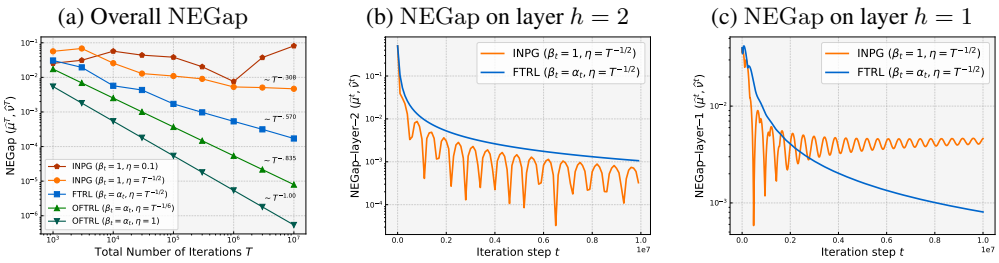

Figure 1: **(a)** NEGap of the final output policies (y-axis) against total # iterations T (x-axis) on the two-layer example (cf. Appendix H.1) in log-log scale. **Each dot represents a different run with its own** (T, η). The scalings of the form $\sim T^{-\alpha}$ are obtained via best linear fits in the log space. **(b,c)** Layer-wise NEGaps (y-axis, log-scale) against iteration count t (x-axis) for {INPG, FTRL} on a **single run** with $T = 10^7$ and $\eta = T^{-1/2}$.

Figure 1b & 1c plot the layer-wise NEGap's of INPG against FTRL, on the single run with $T = 10^7$ and $\eta = T^{-1/2}$. As expected, the NEGap-Layer-2 converges nicely for both algorithms with similar rates (Figure 1b) albeit the oscillation of INPG, whereas their behavior on NEGap-Layer-1 is drastically different: FTRL still converges, whereas INPG seems to be oscillating around a non-zero bias (Figure 1c). This suggests that INPG may indeed be suffer from a non-vanishing bias in the first layer caused by the second layer's learning dynamics. (See Appendix H.2 for additional illustrations.) It would be an interesting open question to investigate the convergence of INPG theoretically.

6 Conclusion

This paper provides a unified framework for analyzing a large class of policy optimization algorithms for two-player zero-sum Markov games. Using our framework, we prove new fast convergence rates for the OFTRL algorithm with smooth value updates: $\widetilde{\mathcal{O}}(T^{-5/6})$ for learning Nash Equilibria two-player zero-sum Markov games, which can be further accelerated to (T^{-1}) by slightly modifying the framework; and $\widetilde{\mathcal{O}}(T^{-3/4})$ for learning Coarse Correlated Equilibria in multi-player general-sum Markov games. We further demonstrate the importance of smooth value updates on a simple numerical example. We believe our work opens up many other interesting directions, such as whether improved rates (e.g. $\widetilde{\mathcal{O}}(T^{-1})$) are available for the unmodified OFTRL algorithm, or further investigation of policy optimization algorithms with eager value updates (such as Independent Natural Policy Gradients). Finally, a limitation of this work is its focus on the full-information setting, and it is an important open question how to generalize our analyses to the sample-based setting.

Acknowledgment

The authors would like to thank Chi Jin, Yuanhao Wang, Tiancheng Yu, Shicong Cen, and Song Mei for the valuable discussions. Runyu Zhang is supported by NSF AI institute: 2112085, ONR YIP: N00014-19-1-2217, NSF CNS: 2003111 and NSF CPS: 2038603.

References

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

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

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

[4] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

[5] J. Bhandari and D. Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

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

[7] S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 2021.

[8] S. Cen, Y. Wei, and Y. Chi. Fast policy extragradient methods for competitive games with entropy regularization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27952–27964. Curran Associates, Inc., 2021.

[9] X. Chen and B. Peng. Hedging in games: Faster convergence of external and swap regrets. *Advances in Neural Information Processing Systems*, 33:18990–18999, 2020.

[10] Z. Chen, D. Zhou, and Q. Gu. Almost optimal algorithms for two-player zero-sum linear mixture markov games. In *International Conference on Algorithmic Learning Theory*, pages 227–261. PMLR, 2022.

[11] C. Daskalakis, D. J. Foster, and N. Golowich. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33: 5527–5540, 2020.

[12] C. Daskalakis, M. Fishelson, and N. Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[14] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[15] D. Ding, K. Zhang, T. Basar, and M. Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33:8378–8390, 2020.

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

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

[18] N. Golowich, S. Pattathil, C. Daskalakis, and A. Ozdaglar. Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems. In *Conference on Learning Theory*, pages 1758–1784. PMLR, 2020.

[19] T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

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

[21] B. Huang, J. D. Lee, Z. Wang, and Z. Yang. Towards general function approximation in zero-sum markov games. *arXiv preprint arXiv:2107.14702*, 2021.

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

[23] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

[24] C. Jin, Q. Liu, Y. Wang, and T. Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021.

[25] C. Jin, Q. Liu, and T. Yu. The power of exploiter: Provable multi-agent rl in large state spaces. *arXiv preprint arXiv:2106.03352*, 2021.

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

[27] S. Khodadadian, P. R. Jhunjhunwala, S. M. Varma, and S. T. Maguluri. On the linear convergence of natural policy gradient algorithm. *arXiv preprint arXiv:2105.01424*, 2021.

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

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

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

[31] M. L. Littman et al. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pages 322–328, 2001.

[32] B. Liu, Q. Cai, Z. Yang, and Z. Wang. Neural trust region/proximal policy optimization attains globally optimal policy. *Advances in neural information processing systems*, 32, 2019.

[33] Q. Liu, T. Yu, Y. Bai, and C. Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021.

[34] Q. Liu, Y. Wang, and C. Jin. Learning markov games with adversarial opponents: Efficient algorithms and fundamental limits. *arXiv preprint arXiv:2203.06803*, 2022.

[35] Y. Liu, K. Zhang, T. Basar, and W. Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

[36] W. Mao and T. Başar. Provably efficient reinforcement learning in decentralized general-sum markov games. *arXiv preprint arXiv:2110.05682*, 2021.

[37] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[38] J. Mei, B. Dai, C. Xiao, C. Szepesvari, and D. Schuurmans. Understanding the effect of stochasticity in policy optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

[39] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[40] J. Peters and S. Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE, 2006.

[41] S. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 26, 2013.

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

[43] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[44] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[45] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[46] L. Shani, Y. Efroni, and S. Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020.

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

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

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

[50] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 28, 2015.

[51] Y. Tian, Y. Wang, T. Yu, and S. Sra. Online learning in unknown markov games. In *International Conference on Machine Learning*, pages 10279–10288. PMLR, 2021.

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

[53] C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive markov games. In *Conference on Learning Theory*, pages 4259–4299. PMLR, 2021.

[54] L. Xiao. On the convergence rates of policy gradient methods. *arXiv preprint arXiv:2201.07443*, 2022.

[55] Q. Xie, Y. Chen, Z. Wang, and Z. Yang. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. *arXiv preprint arXiv:2002.07066*, 2020.

[56] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

[57] T. Yu, Y. Tian, J. Zhang, and S. Sra. Provably efficient algorithms for multi-objective competitive rl. *arXiv preprint arXiv:2102.03192*, 2021.

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

[59] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

[60] R. Zhang, Z. Ren, and N. Li. Gradient play in multi-agent markov stochastic games: Stationary points and convergence. *CoRR*, abs/2106.00198, 2021.

[61] R. Zhang, J. Mei, B. Dai, D. Schuurmans, and N. Li. On the effect of log-barrier regularization in decentralized softmax gradient play in multiagent systems, 2022.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] As this work is primarily theoretical and about understandings, we do not foresee any negative societal impacts of this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Our simulations are deterministic.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Technical tools

A.1 Properties of α_t^i

Throughout this section, the sequence $\{\beta_t^i\}_{i\in[t]}$ is defined through sequence $\{\beta_t\}_{t\geq 1}$ as in (6), and α_t^i is its special case with $\beta_t = \alpha_t$, where $\{\alpha_t\}_{t\geq 1}$ is defined in (3). We present some basic algebraic properties of α_t^i that will be used in later proofs.

Lemma A.1. *Given a sequence $\{\Delta_h^t\}_{h,t}$ defined by*

$$\begin{cases} \Delta_h^t = \sum_{i=1}^t \alpha_t^i \Delta_{h+1}^i + \beta_t, \\ \Delta_{H+1}^t = 0, \ \text{for all } t, \end{cases} \tag{16}$$

where $\{\beta_t\}$ is non-increasing w.r.t. t. Then $\Delta_h^{t+1} \leq \Delta_h^t$ for all $(t,h) \in \mathbb{N} \times [H+1]$.

Proof. We prove by doing backward induction on h. For the base case of induction, notice that the claim is true for $H+1$. Assume the claim is true for $h+1$. At step h, we have

$$\begin{aligned} \Delta_h^{t+1} &= \sum_{i=1}^{t+1} \alpha_{t+1}^i \Delta_{h+1}^i + \beta_{t+1} \\ &= (1-\alpha_{t+1}) \sum_{i=1}^{t} \alpha_t^i \Delta_{h+1}^i + \alpha_{t+1}\Delta_{h+1}^{t+1} + \beta_{t+1} \\ &\leq (1-\alpha_{t+1}) \sum_{i=1}^{t} \alpha_t^i \Delta_{h+1}^i + \alpha_{t+1} \sum_{i=1}^{t} \alpha_t^i \Delta_{h+1}^i + \beta_t = \Delta_h^t, \end{aligned}$$

where the inequality follows from the inductive hypothesis and $\beta_{t+1} \leq \beta_t$. □

The following lemma is taken from [23].

Lemma A.2. *The sequence α_t^i satisfies the following:*

(a) $\sum_{t=i}^{\infty} \alpha_t^i = 1 + 1/H$ *for all* $i \geq 1$.

Lemma A.3 (Convolution of β_T^t with decaying sequences)**.** *From general $\{\beta_t\}_{t\geq 1}$ sequence, we have that*

$$X_T := \sum_{t=1}^{T} \frac{1}{t} \beta_T^t \leq \frac{2c_\beta \log(T)}{T}, \quad \text{for } T \geq 2.$$

Specifically if $\beta_t = \alpha_t$, the following holds for all $T \geq 1$:

(a) $A_T := \sum_{t=1}^T \alpha_T^t \cdot \frac{1}{t^2} \leq \frac{4}{T}$.

(b) $B_T := \sum_{t=1}^T \alpha_T^t \alpha_t \leq \frac{(H+1)^2}{H(H+T)}$

(c) $C_T := \sum_{t=1}^T \alpha_T^t \cdot \alpha_t^2 \leq \frac{4H}{T}$.

Proof. We first prove for the inequality on X_T for general $\{\beta_t\}_{t\geq 1}$ sequence. We start with showing that $X_{t+1} \leq X_t, \ \forall t \geq 1$.

$$\begin{aligned} X_{t+1} &= \sum_{i=1}^{t+1} \frac{1}{i} \beta_{t+1}^i = \sum_{i=1}^{t} \frac{1}{i} \beta_{t+1}^i + \frac{\beta_{t+1}}{t+1} \\ &= (1-\beta_{t+1}) \sum_{i=1}^{t} \frac{1}{i} \beta_t^i + \frac{\beta_{t+1}}{t+1} = (1-\beta_{t+1}) X_t + \frac{\beta_{t+1}}{t+1} \\ \implies \ X_{t+1} - X_t &= \beta_{t+1} \left(\frac{1}{t+1} - X_t \right). \end{aligned}$$

Since

$$\frac{1}{t+1} - X_t = \frac{1}{t+1} - \sum_{i=1}^{t} \frac{1}{i}\beta_t^i = \sum_{i=1}^{t} \beta_t^i \left(\frac{1}{t+1} - \frac{1}{i}\right) \leq 0,$$

we have that $X_{t+1} - X_t \leq 0$.Thus

$$X_T \leq \frac{1}{T}\sum_{t=1}^{T} X_t = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{t} \frac{1}{i}\beta_t^i = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{i}\left(\sum_{t=i}^{T}\beta_t^i\right) \leq \frac{c_\beta}{T}\sum_{i=1}^{T}\frac{1}{i} \leq \frac{2c_\beta \log(T)}{T}.$$

Now we prove for the specific case where $\beta_t = \alpha_t$:

(a) Note that $A_1 = 1$ and we have the recursive relationship

$$A_{T+1} = (1-\alpha_{T+1})A_T + \alpha_{T+1} \cdot \frac{1}{(T+1)^2}$$

by definition of the sequence α_T^t. In particular this implies $A_{T+1} \leq A_T$, since A_T is a weighted average of $1/t^2 \geq 1/(T+1)^2$. Therefore we have

$$\begin{aligned} A_T &\leq \frac{1}{T}\sum_{t=1}^{T} A_t = \frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{t} \alpha_t^s \cdot \frac{1}{s^2} \\ &= \frac{1}{T}\sum_{s=1}^{T} \underbrace{\sum_{t=s}^{T}\alpha_t^s}_{\leq 1+1/H \leq 2} \cdot \frac{1}{s^2} \leq \frac{2}{T}\sum_{s=1}^{T}\frac{1}{s^2} \leq \frac{2}{T}\sum_{s=1}^{\infty}\frac{1}{s^2} \leq \frac{4}{T}. \end{aligned}$$

Above, the step $\sum_{t=s}^{T}\alpha_t^s \leq \sum_{t=s}^{\infty}\alpha_t^s = 1 + 1/H$ follows from Lemma A.2.

(b) From the definition of B_T we have that

$$\begin{aligned} B_{T+1} &= \sum_{t=1}^{T+1}\alpha_{T+1}^t\alpha_t = \sum_{t=1}^{T}(1-\alpha_{T+1})\alpha_T^t\alpha_t + \alpha_{T+1}^2 = (1-\alpha_{T+1}^t)B_T + \alpha_{T+1} \\ \implies B_{T+1} &= \frac{T}{H+T+1}B_T + \frac{(H+1)^2}{(H+T+1)^2} \leq \frac{T}{H+T+1}B_T + \frac{(H+1)^2}{(H+T+1)(H+T)} \\ \implies \left(B_{T+1} - \frac{(H+1)^2}{H(H+T+1)}\right) &\leq \frac{T}{H+T+1}\left(B_T - \frac{(H+1)^2}{H(H+T)}\right). \end{aligned}$$

Since $B_1 = \alpha_1^2 = 1 \leq \frac{(H+1)^2}{H(H+1)}$, we have that $B_T \leq \frac{(H+1)^2}{H(H+T)}$ via proof by induction.

(c) Since $\alpha_t \leq 1$, we have that $C_T \leq B_T$, thus by part (b)

$$C_T \leq B_T \leq \frac{(H+1)^2}{H(H+T)} \leq \frac{4H}{T}.$$

□

Consider the sequence $\{w_t\}_{t\geq 1}$ defined by (cf. also Example 1)

$$w_t = \alpha_t^t/\alpha_t^1. \tag{17}$$

Note that we also have $w_t = \alpha_T^t/\alpha_T^1$ for any $T \geq t$.

Lemma A.4 (Properties of w_t)**.** *The following holds for all $t \geq 2$:*

(a) $w_t/w_{t-1} = (H+t-1)/(t-1)$.

(b) $(\frac{1}{w_{t-1}} - \frac{1}{w_t})\sum_{i=1}^{t-1} w_i = H/(H+1)$.

Proof. (a) We have

$$\frac{w_t}{w_{t-1}} = \frac{\alpha_t}{\alpha_{t-1}(1-\alpha_t)} = \frac{(H+1)/(H+t)}{(H+1)/(H+t-1)\cdot(t-1)/(H+t)} = \frac{H+t-1}{t-1}.$$

(b) We have

$$\left(\frac{1}{w_{t-1}} - \frac{1}{w_t}\right)\sum_{i=1}^{t-1} w_i = \frac{1}{w_{t-1}}\left(1 - \frac{w_{t-1}}{w_t}\right)\sum_{i=1}^{t-1} w_i = \left(1 - \frac{w_{t-1}}{w_t}\right)\cdot\frac{1}{\alpha_{t-1}^{t-1}}$$
$$\overset{(i)}{=} \frac{H}{H+t-1}\cdot\frac{H+t-1}{H+1} = \frac{H}{H+1}.$$

Above, (i) used part (a).

□

A.2 Other technical lemmas

Lemma A.5 (Smoothness of Exponential Weights)**.** *Let $g_1, g_2 \in \mathbb{R}^n$ and*

$$x_1 = \arg\max_{x\in\Delta_{[n]}} \langle x, g_1\rangle - H(x),$$
$$x_2 = \arg\max_{x\in\Delta_{[n]}} \langle x, g_2\rangle - H(x),$$

where $H(x) := \sum_{i=1}^n x_i \log x_i$ is the standard entropy functional. Then $\|x_1 - x_2\|_1 \le 2\|g_1 - g_2\|_\infty$.

Proof. Since H is 1-strongly convex in $\|\cdot\|_1$ (Pinsker's inequality), we have that

$$\begin{aligned}
\frac{\|x_1 - x_2\|_1^2}{2} &\le H(x_1) - H(x_2) - \langle \nabla H(x_2), x_1 - x_2\rangle \\
&= H(x_1) + (\langle x_2, g_1\rangle - H(x_2)) - \langle x_2, g_1\rangle - \langle \nabla H(x_2), x_1 - x_2\rangle \\
&\le H(x_1) + (\langle x_1, g_1\rangle - H(x_1)) - \langle x_2, g_1\rangle - \langle \nabla H(x_2), x_1 - x_2\rangle \\
&= \langle x_1 - x_2, g_1\rangle - \langle \nabla H(x_2), x_1 - x_2\rangle \\
&= \langle x_1 - x_2, g_1 - g_2\rangle + \langle g_2 - \nabla H(x_2), x_1 - x_2\rangle \\
&\le \langle x_1 - x_2, g_1 - g_2\rangle \quad \text{(From the optimality of } x_2\text{)} \\
&\le \|x_1 - x_2\|_1 \|g_1 - g_2\|_\infty,
\end{aligned}$$

which completes the proof. □

B Bound for regret minimization algorithms

B.1 Projected gradient descent

Algorithm 2 Projected gradient descent

Require: Learning rate $\eta > 0$.
1: **Initialize** $x_1 \in \mathcal{X} \subset \mathbb{R}^d$.
2: **for** $t = 1, \ldots, T$ **do**
3: Receive loss $g_t \in \mathbb{R}^d$.
4: Compute update $y_{t+1} = x_t - \eta g_t$, $x_{t+1} = \mathcal{P}_{\mathcal{X}}(y_{t+1})$.

The following weighted regret bound for projected gradient descent is standard. For completeness we provide a proof here. For simplicity of notation, denote the diameter of $\mathcal{X} \subset \mathbb{R}^d$ by R and $G = \max_{t\in[T]} \|g_t\|_2$.

Lemma B.1 (Weighted regret bound for projected gradient descent)**.** *For any weights $\{w_t\}_{t\ge1} \in \mathbb{R}_{>0}$ with $w_t \le w_{t+1}$ for all $t \ge 1$, Algorithm 2 achieves*

$$\max_{z\in\mathcal{X}} \sum_{t=1}^T w_t\langle x_t - z, g_t\rangle \le \frac{w_T}{2\eta}R^2 + \frac{\eta\sum_{t=1}^T w_t\cdot G^2}{2}.$$

Proof. By following the standard GD analysis, we first have

$$\begin{aligned}\langle x_t - z, g_t\rangle =& \frac{1}{\eta}\langle x_t - z, x_t - y_{t+1}\rangle \\ =& \frac{1}{2\eta}\left[\|x_t - z\|^2 + \|x_t - y_{t+1}\|^2 - \|z - y_{t+1}\|^2\right] \\ \le& \frac{1}{2\eta}\left[\|x_t - z\|^2 + \|x_t - y_{t+1}\|^2 - \|z - x_{t+1}\|^2\right] \\ =& \frac{1}{2\eta}\left[\|x_t - z\|^2 - \|z - x_{t+1}\|^2 + \eta^2\|g_t\|^2\right],\end{aligned}$$

where the inequality follows from $z \in \mathcal{X}$ and $x_{t+1} = \mathcal{P}_{\mathcal{X}}(y_{t+1})$. By multiplying both sides with w_t and taking summation over $t \in [T]$, we have

$$\begin{aligned}\sum_{t=1}^{T} w_t\langle x_t - z, g_t\rangle \le& \frac{1}{2\eta}\sum_{t=1}^{T} w_t\left[\|x_t - z\|^2 - \|z - x_{t+1}\|^2 + \eta^2\|g_t\|^2\right] \\ =& \frac{1}{2\eta}\sum_{t=1}^{T-1}(w_{t+1} - w_t)\|z - x_{t+1}\|^2 + w_1\|x_1 - z\|^2 + \frac{\eta\sum_{t=1}^{T} w_t \cdot G^2}{2} \\ \overset{(i)}{\le}& \frac{1}{2\eta}\sum_{t=1}^{T-1}(w_{t+1} - w_t)R^2 + w_1R^2 + \frac{\eta G^2}{2} = \frac{1}{2\eta}w_T R^2 + \frac{\eta\sum_{t=1}^{T} w_t \cdot G^2}{2},\end{aligned}$$

Above, (i) follows as $w_{t+1} \ge w_t$. This completes the proof. □

B.2 Follow-The-Regularized Leader (FTRL)

In this subsection, we consider the following weighted FTRL algorithm over the probability simplex $\Delta_{[A]}$ with the standard (negative) entropy regularizer $\Phi(x) := \sum_{a\in[A]} x(a)\log x(a)$. Below the notation $x_t(a)$ denotes the a-th entry of x_t.

Algorithm 3 Weighted FTRL with changing learning rate

Require: Learning rate $\eta > 0$; Weights $\{w_t\}_{t\ge1} \subset \mathbb{R}_{>0}$.
1: **Initialize** $x_1 \leftarrow \mathbf{1}_A/A$ to be the uniform distribution over $[A]$.
2: **for** $t = 1, \ldots, T$ **do**
3: Receive loss $g_t \in \mathbb{R}^A$.
4: Compute FTRL update

$$x_{t+1} \leftarrow \arg\min_{x\in\Delta_{[A]}} \left\langle x, \sum_{s=1}^{t} w_s g_s \right\rangle + \frac{w_t}{\eta}\Phi(x). \tag{18}$$

Note that (18) has a closed-form solution via exponential weights:

$$x_{t+1}(a) \propto_a \exp\left(-\frac{\eta}{w_t}\sum_{s=1}^{t} w_s g_s(a)\right). \tag{19}$$

Lemma B.2 (Regret bound of weighted FTRL)**.** *Suppose the weights are non-decreasing:* $w_{t+1} \ge w_t$ *for all* $t \ge 1$*, and* $\max_t \|g_t\|_\infty \le G$*. Then Algorithm 3 achieves weighted regret bound*

$$\max_{z\in\Delta_{[A]}} \sum_{t=1}^{T} w_t\langle x_t - z, g_t\rangle \le \frac{w_T}{\eta}\log A + \frac{\eta G^2}{2}\sum_{t=1}^{T} w_t.$$

Proof. Applying standard anytime FTRL analysis (see, e.g., Excercise 28.12 in [28]) with loss sequence $\{w_t g_t\}_{t\ge1}$ and learning rate $\{\eta/w_t\}_{t\ge1}$, we have that

$$\sum_{t=1}^{T} w_t\langle x_t - z, g_t\rangle \le \frac{w_T(\Phi(x_1) - \min_x \Phi(x))}{\eta} + \sum_{t=1}^{T} w_t\left(\langle x_t - x_{t+1}, g_t\rangle - \frac{\mathrm{KL}(x_{t+1}||x_t)}{\eta}\right)$$

$$\leq \frac{w_T \log A}{\eta} + \sum_{t=1}^{T} \left(w_t \langle x_t - x_{t+1}, g_t \rangle - \frac{w_t \|x_t - x_{t+1}\|_1^2}{2\eta} \right) \quad \text{(Pinsker's inequality)}$$

$$\leq \frac{w_T \log A}{\eta} + \sum_{t=1}^{T} \frac{w_t \eta}{2} \|g_t\|_\infty^2$$

$$\leq \frac{w_T \log A}{\eta} + \frac{\eta G^2}{2} \sum_{t=1}^{T} w_t,$$

which completes the proof. □

B.3 Optimistic Follow-The-Regularized-Leader (OFTRL)

We consider the following OFTRL algorithm on the probability simplex $\Delta_{[A]}$ with standard (negative) entropy regularizer $\Phi(x) = \sum_{a \in [A]} x(a) \log x(a)$.

Algorithm 4 Anytime OFTRL

Require: Learning rate $\{\eta_t\}_{t \geq 1}$.

1: **Initialize** $x_1 \leftarrow \mathbf{1}_A / A$ to be the uniform distribution over $[A]$.
2: **for** $t = 1, \ldots, T$ **do**
3: Receive loss $g_t \in \mathbb{R}^A$.
4: Compute a prediction vector $M_{t+1} \in \mathbb{R}^A$ using past observations.
5: Compute OFTRL update

$$x_{t+1} \leftarrow \arg\min_{x \in \Delta_{[A]}} \eta_{t+1} \left\langle x, \sum_{s=1}^{t} g_s + M_{t+1} \right\rangle + \Phi(x). \tag{20}$$

Note that (20) has a closed-form solution via exponential weights:

$$x_{t+1}(a) \propto_a \exp\left(-\eta_{t+1} \left[\sum_{s=1}^{t} g_s(a) + M_{t+1}(a) \right] \right). \tag{21}$$

The following regret bound for OFTRL follows similarly as standard OFTRL analysis, see, e.g. [41, Lemma 1]. For completeness, we provide a proof here.

Lemma B.3 (Regret bound for OFTRL)**.** *Suppose the learning rates are non-increasing: $\eta_t \geq \eta_{t+1}$ for all $t \geq 1$. Then Algorithm 4 achieves the following bound for all $x \in \mathcal{X}$:*

$$\sum_{t=1}^{T} \langle x_t - x, g_t \rangle \leq \frac{\log A}{\eta_T} + \sum_{t=1}^{T} \eta_t \|g_t - M_t\|_\infty^2 - \sum_{t=1}^{T-1} \frac{1}{8\eta_t} \|x_t - x_{t+1}\|_1^2.$$

Proof. Consider a fixed $T \geq 1$. Note that Algorithm 4 is equivalent to Algorithm (25) with regularizer $R_t(\cdot) := (\Phi(\cdot) + \log A)/\eta_{t+1} \geq 0$ for $t \geq 0$, and $R_T(\cdot) := R_{T-1}(\cdot)$ (Note that the shifting by $\log A$ does not affect the algorithm.)

We first decompose the regret into the following three terms

$$\sum_{t=1}^{T} \langle x_t - x, g_t \rangle = \sum_{t=1}^{T} \langle q_{t+1} - x, g_t \rangle + \sum_{t=1}^{T} \langle x_t - q_{t+1}, M_t \rangle + \sum_{t=1}^{T} \langle x_t - q_{t+1}, g_t - M_t \rangle,$$

where $\{q_t\}_{t \geq 1}$ is defined in (26). By Lemma B.5, we can upper bound the first two terms by $R_T(x) - \min_{x'} R_0(x') + S_T \leq R_T(x) + S_T$ and obtain

$$\sum_{t=1}^{T} \langle x_t - x, g_t \rangle$$

$$\leq R_T(x) + \sum_{t=1}^{T} \left(R_{t-1}(q_{t+1}) - R_t(q_{t+1})\right) + \sum_{t=1}^{T} \left(\langle x_t - q_{t+1}, g_t - M_t\rangle - D_{R_{t-1}}(q_{t+1}, x_t) - D_{R_{t-1}}(x_t, q_t)\right)$$

$$\leq R_T(x) + \sum_{t=1}^{T} \left(\|x_t - q_{t+1}\|_1 \|g_t - M_t\|_\infty - \frac{1}{2\eta_t}\|q_{t+1} - x_t\|_1^2 - \frac{1}{2\eta_t}\|x_t - q_t\|_1^2\right),$$

where the second inequality uses $R_{t-1} \leq R_t$ and R_{t-1} is $1/\eta_t$ strongly-convex w.r.t. $\|\cdot\|_1$. Finally, we conclude the proof by applying Cauchy-Schwarz inequality:

$$\|x_t - q_{t+1}\|_1 \|g_t - M_t\|_\infty - \frac{1}{4\eta_t}\|q_{t+1} - x_t\|_1^2 \leq \eta_t \|g_t - M_t\|_\infty^2,$$

and triangle inequality

$$-\frac{1}{4\eta_t}\|q_{t+1} - x_t\|_1^2 - \frac{1}{4\eta_t}\|q_{t+1} - x_{t+1}\|_1^2 \leq -\frac{1}{8\eta_t}\|x_{t+1} - x_t\|_1^2,$$

and the bound $R_T(x) \leq \log A/\eta_{T+1} = \log A/\eta_T$ for any $x \in \Delta_{[A]}$. □

The following lemma bounds the total variation of the iterates in terms of the smoothness of loss vectors and prediction vectors. This can be seen as a generalization of [9, Lemma 3.2] to the case with changing learning rate and arbitrary prediction vectors.

Lemma B.4 (Bounding stability by the smoothness of loss)**.** *Suppose the learning rates are non-increasing: $\eta_t \geq \eta_{t+1}$ for all $t \geq 1$. Then the OFTRL algorithm (20) satisfies (understanding $M_1 := 0$)*

$$\sum_{t=2}^{T} \frac{1}{2\eta_t} \|x_t - x_{t-1}\|_1^2 \leq \frac{\log A}{\eta_T} + \max_{x \in \Delta_{[d]}} \sum_{t=1}^{T-1} \langle x_t - x, g_t\rangle + \sum_{t=2}^{T} \|M_t - M_{t-1}\|_\infty + \|M_T\|_\infty \tag{22}$$

$$\leq \frac{2\log A}{\eta_T} + \sum_{t=1}^{T-1} \eta_t \|g_t - M_t\|_\infty^2 + \sum_{t=2}^{T} \|M_t - M_{t-1}\|_\infty + \|M_T\|_\infty . \tag{23}$$

In particular, choosing the prediction vector $M_t = g_{t-1}$ with $g_0 := 0$, and assume $\|g_t - g_{t-1}\|_\infty \leq G_t$ for all $t \geq 1$, we have

$$\sum_{t=2}^{T} \frac{1}{2\eta_t} \|x_t - x_{t-1}\|_1^2 \leq \frac{2\log A}{\eta_T} + \sum_{t=1}^{T-1} \eta_t \|g_t - g_{t-1}\|_\infty^2 + \sum_{t=2}^{T} \|g_{t-1} - g_{t-2}\|_\infty + \|g_{T-1}\|_\infty$$

$$\leq \frac{2\log A}{\eta_T} + \sum_{t=1}^{T-1} (1 + \eta_t G_t) \|g_t - g_{t-1}\|_\infty + \|g_{T-1}\|_\infty . \tag{24}$$

Proof. We first prove (22). For any $t \geq 2$, the optimality condition of (20) for x_t gives

$$\left\langle \sum_{s=1}^{t-1} g_s + M_t + \frac{\nabla\Phi(x_t)}{\eta_t}, x' - x_t \right\rangle \geq 0$$

for all $x' \in \Delta_{[A]}$. In particular, this holds for $x' = x_{t-1}$, from which we get

$$\frac{1}{2\eta_t} \|x_{t-1} - x_t\|_1^2 \overset{(i)}{\leq} \frac{1}{\eta_t}\mathrm{KL}(x_{t-1}\|x_t) \overset{(ii)}{=} \frac{\Phi(x_{t-1}) - \Phi(x_t)}{\eta_t} - \left\langle \frac{\nabla\Phi(x_t)}{\eta_t}, x_{t-1} - x_t \right\rangle$$

$$\leq \frac{\Phi(x_{t-1}) - \Phi(x_t)}{\eta_t} + \left\langle \sum_{s=1}^{t-1} g_s + M_t, x_{t-1} - x_t \right\rangle,$$

where (i) is by Pinsker's inequality, and (ii) is since the KL divergence is the Bregman divergence of Φ. Summing the above over $t = 2, \ldots, T$ yields

$$\sum_{t=2}^{T} \frac{1}{2\eta_t} \|x_{t-1} - x_t\|_1^2$$

$$\leq \underbrace{-\frac{\Phi(x_T)}{\eta_T}}_{\leq \log A/\eta_T} + \sum_{t=2}^{T} \underbrace{\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)}_{\geq 0} \underbrace{\Phi(x_{t-1})}_{\leq 0} - \left\langle \sum_{s=1}^{T-1} g_s + M_T, x_T \right\rangle + \sum_{t=2}^{T} \langle g_{t-1} + M_t - M_{t-1}, x_{t-1} \rangle$$

$$\leq \frac{\log A}{\eta_T} + \sum_{t=1}^{T-1} \langle x_t, g_t \rangle - \sum_{t=1}^{T-1} \langle x_T, g_t \rangle + \langle M_T, x_T \rangle + \sum_{t=2}^{T} \langle M_t - M_{t-1}, x_{t-1} \rangle$$

$$\leq \frac{\log A}{\eta_T} + \max_{x \in \Delta_{[d]}} \sum_{t=1}^{T-1} \langle x_t - x, g_t \rangle + \sum_{t=2}^{T} \|M_t - M_{t-1}\|_\infty + \|M_T\|_\infty .$$

This proves (22). Then, (23) follows by plugging in the regret bound given by Lemma B.3:

$$\max_{x \in \Delta_{[d]}} \sum_{t=1}^{T-1} \langle x_t - x, g_t \rangle \leq \frac{\log A}{\eta_{T-1}} + \sum_{t=1}^{T-1} \eta_t \|g_t - M_t\|_\infty^2 \leq \frac{\log A}{\eta_T} + \sum_{t=1}^{T-1} \eta_t \|g_t - M_t\|_\infty^2 .$$

Finally, (24) is a direct consequence of (23) by plugging in $M_t = g_{t-1}$ and $\|g_t - g_{t-1}\|_\infty \leq G_t$ for all $t \geq 1$. □

B.3.1 Auxiliary lemma for OFTRL with general regularizers

Consider an OFTRL algorithm with loss function $\{g_t\}_{t \geq 0} \subset \mathbb{R}^d$, parameter space $\mathcal{X} \subset \mathbb{R}^d$, and convex regularizers $R_t : \mathcal{X} \to \mathbb{R}$ for $t \geq 0$:

$$x_{t+1} \leftarrow \arg\min_{x \in \mathcal{X}} \left\langle x, \sum_{s=1}^{t} g_s + M_{t+1} \right\rangle + R_t(x). \tag{25}$$

Define auxiliary sequence

$$q_{t+1} = \arg\min_{x \in \mathcal{X}} \left\langle x, \sum_{s=1}^{t} g_s \right\rangle + R_t(x). \tag{26}$$

Recall the Bregman divergence associated with any convex regularizer $R : \mathcal{X} \to \mathbb{R}$ is given by

$$D_R(x, y) := R(x) - R(y) - \langle \nabla R(y), x - y \rangle \geq 0.$$

Lemma B.5 (Auxiliary lemma for OFTRL with general regularizers)**.** *Algorithm* (25) *achieves the following for any $T \geq 1$ and $x \in \mathcal{X}$:*

$$\sum_{t=1}^{T} \langle q_{t+1}, g_t \rangle + \sum_{t=1}^{T} \langle x_t - q_{t+1}, M_t \rangle \leq \sum_{t=1}^{T} \langle x, g_t \rangle + R_T(x) - \min_{x' \in \mathcal{X}} R_0(x') + S_T,$$

where

$$S_T := \sum_{t=1}^{T} \left(R_{t-1}(q_{t+1}) - R_t(q_{t+1}) \right) - \sum_{t=1}^{T} \left(D_{R_{t-1}}(q_{t+1}, x_t) + D_{R_{t-1}}(x_t, q_t) \right).$$

Proof. We prove the lemma by induction. The above relation holds trivially for $T = 0$. Assume the relation holds for $\tau = T - 1$. For $\tau = T$, we have

$$\sum_{t=1}^{T} \langle q_{t+1}, g_t \rangle + \sum_{t=1}^{T} \langle x_t - q_{t+1}, M_t \rangle$$

$$\leq \min_{x \in \mathcal{X}} \left[\sum_{t=1}^{T-1} \langle x, g_t \rangle + R_{T-1}(x) \right] - \min_{x' \in \mathcal{X}} R_0(x') + S_{T-1} + \langle q_{T+1}, g_T \rangle + \langle x_T - q_{T+1}, M_T \rangle$$

$$= \sum_{t=1}^{T-1} \langle q_T, g_t \rangle + R_{T-1}(q_T) - \min_{x' \in \mathcal{X}} R_0(x')$$

$$
\begin{aligned}
&\quad + S_{T-1} + \langle q_{T+1}, g_T\rangle + \langle x_T - q_{T+1}, M_T\rangle \quad \text{(definition of } q_T\text{)} \\
&\le \sum_{t=1}^{T-1}\langle x_T, g_t\rangle + R_{T-1}(x_T) - D_{R_{T-1}}(x_T, q_T) - \min_{x'\in\mathcal{X}} R_0(x') \\
&\quad + S_{T-1} + \langle q_{T+1}, g_T\rangle + \langle x_T - q_{T+1}, M_T\rangle \quad \text{(optimality of } q_T\text{)} \\
&= \min_{x\in\mathcal{X}}\left[\langle x, \sum_{t=1}^{T-1} g_t + M_T\rangle + R_{T-1}(x)\right] - D_{R_{T-1}}(x_T, q_T) - \min_{x'\in\mathcal{X}} R_0(x') \\
&\quad + S_{T-1} + \langle q_{T+1}, g_T - M_T\rangle \quad \text{(definition of } x_T\text{)} \\
&\le \langle q_{T+1}, \sum_{t=1}^{T-1} g_t + M_T\rangle + R_{T-1}(q_{T+1}) - D_{R_{T-1}}(q_{T+1}, x_T) \\
&\quad - D_{R_{T-1}}(x_T, q_T) - \min_{x'\in\mathcal{X}} R_0(x') + S_{T-1} + \langle q_{T+1}, g_T - M_T\rangle \quad \text{(optimality of } x_T\text{)} \\
&= \min_{x\in\mathcal{X}}\langle x, \sum_{t=1}^{T} g_t\rangle + R_T(x) + (R_{T-1}(q_{T+1}) - R_T(q_{T+1})) \\
&\quad - D_{R_{T-1}}(q_{T+1}, x_T) - D_{R_{T-1}}(x_T, q_T) - \min_{x'\in\mathcal{X}} R_0(x') + S_{T-1} \quad \text{(definition of } q_{T+1}\text{)},
\end{aligned}
$$

which completes the induction. □

C Proofs for Section 3.1

In this section we prove Theorem 2. The proof relies on the following two lemmas.

Lemma C.1 (Performance difference for Markov policies)**.** *In two-player zero-sum Markov games, suppose a Markov policy (μ, ν) satisfies the following for all $h \in [H+1]$:*

$$
\begin{aligned}
&\max_s \max_{\mu^\dagger\in\Delta_{\mathcal{A}}} \left(\left[(\mu^\dagger)^\top Q_h^\star \nu_h\right](s) - V_h^\star(s)\right) \le \varepsilon_h, \\
&\max_s \max_{\nu^\dagger\in\Delta_{\mathcal{B}}} \left(V_h^\star(s) - \left[\mu_h^\top Q_h^\star \nu^\dagger\right](s)\right) \le \varepsilon_h.
\end{aligned}
$$

Then we have for all $h \in [H]$ that

$$
\max\left\{\|V_h^{\dagger,\nu} - V_h^\star\|_\infty, \|V_h^{\mu,\dagger} - V_h^\star\|_\infty\right\} \le \sum_{h'=h}^{H} \varepsilon_{h'}.
$$

Proof. We prove by backward induction over h. The claim is trivial for $h = H+1$. Suppose the claim holds for step $h+1$. At step h,

$$
\begin{aligned}
\|V_h^{\dagger,\nu} - V_h^\star\|_\infty &= \max_s \left|\max_{\mu^\dagger\in\Delta_{\mathcal{A}}} \left[(\mu^\dagger)^\top Q_h^{\dagger,\nu}\nu_h\right](s) - V_h^\star(s)\right| \\
&\le \max_s \left|\max_{\mu^\dagger\in\Delta_{\mathcal{A}}} \left[(\mu^\dagger)^\top Q_h^\star\nu_h\right](s) - V_h^\star(s)\right| + \|Q_h^{\dagger,\nu} - Q_h^\star\|_\infty \\
&\le \varepsilon_h + \|Q_h^{\dagger,\nu} - Q_h^\star\|_\infty.
\end{aligned}
$$

Notice that

$$
\begin{aligned}
&\|Q_h^{\dagger,\nu} - Q_h^\star\|_\infty \le \max_{s,a,b}\left|\left(r_h + \mathbb{P}_h V_{h+1}^{\dagger,\nu}\right)(s,a,b) - \left(r_h + \mathbb{P}_h V_{h+1}^\star\right)(s,a,b)\right| \\
&\le \max_{s,a,b}\left|\mathbb{P}_h\left[V_{h+1}^{\dagger,\nu} - V_{h+1}^\star\right](s,a,b)\right| \le \|V_{h+1}^{\dagger,\nu} - V_{h+1}^\star\|_\infty \le \sum_{h'=h+1}^{H}\varepsilon_{h'}. \quad \text{(by inductive hypothesis)}
\end{aligned}
$$

Combining the two inequalities we get

$$
\|V_h^{\dagger,\nu} - V_h^\star\|_\infty \le \sum_{h'=h}^{H}\varepsilon_{h'}.
$$

This proves the claim for $\|V_h^{\dagger,\nu} - V_h^\star\|_\infty$. The same argument also holds for $\|V_h^{\mu,\dagger} - V_h^\star\|_\infty$, which completes the proof. $\square$

Throughout the rest of this section, we define the following shorthand for the value estimation error:

$$\delta_h^t := \left\|Q_h^t - Q_h^\star\right\|_\infty = \max_{s,a,b} \left|Q_h^t(s,a,b) - Q_h^\star(s,a,b)\right|,$$

where Q_h^t is the estimated value in Algorithm 1.

Lemma C.2 (Recursion of value estimation)**.** *Algorithm 1 guarantees that for all* $(t,h) \in [T] \times [H]$,

$$\delta_h^t \le \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t.$$

Further, suppose that $\mathrm{reg}_h^t \le \overline{\mathrm{reg}}_h^t$ *for all* $(h,t) \in [H] \times [T]$*, where* $\overline{\mathrm{reg}}_h^t$ *is non-increasing in* t*:* $\overline{\mathrm{reg}}_h^t \ge \overline{\mathrm{reg}}_h^{t+1}$ *for all* $t \ge 1$*. Then we have*

$$\delta_h^t \le H c_\beta^{H-1} \cdot \frac{1}{t} \sum_{i=1}^t \max_{h'} \overline{\mathrm{reg}}_{h'}^i,$$

where c_β *is defined in* (9)*.*

Proof. Fix $(h,s,a,b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$. From the definition of $Q_h^\star$ we have that

$$\begin{aligned}
Q_h^\star(s,a,b) &= r_h(s,a,b) + \max_{\mu_{h+1}} \min_{\nu_{h+1}} \mathbb{P}_h\left[\mu_{h+1}^\top Q_{h+1}^\star \nu_{h+1}\right](s,a,b) \\
&\le r_h(s,a,b) + \max_{\mu_{h+1}} \mathbb{P}_h\left[\mu_{h+1}^\top Q_{h+1}^\star \left(\sum_{i=1}^t \beta_t^i \nu_{h+1}^i\right)\right](s,a,b) \\
&= r_h(s,a,b) + \max_{\mu_{h+1}} \sum_{i=1}^t \beta_t^i \mathbb{P}_h\left[\mu_{h+1}^\top Q_{h+1}^\star \nu_{h+1}^i\right](s,a,b) \\
&\le r_h(s,a,b) + \max_{\mu_{h+1}} \sum_{i=1}^t \beta_t^i \left(\mathbb{P}_h\left[\mu_{h+1}^\top Q_{h+1}^i \nu_{h+1}^i\right](s,a,b) + \|Q_{h+1}^i - Q_{h+1}^\star\|_\infty\right) \\
&\le r_h(s,a,b) + \sum_{i=1}^t \beta_t^i \mathbb{P}_h\left[(\mu_{h+1}^i)^\top Q_{h+1}^i \nu_{h+1}^i\right](s,a,b) + \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t \\
&= Q_h^t(s,a,b) + \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t.
\end{aligned}$$

Above, the last equality is derived from the update rule (3), which implies that

$$Q_h^t(s,a,b) = \sum_{i=1}^t \beta_t^i \left(r_h + \mathbb{P}_h\left[(\mu_{h+1}^i)^\top Q_{h+1}^i \nu_{h+1}^i\right]\right)(s,a,b).$$

Therefore we have

$$Q_h^\star(s,a,b) - Q_h^t(s,a,b) \le \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t, \quad \forall\, s,a,b.$$

Apply similar analysis to the min-player, we get

$$Q_h^t(s,a,b) - Q_h^\star(s,a,b) \le \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t, \quad \forall\, s,a,b.$$

Thus we get

$$\delta_h^t \le \sum_{i=1}^t \beta_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1}^t,$$

which completes the proof of the first inequality in the Lemma. Now consider an auxiliary sequence $\{\Delta_h^t\}_{h,t}$ defined by

$$\begin{cases} \Delta_h^t = \sum_{i=1}^t \beta_t^i \Delta_{h+1}^i + \overline{\mathrm{reg}}_{h+1}^t, \\ \Delta_{H+1}^t = 0, \quad \text{for all } t. \end{cases} \tag{27}$$

Where $\overline{\mathrm{reg}}_h^t$ is the upperbound of reg_h^t defined in Theorem 2. Observe that $\{\Delta_h^t\}_{h,t}$ satisfies the following properties

$$\begin{cases} \Delta_h^t \geq \delta_h^t & \text{(by definition)}, \\ \Delta_h^t \leq \Delta_h^{t-1} & \text{(by Lemma A.1)}. \end{cases} \tag{28}$$

Therefore, to control δ_h^t, it suffices to bound $\Delta_h^t \leq \frac{1}{t}\sum_{i=1}^t \Delta_h^i$, which follows from the standard argument in [23]:

$$\begin{aligned} \frac{1}{t}\sum_{i=1}^t \Delta_h^i &= \frac{1}{t}\sum_{i=1}^t\sum_{j=1}^i \beta_i^j \Delta_{h+1}^j + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1}^i \\ &\leq \frac{1}{t}\sum_{j=1}^t \left(\sum_{i=j}^t \beta_i^j\right)\Delta_{h+1}^j + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1}^i \\ &\leq c_\beta \cdot \frac{1}{t}\sum_{i=1}^t \Delta_{h+1}^i + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1}^i \\ &\leq c_\beta^2 \cdot \frac{1}{t}\sum_{i=1}^t \Delta_{h+2}^i + c_\beta \cdot \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+2}^i + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1}^i \\ &\leq \cdots \\ &\leq \left(\sum_{h'=h+1}^H c_\beta^{h'-h}\right) \cdot \frac{1}{t}\sum_{i=1}^t \max_{1\leq h'\leq H} \overline{\mathrm{reg}}_{h'}^i \\ &\leq H c_\beta^{H-1} \cdot \frac{1}{t}\sum_{i=1}^t \max_{1\leq h'\leq H} \overline{\mathrm{reg}}_{h'}^i. \end{aligned}$$

Above, the last step used the fact that $c_\beta \geq 1$. This completes the proof of the second inequality in the Lemma. □

We are now ready to prove the main theorem.

Proof of Theorem 2. Fix any $(h,s) \in [H] \times \mathcal{S}$. We first give a bound for $\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \left[(\mu^\dagger)^\top Q_h^\star \widehat{\nu}_h^T - \left(\widehat{\mu}_h^T\right)^\top Q_h^\star \nu^\dagger\right](s)$, i.e. the per-state duality gap of $(\widehat{\mu}^T, \widehat{\nu}^T)$ with respect to $Q_h^\star$. We have

$$\begin{aligned} &\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \left[(\mu^\dagger)^\top Q_h^\star \widehat{\nu}_h^T - \left(\widehat{\mu}_h^T\right)^\top Q_h^\star \nu^\dagger\right](s) \\ =& \max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \sum_{t=1}^T \beta_T^t \left[(\mu^\dagger)^\top Q_h^\star \nu_h^t - \left(\mu_h^t\right)^\top Q_h^\star \nu^\dagger\right](s) \\ \leq& \underbrace{\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \sum_{t=1}^T \beta_T^t \left[(\mu^\dagger)^\top Q_h^t \nu_h^t - \left(\mu_h^t\right)^\top Q_h^t \nu^\dagger\right](s)}_{\mathrm{reg}_{\mu,h}^T(s)+\mathrm{reg}_{\nu,h}^T(s)} + 2\sum_{t=1}^T \beta_T^t \delta_h^t \\ \leq& 2\overline{\mathrm{reg}}_h^T + 2Hc_\beta^{H-1}\sum_{t=1}^T \beta_T^t \cdot \frac{1}{t}\sum_{i=1}^t \max_{h'} \overline{\mathrm{reg}}_{h'}^i \quad \text{(Lemma C.2)} \\ \leq& 2\overline{\mathrm{reg}}_h^T + 2Hc_\beta^{H-1}\left(\sum_{t=1}^T \frac{1}{t}\beta_T^t\right)\left(\sum_{i=1}^T \max_{h'} \overline{\mathrm{reg}}_{h'}^i\right). \end{aligned}$$

Apply Lemma A.3 into the above inequality, we get that

$$\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}}\left[(\mu^\dagger)^\top Q_h^\star\widehat{\nu}_h^T-\left(\widehat{\mu}_h^T\right)^\top Q_h^\star\nu^\dagger\right](s)\le 2\overline{\mathrm{reg}}_h^T+4Hc_\beta^H\log T\cdot\frac{1}{T}\sum_{t=1}^T\max_{h'}\overline{\mathrm{reg}}_{h'}^t.$$

Since

$$\max_{\mu^\dagger\in\Delta_{\mathcal{A}}}\left(\left[(\mu^\dagger)^\top Q_h^\star\widehat{\nu}_h^T\right](s)-V_h^\star(s)\right),\max_{\nu^\dagger\in\Delta_{\mathcal{B}}}\left(V_h^\star(s)-\left[(\widehat{\mu}_h^T)^\top Q_h^\star\nu^\dagger\right](s)\right)\le\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}}\left[(\mu^\dagger)^\top Q_h^\star\widehat{\nu}_h^T-\left(\widehat{\mu}_h^T\right)^\top Q_h^\star\nu^\dagger\right](s),$$

by applying Lemma C.1 and the preceding per-state duality gap bound, we have

$$\begin{aligned}
&\mathrm{NEGap}(\widehat{\mu}^T,\widehat{\nu}^T)=\left(V_1^{\dagger,\widehat{\nu}^T}(s_1)-V_1^\star(s_1)\right)+\left(V_1^\star(s_1)-V_1^{\widehat{\mu}^T,\dagger}(s_1)\right)\\
&\le 2\sum_{h=1}^H\max_s\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}}\left[(\mu^\dagger)^\top Q_h^\star\widehat{\nu}_h^T-\left(\widehat{\mu}_h^T\right)^\top Q_h^\star\nu^\dagger\right](s)\\
&\le 2\sum_{h=1}^H\left(2\overline{\mathrm{reg}}_h^T+4Hc_\beta^H\log T\cdot\frac{1}{T}\sum_{t=1}^T\max_{h'}\overline{\mathrm{reg}}_{h'}^t\right)\\
&\le 4H\max_h\overline{\mathrm{reg}}_h^T+8H^2c_\beta^H\log T\cdot\frac{1}{T}\sum_{t=1}^T\max_{h'}\overline{\mathrm{reg}}_{h'}^t.
\end{aligned}$$

This completes the proof. □

C.1 Implementation of state-wise average policy

Here we explain how to implement the state-wise average policy (5) efficiently via moving average. Recall that the weights $\{\beta_T^t\}_{t=1}^T$ are defined via $\{\beta_t\}_{t\ge1}$ through (6). Therefore, for each (h,s), we can maintain a set of moving averages

$$\overline{\mu}_h^t(\cdot|s):=(1-\beta_t)\overline{\mu}_h^{t-1}(\cdot|s)+\beta_t\mu_h^t(\cdot|s)$$

for all $t\ge1$ during the execution of Algorithm 1. At time T, we output the moving average $\overline{\mu}_h^T(\cdot|s)$, which is exactly the desired state-wise average policy in (5):

$$\overline{\mu}_h^T(\cdot|s)=\sum_{t=1}^T\beta_T^t\mu_h^t(\cdot|s)=\widehat{\mu}_h^T(\cdot|s).$$

D Algorithm details and proofs for Section 3.2

D.1 Nash V-Learning (full-information version)

The full description of Nash V-Learning (Example 1) using V updates is presented in Algorithm 5.

Proposition D.1 (Equivalence between V update and Q update)**.** *Nash V-learning (full-information version) in Algorithm 5 is equivalent to Algorithm 1 with the* MatrixGameAlg *as weighted FTRL* (10)*.*

Proof. It suffices to show that, for the Q value defined in (4) and the V value defined in (30), the following holds for all (h,s,a,b) and all $t\in[T]$:

$$Q_h^t(s,a,b)=\left[r_h+\mathbb{P}_hV_{h+1}^t\right](s,a,b).\tag{31}$$

Since $\alpha_1=1$, it is not hard to verify that

$$Q_h^1(s,a,b)=\left[r_h+\mathbb{P}_hV_{h+1}^1\right](s,a,b),$$

We now prove by induction on both t and h. Given that $Q_h^1(s,a,b)=\left[r_h+\mathbb{P}_hV_{h+1}^1\right](s,a,b)$, it is not hard to verify that $Q_H^t(s,a,b)=r_H(s,a,b),\ \forall\,k\ge0$. We assume that (31) holds for $(t-1,h)$ and $(t,h+1)$, then for (t,h), from (3)

$$Q_h^t(s,a,b)=(1-\alpha_t)Q_h^{t-1}(s,a,b)+\alpha_t\left(r_h+\mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t\nu_{h+1}^t]\right)(s,a,b)$$

Algorithm 5 Nash V-learning (full-information version)

Require: Learning rate $\{\alpha_t\}_{t\geq 1}$ in (3) and corresponding $\{w_t\}_{t\geq 1}$ (cf. Example 1); $\eta > 0$.
Initialize: Set $V_h^0(s) = H - h + 1$ for all $(h, s) \in [H] \times \mathcal{S}$.
for $t = 1, \ldots, T$ **do**
  for $h = H, \ldots, 1$ **do**
    Update policy for all $s \in \mathcal{S}$ (understanding $w_0 := 1$):

$$\begin{aligned} \mu_h^t(a|s) &\propto_a \exp\left(\frac{\eta}{w_{t-1}} \sum_{i=1}^{t-1} w_i \big[\big(r_h + \mathbb{P}_h V_{h+1}^i\big)\nu_h^i\big](s,a)\right) \\ \nu_h^t(a|s) &\propto_b \exp\left(-\frac{\eta}{w_{t-1}} \sum_{i=1}^{t-1} w_i \Big[\big(r_h + \mathbb{P}_h V_{h+1}^i\big)^\top \mu_h^i\Big](s,b)\right). \end{aligned} \tag{29}$$

    Update V value for all $s \in \mathcal{S}$:

$$V_h^t(s) \leftarrow (1-\alpha_t)V_h^{t-1}(s) + \alpha_t \Big[\big(\mu_h^t\big)^\top \big(r_h + \mathbb{P}_h V_{h+1}^t\big)\nu_h^t\Big](s). \tag{30}$$

$$\leq (1-\alpha_t)\big[r_h + \mathbb{P}_h V_{h+1}^{t-1}\big](s,a,b) + \alpha_t\big(r_h + \mathbb{P}_h\big[(\mu_{h+1}^t)^\top (r_{h+1} + \mathbb{P}_{h+1}V_{h+2})\nu_{h+1}^t\big]\big)(s,a,b)$$

(inductive hypothesis)

$$= \big[r_h + \mathbb{P}_h\big((1-\alpha_t)V_{h+1}^{t-1} + \alpha_t(\mu_{h+1}^t)^\top (r_{h+1} + \mathbb{P}_{h+1}V_{h+2})\nu_{h+1}^t\big)\big](s,a,b)$$

$$= \big[r_h + \mathbb{P}_h V_{h+1}^t\big](s,a,b) \ \text{ (from (30))},$$

which completes the proof by induction. □

Lemma D.1 (Per-state regret bound for Nash V-learning)**.** *Algorithm 5 achieves the following per-state regret bound:*

$$\mathrm{reg}_h^t \leq \frac{(H+1)\log(A \vee B)}{\eta t} + \frac{\eta H^2}{2}, \ \ \forall h \in [H], t \geq 1.$$

Proof. Fix any (h, s). Note that the update of $\{\mu_h^t(\cdot|s)\}_{t\geq 1}$ in (29) is equivalent to the weighted FTRL algorithm (Algorithm 3) with loss vectors $g_i = -\big[Q_h^i \nu_h^i\big](s)$. Thus by Lemma B.2 we get for any $t \geq 1$ that

$$\max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \sum_{i=1}^t w_i \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \big[Q_h^i \nu_h^i\big](s,\cdot)\right\rangle \leq \frac{w_t}{\eta}\log A + \frac{\eta H^2}{2}\sum_{i=1}^t w_i.$$

Further, recalling $\alpha_t^i = w_i \cdot \alpha_t^1$ for $1 \leq i \leq t$, we have

$$\begin{aligned} \mathrm{reg}_{h,\mu}^t &\leq \alpha_t^1\left(\frac{w_t}{\eta}\log A + \frac{\eta H^2}{2}\sum_{i=1}^t w_i\right) = \frac{\alpha_t^1 w_t}{\eta}\log A + \frac{\eta H^2}{2}\sum_{i=1}^t \alpha_t^i \\ &= \frac{\alpha_t}{\eta}\log A + \frac{\eta H^2}{2} \leq \frac{(H+1)\log A}{\eta t} + \frac{\eta H^2}{2}. \end{aligned}$$

The similar bound also holds for $\mathrm{reg}_{h,\nu}^t$, and thus we have that

$$\mathrm{reg}_h^t \leq \frac{(H+1)\log(A \vee B)}{\eta t} + \frac{\eta H^2}{2}.$$

□

Proposition D.2 (Guarantee of Nash V-Learning)**.** *Algorithm 5 achieves*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq 14\eta H^4 \log(T) + \frac{104\log(A \vee B)\log(T)^2 H^3}{\eta T}.$$

Specifically, choosing $\eta = \frac{4}{\sqrt{HT}}$*, we have*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le \frac{82\log(A \vee B)\log(T)^2 H^{7/2}}{\sqrt{T}}.$$

Proof. From Lemma D.1, we can take $\overline{\mathrm{reg}}_h^t$ as $\overline{\mathrm{reg}}_h^t = \frac{(H+1)\log(A\vee B)}{\eta t} + \frac{\eta H^2}{2}$. Then from Theorem 2, we have

$$\begin{aligned}
&\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le 4H \max_h \overline{\mathrm{reg}}_h^T + 8H^2\left(1+\frac{1}{H}\right)^H \log(T) \cdot \frac{1}{T}\sum_{t=1}^T \max_h \overline{\mathrm{reg}}_h^t \\
&\le 4H\left(\frac{(H+1)\log(A\vee B)}{\eta T} + \frac{\eta H^2}{2}\right) + 24H^2\log(T)\cdot\frac{1}{T}\sum_{t=1}^T\left(\frac{(H+1)\log(A\vee B)}{\eta t} + \frac{\eta H^2}{2}\right) \\
&\le 14\eta H^4\log(T) + 24H^2\log(T)\cdot\frac{1}{T}\sum_{t=1}^T\frac{(H+1)\log(A\vee B)}{\eta t} + \frac{8H^2\log(A\vee B)}{\eta T} \\
&\le 14\eta H^4\log(T) + \frac{104\log(A\vee B)\log(T)^2 H^3}{\eta T}.
\end{aligned}$$

Thus, choosing $\eta = \frac{4}{\sqrt{HT}}$, we get

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le \frac{82\log(A \vee B)\log(T)^2 H^{7/2}}{\sqrt{T}}.$$

□

D.2 GDA-Critic

The full description of GDA-Critic (Example 2) using V updates is presented in Algorithm 6.

Algorithm 6 GDA-Critic

Require: Learning rate $\{\alpha_t\}_{t\ge1}$ (defined in (3)), and $\eta > 0$.
Initialize: set $V_h^0(s) = H - h + 1$ and $\mu^0(\cdot|s), \nu^0(\cdot|s)$ to be uniform for all $(h,s) \in [H]\times\mathcal{S}$.
for $t = 1, \dots, T$ **do**
  for $h = H, \dots, 1$ **do**
    Update policy for all $s \in \mathcal{S}$:

$$\begin{aligned}
\mu_h^t(\cdot|s) &\leftarrow \mathcal{P}_{\Delta_\mathcal{A}}\big(\mu_h^{t-1}(\cdot|s) + \eta\big[\big(r_h + \mathbb{P}_h V_{h+1}^{t-1}\big)\nu_h^{t-1}\big](s,\cdot,b)\big) \\
\nu_h^t(\cdot|s) &\leftarrow \mathcal{P}_{\Delta_\mathcal{B}}\Big(\nu_h^{t-1}(\cdot|s) - \eta\Big[\big(r_h + \mathbb{P}_h V_{h+1}^{t-1}\big)^\top \mu_h^{t-1}\Big](s,a,\cdot)\Big)
\end{aligned} \tag{32}$$

    Update V value for all $s \in \mathcal{S}$:

$$V_h^t(s) \leftarrow (1-\alpha_t)V_h^{t-1}(s) + \alpha_t\Big[\big(\mu_h^t\big)^\top\big(r_h + \mathbb{P}_h V_{h+1}^t\big)\nu_h^t\Big](s).$$

Similar as Proposition D.1 (with the same proof), the following equivalence between Q updates and V updates also holds for GDA-Critic.

Proposition D.3 (Equivalence between Q updates and V updates for GDA-Critic)**.** *Algorithm 6 is equivalent to our algorithm framework (Algorithm 1) with the* MatrixGameAlg *instantiated as* (11)*.*

Lemma D.2 (Per-state regret bound for GDA-Critic)**.** *Algorithm 6 achieves the following per-state regret bound:*

$$\mathrm{reg}_h^t \le \frac{2(H+1)}{\eta t} + \frac{\eta(A\vee B)H^2}{2}.$$

Proof. Fix any (h, s) and $t \geq 1$. We apply Lemma B.1 to the projected gradient descent (or ascent) update (32), with weights $w_i = \alpha_t^i$ and loss vectors g_i's $-\left[Q_h^i \nu_h^i\right](s)$ or $\left[\left(Q_h^i\right)^\top \mu_h^i\right](s)$ respectively. For the gradient ascent update for $\mu_h^t(\cdot|s)$, we get

$$\begin{aligned}
\mathrm{reg}_{h,\mu}^t(s) &\leq \frac{\alpha_t^t}{2\eta} \cdot 4 + \frac{\eta\left(\sum_{i=1}^t \alpha_t^i\right) AH^2}{2} = \frac{2}{\eta}\frac{H+1}{H+t} + \frac{\eta A H^2}{2} \\
\mathrm{reg}_{h,\nu}^t(s) &\leq \frac{\alpha_t^t}{2\eta} \cdot 4 + \frac{\eta\left(\sum_{t=1}^T \alpha_t^i\right) BH^2}{2} = \frac{2}{\eta}\frac{H+1}{H+t} + \frac{\eta B H^2}{2} \\
\implies \mathrm{reg}_h^t &\leq \frac{2(H+1)}{\eta t} + \frac{\eta(A \vee B)H^2}{2}.
\end{aligned}$$

□

Proposition D.4 (Guarantee of GDA-Critic)**.** *Algorithm 6 achieves*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq 14\eta(A \vee B)H^4 \log(T) + \frac{208 \log(T)^2 H^3}{\eta T}.$$

Specifically, picking $\eta = \frac{4}{\sqrt{(A \vee B)HT}}$ *yields*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq \frac{108 \log(T)^2 \sqrt{A \vee B} H^{7/2}}{\sqrt{T}}.$$

Proof. From Lemma D.2, we can take $\overline{\mathrm{reg}}_h^t$ as $\frac{2(H+1)}{\eta t} + \frac{\eta(A \vee B)H^2}{2}$, then from Theorem 2

$$\begin{aligned}
&\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq 4H \max_h \overline{\mathrm{reg}}_h^T + 8H^2 \left(1 + \frac{1}{H}\right)^H \log(T) \cdot \frac{1}{T} \sum_{t=1}^T \max_h \overline{\mathrm{reg}}_h^t \\
&\leq 4H \left(\frac{2(H+1)}{\eta T} + \frac{\eta(A \vee B)H^2}{2}\right) + 24H^2 \log(T) \cdot \frac{1}{T} \sum_{t=1}^T \left(\frac{2(H+1)}{\eta t} + \frac{\eta(A \vee B)H^2}{2}\right) \\
&\leq 14\eta(A \vee B)H^4 \log(T) + 48H^2 \log(T) \cdot \frac{1}{T} \sum_{t=1}^T \frac{(H+1)}{\eta t} + \frac{16H^2}{\eta T} \\
&\leq 14\eta(A \vee B)H^4 \log(T) + \frac{208 \log(T)^2 H^3}{\eta T}.
\end{aligned}$$

Thus, pick $\eta = \frac{4}{(A \vee B)\sqrt{HT}}$, we get

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq \frac{108 \log(T)^2 \sqrt{A \vee B} H^{7/2}}{\sqrt{T}}.$$

□

D.3 Nash Q-Learning (full-information version)

The Nash Q-Learning algorithm (Example 3) is described in Algorithm 7.

Lemma D.3 (Per-state regret bound for Nash Q-Learning)**.** *Algorithm 7 achieves the following per-state regret bound:*

$$\mathrm{reg}_h^t \leq \frac{(H+1)^2}{H+t}, \quad \forall\, h \in [H],\ t \geq 1.$$

Proof. We have

$$\mathrm{reg}_{h,\mu}^t(s) = \max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \sum_{i=1}^t \alpha_t^i \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \left[Q_h^i \nu_h^i\right](s, \cdot) \right\rangle$$

Algorithm 7 Nash Q-Learning

Require: Learning rate (For Nash Q-learning) $\{\beta_t = \alpha_t\}$;
Initialize: $Q_h^0(s,a,b) \leftarrow H - h + 1$ for all (h,s,a,b).
for $k = 1, \ldots, K$ **do**
  for $h = H, \ldots, 1$ **do**
    Update policy for all $s \in \mathcal{S}$:

$$(\mu_h^t(\cdot|s), \nu_h^t(\cdot|s)) \leftarrow \mathsf{MatrixNE}(Q_h^{t-1}(s,\cdot,\cdot)). \tag{33}$$

    Update Q value for all $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$:

$$Q_h^t(s,a,b) \leftarrow (1-\alpha_t)Q_h^{t-1}(s,a,b) + \alpha_t\big(r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t \nu_{h+1}^t]\big)(s,a,b).$$

$$\begin{aligned}
&= \max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \sum_{i=1}^t \alpha_t^i \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \left[Q_h^{i-1}\nu_h^i\right](s,\cdot)\right\rangle + \sum_{i=1}^t \alpha_t^i \|Q_h^i - Q_h^{i-1}\|_\infty \\
&\le \sum_{i=1}^t \alpha_t^i \underbrace{\max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \left[Q_h^{i-1}\nu_h^i\right](s,\cdot)\right\rangle}_{=0 \text{ from (33)}} + \sum_{i=1}^t \alpha_t^i \|Q_h^i - Q_h^{i-1}\|_\infty \\
&= \sum_{i=1}^t \alpha_t^i \|Q_h^i - Q_h^{i-1}\|_\infty.
\end{aligned}$$

The same bound also holds for $\mathrm{reg}_{h,\nu}^t(s)$, thus

$$\mathrm{reg}_h^t \le \sum_{i=1}^t \alpha_t^i \|Q_h^i - Q_h^{i-1}\|_\infty.$$

Since

$$\begin{aligned}
&Q_h^i(s,a,b) = (1-\alpha_i)Q_h^{i-1}(s,a,b) + \alpha_i\big(r_h + \mathbb{P}_h[(\mu_{h+1}^i)^\top Q_{h+1}^i \nu_{h+1}^i]\big)(s,a,b) \\
\Longrightarrow &|Q_h^i(s,a,b) - Q_h^{i-1}(s,a,b)| \le \alpha_i \|Q_h^{i-1} - \big(\big(r_h + \mathbb{P}_h[(\mu_{h+1}^i)^\top Q_{h+1}^i \nu_{h+1}^i]\big)\big)\|_\infty \le \alpha_i H \\
\Longrightarrow &\|Q_h^i - Q_h^{i-1}\|_\infty \le \alpha_i H,
\end{aligned}$$

substituting this into the above equations we have that

$$\mathrm{reg}_h^t \le H \sum_{i=1}^t \alpha_t^i \alpha_i \le \frac{(H+1)^2}{H+t} \quad \text{(From Lemma A.3 (b))},$$

which completes the proof. □

Proposition D.5 (Guarantee for Nash Q-Learning)**.** *Algorithm 7 achieves*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le \frac{112 \log(T)^2 H^4}{T}.$$

Proof. From Theorem 2 and Lemma D.3 we have that

$$\begin{aligned}
\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) &\le 4H \max_h \overline{\mathrm{reg}}_h^T + 8H^2 \left(1 + \frac{1}{H}\right)^H \log(T) \cdot \frac{1}{T} \sum_{t=1}^T \max_h \overline{\mathrm{reg}}_h^t \\
&\le 4H \frac{(H+1)^2}{H+T} + 24H^2 \log(T) \cdot \frac{1}{T} \sum_{t=1}^T \frac{(H+1)^2}{H+t} \le \frac{112 \log(T)^2 H^4}{T}.
\end{aligned}$$

□

Algorithm 8 Nash Policy Iteration (Nash-PI)

Initialize: $Q_h^0(s,a,b) \leftarrow H-h+1$ for all (h,s,a,b).
for $t = 1,\dots,T$ **do**
  for $h = H,\dots,1$ **do**
    Update policy for all $s \in \mathcal{S}$:

$$(\mu_h^t(\cdot|s), \nu_h^t(\cdot|s)) \leftarrow \mathsf{MatrixNE}(Q_h^{t-1}(s,\cdot,\cdot)).$$

    Update Q value for all $(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$:

$$Q_h^t(s,a,b) \leftarrow \left[r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t \nu_{h+1}^t]\right](s,a,b). \tag{34}$$

D.4 Nash Policy Iteration

The full description of Nash Policy Iteration (Nash-PI, Example 4) is presented in Algorithm 8.

Note that from (34), we have that Q_h^k equals to $Q_h^{\mu^k\times\nu^k}$. Based on this observation, we have the following lemma.

Lemma D.4 (Exact learning of Q functions)**.** *For Algorithm 8, we have for any $h \in [H]$ and $t \geq H-h+1$ that*

$$Q_h^t(s,a,b) = Q_h^\star(s,a,b), \;\; \forall\,(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}.$$

Proof. We prove this by backward induction over h. For $h = H$, we have that

$$Q_H^t(s,a,b) = r_H(s,a,b), \;\; \forall t \geq 1.$$

Assume that for $h+1$, the condition holds, then for time horizon h and iteration step $t \geq H-h+1$, we have that

$$\begin{aligned} Q_h^t(s,a,b) &= \left[r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t \nu_{h+1}^t]\right](s,a,b) \\ &= \left[r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^\star \nu_{h+1}^t]\right](s,a,b). \end{aligned}$$

Additionally, from the inductive hypothesis

$$(\mu_{h+1}^t(\cdot|s), \nu_{h+1}^t(\cdot|s)) = \mathsf{MatrixNE}(Q_{h+1}^{t-1}(s,\cdot,\cdot)) = \mathsf{MatrixNE}(Q_{h+1}^\star(s,\cdot,\cdot)),$$

we have that

$$[(\mu_{h+1}^t)^\top Q_{h+1}^\star \nu_{h+1}^t](s) = V_{h+1}^\star(s).$$

Thus

$$\begin{aligned} Q_h^t(s,a,b) &= \left[r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^\star \nu_{h+1}^t]\right](s,a,b) \\ &= \left[r_h + \mathbb{P}_h V_{h+1}^\star\right](s,a,b) = Q_h^\star(s,a,b), \end{aligned}$$

which completes the proof. □

Proposition D.6 (Guarantee for Nash-PI)**.** *Algorithm 8 achieves* $\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) = 0$ *for* $T \geq H$.

Proof. For this proposition we will not proof by calling Theorem 2, but instead directly apply Lemma C.1, which is an auxiliary lemma for proving Theorem 2.

Note that Nash-PI corresponds is equivalent to using $\beta_t = 1$ in Algorithm 1, so that $\beta_t^i = \mathbf{1}\{i = t\}$. From Lemma D.4 we have that

$$Q_h^t = Q_h^\star, \;\; \forall\, t \geq H, h \in [H].$$

Thus for $t \geq H$,

$$\max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \left[(\mu^\dagger)^\top Q_h^\star \widehat{\nu}_h^T - \left(\widehat{\mu}_h^T\right)^\top Q_h^\star \nu^\dagger\right](s)$$

Algorithm 9 OFTRL for two-player zero-sum Markov Games

1: **Initialize:** $Q_h^0(s,a,b) \leftarrow H-h+1$ for all (h,s,a,b).
2: **for** $t = 1,\ldots,T$ **do**
3: **for** $h = H,\ldots,1$ **do**
4: Update policies for all $s \in \mathcal{S}$ by OFTRL:

$$\mu_h^t(a|s) \propto_a \exp\Big((\eta/w_t)\cdot\Big[\textstyle\sum_{i=1}^{t-1} w_i(Q_h^i\nu_h^i)(s,a) + w_{t-1}(Q_h^{t-1}\nu_h^{t-1})(s,a)\Big]\Big),$$
$$\nu_h^t(b|s) \propto_b \exp\Big(-(\eta/w_t)\cdot\Big[\textstyle\sum_{i=1}^{t-1} w_i((Q_h^i)^\top\mu_h^i)(s,b) + w_{t-1}((Q_h^{t-1})^\top\mu_h^{t-1})(s,b)\Big]\Big).$$

5: Update Q-value for all $(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$:

$$Q_h^t(s,a,b) \leftarrow (1-\alpha_t)Q_h^{t-1}(s,a,b) + \alpha_t\big(r_h + \mathbb{P}_h[(\mu_{h+1}^t)^\top Q_{h+1}^t\nu_{h+1}^t]\big)(s,a,b). \quad (35)$$

6: Output state-wise average policy:

$$\widehat{\mu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \alpha_T^t\mu_h^t(\cdot|s), \quad \widehat{\nu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \alpha_T^t\nu_h^t(\cdot|s).$$

$$= \max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \Big[(\mu^\dagger)^\top Q_h^T\nu_h^T - \big(\mu_h^T\big)^\top Q_h^T\nu^\dagger\Big](s) + 2\sum_{t=1}^T \beta_T^t\delta_h^t$$
$$= 0, \ \ \forall h.$$

Then applying Lemma C.1, we obtain

$$\mathrm{NEGap}(\widehat{\mu}^T,\widehat{\nu}^T) \le \Big(V_1^{\dagger,\widehat{\nu}^T}(s_1) - V_1^\star(s_1)\Big) + \Big(V_1^\star(s_1) - V_1^{\widehat{\mu}^T,\dagger}(s_1)\Big)$$
$$\le \sum_{h=1}^H \max_{\mu^\dagger\in\Delta_{\mathcal{A}},\nu^\dagger\in\Delta_{\mathcal{B}}} \Big[(\mu^\dagger)^\top Q_h^\star\widehat{\nu}_h^T - \big(\widehat{\mu}_h^T\big)^\top Q_h^\star\nu^\dagger\Big](s) = 0, \quad \text{for } T \ge H.$$

This is the desired result. □

E Proof of Theorem 3

In this section we prove Theorem 3. The full algorithm box of OFTRL for Markov Games is provided in Algorithm 9.

We aim to show that

$$\begin{aligned}&\mathrm{NEGap}(\widehat{\mu}^T,\widehat{\nu}^T)\\ &\le \mathcal{O}\Big(H^{14/3}(\log(A\vee B))^{5/6}(\log T)^{11/6}\cdot T^{-5/6} + H^5\log(A\vee B)(\log T)^2\cdot T^{-1}\Big).\end{aligned} \quad (36)$$

Bounding per-state regret We first bound $\mathrm{reg}_{\nu,h}^t(s)$, i.e. the per-state regret for the min-player, for any fixed $(h,s,t) \in [H]\times[S]\times[T]$. (The bound for $\mathrm{reg}_{\mu,h}^t(s)$ follows similarly.) This is the main part of this proof.

Throughout this part, we will fix (h,s) and omit these subscripts within the policies and Q functions, so that $\nu_h^t(\cdot|s)$ will be abbreviated as ν^t (and similarly for μ^t and Q^t). We will also overload $T \ge 1$ to be any positive integer (instead of the fixed total number of iterations).

Observe that the above update for ν^t is equivalent to the OFTRL algorithm (Algorithm 4) with loss vectors $g_t = w_t(Q^t)^\top\mu^t$ (understanding $g_0 = 0$ and $Q_h^0 = 0$), prediction vector $M_t = g_{t-1} = w_{t-1}(Q^{t-1})^\top\mu^{t-1}$, and learning rate $\eta_t = \eta/w_t$. Therefore we can apply the regret bound for

OFTRL in Lemma B.3 and obtain for any $T \geq 1$ that

$$\begin{aligned}
&\max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} w_t \left\langle \nu^t - \nu^\dagger, (Q^t)^\top \mu^t \right\rangle = \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \left\langle \nu^t - \nu^\dagger, g_t \right\rangle \\
&\leq \frac{\log B}{\eta_T} + \sum_{t=1}^{T} \eta_t \|g_t - M_t\|_\infty^2 - \sum_{t=1}^{T-1} \frac{1}{8\eta_t} \|\nu^t - \nu^{t+1}\|_1^2 \qquad (37) \\
&= \frac{\log B \cdot w_T}{\eta} + \sum_{t=1}^{T} \frac{\eta}{w_t} \|w_t (Q^t)^\top \mu^t - w_{t-1}(Q^{t-1})^\top \mu^{t-1}\|_\infty^2 - \sum_{t=2}^{T} \frac{w_{t-1}}{8\eta} \|\nu^t - \nu^{t-1}\|_1^2.
\end{aligned}$$

We now relate the terms above to the stability of $\{\mu^t\}_{t\geq 1}$ (the other player's policies). Let

$$\Delta_t := \left\| w_t Q^t - w_{t-1} Q^{t-1} \right\|_\infty$$

for all $t \geq 1$ for shorthand, where $\|\cdot\|_\infty$ for a matrix denotes its infinity norm (i.e. entry-wise max absolute value). Then we have

$$\begin{aligned}
&\|w_t (Q^t)^\top \mu^t - w_{t-1}(Q^{t-1})^\top \mu^{t-1}\|_\infty^2 \\
&\leq 2 \left\| (w_t Q^t - w_{t-1} Q^{t-1})^\top \mu^{t-1} \right\|_\infty^2 + 2 \left\| (w_{t-1} Q^{t-1})^\top (\mu^t - \mu^{t-1}) \right\|_\infty^2 \\
&\leq 2\Delta_t^2 + 2 w_{t-1}^2 \left\| Q^{t-1} \right\|_\infty^2 \left\| \mu^t - \mu^{t-1} \right\|_1^2 \\
&\leq 2\Delta_t^2 + 2 w_{t-1}^2 H^2 \left\| \mu^t - \mu^{t-1} \right\|_1^2 \mathbf{1}\{t \geq 2\}.
\end{aligned}$$

By symmetry, the similar bound also holds for $\{\nu^t\}$, from which we obtain for any $t \geq 2$ that

$$-w_{t-1} \left\| \nu^t - \nu^{t-1} \right\|_1^2 \leq \frac{\Delta_t^2}{w_{t-1} H^2} - \frac{1}{2 w_{t-1} H^2} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2.$$

Plugging the above two bounds into (37), we get

$$\begin{aligned}
&\max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} w_t \left\langle \nu^t - \nu^\dagger, (Q^t)^\top \mu^t \right\rangle \\
&\leq \frac{\log B \cdot w_T}{\eta} + \sum_{t=1}^{T} \left[\frac{2\eta}{w_t} \Delta_t^2 + \frac{2\eta H^2 w_{t-1}^2}{w_t} \left\| \mu^t - \mu^{t-1} \right\|_1^2 \mathbf{1}\{t \geq 2\} \right] \\
&\quad + \sum_{t=2}^{T} \left[\frac{\Delta_t^2}{8\eta H^2 w_{t-1}} - \frac{1}{16\eta H^2 w_{t-1}} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2 \right] \\
&\overset{(i)}{\leq} \frac{\log B \cdot w_T}{\eta} + \underbrace{\sum_{t=1}^{T} \left[\frac{2\eta}{w_t} + \frac{1}{8\eta H^2 w_{t-1}} \mathbf{1}\{t \geq 2\} \right] \Delta_t^2}_{:=\mathrm{ERR}_T} + \underbrace{4\eta^2 H^2 \cdot \sum_{t=2}^{T} \frac{1}{2\eta / w_t} \left\| \mu^t - \mu^{t-1} \right\|_1^2}_{:=\mathrm{STAB}_t} \\
&\quad - \sum_{t=2}^{T} \frac{1}{16\eta H^2 w_{t-1}} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2.
\end{aligned} \qquad (38)$$

Above, (i) rearranges terms and used the fact that $w_{t-1} \leq w_t$.

The following lemma (proof deferred to Section E.1) bounds term ERR_T.

Lemma E.1 (Bound on ERR_T)**.** *Suppose $\eta \leq 1/H$. Then for any $T \geq 1$, we have*

$$\alpha_T^1 \cdot \mathrm{ERR}_T \leq \frac{192 H^2}{\eta T}.$$

To bound term STAB_T, note that it is exactly the total distance (in squared L_1 norm) of the sequence $\{\mu^t\}_{t\geq 1}$, which itself follows an OFTRL algorithm with loss sequence $g_t' := -w_t Q^t \nu^t$, $M_t' := g_{t-1}'$, and $\eta_t = \eta / w_t$. Therefore we can apply the stability bound (24) in Lemma B.4 to obtain that

$$\mathrm{STAB}_T = 4\eta^2 H^2 \cdot \sum_{t=2}^{T} \frac{1}{2\eta_t} \left\| \mu^t - \mu^{t-1} \right\|_1^2$$

$$
\begin{aligned}
&\leq 4\eta^2 H^2 \left(\frac{2\log A}{\eta_T} + \sum_{t=1}^{T-1} (1+\eta_t G'_t) \left\| g'_t - g'_{t-1} \right\|_\infty + \left\| g'_{T-1} \right\|_\infty \right) \\
&= 4\eta^2 H^2 \left(\frac{2 w_T \log A}{\eta} + \sum_{t=1}^{T-1} (1+\eta H) \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty + \left\| w_{T-1} Q^{T-1} \nu^{T-1} \right\|_\infty \right) \\
&\overset{(i)}{\leq} 4\eta^2 H^2 \left(\frac{2 w_T \log A}{\eta} + 2 \sum_{t=1}^{T-1} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty + w_{T-1} H \right) \\
&\overset{(ii)}{\leq} 4\eta^2 H^2 \left(\frac{4 w_T \log A}{\eta} + 2 \sum_{t=1}^{T-1} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty \right),
\end{aligned}
$$

where here we take $G'_t = w_t H \geq \left\| g'_t - g'_{t-1} \right\|_\infty$, (i) holds whenever $\eta \leq 1/H$, and (ii) follows as $w_{T-1} H \leq w_T/\eta \leq 2 w_T \log A/\eta$.

Plugging the above bounds into (38) yields that for any $T \geq 1$,

$$
\begin{aligned}
&\mathrm{reg}^T_{\nu,h}(s) = \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \underbrace{\alpha_T^t}_{\alpha_T^1 \cdot w_t} \left\langle \nu^t - \nu^\dagger, (Q^t)^\top \mu^t \right\rangle = \alpha_T^1 \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} w_t \left\langle \nu^t - \nu^\dagger, (Q^t)^\top \mu^t \right\rangle \\
&\leq \frac{\log B \cdot (\alpha_T^1 w_T)}{\eta} + \alpha_T^1 \mathrm{ERR}_T + \alpha_T^1 \left[\mathrm{STAB}_T - \sum_{t=2}^{T} \frac{1}{16\eta H^2 w_{t-1}} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2 \right] \\
&\leq \frac{\log B \cdot \alpha_T^T}{\eta} + \frac{192 H^2}{\eta T} + \alpha_T^1 \Bigg[16\eta H^2 w_T \log A \\
&\qquad + 8\eta^2 H^2 \sum_{t=1}^{T-1} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty - \sum_{t=2}^{T} \frac{1}{16\eta H^2 w_{t-1}} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2 \Bigg] \\
&\overset{(i)}{\leq} \frac{\log B \cdot \alpha_T^T}{\eta} + \frac{192 H^2}{\eta T} + \alpha_T^1 \Bigg[32 \underbrace{\eta H^2}_{\leq 1/\eta} w_T \log A \\
&\qquad + \sum_{t=2}^{T-1} \left(8\eta^2 H^2 \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty - \frac{1}{16\eta H^2 w_{t-1}} \left\| w_t Q^t \nu^t - w_{t-1} Q^{t-1} \nu^{t-1} \right\|_\infty^2 \right) \Bigg] \\
&\overset{(ii)}{\leq} \frac{33 \log(A \vee B) \cdot \alpha_T^T}{\eta} + \frac{192 H^2}{\eta T} + \alpha_T^1 \sum_{t=2}^{T-1} 256 \eta^5 H^6 w_{t-1} \\
&\leq \frac{33 \log(A \vee B) \cdot \alpha_T^T}{\eta} + \frac{192 H^2}{\eta T} + 256 \eta^5 H^6 \underbrace{\sum_{t=2}^{T-1} \alpha_T^{t-1}}_{\leq 1} \\
&\overset{(iii)}{\leq} C \left[\frac{H^2 \log(A \vee B)}{\eta T} + \eta^5 H^6 \right].
\end{aligned}
$$

Above, (i) used the fact that $8\eta^2 H^2 \left\| w_1 Q^1 \nu^1 \right\|_\infty \leq 8\eta^2 H^3 w_1 \leq 8\eta^2 H^3 w_T \leq 16\eta H^2 w_T \log A$, (ii) used the fact that $8\eta^2 H^2 z - z^2/(16\eta H^2 w_{t-1}) \leq 256\eta^5 H^6 w_{t-1}$ by the AM-GM inequality, and (iii) used the fact that $\alpha_T^T = \alpha_T = (H+1)/(H+T) \leq 2H/T$, where $C \leq 256$ is an absolute constant.

By symmetry, the same regret bound also holds for $\mathrm{reg}^T_{\mu,h}(s)$, which gives that for any $t \geq 1$

$$
\mathrm{reg}^t_h := \max_{s \in \mathcal{S}} \max \left\{ \mathrm{reg}^t_{\mu,h}(s), \mathrm{reg}^t_{\nu,h}(s) \right\} \leq \underbrace{C \left[\frac{H^2 \log(A \vee B)}{\eta t} + \eta^5 H^6 \right]}_{:= \overline{\mathrm{reg}}^t_h}.
$$

Note that $\overline{\mathrm{reg}}^t_h$ is decreasing in t. This is the desired regret bound.

Performance of output policy As our algorithm chooses $\beta_t = \alpha_t = (H+1)/(H+t)$, we can invoke Theorem 2 with $c_\beta = 1 + 1/H \geq \sum_{t=j}^{\infty} \alpha_t^j$ (by Lemma A.2) so that $c_\beta^H = (1+1/H)^H \leq e \leq 3$. Further, by the above regret bound,

$$\max_{h\in[H]} \overline{\mathrm{reg}}_h^t \leq C\left[\frac{H^2 \log(A \vee B)}{\eta t} + \eta^5 H^6\right].$$

Plugging this into Theorem 2 yields that the output policy $(\widehat{\mu}^T, \widehat{\nu}^T)$ satisfies

$$\begin{aligned}
&\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \\
&\leq \mathcal{O}\left(H \max_{h\in[H]} \overline{\mathrm{reg}}_h^T + H^2 c_\beta^H \cdot \frac{\log T}{T} \sum_{t=1}^{T} \max_{h\in[H]} \overline{\mathrm{reg}}_h^t\right) \\
&\leq H \cdot \mathcal{O}\left(\frac{H^2 \log(A\vee B)}{\eta T} + \eta^5 H^6\right) + H^2 \frac{\log T}{T} \cdot \mathcal{O}\left(\frac{H^2 \log(A\vee B)\log T}{\eta} + \eta^5 H^6 T\right) \\
&= \mathcal{O}\left(\frac{H^4 \log(A\vee B)(\log T)^2}{\eta T} + \eta^5 H^8 \log T\right).
\end{aligned}$$

Choosing $\eta = (\log T \log(A\vee B)/H^4 T)^{1/6} \wedge (1/H)$, we get

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \leq \mathcal{O}\Big(H^{14/3}(\log(A\vee B))^{5/6}(\log T)^{11/6} \cdot T^{-5/6} + H^5 \log(A\vee B)(\log T)^2/T\Big).$$

This proves (36) and thus Theorem 3.

E.1 Proof of Lemma E.1

Recall our notation $Q^t := Q_h^t(s,\cdot,\cdot) \in [0,H]^{A\times B}$ for some fixed $(h,s) \in [H]\times\mathcal{S}$. We first note that, for any $t \geq 2$,

$$\begin{aligned}
&\left\|w_t Q^t - w_{t-1}Q^{t-1}\right\|_\infty^2 \leq 2\left\|w_t Q^t - w_{t-1}Q^t\right\|_\infty^2 + 2\left\|w_{t-1}(Q^t - Q^{t-1})\right\|_\infty^2 \\
&\leq 2(w_t - w_{t-1})^2 H^2 + 2w_{t-1}^2 \alpha_t^2 H^2 \\
&= 2w_{t-1}^2 H^2\left[\alpha_t^2 + \frac{H^2}{(t-1)^2}\right] \leq 2w_{t-1}^2 H^2 \cdot \frac{8H^2}{t^2} = 16 w_{t-1}^2 H^4/t^2.
\end{aligned}$$

For $t=1$, we have $\left\|w_t Q^t - w_{t-1}Q^{t-1}\right\|_\infty^2 \leq w_1^2 H^2 = H^2$. Substituting this into the expression of ERR_T gives

$$\begin{aligned}
&\alpha_T^1 \mathrm{ERR}_T \\
&= \alpha_T^1 \sum_{t=1}^{T}\left(\frac{2\eta}{w_t} + \frac{1}{8\eta w_{t-1} H^2}\mathbf{1}\{t\geq 2\}\right)\cdot\left(H^2\mathbf{1}\{t=1\} + 16w_{t-1}^2 H^4/t^2 \cdot \mathbf{1}\{t\geq 2\}\right) \\
&= 2\eta\alpha_T^1 H^2 + \alpha_T^1 \sum_{t=2}^{T}\left(\frac{2\eta w_{t-1}^2}{w_t} + \frac{w_{t-1}}{8\eta H^2}\right)\cdot\frac{16H^4}{t^2} \\
&\overset{(i)}{\leq} 2\eta\alpha_T^1 H^2 + \sum_{t=2}^{T}\left(2\eta\alpha_T^t H^2 + \frac{\alpha_T^t}{8\eta}\right)\cdot\frac{16H^2}{t^2} \\
&\overset{(ii)}{\leq} 2\eta\alpha_T^1 H^2 + \sum_{t=2}^{T}\alpha_T^t \cdot \frac{3}{\eta}\cdot\frac{16H^2}{t^2} \\
&\overset{(iii)}{\leq} \frac{48H^2}{\eta}\sum_{t=1}^{T}\alpha_T^t \cdot \frac{1}{t^2} \overset{(iv)}{\leq} \frac{192H^2}{\eta T}.
\end{aligned}$$

Above, (i) used $w_{t-1} \leq w_t$ and $\alpha_T^1 w_t = \alpha_T^t$; (ii) used the fact that $2\eta H^2 \leq 2/\eta$ (as $\eta \leq 1/H$) and thus $2\eta H^2 + 1/(8\eta) \leq (2+1/8)/\eta \leq 3/\eta$; (iii) used the fact that $2\eta H^2 \leq 48H^2/\eta$ which also follows from $\eta \leq 1/H \leq 1$; (iv) used Lemma A.3(a). This is the desired result. □

F A modified OFTRL algorithm with $\widetilde{\mathcal{O}}(T^{-1})$ rate

In this section we show that a slightly modified OFTRL algorithm (described Algorithm 10) achieves $\widetilde{\mathcal{O}}(T^{-1})$ convergence rate for finding NE in two-player zero-sum Markov Games, improving over the $\widetilde{\mathcal{O}}(T^{-5/6})$ of Algorithm 9.

Algorithm 10 Modified OFTRL

1: **Initialize:** $\overline{Q}_h^0(s,a,b) \leftarrow H-h+1, \underline{Q}_h^0 \leftarrow 0$ for all (h,s,a,b).
2: **for** $t = 1,\ldots,T$ **do**
3: **for** $h = H,\ldots,1$ **do**
4: Update policies for all $s \in \mathcal{S}$ by OFTRL:

$$\mu_h^t(a|s) \propto_a \exp\left((\eta/w_t)\cdot\left[\sum_{i=1}^{t-1} w_i(\overline{Q}_h^i \nu_h^i)(s,a) + w_{t-1}(\overline{Q}_h^{t-1}\nu_h^{t-1})(s,a)\right]\right);$$

$$\nu_h^t(b|s) \propto_b \exp\left(-(\eta/w_t)\cdot\left[\sum_{i=1}^{t-1} w_i((\underline{Q}_h^i)^\top \mu_h^i)(s,b) + w_{t-1}((\underline{Q}_h^{t-1})^\top \mu_h^{t-1})(s,b)\right]\right).$$

5: Update Q-values for all $(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$:

$$\overline{Q}_h^t(s,a,b) \leftarrow r_h(s,a,b) + \mathbb{P}_h\left[\max_{\mu^\dagger\in\Delta_{\mathcal{A}}}\left\langle \mu^\dagger, \sum_{i=1}^t \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i\right\rangle\right](s,a,b);$$

$$\underline{Q}_h^t(s,a,b) \leftarrow r_h(s,a,b) + \mathbb{P}_h\left[\min_{\nu^\dagger\in\Delta_{\mathcal{B}}}\left\langle \nu^\dagger, \sum_{i=1}^t \alpha_t^i \left(\underline{Q}_{h+1}^i\right)^\top \mu_{h+1}^i\right\rangle\right](s,a,b). \tag{39}$$

6: Output state-wise average policy for all (h,s):

$$\widehat{\mu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \alpha_T^t \mu_h^t(\cdot|s), \quad \widehat{\nu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \alpha_T^t \nu_h^t(\cdot|s).$$

Algorithm 10 keeps track of a series of $\overline{Q}_h^t, \underline{Q}_h^t$'s that are upper-bounds and lower-bounds of $Q_h^\star$ respectively. The policy update is similar to the update as the OFTRL algorithm (Algorithm 4), but here μ is performing OFTRL with respect to $\overline{Q}_h^t$'s while ν with respect to $\underline{Q}_h^t$'s. The value updates (39) are slightly different from the value update in our unified framework, however, we remark that it is still an incremental update because the terms inside the inner product $\sum_{i=1}^t \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i, \sum_{i=1}^t \alpha_t^i \left(\underline{Q}_{h+1}^i\right)^\top \mu_{h+1}^i$ are incremental updates, which leads to that fact that $\overline{Q}_h^t, \underline{Q}_h^t$'s are also updating incrementally.. Further, the algorithm can be performed in a decentralized manner, which is stated in Algorithm 11. The convergence result is stated in Theorem F.1.

Theorem F.1 (Convergence rate of modified OFTRL)**.** *Algorithm 10 with $\eta = \frac{1}{16H}$ guarantees that*

$$\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le C\left[\frac{H^4 \log(A\vee B)(\log T)^2}{T}\right],$$

where C is some absolute constant.

F.1 Proof of Theorem F.1

In this section, we consider the following definitions of regret, which is slightly different from the definition in (7):

$$\mathrm{reg}_{h,\mu}^t(s) := \max_{\mu^\dagger\in\Delta_{\mathcal{A}}} \sum_{i=1}^t \alpha_t^i \left\langle \mu^\dagger - \mu_h^i(\cdot|s), \left[\overline{Q}_h^i \nu_h^i\right](s,\cdot)\right\rangle,$$

$$\mathrm{reg}_{h,\nu}^t(s) := \max_{\nu^\dagger\in\Delta_{\mathcal{B}}} \sum_{i=1}^t \alpha_t^i \left\langle \nu_h^i(\cdot|s) - \nu^\dagger, \left[(\underline{Q}_h^i)^\top \mu_h^i\right](s,\cdot)\right\rangle,$$

$$\mathrm{reg}_{h,\mu+\nu}^t := \max_{s\in\mathcal{S}} \mathrm{reg}_{h,\mu}^t(s) + \mathrm{reg}_{h,\nu}^t(s).$$

We first prove that $\underline{Q}_h^t$ and $\overline{Q}_h^t$ upper and lower bounds $Q_h^\star$ respectively.

Lemma F.1.

$$\underline{Q}_h^t(s,a,b) \le Q_h^\star(s,a,b) \le \overline{Q}_h^t(s,a,b).$$

Algorithm 11 Modified OFTRL (Equivalent V-form)

1: **Initialize:** $\overline{V}_h^1(s) \leftarrow H-h+1, \underline{V}_h^1(s) \leftarrow 0$ for all (h,s,a,b).
2: **for** $t = 1, \ldots, T$ **do**
3: **for** $h = H, \ldots, 1$ **do**
4: Update policies for all $s \in \mathcal{S}$ by OFTRL:

$$\mu_h^t(a|s) \propto_a \exp\left((\eta/w_t)\cdot\left[\textstyle\sum_{i=1}^{t-1} w_i \overline{L}_h^i(s,a) + w_{t-1}\overline{L}_h^{t-1}(s,a)\right]\right)$$

$$\nu_h^t(b|s) \propto_b \exp\left(-(\eta/w_t)\cdot\left[\textstyle\sum_{i=1}^{t-1} w_i((Q_h^i)^\top \mu_h^i)(s,b) + w_{t-1}((Q_h^{t-1})^\top \mu_h^{t-1})(s,b)\right]\right).$$

5: Update losses for all $(s,a) \in \mathcal{S}\times\mathcal{A}$:

$$\overline{L}_h^t(s,a) \leftarrow \left\langle r_h(s,a,\cdot) + \left[\mathbb{P}_h \overline{V}_{h+1}^t\right](s,a,\cdot), \nu_h^t(\cdot|s)\right\rangle,$$

$$\underline{L}_h^t(s,a) \leftarrow \left\langle \left[r_h(s,a,\cdot) + \left[\mathbb{P}_h \underline{V}_{h+1}^t\right](s,a,\cdot)\right]^\top, \mu_h^t(\cdot|s)\right\rangle.$$

6: Update V-value for all $s \in \mathcal{S}$:

$$\overline{V}_h^t(s) \leftarrow \max_{\mu^\dagger\in\Delta_\mathcal{A}} \left\langle \mu^\dagger, \textstyle\sum_{i=1}^t \alpha_t^i \overline{L}_h^i(s,\cdot)\right\rangle, \quad \underline{V}_h^t(s) \leftarrow \min_{\nu^\dagger\in\Delta_\mathcal{B}} \left\langle \nu^\dagger, \textstyle\sum_{i=1}^t \alpha_t^i \underline{L}_h^i(s,\cdot)\right\rangle. \tag{40}$$

7: Output state-wise average policy for all (h,s):

$$\widehat{\mu}_h^T(\cdot|s) \leftarrow \textstyle\sum_{t=1}^T \alpha_T^t \mu_h^t(\cdot|s), \quad \widehat{\nu}_h^T(\cdot|s) \leftarrow \sum_{t=1}^T \alpha_T^t \nu_h^t(\cdot|s).$$

Proof. We prove by induction on (h,t). Given the initialization, for $t=0$ the condition holds. Since $\overline{Q}_{H+1}^t, \underline{Q}_{H+1}^t = 0$, we have that for $h = H+1$ the condition holds. Assume that the condition hold for $(i, h+1), i \le t$, then

$$\begin{aligned}
\overline{Q}_h^t(s,a,b) &= r_h(s,a,b) + \mathbb{P}_h\left[\max_{\mu^\dagger\in\Delta_\mathcal{A}} \left\langle \mu^\dagger, \sum_{i=1}^t \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i\right\rangle\right](s,a,b) \\
&\ge r_h(s,a,b) + \mathbb{P}_h\left[\max_{\mu^\dagger\in\Delta_\mathcal{A}} \left\langle \mu^\dagger, Q_{h+1}^\star\left(\sum_{i=1}^t \alpha_t^i \nu_{h+1}^i\right)\right\rangle\right](s,a,b) \\
&\ge r_h(s,a,b) + \mathbb{P}_h\left[\max_{\mu^\dagger\in\Delta_\mathcal{A}} \min_{\nu^\dagger\in\Delta_\mathcal{B}} \left\langle \mu^\dagger, Q_{h+1}^\star \nu^\dagger\right\rangle\right](s,a,b) \\
&= Q_h^\star(s,a,b).
\end{aligned}$$

Using similar strategy, we can also show that $\underline{Q}_h^t(s,a,b) \le Q_h^\star(s,a,b)$, which implies that the condition hold for (t,h), and thus finishes the proof by induction. □

Throughout the rest of this section, we define the following shorthand for the gap between $\overline{Q}_h^t, \underline{Q}_h^t$ defined in (39):

$$\delta_h^t := \|\overline{Q}_h^t - \underline{Q}_h^t\|_\infty = \max_{s,a,b}\left[\overline{Q}_h^t(s,a,b) - \underline{Q}_h^t(s,a,b)\right],$$

Lemma F.2 (Recursion of δ_h^t)**.** *Algorithm 10 guarantees that for all $(t,h) \in [T]\times[H]$,*

$$\delta_h^t \le \sum_{i=1}^t \alpha_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1,\mu+\nu}^t.$$

Further, suppose that $\mathrm{reg}_{h,\mu+\nu}^t \le \overline{\mathrm{reg}}_{h,\mu+\nu}^t$ for all $(h,t) \in [H]\times[T]$, where $\overline{\mathrm{reg}}_{h,\mu+\nu}^t$ is non-increasing in t: $\overline{\mathrm{reg}}_{h,\mu+\nu}^t \ge \overline{\mathrm{reg}}_{h,\mu+\nu}^{t+1}$ for all $t \ge 1$. Then we have

$$\delta_h^t \le 2H \cdot \frac{1}{t}\sum_{i=1}^t \max_{h'} \overline{\mathrm{reg}}_{h',\mu+\nu}^i.$$

Proof. The proof structure resembles Lemma C.2. From the definition of $\overline{Q}_h^t, \underline{Q}_h^t$, we have that

$$
\begin{aligned}
&\overline{Q}_h^t(s,a,b) - \underline{Q}_h^t(s,a,b) \\
&\le \mathbb{P}_h \max_{\mu^\dagger \in \Delta_{\mathcal{A}}, \nu^\dagger \in \Delta_{\mathcal{B}}} \left\langle \mu^\dagger, \sum_{i=1}^t \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i \right\rangle - \left\langle \nu^\dagger, \sum_{i=1}^t \alpha_t^i (\underline{Q}_{h+1}^i)^\top \mu_{h+1}^i \right\rangle \\
&= \mathbb{P}_h \Bigg[\max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \left\langle \mu^\dagger, \sum_{i=1}^t \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i \right\rangle - \sum_{i=1}^t \alpha_t^i (\mu_{h+1}^i)^\top \overline{Q}_{h+1}^i \nu_{h+1}^i \\
&+ \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{i=1}^t \alpha_t^i (\mu_{h+1}^i)^\top \overline{Q}_{h+1}^i \nu_{h+1}^i - \left\langle \nu^\dagger, \sum_{i=1}^t \alpha_t^i (\underline{Q}_{h+1}^i)^\top \mu_{h+1}^i \right\rangle \\
&+ \sum_{i=1}^t \alpha_t^i (\mu_{h+1}^i)^\top \overline{Q}_{h+1}^i \nu_{h+1}^i - \sum_{i=1}^t \alpha_t^i (\mu_{h+1}^i)^\top \overline{Q}_{h+1}^i \nu_{h+1}^i \Bigg] \\
&\le \mathrm{reg}_{h+1,\mu+\nu}^t + \sum_{i=1}^t \alpha_t^i \|\overline{Q}_{h+1}^i - \underline{Q}_{h+1}^i\|_\infty = \sum_{i=1}^t \alpha_t^i \delta_{h+1}^i + \mathrm{reg}_{h+1,\mu+\nu}^t.
\end{aligned}
$$

Then using the same argument as Lemma C.2, we can consider an auxiliary sequence

$$
\begin{cases} \Delta_h^t = \sum_{i=1}^t \alpha_t^i \Delta_{h+1}^i + \overline{\mathrm{reg}}_{h+1,\mu+\nu}^t, \\ \Delta_{H+1}^t = 0, \quad \text{for all } t. \end{cases} \tag{41}
$$

Observe that $\{\Delta_h^t\}_{h,t}$ satisfies the following properties

$$
\begin{cases} \Delta_h^t \ge \delta_h^t & \text{(by definition)}, \\ \Delta_h^t \le \Delta_h^{t-1} & \text{(by Lemma A.1)}. \end{cases} \tag{42}
$$

Therefore, to control δ_h^t, it suffices to bound $\Delta_h^t \le \frac{1}{t}\sum_{i=1}^t \Delta_h^i$, which follows from the standard argument in [23]:

$$
\begin{aligned}
\frac{1}{t}\sum_{i=1}^t \Delta_h^i &= \frac{1}{t}\sum_{i=1}^t \sum_{j=1}^i \alpha_i^j \Delta_{h+1}^j + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1,\mu+\nu}^i \\
&\le \frac{1}{t}\sum_{j=1}^t \left(\sum_{i=j}^t \alpha_i^j\right) \Delta_{h+1}^j + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1,\mu+\nu}^i \\
&\le \left(1+\frac{1}{H}\right)\cdot \frac{1}{t}\sum_{i=1}^t \Delta_{h+1}^i + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1,\mu+\nu}^i \\
&\le \left(1+\frac{1}{H}\right)^2 \cdot \frac{1}{t}\sum_{i=1}^t \Delta_{h+2}^i + \left(1+\frac{1}{H}\right)\cdot \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+2,\mu+\nu}^i + \frac{1}{t}\sum_{i=1}^t \overline{\mathrm{reg}}_{h+1,\mu+\nu}^i \\
&\le \cdots \\
&\le \left(\sum_{h'=h}^H \left(1+\frac{1}{H}\right)^{h'-h}\right)\cdot \frac{1}{t}\sum_{i=1}^t \max_{1\le h'\le H} \overline{\mathrm{reg}}_{h',\mu+\nu}^i \\
&\le (e-1)H \cdot \frac{1}{t}\sum_{i=1}^t \max_{1\le h'\le H} \overline{\mathrm{reg}}_{h',\mu+\nu}^i \le 2H \cdot \frac{1}{t}\sum_{i=1}^t \max_{1\le h'\le H} \overline{\mathrm{reg}}_{h',\mu+\nu}^i.
\end{aligned}
$$

which completes the proof. □

Lemma F.3 (Bound the NEGap by $\mathrm{reg}_{h,\mu+\nu}$)**.** *Suppose that the per-state regrets (summing over the two agents) can be upper-bounded as* $\mathrm{reg}_{h,\mu+\nu}^t \le \overline{\mathrm{reg}}_{h,\mu+\nu}^t$ *for all* $(h,t) \in [H]\times[T]$ *where* $\overline{\mathrm{reg}}_{h,\mu+\nu}^t$ *is non-increasing in* t*:* $\overline{\mathrm{reg}}_{h,\mu+\nu}^t \ge \overline{\mathrm{reg}}_{h,\mu+\nu}^{t+1}$ *for all* $t \ge 1$*. Then, the output policy* $(\widehat{\mu}^T, \widehat{\nu}^T)$ *of Algorithm 10 satisfies*

$$
\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) \le 2H \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 24H^2 \log T \cdot \frac{1}{T}\sum_{t=1}^T \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^t
$$

Proof. From Lemma C.1 we have that

$$
\begin{aligned}
&\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) = \left(V_1^{\dagger,\widehat{\nu}^T}(s_1) - V_1^{\star}(s_1)\right) + \left(V_1^{\star}(s_1) - V_1^{\widehat{\mu}^T,\dagger}(s_1)\right) \\
&\leq 2\sum_{h=1}^{H} \max_s \max_{\mu^\dagger \in \Delta_{\mathcal{A}}, \nu^\dagger \in \Delta_{\mathcal{B}}} \left[(\mu^\dagger)^\top Q_h^\star \widehat{\nu}_h^T - \left(\widehat{\mu}_h^T\right)^\top Q_h^\star \nu^\dagger\right](s) \\
&= 2\sum_{h=1}^{H} \max_s \max_{\mu^\dagger \in \Delta_{\mathcal{A}}, \nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \alpha_T^t \left[(\mu^\dagger)^\top Q_h^\star \nu_h^t - \left(\mu_h^t\right)^\top Q_h^\star \nu^\dagger\right](s) \\
&\leq 2\sum_{h=1}^{H} \max_s \max_{\mu^\dagger \in \Delta_{\mathcal{A}}, \nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \alpha_T^t \left[(\mu^\dagger)^\top \overline{Q}_h^t \nu_h^t - \left(\mu_h^t\right)^\top \underline{Q}_h^t \nu^\dagger\right](s) \\
&\leq 2\sum_{h=1}^{H} \max_s \left(\max_{\mu^\dagger \in \Delta_{\mathcal{A}}} \sum_{t=1}^{T} \alpha_T^t \left[(\mu^\dagger)^\top \overline{Q}_h^t \nu_h^t - \left(\mu_h^t\right)^\top \overline{Q}_h^t \nu_h^t\right](s) \right. \\
&\qquad \left. + \max_{\nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \alpha_T^t \left[(\mu_h^t)^\top \underline{Q}_h^t \nu_h^t - \left(\mu_h^t\right)^\top \underline{Q}_h^t \nu^\dagger\right](s) \right) \\
&+ 2\sum_{h=1}^{H} \max_s \max_{\mu^\dagger \in \Delta_{\mathcal{A}}, \nu^\dagger \in \Delta_{\mathcal{B}}} \sum_{t=1}^{T} \alpha_T^t \left[(\mu_h^t)^\top \overline{Q}_h^t \nu_h^t - \left(\mu_h^t\right)^\top \underline{Q}_h^t \nu_h^t\right](s) \\
&\leq 2\sum_{h=1}^{H} \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 2\sum_{h=1}^{H}\sum_{t=1}^{T} \alpha_T^t \delta_h^t \\
&\leq 2H \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 4H^2 \sum_{t=1}^{T} \alpha_T^t \frac{1}{t} \sum_{i=1}^{t} \max_h \mathrm{reg}_{h,\mu+\nu}^i \quad \text{(Lemma F.1)} \\
&\leq 2H \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 4H^2 \left(\sum_{t=1}^{T} \frac{1}{t}\alpha_T^t\right)\left(\sum_{i=1}^{T} \max_h \mathrm{reg}_{h,\mu+\nu}^i\right) \\
&\leq 2H \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 24H^2 \log T \cdot \frac{1}{T}\sum_{t=1}^{T} \max_h \mathrm{reg}_{h,\mu+\nu}^t \quad \text{(Lemma A.3)},
\end{aligned}
$$

□

Lemma F.4 (Bound $\mathrm{reg}_{h,\mu+\nu}^t$)**.** *Running Algorithm 10 with $\eta = \frac{1}{16H}$ can guarantee that*

$$
\mathrm{reg}_{h,\mu+\nu}^T(s) \leq \frac{36H^2 \log(A \vee B)}{T}
$$

Proof. From Lemma B.3, substituting $g_t = w_t \overline{Q}_h^t \nu_h^t(s), M_t = w_t \overline{Q}_h^{t-1} \nu_h^{t-1}(s), \eta_t = \frac{\eta}{w_t}$, we can get that

$$
\begin{aligned}
&\sum_{t=1}^{T} w_t \left[\left\langle \mu^\dagger, \overline{Q}_h^t \nu_h^t \right\rangle - \left\langle \mu_h^t, \overline{Q}_h^t \nu_h^t \right\rangle\right] \\
&\leq \frac{w_T \log A}{\eta} + \eta \sum_{t=1}^{T} w_t \|\overline{Q}_h^t \nu_h^t(s) - \overline{Q}_h^{t-1} \nu_h^{t-1}(s)\|_\infty^2 - \sum_{t=2}^{T} \frac{w_t}{8\eta} \|\mu_h^t(\cdot|s) - \mu_h^{t-1}(\cdot|s)\|_1^2 \\
&\implies \mathrm{reg}_{h,\mu}^T \leq \alpha_T^1 \sum_{t=1}^{T} w_{t-1} \left\langle \mu^\dagger, \overline{Q}_h^t \nu_h^t \right\rangle - \left\langle \mu_h^t, \overline{Q}_h^t \nu_h^t \right\rangle \\
&\qquad \leq \frac{\alpha_T \log A}{\eta} + \eta \sum_{t=1}^{T} \alpha_T^t \|\overline{Q}_h^t \nu_h^t(s) - \overline{Q}_h^{t-1} \nu_h^{t-1}(s)\|_\infty^2 - \sum_{t=2}^{T} \frac{\alpha_T^{t-1}}{8\eta} \|\mu_h^t(\cdot|s) - \mu_h^{t-1}(\cdot|s)\|_1^2.
\end{aligned}
$$

Further we have that

$$\|\overline{Q}_h^t \nu_h^t(s) - \overline{Q}_h^{t-1} \nu_h^{t-1}(s)\|_\infty^2 \le 2\|\overline{Q}_h^t - \overline{Q}_h^{t-1}\|_\infty^2 + 2\|\nu_h^t(s) - \nu_h^{t-1}(s)\|_1^2.$$

From the definition of $\overline{Q}_h^t$ we have that

$$\begin{aligned}
\|\overline{Q}_h^t - \overline{Q}_h^{t-1}\| &\le \left\| \sum_{i=1}^{t} \alpha_t^i \overline{Q}_{h+1}^i \nu_{h+1}^i - \sum_{i=1}^{t-1} \alpha_{t-1}^i \overline{Q}_{h+1}^i \nu_{h+1}^i \right\|_\infty \\
&= \left\| \alpha_t \overline{Q}_{h+1}^t \nu_{h+1}^t + (1-\alpha_t) \sum_{i=1}^{t-1} \alpha_{t-1}^i \overline{Q}_{h+1}^i \nu_{h+1}^i - \sum_{i=1}^{t-1} \alpha_{t-1}^i \overline{Q}_{h+1}^i \nu_{h+1}^i \right\|_\infty \\
&= \left\| \alpha_t \overline{Q}_{h+1}^t \nu_{h+1}^t - \alpha_t \sum_{i=1}^{t-1} \alpha_{t-1}^i \overline{Q}_{h+1}^i \nu_{h+1}^i \right\|_\infty \le \alpha_t H.
\end{aligned}$$

Substitute this inequality to the regret bound we have

$$\begin{aligned}
&\mathrm{reg}_{h,\mu}^T(s) \\
&\le \frac{\alpha_T \log A}{\eta} + 2\eta \sum_{t=1}^{T} \alpha_T^t \alpha_t^2 H^2 + 2\eta \sum_{t=1}^{T} \alpha_T^t \|\nu_h^t(s) - \nu_h^{t-1}(s)\|_1^2 - \sum_{t=2}^{T} \frac{\alpha_T^{t-1}}{8\eta} \|\mu_h^t(\cdot|s) - \mu_h^{t-1}(\cdot|s)\|_1^2 \\
&\overset{\text{(Lemma A.3)}}{\le} \frac{\alpha_T \log A}{\eta} + \frac{8\eta H^3}{T} + 2\eta \sum_{t=1}^{T} \alpha_T^t \|\nu_h^t(s) - \nu_h^{t-1}(s)\|_1^2 + - \sum_{t=2}^{T} \frac{\alpha_T^{t-1}}{8\eta} \|\mu_h^t(\cdot|s) - \mu_h^{t-1}(\cdot|s)\|_1^2.
\end{aligned}$$

Similar bound holds for $\mathrm{reg}_{h,\nu}^T$:

$$\mathrm{reg}_{h,\nu}^T(s) \le \frac{\alpha_T \log B}{\eta} + \frac{8\eta H^3}{T} + 2\eta \sum_{t=1}^{T} \alpha_T^t \|\mu_h^t(s) - \mu_h^{t-1}(s)\|_1^2 + - \sum_{t=2}^{T} \frac{\alpha_T^{t-1}}{8\eta} \|\nu_h^t(\cdot|s) - \nu_h^{t-1}(\cdot|s)\|_1^2.$$

Summing $\mathrm{reg}_{h,\mu}^T(s), \mathrm{reg}_{h,\nu}^T(s)$ together we get

$$\begin{aligned}
\mathrm{reg}_{h,\mu+\nu}^T(s) &\le \frac{2\alpha_T \log(A \vee B)}{\eta} + \frac{16\eta H^3}{T} + 16\eta \alpha_T^1 \\
&\quad + \sum_{t=2}^{T} \left(2\eta \alpha_T^t - \frac{\alpha_T^{t-1}}{8\eta} \right) \left(\|\mu_h^t(\cdot|s) - \mu_h^{t-1}(\cdot|s)\|_1^2 + \nu_h^t(\cdot|s) - \nu_h^{t-1}(\cdot|s)\|_1^2 \right).
\end{aligned}$$

Since $\frac{\alpha_T^{t-1}}{\alpha_T^t} \ge \frac{1}{H}$ for $t \ge 2$, by setting $\eta = \frac{1}{16H}$ we can guarantee that $2\eta\alpha_T^t - \frac{\alpha_T^{t-1}}{8\eta} \le 0$, thus

$$\begin{aligned}
\mathrm{reg}_{h,\mu+\nu}^T(s) &\le \frac{2\alpha_T \log(A \vee B)}{\eta} + \frac{16\eta H^3}{T} + 16\eta \alpha_T^1 \le \frac{32H^2 \log(A \vee B)}{T} + \frac{H^2}{T} + \frac{1}{T} \\
&\le \frac{36H^2 \log(A \vee B)}{T}
\end{aligned}$$

□

Given Lemma F.3 and F.4, we are now ready to prove Theorem F.1.

Proof Theorem F.1. From Lemma F.3 and F.4 we have that:

$$\begin{aligned}
\mathrm{NEGap}(\widehat{\mu}^T, \widehat{\nu}^T) &\le 2H \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^T + 24H^2 \log T \cdot \frac{1}{T} \sum_{t=1}^{T} \max_h \overline{\mathrm{reg}}_{h,\mu+\nu}^t \\
&\le 2H \frac{36H^2 \log(A \vee B)}{T} + 24H^2 \log T \cdot \frac{1}{T} \sum_{t=1}^{T} \frac{36H^2 \log(A \vee B)}{t} \\
&\le \frac{936H^4 \log(A \vee B)(\log T + 1)^2}{T},
\end{aligned}$$

which completes the proof. □

G Optimistic policy optimization for general-sum Markov Games

G.1 Preliminaries

Here we formally present the preliminaries for multi-player general-sum Markov games, parallel to the zero-sum setting considered in Section 2.

Multi-player general-sum Markov games We consider tabular episodic (finite-horizon) m-player general-sum Markov games (MGs), which can be denoted as $\mathcal{M}(H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathbb{P}, \{r_i\}_{i=1}^m)$, where H is the horizon length; $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$; $\mathcal{A}_i$ is the action space of the i-th player, with $|\mathcal{A}_i| = A_i$. We use $\mathbf{a} = (a_1, \ldots, a_m) \in \prod_{i\in[m]} \mathcal{A}_i =: \mathcal{A}$ to denote a joint action taken by all players; $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is the transition probabilities, where each $\mathbb{P}_h(s'|s, \mathbf{a})$ gives the probability of transition to state s' from state-action $(s, \mathbf{a})$; $r_i = \{r_{i,h}\}_{h=1}^H$ are the reward functions, where each $r_{i,h}(s, \mathbf{a})$ is the deterministic reward function of the i-th player at time step h and state-action $(s, \mathbf{a})$. In each episode, the MG starts with a deterministic initial state s_1. Then at each time step $1 \le h \le H$, all players observes the state s_h, each player takes an action $a_{i,h} \in \mathcal{A}_i$. Then, each player receive their rewards $r_{i,h}(s_h, \mathbf{a}_h)$, and the game transitions to the next state $s_{t+1} \sim \mathbb{P}_h(\cdot|s_h, \mathbf{a}_h)$.

Policies & value functions A (Markov) policy π_i of the i-th player is a collection of policies $\pi_i = \{\pi_{i,h} : \mathcal{S} \to \Delta_{\mathcal{A}_i}\}_{h=1}^H$, where each $\pi_{i,h}(\cdot|s_h) \in \Delta_{\mathcal{A}_i}$ specifies the probability of taking action $a_{i,h}$ at (h, s_h). We use $\pi = \{\pi_i\}_{i\in[m]}$ to denote a product policy of all players. For any joint policy π (not necessarily a product policy), we use $V_{i,h}^\pi : \mathcal{S} \to \mathbb{R}$ and $Q_{i,h}^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to denote the (i-th player's) value function and Q-function at time step h, respectively, i.e.

$$V_{i,h}^\pi(s) := \mathbb{E}_\pi\Big[\textstyle\sum_{h=h'}^H r_{i,h'}(s_{h'}, \mathbf{a}_{h'}) \mid s_h = s\Big], \tag{43}$$

$$Q_{i,h}^\pi(s, \mathbf{a}) := \mathbb{E}_\pi\Big[\textstyle\sum_{h=h'}^H r_{i,h'}(s_{h'}, \mathbf{a}_{h'}) \mid s_h = s, \mathbf{a}_h = \mathbf{a}\Big]. \tag{44}$$

For notational simplicity, we use the following abbreviation: $[\mathbb{P}_h V](s, \mathbf{a}) := \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,\mathbf{a})} V(s')$ for any value function V. By definition of the value functions and Q-functions, we have the following Bellman equations for all Markov product policy π and all $(i, h, s, \mathbf{a})$:

$$\begin{aligned} Q_{i,h}^\pi(s, \mathbf{a}) &= \big(r_{i,h} + \mathbb{P}_h V_{i,h+1}^\pi\big)(s, \mathbf{a}), \\ V_{i,h}^\pi(s, \mathbf{a}) &= \mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\big[Q_{i,h}^\pi(s, \mathbf{a})\big] = \big\langle Q_{i,h}^\pi(s, \cdot), \pi_h(\cdot|s)\big\rangle. \end{aligned}$$

The goal for the i-th player is to maximize their own value function.

Correlated policy & best response A (general) correlated policy π is any policy for which players may take actions in a history-dependent and correlated fashion. More precisely, a correlated policy π is a mapping $\big\{\pi_h : \Omega \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}}\big\}$, and executes as follows. At the beginning of an episode, a random seed $w \in \Omega$ is sampled from some distribution (also denoted as Ω with slight abuse of notation). Then, at each step h and state s_h, suppose the history so far is $(s_1, \mathbf{a}_1, \ldots, s_{h-1}, \mathbf{a}_{h-1})$. Then, π samples a joint action $\mathbf{a}_h \sim \pi_h(\cdot|\omega, (s_1, \mathbf{a}_1, \ldots, s_{h-1}, \mathbf{a}_{h-1}); s_h)$. This formulation allows each $\pi_h(\cdot|\omega, \cdot, \cdot)$ to be a Markov product policy for any *fixed* ω while still making π to be a correlated policy, due to the correlation introduced by ω.

For any correlated policy π, let π_{-i} denote the (marginal) policy of all but the i-th player. Then, the (i-th) player's best-response value function is

$$V_{i,1}^{\dagger,\pi_{-i}}(s_1) := \max_{\pi_i^\dagger} V_{i,1}^{\pi_i^\dagger\times\pi_{-i}}(s_1),$$

where the max is over all (potentially history-dependent) policy $\pi_i^\dagger$ for the i-th player.

Coarse Correlated Equilibrium (CCE) For general-sum MGs, we consider learning an approximate Coarse Correlated Equilibrium [33, 49] defined as follows.

Definition G.1 (ε-approximate Coarse Correlated Equilibrium)**.** *For any $\varepsilon \ge 0$, a correlated policy π is an ε-approximate Coarse Correlated Equilibrium (ε-CCE) if*

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

Algorithm 12 OFTRL for multi-player general-sum Markov games

1: **Initialize:** $Q_h^0(s, \mathbf{a}) \leftarrow H - h + 1$ for all (h, s, a, b).
2: **for** $t = 1, \ldots, T$ **do**
3: **for** $h = H, \ldots, 1$ **do**
4: Update policies for all $s \in \mathcal{S}$ and $i \in [m]$ by OFTRL

$$\pi_{i,h}^t(a_i|s) \propto_{a_i} \exp\Big((\eta/w_t) \cdot \Big[\textstyle\sum_{j=1}^{t-1} w_j (Q_{i,h}^j \pi_{-i,h}^j)(s, a_i) + w_t (Q_{i,h}^{t-1} \pi_{-i,h}^{t-1})(s, a_i)\Big]\Big). \tag{45}$$

5: Update Q-value for all $(i, s, \mathbf{a}) \in [m] \times \mathcal{S} \times \mathcal{A}$:

$$Q_{i,h}^t(s, \mathbf{a}) \leftarrow (1 - \alpha_t) Q_{i,h}^{t-1}(s, \mathbf{a}) + \alpha_t \big(r_h + \mathbb{P}_h[Q_{i,h+1}^t \pi_{h+1}^t]\big)(s, \mathbf{a}). \tag{46}$$

6: Output policy $\widehat{\pi}^T = \widehat{\pi}_1^T$, where $\widehat{\pi}_1^T$ is defined in Algorithm 13.

Additional notation For any Q function $Q_{i,h}(s, \cdot) : \mathcal{S} \times (\prod_{i=1}^m \mathcal{A}_i) \to \mathbb{R}$ and joint policy $\pi_h(\cdot|s)$, we use $[Q_{i,h}\pi_h](s) := \langle Q_{i,h}(s, \cdot), \pi_h(\cdot|s)\rangle$ for shorthand. Similarly, for any joint policy $\pi_{-i,h}(\cdot|s)$ over all but the i-th player, $[Q_{i,h}\pi_{-i,h}](s, a_i) := \langle Q_{i,h}(s, a_i, \cdot), \pi_{-i,h}(\cdot|s)\rangle$.

G.2 Algorithm and formal statement of result

Algorithm 13 Policy $\widehat{\pi}_h^t$

Require: Product policies $\pi_{h'}^{t'}(\cdot|s') = \prod_{i=1}^m \pi_{i,h'}^{t'}(\cdot|s')$ for all $(h', t', s') \in [H] \times [T] \in \mathcal{S}$.
1: Sample $j \in [t]$ with probability $\mathbb{P}(j = i) = \alpha_t^i$.
2: Play policy π_h^j at the h-th step of the game.
3: Play policy $\widehat{\pi}_{h+1}^j$ for step $h + 1$ onward.

Theorem G.1 (Formal version of Theorem 4)**.** *Suppose Algorithm 12 is run for T rounds. Then the per-state regret can be bounded as follows for some absolute constant $C > 0$:*

$$\mathrm{reg}_h^t \le \overline{\mathrm{reg}}_h^t := C\left[\frac{H \log A_{\max}}{\eta t} + \frac{\eta H^3}{t} + (m-1)^2 \eta^3 H^4\right] \quad \textit{for all } (h, t) \in [H] \times [T].$$

Further, choosing $\eta = (\log A_{\max} \log T/(H^3 T))^{1/4}(m-1)^{-1/2}$, the output policy $\widehat{\pi}^T$ achieves

$$\begin{aligned}\mathrm{CCEGap}(\widehat{\pi}^T) \le \mathcal{O}\Big(& H^{11/4}(\log A_{\max} \log T)^{3/4}\sqrt{m-1} \cdot T^{-3/4} \\ &+ H^{13/4}(\log A_{\max})^{1/4}(\log T)^{5/4}(m-1)^{-1/2} \cdot T^{-5/4}\Big).\end{aligned}$$

Proof overview and remarks The proof of Theorem G.1 also follows by relating the performance of the output policy by per-state regrets via performance difference (Lemma G.2, similar as Theorem 2), and bounding per-state regrets as $\mathrm{reg}_h^t \le \overline{\mathrm{reg}}_h^t := \widetilde{\mathcal{O}}(1/(\eta t) + \eta^3(m-1)^2)$ which gives the theorem. The latter builds upon the fast convergence analysis of OFTRL in multi-player normal-form games [50] as well as additional handling of the changing game rewards, similar as in Theorem 3. Note that the $\widetilde{\mathcal{O}}(T^{-3/4})$ rate here is worse than $\widetilde{\mathcal{O}}(T^{-5/6})$ for the zero-sum setting in Theorem 3. This happens as the fine-grained analysis of OFTRL [9] used there relies critically on the game having two players (for translating between the iterate stabilities and loss stabilities between each other), and becomes infeasible when there are more than 2 players.

We first present some lemmas in Section G.3. The proof of Theorem G.1 is then provided in Section G.4.

G.3 Useful lemmas

We additionally define the V-values maintained by Algorithm 12 as

$$V_{i,h}^t(s) := \sum_{j=1}^t \alpha_t^j \left[Q_{i,h}^j \pi_h^j\right](s) \tag{47}$$

for all $(i,h,t,s) \in [m] \times [H] \times [T] \times \mathcal{S}$, where $Q_{i,h}^t$ and π_h^t are the Q-functions and joint policies maintained within Algorithm 12. Note that by (46), we immediately have

$$\begin{aligned} Q_{i,h}^t(s,\mathbf{a}) &= \sum_{j=1}^t \alpha_t^j \left[r_h + \mathbb{P}_h[Q_{i,h+1}^j \pi_{h+1}^j]\right](s,\mathbf{a}) \\ &= \left(r_h + \mathbb{P}_h\left[\sum_{j=1}^t \alpha_t^j Q_{i,h+1}^j \pi_{h+1}^j\right]\right)(s,\mathbf{a}) = \left(r_h + \mathbb{P}_h V_{i,h+1}^t\right)(s,\mathbf{a}). \end{aligned} \tag{48}$$

We also define the value functions of $\widehat{\pi}_h^t$ and of its best response for any (i,h,t,s) as (see e.g. [49, Definition C.4 & Eq.(8)]):

$$\begin{aligned} V_{i,h}^{\widehat{\pi}_h^t}(s) &:= \mathbb{E}_{\widehat{\pi}_h^t}\left[\sum_{h'=h}^H r_{i,h'} | s_h = s\right], \\ V_{i,h}^{\dagger,\widehat{\pi}_{-i,h}^t}(s) &:= \max_{\pi_{i,h:H}} \mathbb{E}_{\pi_{i,h:H} \times \widehat{\pi}_{-i,h}^t}\left[\sum_{h'=h}^H r_{i,h'} | s_h = s\right]. \end{aligned}$$

Lemma G.1 (Equivalence of value functions)**.** *For Algorithm 12, we have for all $i \in [m]$ and all $(h,s,t) \in [H+1] \times \mathcal{S} \times [T]$ that*

$$V_{i,h}^t(s) = V_{i,h}^{\widehat{\pi}_h^t}(s).$$

Proof. We prove this by backward induction over $h \in [H+1]$. The claim trivially holds for $h = H+1$. Suppose the claim holds for steps $h+1$ onward and all $(s,t) \in \mathcal{S} \times [T]$. For step h and any fixed $(s,t) \in \mathcal{S} \times [T]$, note that

$$\begin{aligned} V_{i,h}^t(s) &= \sum_{j=1}^t \alpha_t^j \left[Q_{i,h}^j \pi_h^j\right](s) \overset{(i)}{=} \sum_{j=1}^t \alpha_t^j \left[(r_h + \mathbb{P}_h V_{i,h+1}^j)\pi_h^j\right](s) \\ &\overset{(ii)}{=} \sum_{j=1}^t \alpha_t^j \left[(r_h + \mathbb{P}_h V_{i,h+1}^{\widehat{\pi}_{h+1}^j})\pi_h^j\right](s) \overset{(iii)}{=} V_{i,h}^{\widehat{\pi}_h^t}(s). \end{aligned}$$

Above, (i) follows by (48); (ii) uses the inductive hypothesis; (iii) uses the definition of the output policy $\widehat{\pi}_h^t$ (cf. Algorithm 13), which samples $j \in [t]$ with probability α_t^j, plays $\pi_h^j(\cdot|s)$, and plays $\widehat{\pi}_{h+1}^j$ for the rest of the game. This proves the case for step h and thus the lemma. □

Define the weighted per-state regrets as

$$\mathrm{reg}_{h,i}^t(s) := \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{j=1}^t \alpha_t^j \left\langle Q_h^j(s,\cdot), (\pi_i^\dagger \times \pi_{-i,h}^j)(\cdot|s) - \pi_h^j(\cdot|s)\right\rangle, \tag{49}$$

$$\mathrm{reg}_h^t := \max_{s \in \mathcal{S}} \max_{i \in [m]} \mathrm{reg}_{h,i}^t(s). \tag{50}$$

The following lemma bounds the difference between the values of the certified policy π_h^t (Algorithm 13) and its best-response.

Lemma G.2 (Recursion of best-response values)**.** *For the policy $\widehat{\pi}_h^t$ defined in Algorithm 13, we have for all $(i,h,t) \in [m] \times [H] \times [T]$ that*

$$\max_{s \in \mathcal{S}} \left(V_{i,h}^{\dagger,\widehat{\pi}_{-i,h}^t}(s) - V_{i,h}^{\widehat{\pi}_h^t}(s)\right) \le \mathrm{reg}_h^t + \sum_{j=1}^t \alpha_t^j \max_{s \in \mathcal{S}} \left(V_{i,h+1}^{\dagger,\widehat{\pi}_{-i,h+1}^j}(s) - V_{i,h+1}^{\widehat{\pi}_{h+1}^j}(s)\right).$$

Proof. Fix $(i,h,t) \in [m] \times [H] \times [T]$. We have for any $s \in \mathcal{S}$ that

$$
\begin{aligned}
& V_{i,h}^{\dagger,\widehat{\pi}^t_{-i,h}}(s) - V_{i,h}^{\widehat{\pi}^t_h}(s) \\
&= \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \left\langle \pi_i^\dagger, \sum_{j=1}^t \alpha_t^j \left[(r_h + \mathbb{P}_h V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}) \pi^j_{-i,h} \right](s,\cdot) \right\rangle - \sum_{j=1}^t \alpha_t^j \left\langle \pi^j_{i,h}, \left[(r_h + \mathbb{P}_h V_{i,h+1}^{\widehat{\pi}^j_{h+1}}) \pi^j_{-i,h} \right](s,\cdot) \right\rangle \\
&\overset{(i)}{\le} \sum_{j=1}^t \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}(s') - V_{i,h+1}^{\widehat{\pi}^j_{h+1}}(s') \right) + \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{j=1}^t \alpha_t^j \left\langle \pi_i^\dagger - \pi^j_{i,h}, \left[(r_h + \mathbb{P}_h V_{i,h+1}^{\widehat{\pi}^j_{h+1}}) \pi^j_{-i,h} \right](s,\cdot) \right\rangle \\
&\overset{(ii)}{=} \sum_{j=1}^t \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}(s') - V_{i,h+1}^{\widehat{\pi}^j_{h+1}}(s') \right) + \underbrace{\max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{j=1}^t \alpha_t^j \left\langle \pi_i^\dagger - \pi^j_{i,h}, \left[(r_h + \mathbb{P}_h V_{i,h+1}^{j}) \pi^j_{-i,h} \right](s,\cdot) \right\rangle}_{\mathrm{reg}^t_{i,h}(s)} \\
&\le \sum_{j=1}^t \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}(s') - V_{i,h+1}^{\widehat{\pi}^j_{h+1}}(s') \right) + \mathrm{reg}^t_h.
\end{aligned}
$$

Above, (i) follows by substituting $V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}$ with $V_{i,h+1}^{\widehat{\pi}^j_{h+1}}$ and paying the additive error; (ii) follows from Lemma G.1. This proves the desired result. □

Lemma G.3 (Guarantee of Algorithm 12 via per-state regrets)**.** *Suppose that the per-state regrets (50) can be upper-bounded as* $\mathrm{reg}^t_h \le \overline{\mathrm{reg}}^t_h$ *for all* $(h,t) \in [H] \times [T]$*, where* $\overline{\mathrm{reg}}^t_h$ *is non-increasing in* t*:* $\overline{\mathrm{reg}}^t_h \ge \overline{\mathrm{reg}}^{t+1}_h$ *for all* $t \ge 1$*. Then running Algorithm 12 will guarantee that*

$$
\mathrm{CCEGap}(\widehat{\pi}^T) \le CH \cdot \frac{1}{T} \sum_{t=1}^T \max_{h \in [H]} \overline{\mathrm{reg}}^t_h. \tag{51}
$$

for all $T \ge 2$*, where* $C > 0$ *is an absolute constant.*

Proof. For any $(h,t) \in [H+1] \times [T]$, define

$$
\delta^t_h := \max_{i \in [m]} \max_{s \in \mathcal{S}} \left(V_{i,h+1}^{\dagger,\widehat{\pi}^j_{-i,h+1}}(s) - V_{i,h+1}^{\widehat{\pi}^j_{h+1}}(s) \right).
$$

Then Lemma G.2 implies the recursive relationship

$$
\delta^t_h \le \mathrm{reg}^t_h + \sum_{j=1}^t \alpha_t^j \delta^j_{h+1}.
$$

(With $\delta^t_{H+1} \equiv 0$ for all $t \in [T]$.) Therefore we can imitate the proof of Lemma C.2 and obtain that, for any $\overline{\mathrm{reg}}^t_h$ such that $\mathrm{reg}^t_h \le \overline{\mathrm{reg}}^t_h$ and $\overline{\mathrm{reg}}^t_h \ge \overline{\mathrm{reg}}^{t+1}_h$,

$$
\delta^t_h \le H c_\beta^{H-1} \cdot \frac{1}{t} \sum_{j=1}^t \max_{h' \in [H]} \overline{\mathrm{reg}}^j_{h'},
$$

where $c_\beta = 1 + 1/H = \sup_{j \ge 1} \sum_{t=j}^\infty \alpha_t^j$ by Lemma A.2(a).

Further, by definition of the output policy $\widehat{\pi}^T = \widehat{\pi}^T_1$ (cf. Algorithm 13), we have

$$
\begin{aligned}
\mathrm{CCEGap}(\widehat{\pi}^T) &= V_{i,1}^{\dagger,\widehat{\pi}^T_{-i,1}}(s_1) - V_{i,1}^{\widehat{\pi}^T_1}(s_1) \le \delta^T_1 \le H c_\beta^{H-1} \cdot \frac{1}{T} \sum_{t=1}^T \max_{h \in [H]} \overline{\mathrm{reg}}^t_h \\
&\le CH \cdot \frac{1}{T} \sum_{t=1}^T \max_{h \in [H]} \overline{\mathrm{reg}}^t_h,
\end{aligned}
$$

where $C \le 3$ as $c_\beta^{H-1} \le (1+1/H)^H \le e \le 3$. This is the desired result. □

G.4 Proof of Theorem G.1

Bounding per-state regret We first bound $\mathrm{reg}_{i,h}^t(s)$ (cf. definition in (49)), i.e. the per-state regret for the i-th player, for any fixed $(i,h,s,t) \in [m] \times [H] \times [S] \times [T]$. This is the main part of this proof.

Throughout this part, we will fix (i,h,s,t), and omit the subscript (h,s) within the policies and Q functions, so that $\pi_{i,h}^t(\cdot|s)$ will be abbreviated as π_i^t, and $Q_{i,h}^t(s,\cdot)$ will be abbreviated as Q_i^t. We will also reload $T \geq 1$ to be any positive integer (instead of the fixed total number of iterations).

We first observe that the update (45) for $\pi_{i,h}^t(\cdot|s)$ is exactly equivalent to the OFTRL algorithm (Algorithm 4) with loss vectors $g_t = w_t(Q_i^t)^\top \pi_{-i}^t$ (understanding $g_0 = 0$ and $Q_i^0 = 0$), prediction vector $M_t = w_t(Q_i^{t-1})^\top \pi_{-i}^{t-1}$, and learning rate $\eta_t = \eta/w_t$. Therefore we can apply the regret bound for OFTRL in Lemma B.3 and obtain for any $T \geq 1$ that

$$\begin{aligned}
&\max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{t=1}^T w_t \left\langle \pi_i^t - \pi_i^\dagger, Q_i^t \pi_{-i}^t \right\rangle = \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{t=1}^T \left\langle \pi_i^t - \pi_i^\dagger, g_t \right\rangle \\
&\leq \frac{\log A_i}{\eta_T} + \sum_{t=1}^T \eta_t \|g_t - M_t\|_\infty^2 - \sum_{t=1}^{T-1} \frac{1}{8\eta_t} \|\pi_i^t - \pi_i^{t+1}\|_1^2 \\
&\leq \frac{\log A_i \cdot w_T}{\eta} + \sum_{t=1}^T \eta w_t \|Q_i^t \pi_{-i}^t - Q_i^{t-1} \pi_{-i}^{t-1}\|_\infty^2 \\
&\leq \frac{\log A_{\max} \cdot w_T}{\eta} + \underbrace{\sum_{t=1}^T 2\eta w_t \left\|Q_i^t - Q_i^{t-1}\right\|_\infty^2}_{\mathrm{I}} + \underbrace{\sum_{t=2}^T 2\eta w_t H^2 \left\|\pi_{-i}^t - \pi_{-i}^{t-1}\right\|_1^2}_{\mathrm{II}}.
\end{aligned} \tag{52}$$

Above, the last inequality uses the fact that $\left\|(Q_i^t - Q_i^{t-1})\pi_{-i}^t\right\|_\infty \leq \left\|Q_i^t - Q_i^{t-1}\right\|_\infty$ for $t \geq 1$, and $\left\|Q_i^{t-1}(\pi_{-i}^t - \pi_{-i}^{t-1})\right\|_\infty \leq H \left\|\pi_{-i}^t - \pi_{-i}^{t-1}\right\|_1$ for $t \geq 2$.

For term I, noticing that $\left\|Q_i^t - Q_i^{t-1}\right\|_\infty \leq \alpha_t H$ by (46), we have

$$\mathrm{I} \leq \sum_{t=1}^T 2\eta w_t \alpha_t^2 H^2 = 2\eta H^2 \sum_{t=1}^T w_t \alpha_t^2.$$

Bounding term II requires the following lemma on the stability of the iterates. The proof can be found in Section G.5.

Lemma G.4 (Stability of iterates)**.** *We have for any $i \in [m]$ and any $t \geq 2$ that (recall the subscripts (h,s) are omitted below):*

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \leq 4\eta H. \tag{53}$$

Consequently,

$$\left\|\pi_{-i}^t - \pi_{-i}^{t-1}\right\|_1 \leq 4(m-1)\eta H. \tag{54}$$

Using Lemma G.4, we have

$$\mathrm{II} \leq \sum_{t=2}^T 2\eta w_t H^2 \cdot 16(m-1)^2 \eta^2 H^2 = 32\eta^3 H^4 (m-1)^2 \sum_{t=2}^T w_t.$$

Plugging the preceding bounds into (52) and using $\alpha_T^1 \cdot w_t = \alpha_T^t$ yields that

$$\mathrm{reg}_{i,h}^T(s) = \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{t=1}^T \underbrace{\alpha_T^t}_{\alpha_T^1 \cdot w_t} \left\langle \pi_i^t - \pi_i^\dagger, Q_i^t \pi_{-i}^t \right\rangle = \alpha_T^1 \max_{\pi_i^\dagger \in \Delta_{\mathcal{A}_i}} \sum_{t=1}^T w_t \left\langle \pi_i^t - \pi_i^\dagger, Q_i^t \pi_{-i}^t \right\rangle$$

$$\leq \frac{\log A_{\max} \cdot \alpha_T^T}{\eta} + 2\eta H^2 \sum_{t=1}^{T} \alpha_T^t \alpha_t^2 + 32\eta^3 H^4 (m-1)^2 \underbrace{\sum_{t=2}^{T} \alpha_T^t}_{\leq 1}$$

$$\overset{(i)}{\leq} \frac{2H \log A_{\max}}{\eta T} + \frac{8\eta H^3}{T} + 32\eta^3 H^4 (m-1)^2$$

$$\leq C\left[\frac{H \log A_{\max}}{\eta T} + \frac{\eta H^3}{T} + \eta^3 H^4 (m-1)^2\right] =: \overline{\mathrm{reg}}_h^T.$$

Above, (i) used the fact that $\alpha_T^T = \alpha_T = (H+1)/(H+T) \leq 2H/T$, and $\sum_{t=1}^{T} \alpha_T^t \alpha_t^2 \leq 4H/T$ by Lemma A.3(c), and $C \leq 32$ is an absolute constant. This proves the per-state regret bounds claimed in Theorem G.1.

Overall policy guarantee Plugging the above per-state regret bounds into Lemma G.3 yields that, the output policy $\widehat{\pi}^T$ of Algorithm 12 achieves

$$\mathrm{CCEGap}(\widehat{\pi}^T) \leq CH \cdot \frac{1}{T} \sum_{t=1}^{T} \max_{h \in [H]} \overline{\mathrm{reg}}_h^T$$

$$\leq \mathcal{O}\left(\frac{H^2 \log A_{\max} \log T}{\eta T} + \frac{\eta H^4 \log T}{T} + \eta^3 H^5 (m-1)^2\right).$$

Choosing $\eta = (\log A_{\max} \log T/(H^3 T))^{1/4} (m-1)^{-1/2}$, the above can be upper bounded as

$$\mathcal{O}\left(H^{11/4} (\log A_{\max} \log T)^{3/4} \sqrt{m-1} \cdot T^{-3/4} + H^{13/4} (\log A_{\max})^{1/4} (\log T)^{5/4} (m-1)^{-1/2} \cdot T^{-5/4}\right),$$

which is the desired result. □

G.5 Proof of Lemma G.4

We first prove (53). By the OFTRL update (45) and the smoothness of exponential weights (Lemma A.5), we have for any $t \geq 2$ that

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \leq 2 \left\|G^t - G^{t-1}\right\|_\infty,$$

where G^t, G^{t-1} are the (weighted) total losses in (45):

$$G^t := \frac{\eta}{w_t}\left[\sum_{j=1}^{t-1} w_j Q_i^j \pi_{-i}^j + w_t Q_i^{t-1} \pi_{-i}^{t-1}\right],$$

$$G^{t-1} := \frac{\eta}{w_{t-1}}\left[\sum_{j=1}^{t-2} w_j Q_i^j \pi_{-i}^j + w_{t-1} Q_i^{t-2} \pi_{-i}^{t-2}\right].$$

Therefore we have

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \leq 2 \left\|G^t - G^{t-1}\right\|_\infty$$

$$\leq 2\eta \left\|\left(\frac{1}{w_t} - \frac{1}{w_{t-1}}\right) \sum_{j=1}^{t-1} w_j Q_i^j \pi_{-i}^j\right\|_\infty + 2\eta \left\|2Q_i^{t-1}\pi_{-i}^{t-1} - Q_i^{t-2}\pi_{-i}^{t-2}\right\|_\infty$$

$$\overset{(i)}{\leq} 2\eta H \cdot \underbrace{\left(\frac{1}{w_{t-1}} - \frac{1}{w_t}\right) \sum_{j=1}^{t-1} w_j}_{\overset{(ii)}{=} H/(H+1) \leq 1} + 2\eta H \leq 4\eta H.$$

Above, (i) uses the fact that $Q_i^j \pi_{-i}^j \in [0, H]$ entry-wise for all $j \geq 1$, and (ii) uses Lemma A.4(b). This proves (53).

The above directly implies (54) by the following bound on the TV distance (or L_1 norm) of product distributions [50]:

$$\left\|\pi_{-i}^t - \pi_{-i}^{t-1}\right\|_1 \leq \sum_{j\neq i} \left\|\pi_j^t - \pi_j^{t-1}\right\|_1 .$$

This completes the proof. □

H Experimental details and additional studies

H.1 Experimental details for Section 5

Details about the game The simulations in Section 5 is performed on the following two-player-zero-sum Markov game with $H = 2$. The state space at $h = 1$ only consists of a single state $\mathcal{S}_1 = \{s_0\}$. The state space at $h = 2$ consists of four different states $\mathcal{S}_2 = \{s_{11}, s_{12}, s_{21}, s_{22}\}$. The action spaces are the same for every state, namely $\mathcal{A} = \{a_1, a_2\}, \mathcal{B} = \{b_1, b_2\}$, i.e. each player has two actions. The transition from $\mathcal{S}_1 \times \mathcal{A} \times \mathcal{B} \to \mathcal{S}_2$ is deterministic, which takes the following form:

$$s_0 \times a_i \times b_j \to s_{ij}, \quad 1 \leq i, j \leq 2.$$

The instantaneous reward r_h depends only on the action (and not the state), i.e., $r_h : \mathcal{A} \times \mathcal{B} \to [0, 1]$, which takes values as (scaled) identity matrices:

$$r_1(\cdot,\cdot) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad r_2(\cdot,\cdot) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{55}$$

Direct calculation yields that the Nash values and policies for this game is given by

$$\begin{aligned} &V_2^\star(s) = 0.5 \ \text{ for } s \in \{s_{11}, s_{12}, s_{21}, s_{22}\}, \\ &Q_1^\star(s_0,\cdot,\cdot) = \begin{bmatrix} 0.6 & 0.5 \\ 0.5 & 0.6 \end{bmatrix}, \quad \mu_1^\star(\cdot|s_0) = \nu_1^\star(\cdot|s_0) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad V_1^\star(s_0) = 0.55. \end{aligned} \tag{56}$$

Initialization All algorithms in Figure 1 use the following initialization (μ^0, ν^0):

$$\begin{aligned} \text{At } h = 1: \ & \mu_1^0(a_1|s_0) = 0.3, \quad \mu_1^0(a_2|s_0) = 0.7, \quad \nu_1^0(b_1|s_0) = 0.7, \quad \nu_1^0(b_2|s_0) = 0.3; \\ \text{At } h = 2: \ & \mu_2^0(a_1|s_{11}) = 0.248, \ \mu_2^0(a_2|s_{11}) = 0.752, \ \nu_2^0(b_1|s_{11}) = 0.248, \ \nu_2^0(b_2|s_{11}) = 0.752; \\ & \mu_2^0(a_1|s_{12}) = 0.500, \ \mu_2^0(a_2|s_{12}) = 0.500, \ \nu_2^0(b_1|s_{12}) = 0.168, \ \nu_2^0(b_2|s_{12}) = 0.832; \\ & \mu_2^0(a_1|s_{21}) = 0.500, \ \mu_2^0(a_2|s_{21}) = 0.500, \ \nu_2^0(b_1|s_{21}) = 0.168, \ \nu_2^0(b_2|s_{21}) = 0.832; \\ & \mu_2^0(a_1|s_{22}) = 0.752, \ \mu_2^0(a_2|s_{22}) = 0.248, \ \nu_2^0(b_1|s_{22}) = 0.248, \ \nu_2^0(b_2|s_{22}) = 0.752. \end{aligned}$$

Standard FTRL (Algorithm 3) and OFTRL (Algorithm 4) by default uses the uniform distribution as the initialization, as it minimizes the (neg)entropy $\Phi(\cdot)$. To make them initialized at μ^0, we change the regularizers for $\mu_h(\cdot|s)$ to be $\mathrm{KL}(\cdot\|\mu_h^0(\cdot|s))$ for the max-player. (And similarly $\mathrm{KL}(\cdot\|\nu_h^0(\cdot|s))$ for the min-player.) Note that our actual initialization above satisfies the property that all its values are bounded within the interval $[0.15, 0.85]$. In particular, $\mathrm{KL}(\mu'\|\mu_h^0(\cdot|s))$ for any other $\mu' \in \Delta_{\mathcal{A}}$ is bounded by $O(\log(1/0.15)) = O(1)$, and thus all the convergence theorems will still hold with this modified regularizer, with at most a larger (multiplicative) constant than with the $\Phi(\cdot)$ regularizer.

Remark on runtime Running all our experiments takes approximately 6.46 hours CPU running time (Intel(R) Core(TM) i5-8250U CPU).[5]

H.2 Additional visualizations for the INPG algorithm

Figure 1 shows that the INPG algorithm (with $\eta = 1/\sqrt{T}$) appears to converge much slower than $O(T^{-1/2})$ (which is the rate for FTRL with $\eta = 1/\sqrt{T}$). Here we present some further understandings of this phenomenon by visualizing the optimization trajectories of the INPG algorithm.

[5] Code: https://github.com/DianYu420376/NeurIPS2022-Policy-Optimization-MGs

(a) Value functions at $h = 2$ (b) Policy at $h = 1$ ($\mu_1^t(a_1|s_0)$)

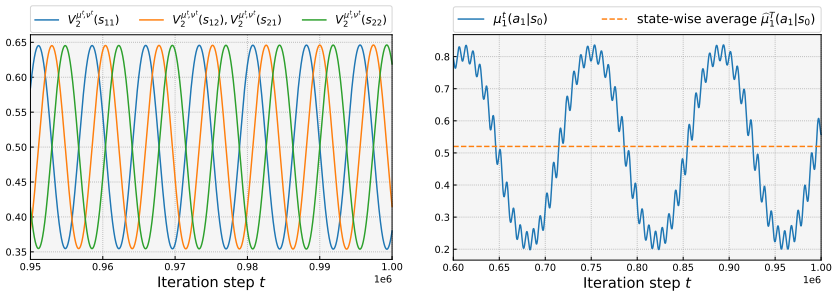

Figure 2: Visualizations of the optimization trajectory of the INPG algorithm **along a single run** with $T = 10^6$ and $\eta = 1/\sqrt{T}$. **(a)** Value functions in the second layer $V_2^{\mu^t,\nu^t}(s)$ for all four states $s \in \{s_{11}, s_{12}, s_{21}, s_{22}\}$ over the last 5×10^4 steps. **(b)** Policy in the first layer, i.e. $\mu_1^t(a_1|s_0)$ over the last 4×10^5 steps. The horizontal line plots the value of the final averaged policy $\widehat{\mu}_1^T(a_1|s_0)$ (where the averaging is over the **entire run** $t \in [T]$).

Figure 2a shows the evolution of the value functions at $h = 2$ over iteration step t, for the last 5×10^4 steps. For all four states, the policy optimization is equivalent to Hedge on the matrix game with identity reward matrix 55, and thus exhibits an expected cyclic behavior and leads to the sinusoidal-like curves shown in Figure 2a. However, due to the choice of our specific initialization (μ^0, ν^0), the four curves behave like the same periodic curves with different *"phases"*.

Figure 2b shows the evolution of the policy at $h = 1$ (specifically, $\mu_1^t(a_1|s_0)$ which is the probability of the max-player taking action a_1) over t, for the last 4×10^5 steps. (The result for the min-player is similar.) The curve also behaves periodically, and appears to be a superposition of two waves, one *main waive* with larger magnitude and period, and another *oscillation* with smaller magnitude and period. Qualitatively, the main wave is caused by the intrinsic cyclic behavior of learning with respect to the (fixed) reward at the $h = 1$, while the oscillation is caused by the changing reward that is backed-up from $h = 2$. Further, as the reward in the second layer has much higher magnitude than the first layer in this game, the oscillation has a non-negligible magnitude.

The horizontal line in Figure 2b plots the final output policy $\widehat{\mu}_1^T(a_1|s_0) \approx 0.52$, which we recall is the average of $\mu_1^t(a_1|s_0)$ over the entire run $t \in [T]$ (cf. Section 5). Note that the unique Nash equilibrium satisfies $\mu_1^\star(a_1|s_0) = 0.5$ (56), and the error $\widehat{\mu}_1^T(a_1|s_0) - \mu_1^\star(a_1|s_0) \approx 0.02$. We suspect that this may be an intrinsic bias caused by the aforementioned correlation between the two layers' learning processes (in particular, the different "phases" of the second-layer's learning over the four states), and may also be the cause of the slow convergence for INPG shown in Figure 1.

H.3 Additional theoretical justifications

INPG as an instantiation of Algorithm 1 Here we show why the instantiation of Algorithm 1 with $\beta_t = 1$ and

$$\mu_h^t(a|s) \propto_a \mu_h^{t-1}(a|s) \exp\big(\eta\big[Q_h^{t-1}\nu_h^{t-1}\big](s)\big), \;\; \nu_h^t(b|s) \propto_b \nu_h^{t-1}(b|s) \exp\Big(-\eta\Big[\big(Q_h^{t-1}\big)^\top \mu_h^{t-1}\Big](s)\Big).$$

considered in Section 5 is equivalent to the Independent Natural Policy Gradient (INPG) algorithm. Indeed, choosing $\beta_t = 1$ in Algorithm 1 ensures that $Q_h^t = Q_h^{\mu^t,\nu^t}$ (the true value function of (μ^t, ν^t)). Therefore, the above update is equivalent to

$$\mu_h^t(a|s) \propto_a \mu_h^{t-1}(a|s) \exp\Big(\eta\Big[Q_h^{\mu^{t-1},\nu^{t-1}}\nu_h^{t-1}\Big](s)\Big), \;\; \nu_h^t(b|s) \propto_b \nu_h^{t-1}(b|s) \exp\Big(-\eta\Big[\Big(Q_h^{\mu^{t-1},\nu^{t-1}}\Big)^\top \mu_h^{t-1}\Big](s)\Big).$$

This is exactly an independent two-player version of the Natural Policy Gradient algorithm (e.g. [1]), where each player plays an NPG algorithm as if they are facing their own Markov Decision Process, with the opponent fixed.

NEGap-Layer-1 **lower bounds** NEGap Here we show NEGap-Layer-1$(\mu, \nu) \le$ NEGap(μ, ν) for any (μ, ν). From the definition of $V_h^\star$ we have that

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

$$\implies V_h^{\mu,\dagger} \le V_h^\star(s) \le V_h^{\dagger,\nu}(s), \quad \forall \mu, \nu.$$

Thus

$$\begin{aligned}
Q_h^\star(s,a,b) &= \left[r_h + \mathbb{P}_h V_{h+1}^\star\right](s,a,b) \le \left[r_h + \mathbb{P}_h V_{h+1}^{\dagger,\nu}\right](s,a,b) = Q_h^{\dagger,\nu}(s,a,b), \\
Q_h^\star(s,a,b) &= \left[r_h + \mathbb{P}_h V_{h+1}^\star\right](s,a,b) \ge \left[r_h + \mathbb{P}_h V_{h+1}^{\mu,\dagger}\right](s,a,b) = Q_h^{\mu,\dagger}(s,a,b) \\
\implies \quad & Q_h^{\mu,\dagger}(s,a,b) \le Q_h^\star(s,a,b) \le Q_h^{\dagger,\nu}(s,a,b).
\end{aligned}$$

Thus for our example

$$\begin{aligned}
\text{NEGap-Layer-1}(\mu,\nu) &= \max_{\mu_1^\dagger} \left[(\mu_1^\dagger)^\top Q_1^\star \nu_1\right](s_0) - \min_{\nu_1^\dagger} \left[\mu_1^\top Q_1^\star \nu_1^\dagger\right](s_0) \\
&\le \max_{\mu_1^\dagger} \left[(\mu_1^\dagger)^\top Q_1^{\dagger,\nu} \nu_1\right](s_0) - \min_{\nu_1^\dagger} \left[\mu_1^\top Q_1^{\mu,\dagger} \nu_1^\dagger\right](s_0) \\
&= V_1^{\dagger,\nu}(s_0) - V_1^{\mu,\dagger}(s_0) = \text{NEGap}(\mu,\nu).
\end{aligned}$$

I Additional details

I.1 Learning setting

In this paper we consider the full-information setting for learning Markov Games formally defined via the following oracle: Given any $h \in [H]$, policies $\mu_{h+1} = \{\mu_{h+1}(\cdot|s) \in \Delta_{\mathcal{A}}\}_{s\in\mathcal{S}}$, $\nu_{h+1} = \{\nu_{h+1}(\cdot|s) \in \Delta_{\mathcal{B}}\}_{s\in\mathcal{S}}$, and V function $V_{h+1} \in \mathbb{R}^{\mathcal{S}}$, we can query the exact value of

$$r_h + \mathbb{P}_h V_{h+1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}\times\mathcal{B}}. \tag{57}$$

Note that our algorithm framework (Algorithm 1) and all its subsequent instantiations can execute under the above oracle: Each iteration of Algorithm 1 makes H queries to this oracle, one for each $h \in [H]$ with value function $V_{h+1} = (\mu_{h+1}^t)^\top Q_{h+1}^t \nu_{h+1}^t$ (cf. (4)).

For the purpose of comparing policy optimization type algorithms, we also consider the following weaker oracle: Given any $h \in [H]$, policies $\mu_h = \{\mu_h(\cdot|s) \in \Delta_{\mathcal{A}}\}_{s\in\mathcal{S}}$, $\nu_h = \{\nu_h(\cdot|s) \in \Delta_{\mathcal{B}}\}_{s\in\mathcal{S}}$ and V function $V_{h+1} \in \mathbb{R}^{\mathcal{S}}$, we can query the expected value of its one-step backup under each player's policy:

$$\begin{cases} (r_h + \mathbb{P}_h V_{h+1})\nu_h \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}, \\ (r_h + \mathbb{P}_h V_{h+1})^\top \mu_h \in \mathbb{R}^{\mathcal{S}\times\mathcal{B}}. \end{cases} \tag{58}$$

This oracle is also considered in [53, Section 3]. It is clear that one query to (57) can implement one query to (58), and thus (58) is a weaker oracle.

I.2 Comparison of algorithms

We first remark that all algorithms considered in Section 3.2, 4, & 5 fall into the unified framework of Algorithm 1 and thus are implementable using the full-information oracle (57).

Second, most algorithms we consider: Nash V-Learning (Example 1), GDA (Example 2), OFTRL (Algorithm 4 & Theorem 3), as well as INPG (Section 5) are policy optimization type algorithms which can be implemented using the weaker oracle (58) (see implementation details in Algorithm 5,6,9). In this paper, we are interested in comparing the rates of the above algorithms, where OFTRL achieves a faster $\widetilde{O}(T^{-5/6})$ rate, and Nash V-Learning and GDA achieve the standard $\widetilde{O}(T^{-1/2})$ rate. (The convergence rate of INPG in theory is still open.)

By contrast, Nash Q-Learning (Example 3) and Nash Policy Iteration (Example 4) are not typically considered as policy optimization algorithms, and they cannot be implemented using (58). Our unified framework (Algorithm 1) allows using (57) and thus encapsulates these two algorithms, but due to the usage of a stronger oracle, their convergence rates are not comparable with the aforementioned policy optimization algorithms.