Multi-Agent Meta-Reinforcement Learning: Sharper Convergence Rates with Task Similarity

Weichao Mao
University of Illinois Urbana-Champaign
weichao2@illinois.edu

Haoran Qiu
University of Illinois Urbana-Champaign
haoranq4@illinois.edu

Chen Wang
IBM Research
chen.wang1@ibm.com

Hubertus Franke
IBM Research
frankeh@us.ibm.com

Zbigniew Kalbarczyk
University of Illinois Urbana-Champaign
kalbarcz@illinois.edu

Ravishankar K. Iyer
University of Illinois Urbana-Champaign
rkiyer@illinois.edu

Tamer Başar
University of Illinois Urbana-Champaign
basar1@illinois.edu

Abstract

Multi-agent reinforcement learning (MARL) has primarily focused on solving a single task in isolation, while in practice the environment is often evolving, leaving many related tasks to be solved. In this paper, we investigate the benefits of meta-learning in solving multiple MARL tasks collectively. We establish the first line of theoretical results for meta-learning in a wide range of fundamental MARL settings, including learning Nash equilibria in two-player zero-sum Markov games and Markov potential games, as well as learning coarse correlated equilibria in general-sum Markov games. Under natural notions of task similarity, we show that meta-learning achieves provable sharper convergence to various game-theoretical solution concepts than learning each task separately. As an important intermediate step, we develop multiple MARL algorithms with initialization-dependent convergence guarantees. Such algorithms integrate optimistic policy mirror descents with stage-based value updates, and their refined convergence guarantees (nearly) recover the best known results even when a good initialization is unknown. To our best knowledge, such results are also new and might be of independent interest. We further provide numerical simulations to corroborate our theoretical findings.

1 Introduction

Many real-world sequential decision-making problems involve multiple agents interacting in a shared environment, a scenario commonly captured by game theory and addressed using multi-agent reinforcement learning (MARL). Existing research in MARL has primarily focused on solving a single task (i.e., a game) independently. In practice, however, one often needs to collectively solve a set of similar tasks due to the dynamically evolving environment. For example, in sponsored search auctions [48], the advertising spaces and search results are dynamic, and each bidder with an active bid will participate in a sequence of related auctions. In multi-robot cooperation [31, 26], the learning agents are often first pre-trained in simplified environments and are then asked to quickly adapt to more complicated ones. In cloud computing [53, 70], a learning-based autoscaling policy needs to achieve fast model adaptation to deal with varied application workloads or constantly evolving cloud infrastructures. All of these intriguing applications call for the development of intelligent multi-agent systems that can continuously build on previous experiences to enhance the learning of new tasks.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Meta-learning, or learning-to-learn [64, 56, 65, 58], is a rapidly developing approach that is particularly suitable for learning in a set of related tasks. In essence, meta-learning studies the use of data from existing tasks to learn representations or model parameters that enable quick adaptation to new tasks. By exploiting the knowledge obtained from prior tasks, the meta-learner can ideally solve an unseen task using much fewer training samples than learning from scratch, especially when the tasks share some inherent similarities. Despite many empirical successes [71, 31, 26], the theoretical results of meta-learning in multi-agent scenarios are still relatively lacking. It remains elusive whether meta-learning can provably expedite the convergence of MARL, and if so, what the proper task similarity assumptions to impose are. In fact, it is even unclear whether a meta-learner converges at all in a highly non-stationary system with loosely-coupled learning agents and diverse task setups.

In this paper, we make an initial attempt toward characterizing some of the central theoretical properties of meta-learning in a wide range of fundamental MARL settings. We focus on the classic model-agnostic meta-learning (MAML) [20] type of algorithms that aim to learn a good initialization for quick adaptation to new tasks. To study the convergence rate of MAML, an important prerequisite is to understand how the convergence of MARL algorithms depends on the quality of policy initialization. However, the convergence guarantees of most existing MARL algorithms are initialization-independent: They fail to track how the sub-optimality of the initial policy propagates during the learning process, and only provide pessimistic guarantees with respect to worst-case initialization. As a crucial intermediate step to meta-MARL, we need to establish refined *initialization-dependent* convergence guarantees for MARL. Our main contributions are thus summarized as follows.

Contributions. 1) For learning Nash equilibria (NE) in two-player zero-sum Markov games, we first propose an MARL algorithm blessed with a refined convergence analysis that explicitly characterizes the dependence on policy initialization (Section 3.1). Our algorithm runs optimistic online mirror descent for policy optimization and performs stage-based value function updates. Even when initialized with random policies, our algorithm still matches the best-known convergence rates in the literature except for an extra logarithmic term. Our algorithm and analysis appear to be new and might be of independent interest. 2) Based on such refined analysis, we show that meta-learning provably achieves faster convergence to NE when learning a sequence of "similar" zero-sum games collectively, where our similarity metric naturally depends on the closeness of the games' NE policies (Section 3.2). 3) For learning NE in Markov potential games (MPGs), we show that a simple refinement of an existing algorithm suffices to provide initialization-dependent guarantees. We establish sharper convergence rates of meta-learning when the MPGs have similar potential functions (Section 4.1). In addition, with a properly chosen policy update rule, we prove the non-asymptotic convergence of the exact MAML algorithm in MPGs (Section 4.2), despite the convoluted learning dynamics of multiple loosely-coupled agents. 4) For learning coarse correlated equilibria (CCE) in general-sum Markov games (Section 5), we analogously start by designing an initialization-dependent MARL algorithm, and then establish the sharper convergence rate of meta-learning under natural similarity metrics. 5) We provide numerical results to corroborate our theoretical findings (Section 6).

Related Work. Gradient-based meta-learning is a simple and effective approach that can be easily applied to any learning problem trained with gradient descent. The seminal MAML method [20] tries to learn a good model parameter initialization that leads to quick model adaptation. Theoretical properties of MAML have been investigated in a series of works [54, 17, 66, 18, 30]. In particular, [17, 30] have established the convergence of MAML to first-order stationarity for non-convex objectives. [18] has designed an unbiased gradient estimator for MAML in reinforcement learning tasks. Various first-order approximations [20, 49, 17] of MAML have been proposed to avoid the heavy computation of the Hessian. Meta-learning has also been studied in online convex optimization [21, 4, 14, 35], where regret bounds have been established under different metrics of task similarity. Another line of research [29, 51, 41] views meta-learning through the lens of task inference, where an RL policy is conditioned on a belief over tasks and perform Bayesian updates through interactions to adapt to different tasks.

MARL has been widely studied under the formulation of stochastic games (i.e., Markov games) [57]. Due to the fundamental difficulty of computing NE in generic games [10], most MARL research has focused on learning NE in games with special structures (such as zero-sum Markov games [67, 2, 69, 3, 12, 68, 7, 72, 73] and Markov potential games [43, 38, 76, 25, 22, 15, 78]) or learning weaker solution concepts such as (coarse) correlated equilibria [40, 44, 59, 33, 47, 16, 13]. The most relevant works are [77, 72], which have studied the convergence of optimistic no-regret learning and smooth value updates in MARL with full-information feedback. For learning NE in MPGs, [38, 76, 15] have studied independent policy gradient methods and established their sample complexity results. These works have focused on learning a single game in isolation but have not considered exploiting the connections between multiple games to expedite the learning process.

Most related to ours, [27] has studied meta-learning in normal-form games. Under different notions of game similarities, [27] has shown faster convergences of meta-learning in zero-sum, general-sum, and Stackelberg games. [75] has investigated no-regret learning in time-varying zero-sum normal-form games. Compared to [27, 75], we consider meta-learning in the more generic and challenging Markov game setup with state transitions. Other related works include meta-learning for regret minimization in a distribution of games [61] and meta-safe RL for quick adaptation in constrained Markov decision processes (CMDPs) under task similarity [34]. Finally, meta-learning has also been empirically applied to many important MARL scenarios, including multi-intersection traffic signal control [71], multi-agent communication with natural language [26], and multi-agent collaboration with first-person pixel observations in open-ended tasks [63].

2 Preliminaries

Markov game. An N-player episodic Markov game is defined by a tuple $\mathbb{G} = (\mathcal{N}, H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{r_i\}_{i=1}^N, P)$, where (1) $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $\mathcal{S}$ is the finite state space; (4) $\mathcal{A}_i$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r_i : [H] \times \mathcal{S} \times \mathcal{A}_{\text{all}} \to [0, 1]$ is the reward function for agent i, where $\mathcal{A}_{\text{all}} = \times_{i=1}^N \mathcal{A}_i$ is the joint action space; and (6) $P : [H] \times \mathcal{S} \times \mathcal{A}_{\text{all}} \to \Delta(\mathcal{S})$ is the transition kernel. The agents interact in an unknown environment for T episodes. Without loss of generality, we make a standard assumption [33, 59] that each episode starts from a fixed initial state s_1. Our results can be easily generalized to the setting where the initial state is sampled from a fixed distribution. At each time step $h \in [H]$, the agents observe the state $s_h \in \mathcal{S}$, and take actions $a_{h,i} \in \mathcal{A}_i, i \in \mathcal{N}$ simultaneously. Agent i then receives its reward $r_{h,i}(s_h, \boldsymbol{a}_h)$, where $\boldsymbol{a}_h = (a_{h,1}, \dots, a_{h,N})$, and the environment transitions to the next state $s_{h+1} \sim P_h(\cdot|s_h, \boldsymbol{a}_h)$. Let $S = |\mathcal{S}|$, $A_i = |\mathcal{A}_i|, \forall i \in \mathcal{N}$, and $A_{\max} = \max_{i \in \mathcal{N}} A_i$.

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

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

For agent i, a policy $\pi_i^{\dagger}$ is a *best response* to π_{-i} if $V_{1,i}^{\pi_i^{\dagger}, \pi_{-i}}(s_1) = \sup_{\pi_i} V_{1,i}^{\pi_i, \pi_{-i}}(s_1)$. A joint (product) policy $\pi = (\pi_i, \pi_{-i}) \in \Pi$ is a *Nash equilibrium* (NE) if π_i is a best response to π_{-i} for all $i \in \mathcal{N}$. Similarly, for any $\varepsilon > 0$, a joint policy $\pi = (\pi_i, \pi_{-i})$ is an ε-approximate NE if $V_{1,i}^{\pi_i, \pi_{-i}}(s_1) \geq V_{1,i}^{\pi_i^{\dagger}, \pi_{-i}}(s_1) - \varepsilon, \forall i \in \mathcal{N}$.

Correlated policy and coarse correlated equilibrium. We define $\pi = \{\pi_h : \mathbb{R} \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S} \to \Delta(\mathcal{A})\}_{h \in [H]}$ as a (non-Markov) *correlated policy*, where for each $h \in [H]$, π_h maps from a coordination device $z \in \mathbb{R}$ and a history of length $h - 1$ to a distribution over the joint action space. Let π_i and π_{-i} be the proper marginal distributions of π whose outputs are restricted to $\Delta(\mathcal{A}_i)$ and $\Delta(\mathcal{A}_{-i})$, respectively. The value functions for non-Markov correlated policies at step $h = 1$ are defined in a similar way as for product policies. Given the PPAD-hardness of calculating NE in general [11], people often study a relaxed solution concept named coarse correlated equilibrium (CCE), which allows possible correlations in the policies: In particular, for any $\varepsilon > 0$, a *correlated policy* $\pi = (\pi_i, \pi_{-i})$ is an ε-approximate CCE if $V_{1,i}^{\pi_i, \pi_{-i}}(s_1) \geq V_{1,i}^{\pi_i^{\dagger}, \pi_{-i}}(s_1) - \varepsilon, \forall i \in \mathcal{N}$.

Two-player zero-sum Markov game. An important special case of Markov games is (two-player) zero-sum Markov games, where there are two players ($N = 2$) with exactly opposite rewards

($r_1 = -r_2$). In a zero-sum game, we simply use r, V, and Q to denote the reward and (Q-)value functions for the max-player, i.e., agent 1. Correspondingly, the min-player has $-r, -V$, and $-Q$. For notational convenience, we denote the action space for the max-player (resp. min-player) by $\mathcal{A}$ (resp. $\mathcal{B}$), and let $A = |\mathcal{A}|$, $B = |\mathcal{B}|$. We also write their policies (π_1, π_2) as (μ, ν) for short. In zero-sum games, it is known that although the NE policy $(\mu^\star, \nu^\star)$ may not be unique, all the NE have the same values. We use $V_h^\star$ and $Q_h^\star$ to denote the NE value function and the NE Q-function. For any fixed $(h, s) \in [H] \times \mathcal{S}$ and an arbitrary function $Q : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$, we may consider $Q(s, \cdot, \cdot)$ as an $A \times B$ matrix. Then, for any policy pair (μ_h, ν_h) at step $h \in [H]$, we can write in shorthand:

$$\left[\mu_h^\top Q \nu_h\right](s) := \mathbb{E}_{a \sim \mu_h(\cdot|s), b \sim \nu_h(\cdot|s)}[Q(s, a, b)] = \langle \mu_h, Q\nu_h \rangle (s),$$
$$\left[\mu_h^\top Q\right](s, \cdot) := \mathbb{E}_{a \sim \mu_h(\cdot|s)}[Q(s, a, \cdot)], \text{ and } [Q\nu_h](s, \cdot) := \mathbb{E}_{b \sim \nu_h(\cdot|s)}[Q(s, \cdot, b)].$$

Given the transition function P and an arbitrary function $V : \mathcal{S} \to \mathbb{R}$, we define

$$[P_h V](s, a, b) := \mathbb{E}_{s' \sim P_h(\cdot|s,a,b)}[V(s')].$$

The Bellman equations can hence be rewritten more succinctly as

$$V_h^{\mu,\nu}(s) = \left[\mu_h^\top Q_h^{\mu,\nu} \nu_h\right](s), \text{ and } Q_h^{\mu,\nu}(s, a, b) = r_h(s, a, b) + \left[P_h V_{h+1}^{\mu,\nu}\right](s, a, b).$$

Markov potential game. Another important class of games is Markov potential games [43, 39, 76]. MPGs cover Markov teams [36], a fully cooperative setting where all agents share the same rewards. A Markov game is an MPG if there exists a global potential function $\Phi : \Pi \times \mathcal{S} \to [0, \Phi_{\max}]$ that can capture the variations of the agents' individual values: Specifically, $\forall i \in \mathcal{N}$ and $s \in \mathcal{S}$,

$$\Phi_s(\pi_i, \pi_{-i}) - \Phi_s(\pi_i', \pi_{-i}) = V_{1,i}^{\pi_i, \pi_{-i}}(s) - V_{1,i}^{\pi_i', \pi_{-i}}(s), \forall \pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}.$$

Throughout the paper, we consider the classic full-information feedback setting [23, 8, 62, 68, 7], where the players are assumed to have exact information of the consequences of each of their candidate actions. In the case of zero-sum games, this implies that for any (h, s), the max-player and min-player can query $[Q_h \nu_h](s, \cdot)$ and $[\mu_h^\top Q_h](s, \cdot)$, respectively. Our meta-learning results can be easily extended to the stochastic bandit feedback setting using standard techniques as in [3, 44, 33, 15].

Meta-learning. Let $\mathcal{G} = \{\mathbb{G}^k\}$ be a set of different Markov games. Each game is defined by $\mathbb{G}^k = (\mathcal{N}, H, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{r_i^k\}_{i=1}^N, P^k)$, where we assume without loss of generality that the games share the same agent set and state & action spaces, but can have different transition and reward functions. Most of our results are established in the online learning setting where we encounter a sequence of K games $(\mathbb{G}^1, \ldots, \mathbb{G}^K)$ one by one. To achieve faster convergence, the learning agents should use the knowledge obtained from previous games to expedite the learning process in future games.

The underlying principle of MAML [20] is to learn a good initialization such that running a few training steps from this initialization can lead to well-performing model parameters on any new task. An MAML-type algorithm in the context of RL typically involves two nested stages. The inner stage (or "base algorithm") ψ performs T iterations of policy updates to optimize for an individual task $\mathbb{G}^k$:

$$\pi^{k,t} \leftarrow \psi(\pi^{k,t-1}; \mathbb{G}^k), \forall t \in [T]. \tag{1}$$

When task $\mathbb{G}^k$ is completed, the outer stage (or "meta-algorithm") Ψ learns to form a good initialization $\pi^{k+1,0}$ for a new task $\mathbb{G}^{k+1}$ using all the knowledge obtained from all previous tasks:

$$\pi^{k+1,0} \leftarrow \Psi(\{\pi^{k',t}\}_{k' \in [k], t \in [T]}; \mathbb{G}^1, \ldots, \mathbb{G}^k). \tag{2}$$

In this paper, we seek to properly instantiate both the base algorithm ψ and the meta-algorithm Ψ for a variety of MARL problems. We aim to show that a proper design of the meta-learning procedure (ψ, Ψ) can largely reduce the number of iterations T required to find NE or CCE in a new game.

3 Meta-Learning for Two-Player Zero-Sum Markov Games

In this section, we study meta-learning for Nash equilibria in zero-sum Markov games, where players are fully competitive. Since MAML-type algorithms seek to learn a good initialization for quick adaptation, it is crucial to explicitly characterize how the convergence behavior of an MARL algorithm depends on the initial policy. To our best knowledge, such results are not directly achievable using existing algorithms. For this reason, in Section 3.1, we start by proposing a new base algorithm (1) for zero-sum Markov games that has a refined initialization-dependent convergence guarantee. Based on that, we present our meta-algorithm (2) in Section 3.2 and establish its sharper convergence rates.

3.1 Initialization-Dependent Convergence in an Individual Zero-Sum Markov Game

Algorithm 1 presents our optimistic online mirror descent algorithm with stage-based value updates for learning NE in a zero-sum Markov game. To establish initialization-dependent convergence, Algorithm 1 performs optimistic online mirror descent (OMD) [55, 62] for policy updates (Lines 5 and 6), in contrast to the popular optimistic follow the regularized leader (FTRL) method in recent MARL policy optimization [77, 72]. We choose the negative entropy as our regularizer R, in which case the Bregman divergence $D_R(\cdot,\cdot)$ reduces to the Kullback–Leibler divergence and optimistic OMD becomes an optimistic variant of the classic multiplicative weights update (MWU) algorithm.

Algorithm 1: Optimistic Online Mirror Descent for Zero-Sum Markov Games

---

1 **Input:** Initial policies $\tilde{\mu} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A})$ and $\tilde{\nu} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{B})$;

2 Set stage index $\tau \leftarrow 1$, $t_\tau^{\text{start}} \leftarrow 1$, and $L_\tau \leftarrow H$;

3 **Initialize:** $\mu_h^0 = \hat{\mu}_h^0 \leftarrow \tilde{\mu}_h^1$, $\nu_h^0 = \hat{\nu}_h^0 \leftarrow \tilde{\nu}_h^1$, and $Q_h^\tau \leftarrow \mathbf{0}, \forall h \in [H]$;

4 **for** *iteration* $t \leftarrow 1$ *to* T **do**

5   **Auxiliary policy update:** for each step $h \in [H]$ and state $s \in \mathcal{S}$:

$$\hat{\mu}_h^t(\cdot|s) \leftarrow \operatorname*{argmax}_{\hat{\mu}\in\Delta(\mathcal{A})} \eta \left\langle \hat{\mu}, [Q_h^\tau \nu_h^{t-1}](s,\cdot)\right\rangle - D_R(\hat{\mu}, \hat{\mu}_h^{t-1}(\cdot|s));$$

$$\hat{\nu}_h^t(\cdot|s) \leftarrow \operatorname*{argmax}_{\hat{\nu}\in\Delta(\mathcal{B})} \eta \left\langle \hat{\nu}, [(\mu_h^{t-1})^\top Q_h^\tau](s,\cdot)\right\rangle - D_R(\hat{\nu}, \hat{\nu}_h^{t-1}(\cdot|s));$$

6   **Policy update:** for each step $h \in [H]$ and state $s \in \mathcal{S}$:

$$\mu_h^t(\cdot|s) \leftarrow \operatorname*{argmax}_{\mu\in\Delta(\mathcal{A})} \eta \left\langle \mu, [Q_h^\tau \nu_h^{t-1}](s,\cdot)\right\rangle - D_R(\mu, \hat{\mu}_h^{t}(\cdot|s));$$

$$\nu_h^t(\cdot|s) \leftarrow \operatorname*{argmax}_{\nu\in\Delta(\mathcal{B})} \eta \left\langle \nu, [(\mu_h^{t-1})^\top Q_h^\tau](s,\cdot)\right\rangle - D_R(\nu, \hat{\nu}_h^{t}(\cdot|s));$$

7   **if** $t - t_\tau^{start} + 1 \geq L_\tau$ **then**

8         $t_\tau^{\text{end}} \leftarrow t, t_{\tau+1}^{\text{start}} \leftarrow t+1, L_{\tau+1} \leftarrow \lfloor (1+1/H) L_\tau \rfloor$;

9   **Value update:** for each $h \in [H], s \in \mathcal{S}, a \in \mathcal{A}, b \in \mathcal{B}$:

$$Q_h^{\tau+1}(s,a,b) \leftarrow \frac{1}{L_\tau} \sum_{t'=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left(r_h + P_h[(\mu_{h+1}^{t'})^\top Q_{h+1}^\tau (\nu_{h+1}^{t'})]\right)(s,a,b);$$

10         $\tau \leftarrow \tau + 1$; $\mu_h^t = \hat{\mu}_h^t \leftarrow \tilde{\mu}_h^\tau$, $\nu_h^t = \hat{\nu}_h^t \leftarrow \tilde{\nu}_h^\tau, \forall h \in [H]$;

11 **Output policy:** $\bar{\mu}_h(\cdot|s) = \frac{1}{T}\sum_{t=1}^T \mu_h^t(\cdot|s)$ and $\bar{\nu}_h(\cdot|s) = \frac{1}{T}\sum_{t=1}^T \nu_h^t(\cdot|s), \forall s \in \mathcal{S}, h \in [H]$.

In order to establish convergence to (approximate) NE, we need to show that our optimistic OMD policy updates achieve "no regret" with respect to the value estimate sequence at each state, i.e., to upper bound (3). If we were to use the celebrated $\alpha_t = \frac{H+1}{H+t}$ learning rate [32] to update the value function estimates, we will inevitably need to show a no-weighted-regret guarantee for optimistic OMD, because such a time-varying learning rate assigns non-uniform weights to each history step. However, incorporating OMD with a dynamic learning rate is known to be challenging and can easily lead to linear regret [50]. While a stabilization technique [19] has been introduced to tackle this challenge, we take a different route by resorting to an alternative value update method, namely *stage-based* value updates [79]. Specifically, we divide the total T iterations into multiple stages and only update our value estimates at the end of a stage (Line 9). We let the lengths of the stages grow exponentially at a rate of $(1 + 1/H)$ (Line 8) [79, 45]. The exponential growth ensures that the total T iterations can be covered by a small number of stages, while the $(1 + 1/H)$ growth rate guarantees that the value estimation error does not blow up during the H steps of recursion (Lemma 9). Compared with the incremental $\alpha_t = \frac{H+1}{H+t}$ update rule that modifies the value estimates at every step, stage-based updates are more stationary and allow us to assign uniform weights to each history step. This leads to a simpler no(-average)-regret problem [47] that can be easily addressed by (optimistic) OMD.

We introduce a few notations before presenting the convergence analysis of Algorithm 1. Let $\tau(t)$ denote the index of the stage that iteration t belongs to. We denote by $\bar{\tau}$ the total number of stages, i.e., $\bar{\tau} := \tau(T)$. For any $(\tau, h, s) \in [\bar{\tau}] \times [H] \times \mathcal{S}$, define the per-state regrets for the max-player as

$$\mathrm{reg}_{h,1}^{\tau}(s) := \max_{\mu_h^{\tau,\dagger}(\cdot|s) \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^\tau \nu_h^j \right\rangle (s). \tag{3}$$

The per-state regret $\mathrm{reg}_{h,2}^{\tau}(s)$ for the min-player can be defined symmetrically (see (14) in Appendix B). We define the maximal regret (over the states and the two players) as $\mathrm{reg}_h^\tau := \max_{s \in \mathcal{S}} \max_{i=1,2}\{\mathrm{reg}_{h,i}^\tau(s)\}$. An upper bound for the per-state regrets is provided in Lemma 8 of Appendix B, which is useful in the analysis of Algorithm 1. We use the standard notion of

$$\text{NE-gap}(\mu, \nu) := V_1^{\dagger,\nu}(s_1) - V_1^{\mu,\dagger}(s_1)$$

to measure the optimality of a policy pair (μ, ν). The initialization-dependent convergence rate of Algorithm 1 is as follows.

Theorem 1. *If Algorithm 1 is run on a two-player zero-sum Markov game for T iterations with a learning rate $\eta \leq 1/(8H^2)$, the output policy pair $(\bar{\mu}, \bar{\nu})$ satisfies:*

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{192H^3}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \max_s \left(D_R(\mu_h^{\tau,\dagger}(\cdot|s), \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}(\cdot|s), \tilde{\nu}_h^\tau(\cdot|s)) \right).$$

In addition, if the players' policies are initialized to be uniform policies, i.e., $\tilde{\mu}_h^\tau(\cdot|s) = \mathbf{1}/A$ and $\tilde{\nu}_h^\tau(\cdot|s) = \mathbf{1}/B, \forall s \in \mathcal{S}, \tau \in [\bar{\tau}], h \in [H]$, we further have

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{768H^5 \log T \log(AB)}{T}. \tag{4}$$

Compared to existing results [77, 72], Theorem 1 directly associates the convergence rate with the quality of the initial policy $(\tilde{\mu}, \tilde{\nu})$. Even when a good policy initialization is unknown and the algorithm is initialized with uniformly random policies, our convergence rate in (4) still matches the best-known result in the literature [72] except for an extra factor of $O(\log T)$. When suppressing the logarithmic terms, Theorem 1 immediately implies that for any $\varepsilon > 0$, Algorithm 1 takes no more than $T = \widetilde{O}(H^5/\varepsilon)$ steps to learn an ε-approximate NE in an individual zero-sum Markov game.

3.2 Sharper Convergence with Meta-Learning

Having settled the initialization-dependent convergence in a zero-sum game, we proceed to show how meta-learning can learn a set of related games collectively and more rapidly. We consider an online setting with a sequence of K games $\mathcal{G} = (\mathbb{G}^1, \ldots, \mathbb{G}^K)$. For the max-player, let $\tilde{\mu}^k$ and $\bar{\mu}^k$, respectively, denote the initial policy and output policy of Algorithm 1 on game $\mathbb{G}^k$. By putting together $\mu_h^{\tau,\dagger}(\cdot|s)$ over all $(\tau, h, s) \in [\bar{\tau}] \times [H] \times \mathcal{S}$, we let $\mu^{k,\dagger} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A})$ denote the best fixed policies in hindsight on $\mathbb{G}^k$. Define $\tilde{\nu}^k, \bar{\nu}^k$ and $\nu^{k,\dagger}$ analogously for the min-player. Let $\mu^\star = \frac{1}{K}\sum_{k=1}^K \mu^{k,\dagger}$ and $\nu^\star = \frac{1}{K}\sum_{k=1}^K \nu^{k,\dagger}$ be the empirical averages of the best response policies. To ensure that the knowledge gained from previous games is useful for learning future tasks, we need to impose some similarity assumptions on the games $\mathcal{G}$. We consider the following similarity metric:

$$\Delta_{\mu,\nu} := \sum_{k=1}^{K} \left(\mathrm{KL}\left(\mu^{k,\dagger} \| \mu^\star\right) + \mathrm{KL}\left(\nu^{k,\dagger} \| \nu^\star\right) \right).$$

Intuitively, since $\{\nu^{k,t}\}_{t \in [T]}$ converges to an equilibrium policy for $\mathbb{G}^k$ when T is large, the best fixed responses $\mu^{k,\dagger}$ can be considered as an approximation of the max-player's NE policy on $\mathbb{G}^k$. In this sense, $\Delta_{\mu,\nu}$ essentially measures the distances between the NE policies of different games. It considers a set of games $\mathcal{G}$ to be "similar" if their NE policies lie in a close neighborhood of each other. We remark that there might be multiple NE policies (with the same value) in a zero-sum game, and $\Delta_{\mu,\nu}$ only takes into account the NE policy pairs that Algorithm 1 actually delivers.

Our meta-learning procedure proceeds as follows: Within each game $\mathbb{G}^k$, we run Algorithm 1 as our base algorithm (1) to find a NE of $\mathbb{G}^k$. In a new game $\mathbb{G}^{k+1}$, the initial policy of Algorithm 1 is given by the following meta-updates in the outer loop (2), which essentially averages the best response policies of the previous tasks under α-greedy parameterization:

$$\tilde{\mu}^{k+1} = \frac{1}{k}\sum_{k'=1}^{k}[\mu^{k',\dagger}]_\alpha, \quad \text{and} \quad \tilde{\nu}^{k+1} = \frac{1}{k}\sum_{k'=1}^{k}[\nu^{k',\dagger}]_\alpha. \tag{5}$$

In particular, for any vector $\mathbf{x} \in \mathbb{R}^d$, we define its α-greedy parameterization $[\mathbf{x}]_\alpha := (1-\alpha)\mathbf{x} + \frac{\alpha}{d}\mathbf{1}$ to be a weighted average with a uniform vector $\mathbf{1}/d \in \mathbb{R}^d$ of a proper dimension, where $\alpha \in (0, 1/2)$. Since $\mu^{k,\dagger}$ denotes a set of vectors, we apply the operator $[\cdot]_\alpha$ element-wise to each of the vectors. The reason for using α-greedy is mainly technical: $\mathrm{KL}(\cdot\|\cdot)$ is not Lipschitz continuous near the boundary of the probability simplex, and α-greedy parameterization helps to stay α-distance away from the boundary. We are now ready to present our sharper convergence rates for meta-learning.

Theorem 2. *In a sequence of K two-player zero-sum Markov games, if Algorithm 1 is run for T iterations as the base algorithm and* (5) *with $\alpha = 1/\sqrt{K}$ as the meta-updates, we have*

$$\frac{1}{K}\sum_{k=1}^{K}\text{NE-gap}(\bar{\mu}^k, \bar{\nu}^k) \leq \frac{192H^5}{T}\left(\frac{\Delta_{\mu,\nu}}{KH^2} + \frac{10(A+B)\log K}{\sqrt{K}H^2} + \frac{16\log T\log(ABK)}{\sqrt{K}}\right). \tag{6}$$

Consequently, for any $\varepsilon > 0$, $T = \tilde{O}(\frac{H^3}{\varepsilon}(\frac{\Delta_{\mu,\nu}}{K} + \frac{A+B+H^2}{\sqrt{K}}))$ steps on average suffice to find an ε-approximate Nash equilibrium in each game.

When the number of games K is large, the last two terms on the RHS of (6) become negligible. Hence, compared to the best-known results $\widetilde{O}(H^5/T)$ of learning each game individually, Theorem 2 implies a significantly sharper convergence rate when the games are similar, i.e., when $\Delta_{\mu,\nu} \ll KH^2$.

4 Meta-Learning for Markov Potential Games

In this section, we study meta-learning for NE in Markov potential games. We show that a straightforward refinement to the analysis of an existing algorithm [15] provides initialization-dependent bounds. Building on it, in Section 4.1, we first investigate the sharper convergence of meta-learning in a sequence of similar MPGs. Further, since there exists an optimization objective universally agreed on by all the players in an MPG (i.e., the potential function), we can formulate the meta-learning problem in the same way as MAML [20]. In Section 4.2, by choosing a proper base algorithm, we establish the non-asymptotic convergence of MAML in the highly non-stationary multi-agent scenario, without even imposing any smoothness assumptions as in existing works [17, 18, 30].

4.1 Sharper Rates in Similar Games

To be consistent with existing results in the literature, in this section, we consider an infinite-horizon γ-discounted reward setting for MPGs [43, 39, 76, 15]. A detailed description of the setup is provided in Appendix C for completeness. Equivalent results for the finite-horizon episodic setting (as we defined in Section 2) can be derived in a straightforward way. We choose an existing state-of-the-art algorithm, namely independent projected Q-descent [15], as our base algorithm (1). Specifically, in an MPG $\mathbb{G}^k$, each agent independently runs policy gradient ascents to update its own policy for T iterations:

$$\pi_i^{k,t}(\cdot|s) \leftarrow \mathrm{Proj}_{\Delta(\mathcal{A}_i)}\left(\pi_i^{k,t-1}(\cdot|s) + \alpha\bar{Q}_i^{\pi^{k,t-1}}(s,\cdot)\right), \forall t \in [T], \tag{7}$$

where $\bar{Q}_i^\pi$ is the "averaged" Q-function formally defined in Appendix C. Let $\Phi(\cdot\,;\mathbb{G}^k)$ denote the potential function of $\mathbb{G}^k$. Through a simple refinement of the analysis in [15], we can establish the following initialization-dependence bound for our base algorithm (7).

Proposition 1. *(Theorem 1 in [15]) Suppose that all players in a Markov potential game $\mathbb{G}^k$ run independent projected Q-descent* (7) *for T iterations with $\alpha \leq \frac{(1-\gamma)^4}{8\kappa^3 NA_{\max}}$. Then, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\text{NE-gap}(\pi^{k,t}) \leq \sqrt{\frac{\kappa(\mathbb{G}^k)(\Phi(\pi^{k,T};\mathbb{G}^k) - \Phi(\pi^{k,0},\mathbb{G}^k))}{\alpha T(1-\gamma)^2}},$$

where $\kappa(\mathbb{G}^k)$ is the standard distribution mismatch coefficient for $\mathbb{G}^k$ formally defined in Appendix C.

Proposition 1 immediately implies that if we learn each MPG individually, it takes $T = O(\frac{NA_{\max}\kappa^4\Phi_{\max}}{(1-\gamma)^6\varepsilon^2})$ steps to find an ε-approximate NE. To show the effectiveness of meta-learning, we consider the following similarity metric for a sequence of K games, which measures the maximal point-wise deviations of the potential functions:

$$\Delta_\Phi := \sum_{k=1}^{K-1} \max_\pi \left(\Phi(\pi; \mathbb{G}^k) - \Phi(\pi; \mathbb{G}^{k+1})\right). \tag{8}$$

As for the meta-updates, we simply instantiate (2) as $\pi_i^{k,0} \leftarrow \pi_i^{k-1,T}$, which lets each agent play the converged policy in the previous game. The intuition is that after running T steps on $\mathbb{G}^{k-1}$, the agents will converge to an approximate NE policy of $\mathbb{G}^{k-1}$. Since (8) requires the potential functions to be close, the converged policy $\pi^{k-1,T}$ should serve as a good starting point to search for NE in $\mathbb{G}^k$. We formally characterize such an intuition in the following theorem, which shows the sharper convergence of meta-learning in a large set of similar MPGs (i.e., when K is large and Δ_Φ is small):

Theorem 3. *In a sequence of K Markov potential games, if* (7) *is run for T iterations as the base algorithm and $\pi_i^{k,0} \leftarrow \pi_i^{k-1,T}$ as the meta-updates, then, for any $\varepsilon > 0$, $T = O(\frac{NA_{\max}\kappa^4(\Phi_{\max}+\Delta_\Phi)}{K(1-\gamma)^6\varepsilon^2})$ steps on average suffice to find an ε-approximate Nash equilibrium in each game.*

4.2 Convergence to MAML Objective

In this subsection, we study meta-learning for MPGs under exactly the same formulation as in the seminal work of MAML [20]. Let $\mathcal{G} = \{\mathbb{G}^j\}$ be a set of different MPGs, where the games are now drawn from a fixed distribution p that we can sample from. We consider parametric policy classes where agent i's policy is parameterized by $\theta_i = \{\theta_i(a_i|s) \in \mathbb{R}\}_{s\in\mathcal{S}, a_i\in\mathcal{A}_i}$. We focus on softmax parameterization where

$$\pi_{\theta_i}(a_i|s) = \frac{\exp(\theta_i(a_i|s))}{\sum_{a_i'\in\mathcal{A}_i}\exp(\theta_i(a_i'|s))}.$$

Let $\zeta(\cdot\,; \mathbb{G})$ denote the operator of performing one step of policy gradient update on game $\mathbb{G}$, i.e., $\zeta(\theta; \mathbb{G}) := \theta + \alpha\nabla\Phi(\theta; \mathbb{G})$, where $\alpha > 0$ is the learning rate. The T-step MAML objective [20, 18, 30] can be formulated as

$$\max_{\theta\in\Theta} F_T(\theta) := \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\zeta(\ldots(\zeta(\theta; \mathbb{G}))\ldots); \mathbb{G}\right)\right], \tag{9}$$

where $\theta = (\theta_1, \ldots, \theta_N) \in \Theta$, and the operator $\zeta(\cdot\,; \mathbb{G})$ is applied T times. Intuitively, MAML tries to find a good parameter initialization from which running T steps of gradient ascents on any new task $\mathbb{G}$ leads to well-performing policy parameters.

Similar to Section 2, the MAML procedure consists of two nested stages. For the inner stage (1), we let each agent independently run T steps of policy gradient ascents to update its policy parameter $\theta_i^{(t)}$ on each encountered MPG. It is known (Theorem 5 of [78]) that $T = O(1/\varepsilon^2)$ steps will find an ε-approximate NE for each individual MPG. For the outer stage (2), MAML directly performs gradient ascents with respect to the meta-objective (9). The gradient of F_T can be written in closed-form as

$$\nabla F_T(\theta) = \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Big(\prod_{t=0}^{T-1}\left(I + \alpha\nabla^2\Phi(\theta^{(t)}; \mathbb{G})\right)\Big)\nabla\Phi(\theta^{(T)}; \mathbb{G})\right]. \tag{10}$$

A detailed discussion of MAML and its instantiation in our problem are provided in Appendix D. Most importantly, Appendix D shows that both the policy gradient $\nabla\Phi(\theta)$ and the policy Hessian $\nabla^2\Phi(\theta)$ can be written in closed-form, which allows us to construct unbiased estimators of (10) from samples. Despite the fact that the learning agents update their policies independently in an intertwined multi-agent system, our next result shows that the MAML updates converge to a stationary point of the meta-objective (9) in a non-asymptotic manner. A key step of the proof is to prove (rather than assume, as in existing works [17, 30]) that the meta-objective is Lipschitz smooth in the policy parameter θ. The smoothness constant can also be written in a closed form (Lemma 14).

Theorem 4. *Suppose that the agents run independent policy gradient ascents with softmax parameterization on each encountered MPG as the inner stage, and perform gradient ascents w.r.t the MAML objective as the outer stage. For any $\varepsilon > 0$, $K = \frac{4NL_F}{(1-\gamma)\varepsilon^2}$ iterations of MAML updates can find a policy $\theta^\star$ such that $\|\nabla F_T(\theta^\star)\| \le \varepsilon$, where L_F is given in Lemma 14 of Appendix D.*

5 Meta-Learning for General-Sum Markov Games

In this section, we consider learning coarse correlated equilibria in general-sum Markov games with no assumption on reward structures. Similar to Section 3, we start by developing an initialization-dependent algorithm, followed by investigating the sharper convergence of meta-learning.

Our base algorithm for learning CCE also uses optimistic OMD with stage-based value updates. Detailed descriptions are deferred to Algorithm 2 in Appendix E due to space limitations. Algorithm 2 follows a similar structure as Algorithm 1, but the output policy $\bar{\pi}$ of Algorithm 2 is no longer a state-wise average policy and is instead a correlated policy [3, 44, 33]. For any correlated policy π, we use the notion

$$\text{CCE-gap}(\pi) := \max_{i\in\mathcal{N}} V_{1,i}^{\dagger,\pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1)$$

to measure its distance to a CCE. Let $\bar{\tau}$ denote the total number of stages of Algorithm 2. Similar to zero-sum games (3), for any $(\tau, h, s) \in [\bar{\tau}] \times [H] \times \mathcal{S}$, we define the per-state regret for each player $i \in \mathcal{N}$ as

$$\text{reg}_{h,i}^{\tau}(s) := \max_{\pi_{h,i}^{\tau,\dagger}(\cdot|s)\in\Delta(\mathcal{A}_i)} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \pi_{h,i}^{\tau,\dagger} - \pi_{h,i}^{j}, Q_{h,i}^{\tau} \pi_{h,-i}^{j} \right\rangle (s),$$

where $Q_{h,i}^{\tau}$ is player i's Q-function estimate at stage τ. We define the maximal regret (over all states and players) as $\text{reg}_h^{\tau} := \max_{s\in\mathcal{S}} \max_{i\in\mathcal{N}}\{\text{reg}_{h,i}^{\tau}(s)\}$. The initialization-dependent convergence rate of Algorithm 2 is established in the following theorem.

Theorem 5. *If Algorithm 2 is run on a general-sum Markov game for T iterations with a learning rate $\eta > 0$, the output policy $\bar{\pi}$ satisfies:*

$$\text{CCE-gap}(\bar{\pi}) \le \frac{3}{\eta T} \sum_{\tau=1}^{\bar{\tau}} \sum_{h=1}^{H} \max_{i\in\mathcal{N}, s\in\mathcal{S}} D_R(\pi_{h,i}^{\tau,\dagger}(\cdot|s), \tilde{\pi}_{h,i}^{\tau}(\cdot|s)) + 36N^2\eta^2H^4.$$

In addition, if the players' policies are initialized to be uniform policies $\tilde{\pi}_{h,i}^{\tau}(\cdot|s) = \mathbf{1}/A_i, \forall i \in \mathcal{N}$ and η is chosen as $\eta = H^{-2/3}T^{-1/3}(N-1)^{-2/3}$, then we have

$$\text{CCE-gap}(\bar{\pi}) \le \frac{12N^{\frac{2}{3}}H^{\frac{8}{3}} \log T \log A_{\max}}{T^{\frac{2}{3}}}. \tag{11}$$

Compared to existing results, Theorem 5 directly associates the convergence rate with the quality of the initial policy $\tilde{\pi}$. With uniform initialization, the convergence rate in (11) has a slightly worse dependence on T than the best known result $\widetilde{O}(\sqrt{N}H^{11/4}/T^{3/4})$ [77]. Such deterioration is due to the potential lack of a smoothness condition for optimistic OMD that directly connects the stability of policies to the stability of utility functions (Lemma 18), unlike in optimistic FTRL. Although we believe that our rate in (11) can almost certainly be improved via a refined stability analysis, we leave the tightening of it to our future work as it would be a departure from the main focus of this work.

Let $\tilde{\pi}^k$ and $\bar{\pi}^k$, respectively, denote the initial policy and output policy of Algorithm 2 on game $\mathbb{G}^k$. For player $i \in \mathcal{N}$, by putting together $\pi_{h,i}^{\tau,\dagger}(\cdot|s)$ over all (τ, h, s), we use $\pi_i^{k,\dagger} : [\bar{\tau}]\times[H]\times\mathcal{S} \to \Delta(\mathcal{A}_i)$ to denote the best fixed policies in hindsight on $\mathbb{G}^k$. We consider a game similarity metric defined as

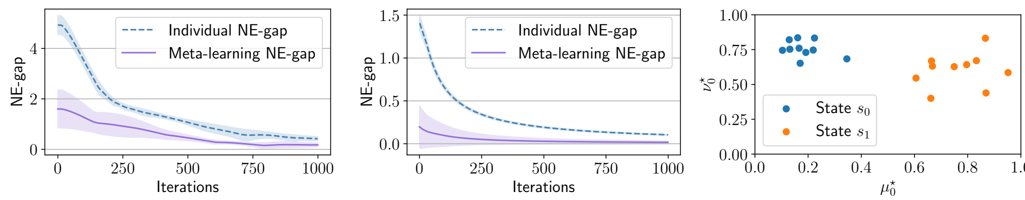
$$\Delta_\pi := \sum_{k=1}^{K}\sum_{i=1}^{N} \text{KL}(\pi_i^{k,\dagger} \| \pi_i^{\star}), \text{ where } \pi_i^{\star} = \frac{1}{K}\sum_{k=1}^{K} \pi_i^{k,\dagger}.$$

The following theorem presents the convergence rate of meta-learning, which again is sharper than learning each game individually when the games are similar, i.e., when Δ_π is sufficiently small.

Theorem 6. *In a sequence of K general-sum Markov games, if Algorithm 2 is run for T iterations as the base algorithm and the meta-updates $\tilde{\pi}_i^k = \frac{1}{k-1}\sum_{k'=1}^{k-1}[\pi_i^{k',\dagger}]_\alpha, \forall i \in \mathcal{N}$ are used with $\alpha = 1/\sqrt{K}$ for policy initializations, then, for any $\varepsilon > 0$, $T = \tilde{O}(\frac{HN}{\varepsilon^{3/2}}(\frac{\Delta_\pi^{3/2}}{K^{5/4}} + \frac{A_{\max}^{3/2}+H^3}{K^{1/2}}))$ steps on average suffice to find an ε-approximate CCE in each game.*

6 Simulations

We numerically evaluate our meta-learning algorithms from Sections 3 and 4 on a sequence of K games. In this section, we evaluate on a sequence of $K = 10$ zero-sum Markov games and Markov potential games with two states, two players, and two candidate actions for each player. In Appendix F, we further demonstrate the scalability of our methods by providing numerical results on larger-scale tasks, including a simplified version of the Poker endgame considered in [27] and a 1D linear-quadratic tracking problem [37] with 4 cooperative players.



(a) Zero-sum Markov game (b) Markov potential game (c) NE visualization

Figure 1: NE-gap of policies output by individual learning and meta-learning in (a) zero-sum Markov games, and (b) Markov potential games. Shaded areas denote the standard deviations. (c) visualizes the NE policies of the K games in the normalized space $[0, 1] \times [0, 1]$ to illustrate their closeness.

We generate the $K = 10$ games by first specifying a "base game" and then adding random perturbations to its reward function to get K slightly different games. Each of the K games is run for $T = 1000$ iterations. To better visualize the similarity level of these games, in Figure 1(c), we plot the NE policies of the perturbed zero-sum matrix games at each of the two states for the $K = 10$ games. We remark that due to the existence of state transitions, the NE policies with respect to the stage Q-functions can be more diversified than Figure 1(c). Detailed descriptions of the simulation setup are deferred to Appendix F.

We evaluate the convergences of the algorithms in terms of NE-gap. Figures 1(a) and 1(b) compare the average NE-gap over the K games between individual learning and meta-learning for zero-sum Markov games and Markov potential games, respectively. We see that meta-learning can utilize knowledge from previous tasks to attain better policy initialization in a new task and converges to an approximate NE policy using much fewer iterations.

7 Concluding Remarks

In this paper, we have introduced meta-learning to solve multiple MARL tasks collectively. Under natural similarity metrics, we have shown that meta-learning achieves provably sharper convergence for learning NE in zero-sum and potential games and for learning CCE in general-sum games. Along the way, we have proposed new MARL algorithms with fine-grained initialization-dependent convergence guarantees. Our work appears to be the first to investigate the theoretical properties of meta-learning in MARL and provide reliable justifications for its usage. As for the limitations, our convergence rate for learning CCE (Theorem 5) is slightly less competitive than the best-known results when our policies are initialized conservatively, which might be improved via a refined policy stability analysis. Other future directions include further generalization of our results to alternative game similarity metrics and broader types of games (e.g., stochastic Stackelberg games).

Acknowledgments and Disclosure of Funding

This work was partially supported by the National Science Foundation (NSF) under grant CCF 20-29049; by the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR), a research collaboration that is part of the IBM AI Horizon Network; by the IBM-ILLINOIS Discovery Accelerator Institute (IIDAI); and by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-19-1-0353. Any opinions, findings or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or IBM or AFOSR.

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: 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, 2020.

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

[4] M.-F. Balcan, M. Khodak, and A. Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, pages 424–433. PMLR, 2019.

[5] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6(10), 2005.

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

[7] S. Cen, Y. Wei, and Y. Chi. Fast policy extragradient methods for competitive games with entropy regularization. *Advances in Neural Information Processing Systems*, 34:27952–27964, 2021.

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

[9] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

[10] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):1–57, 2009.

[11] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[12] 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.

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

[14] G. Denevi, C. Ciliberto, R. Grazzi, and M. Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pages 1566–1575. PMLR, 2019.

[15] D. Ding, C.-Y. Wei, K. Zhang, and M. Jovanovic. Independent policy gradient for large-scale Markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *International Conference on Machine Learning*, pages 5166–5220. PMLR, 2022.

[16] L. Erez, T. Lancewicki, U. Sherman, T. Koren, and Y. Mansour. Regret minimization and convergence to equilibria in general-sum Markov games. *arXiv preprint arXiv:2207.14211*, 2022.

[17] A. Fallah, A. Mokhtari, and A. Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092, 2020.

[18] A. Fallah, K. Georgiev, A. Mokhtari, and A. Ozdaglar. On the convergence theory of debiased model-agnostic meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3096–3107, 2021.

[19] H. Fang, N. J. Harvey, V. S. Portella, and M. P. Friedlander. Online mirror descent and dual averaging: Keeping pace in the dynamic case. *Journal of Machine Learning Research*, 23(1): 5271–5308, 2022.

[20] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[21] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR, 2019.

[22] R. Fox, S. M. Mcaleer, W. Overman, and I. Panageas. Independent natural policy gradient always converges in Markov potential games. In *International Conference on Artificial Intelligence and Statistics*, pages 4414–4425, 2022.

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

[24] T. Furmston, G. Lever, and D. Barber. Approximate Newton methods for policy search in Markov decision processes. *Journal of Machine Learning Research*, 17, 2016.

[25] Z. Gao, Q. Ma, T. Başar, and J. R. Birge. Finite-sample analysis of decentralized Q-learning for stochastic games. *arXiv preprint arXiv:2112.07859*, 2021.

[26] A. Gupta, M. Lanctot, and A. Lazaridou. Dynamic population-based meta-learning for multi-agent communication with natural language. *Advances in Neural Information Processing Systems*, 34:16899–16912, 2021.

[27] K. Harris, I. Anagnostides, G. Farina, M. Khodak, Z. S. Wu, and T. Sandholm. Meta-learning in games. *arXiv preprint arXiv:2209.14110*, 2022.

[28] E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[29] J. Humplik, A. Galashov, L. Hasenclever, P. A. Ortega, Y. W. Teh, and N. Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

[30] K. Ji, J. Yang, and Y. Liang. Theoretical convergence of multi-step model-agnostic meta-learning. *Journal of Machine Learning Research*, 2022.

[31] H. Jia, B. Ding, H. Wang, X. Gong, and X. Zhou. Fast adaptation via meta learning in multi-agent cooperative tasks. In *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pages 707–714, 2019.

[32] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? *Advances in Neural Information Processing Systems*, 31, 2018.

[33] C. Jin, Q. Liu, Y. Wang, and T. Yu. V-learning–A simple, efficient, decentralized algorithm for multiagent RL. In *ICLR Workshop on Gamification and Multiagent Solutions*, 2022.

[34] V. Khattar, Y. Ding, B. Sel, J. Lavaei, and M. Jin. A CMDP-within-online framework for meta-safe reinforcement learning. In *International Conference on Learning Representations*, 2022.

[35] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[37] M. Lauriere, S. Perrin, S. Girgin, P. Muller, A. Jain, T. Cabannes, G. Piliouras, J. Pérolat, R. Elie, O. Pietquin, et al. Scalable deep reinforcement learning algorithms for mean field games. In *International Conference on Machine Learning*, pages 12078–12095. PMLR, 2022.

[38] 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.

[39] S. Leonardos, W. Overman, I. Panageas, and G. Piliouras. Global convergence of multi-agent policy gradient in Markov potential games. In *International Conference on Learning Representations*, 2021.

[40] 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*, 2021.

[41] S. Liu, M. Lanctot, L. Marris, and N. Heess. Simplex neural population learning: Any-mixture bayes-optimality in symmetric zero-sum games. In *International Conference on Machine Learning*, pages 13793–13806. PMLR, 2022.

[42] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30, 2017.

[43] S. V. Macua, J. Zazo, and S. Zazo. Learning parametric closed-loop policies for Markov potential games. In *International Conference on Learning Representations*, 2018.

[44] W. Mao and T. Başar. Provably efficient reinforcement learning in decentralized general-sum Markov games. *Dynamic Games and Applications*, 13:165–186, 2023.

[45] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Başar. Near-optimal model-free reinforcement learning in non-stationary episodic MDPs. In *International Conference on Machine Learning*, pages 7447–7458. PMLR, 2021.

[46] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, R. Iyer, and T. Basar. A mean-field game approach to cloud resource management with function approximation. *Advances in Neural Information Processing Systems*, 35:36243–36258, 2022.

[47] W. Mao, L. Yang, K. Zhang, and T. Başar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049, 2022.

[48] D. Nekipelov, V. Syrgkanis, and E. Tardos. Econometrics for learning agents. In *Proceedings of the 16th ACM Conference on Economics and Computation*, pages 1–18, 2015.

[49] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[50] F. Orabona and D. Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.

[51] P. A. Ortega, J. X. Wang, M. Rowland, T. Genewein, Z. Kurth-Nelson, R. Pascanu, N. Heess, J. Veness, A. Pritzel, P. Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

[52] S. Perrin, J. Pérolat, M. Laurière, M. Geist, R. Elie, and O. Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. *Advances in Neural Information Processing Systems*, 33:13199–13213, 2020.

[53] H. Qiu, W. Mao, C. Wang, H. Franke, A. Youssef, Z. T. Kalbarczyk, T. Başar, and R. K. Iyer. AWARE: Automate workload autoscaling with reinforcement learning in production cloud systems. In *USENIX Annual Technical Conference*, 2023.

[54] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[56] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2016.

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

[58] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[59] Z. Song, S. Mei, and Y. Bai. When can we learn general-sum Markov games with a large number of players sample-efficiently? In *International Conference on Learning Representations*, 2022.

[60] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

[61] D. Sychrovsky, M. Sustr, E. Davoodi, M. Lanctot, and M. Schmid. Learning not to regret. *arXiv preprint arXiv:2303.01074*, 2023.

[62] 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.

[63] A. A. Team, J. Bauer, K. Baumli, S. Baveja, F. Behbahani, A. Bhoopchand, N. Bradley-Schmieg, M. Chang, N. Clay, A. Collister, et al. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*, 2023.

[64] S. Thrun and L. Pratt. *Learning to Learn*. Springer Science & Business Media, 1998.

[65] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[66] L. Wang, Q. Cai, Z. Yang, and Z. Wang. On the global optimality of model-agnostic meta-learning. In *International Conference on Machine Learning*, pages 9837–9846, 2020.

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

[68] 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, 2021.

[69] Q. Xie, Y. Chen, Z. Wang, and Z. Yang. Learning zero-sum simultaneous-move Markov games using function approximation and correlated equilibrium. In *Conference on Learning Theory*, pages 3674–3682, 2020.

[70] S. Xue, C. Qu, X. Shi, C. Liao, S. Zhu, X. Tan, L. Ma, S. Wang, S. Wang, Y. Hu, et al. A meta reinforcement learning approach for predictive autoscaling in the cloud. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4290–4299, 2022.

[71] S. Yang and B. Yang. A meta multi-agent reinforcement learning algorithm for multi-intersection traffic signal control. In *IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pages 18–25, 2021.

[72] Y. Yang and C. Ma. $O(T^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum Markov games. *arXiv preprint arXiv:2209.12430*, 2022.

[73] S. Zeng, T. Doan, and J. Romberg. Regularized gradient descent ascent for two-player zero-sum Markov games. *Advances in Neural Information Processing Systems*, 35:34546–34558, 2022.

[74] K. Zhang, A. Koppel, H. Zhu, and T. Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383*, 2019.

[75] M. Zhang, P. Zhao, H. Luo, and Z.-H. Zhou. No-regret learning in time-varying zero-sum games. In *International Conference on Machine Learning*, pages 26772–26808, 2022.

[76] R. Zhang, Z. Ren, and N. Li. Gradient play in stochastic games: Stationary points, convergence, and sample complexity. *arXiv preprint arXiv:2106.00198*, 2021.

[77] R. Zhang, Q. Liu, H. Wang, C. Xiong, N. Li, and Y. Bai. Policy optimization for Markov games: Unified framework and faster convergence. In *Advances in Neural Information Processing Systems*, 2022.

[78] R. Zhang, J. Mei, B. Dai, D. Schuurmans, and N. Li. On the global convergence rates of decentralized softmax gradient play in Markov potential games. In *Advances in Neural Information Processing Systems*, 2022.

[79] Z. Zhang, Y. Zhou, and X. Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020.

Supplementary Materials for "Multi-Agent Meta-Reinforcement Learning"

A Technical Lemmas

Lemma 1. *Let $x, y \in \mathbb{R}^d$ be two probability distributions lying in the d-dimensional simplex for $d \geq 2$. For $\alpha \in (0, 1/2)$, let $[x]_\alpha = (1-\alpha)x + \frac{\alpha}{d}\mathbf{1}$ denote a weighted average between x and a uniform vector $\mathbf{1}/d \in \mathbb{R}^d$ of a proper dimension. Denote by $\mathrm{KL}(x\|y)$ the Kullback–Leibler divergence between x and y. If $y_i \geq \alpha/d, \forall i \in [d]$, then we have*

$$\mathrm{KL}(x\|y) \leq \mathrm{KL}(\tilde{x}\|y) + 4\alpha \ln \frac{d}{\alpha}.$$

Proof. From the three-points identity of the Bregman divergence (Lemma 3.1 of [9]),

$$\mathrm{KL}(x\|y) - \mathrm{KL}(\tilde{x}\|y) = \mathrm{KL}(x\|\tilde{x}) + \langle \ln \tilde{x} - \ln y, x - \tilde{x} \rangle \tag{12}$$

The first term in (12) can be bounded by

$$\mathrm{KL}(x\|\tilde{x}) = \sum_{i=1}^d x_i \ln \frac{x_i}{\tilde{x}_i} = \sum_{i=1}^d x_i \ln \frac{x_i}{(1-\alpha)x_i + \frac{\alpha}{d}} \leq \sum_{i=1}^d x_i \ln \frac{1}{1-\alpha} \leq \ln \frac{1}{1-\alpha}.$$

By the Hölder's inequality, the second term in (12) is bounded as

$$\langle \ln \tilde{x} - \ln y, x - \tilde{x} \rangle \leq \|\ln \tilde{x} - \ln y\|_\infty \|x - \tilde{x}\|_1 . \tag{13}$$

We handle the two terms in (13) separately. First,

$$\|\ln \tilde{x} - \ln y\|_\infty = \sup_{i \in [d]} \left| \ln \frac{\tilde{x}_i}{y_i} \right| \leq \sup_{i \in [d]} \max \left\{ \ln \frac{\tilde{x}_i}{y_i}, \ln \frac{y_i}{\tilde{x}_i} \right\} \leq \ln \frac{1 - \alpha + \frac{\alpha}{d}}{\alpha/d} \leq \ln \frac{d}{\alpha},$$

where the second to last step uses the facts that $\alpha/d \leq \tilde{x}_i \leq 1$ and $\alpha/d \leq y_i \leq 1, \forall i \in [d]$. The last step is simply due to the fact that $d \geq 1$. To bound the second term in (13), notice that

$$\|x - \tilde{x}\|_1 = \|x - (1-\alpha)x - \alpha\mathbf{1}/d\|_1 = \alpha \|x - \mathbf{1}/d\|_1 \leq 2\alpha.$$

Putting everything together, (12) can be bounded by

$$\mathrm{KL}(x\|\tilde{x}) + \langle \ln \tilde{x} - \ln y, x - \tilde{x} \rangle \leq \ln \frac{1}{1-\alpha} + 2\alpha \ln \frac{d}{\alpha} \leq \alpha^2 + \alpha + 2\alpha \ln \frac{d}{\alpha} \leq 4\alpha \ln \frac{d}{\alpha},$$

where the second to last step is derived using the Taylor expansion, and the last step holds due to the assumptions that $\alpha \in (0, 1/2)$ and $d \geq 2$. This completes the proof of the lemma. $\square$

Lemma 2. *(Proposition B.1 of [35]) Let $R : \Theta \to \mathbb{R}$ be 1-strongly convex with respect to $\|\cdot\|$ and consider any $\theta_1, \ldots, \theta_K \in \Theta$. Then, when run on the loss sequence $\alpha_1 D_R(\theta_1, \dot{)}, \ldots, \alpha_K D_R(\theta_K, \dot{)}$ for any positive scalars $\alpha_1, \ldots, \alpha_K \in \mathbb{R}_+$, the follow-the-leader (FTL) algorithm obtains regret*

$$\mathrm{reg}_K \leq 2CD \sum_{k=1}^K \frac{\alpha_k^2 G_k}{\alpha_k + 2\sum_{k'=1}^{k-1} \alpha_{k'}},$$

for C such that $\|\theta\| \leq C \|\theta\|_2, \forall \theta \in \Theta$, $D = \max_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2$ the L2 diameter of Θ, and G_k the Lipschitz constant of $D_R(\theta_k, \cdot)$ over Θ with respect to $\|\cdot\|$.

Lemma 3. *(Lemma 2 of [18]) For any $i \in \{1, \ldots, n\}$, let $f_i : \mathbb{R}^d \to W_i$ be a continuous function with $W_i \in \{\mathbb{R}, \mathbb{R}^d, \mathbb{R}^{1\times d}, \mathbb{R}^{d \times d}\}$ such that $g(\theta) = f_n(\theta) \ldots f_1(\theta)$ is well-defined. Suppose f_i is B_i-bounded and L_i-Lipschitz, i.e., $\|f_i(\theta)\| \leq B_i$ and $\|f_i(\theta) - f_i(\theta')\| \leq L_i \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d$ for some non-negative constants B_i and L_i. Then, $g(\theta)$ is Lipschitz with constant $L_g = \sum_{i=1}^n (L_i \prod_{j \neq i} B_j)$, i.e., $\|g(\theta) - g(\theta')\| \leq L_g \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d$.*

Lemma 4. *(Lemma 3 of [18]) For any $i \in \{1, \ldots, n\}$, let $f_i : \mathbb{R}^d \to \mathbb{R}^m$ be a continuously differentiable function that is B_f-bounded and L_f-Lipschitz continuous. Let $p(\cdot;\theta)$ be a distribution on $\{f_i\}_{i=1}^n$ where the probability of drawing f_i is $p(i;\theta)$. Suppose there exists a non-negative constant B_p such that $\|\nabla_\theta \log p(i;\theta)\| \le B_p$ for any i and θ. Then, the function $g(\theta) = \mathbb{E}_{p(i;\theta)}[f(i;\theta)]$ is Lipschitz continuous with constant $B_f B_p + L_f$.*

Lemma 5. *Consider a block diagonal matrix C that is a square matrix such that the main-diagonal consists of N block matrices $A_1 \in \mathbb{R}^{d_1 \times d_1}, \ldots, A_N \in \mathbb{R}^{d_N \times d_N}$ and all off-diagonal blocks are zero matrices. Then, it holds that $\|C\| \le \max_{1 \le i \le N} \|A_i\|$.*

Proof. We prove the lemma via induction on N. For the induction basis $N = 2$, we need to show

$$\|C\| = \left\| \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix} \right\| \le \max\{\|A_1\|, \|A_2\|\}.$$

To see this, let $x \in \mathbb{R}^{d_1}$ and $y \in \mathbb{R}^{d_2}$ be such that $\left\| \begin{bmatrix} x \\ y \end{bmatrix} \right\|^2 = \|x\|^2 + \|y\|^2 = 1$. Then, by the definition of the matrix norm,

$$\left\| C \begin{bmatrix} x \\ y \end{bmatrix} \right\|^2 = \|A_1 x\|^2 + \|A_2 y\|^2 \le \|A_1\|^2 \|x\|^2 + \|A_2\|^2 \|y\|^2 \le \max\{\|A_1\|^2, \|A_2\|^2\},$$

where the last step uses the fact that $\|x\|^2 + \|y\|^2 = 1$. This completes the proof of the induction basis $N = 2$. Now, suppose that the lemma holds for $N = k-1$. We next show that it also holds for $N = k$. Let $C = \begin{bmatrix} A_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & A_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & A_k \end{bmatrix}$. Note that we can rewrite the matrix as $C = \begin{bmatrix} C_{k-1} & \mathbf{0} \\ \mathbf{0} & A_k \end{bmatrix}$, where $C_{k-1} = \begin{bmatrix} A_1 & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & A_{k-1} \end{bmatrix}$ is a block diagonal matrix consisting of $k-1$ matrices. Invoking the induction hypothesis for $N = k-1$, we know that $\|C_{k-1}\| \le \max_{1 \le i \le k-1} \|A_i\|$. Finally, using the induction hypothesis for $N = 2$, we conclude that $\|C\| \le \max\{\|C_{k-1}\|, \|A_k\|\} \le \max_{1 \le i \le k} \|A_i\|$. This completes the induction proof. □

Lemma 6. *Consider a block matrix $A(\theta)$ with $N \times N$ blocks parameterized by $\theta \in \mathbb{R}^d$:*

$$A(\theta) = \begin{bmatrix} A_{1,1}(\theta) & \ldots & A_{1,N}(\theta) \\ \vdots & \ddots & \vdots \\ A_{N,1}(\theta) & \ldots & A_{N,N}(\theta) \end{bmatrix},$$

where $A_{i,j}(\theta) \in \mathbb{R}^{d_i \times d_j}, \forall 1 \le i, j \le N$ and $d = \sum_{i=1}^N d_i$. Suppose that the norm of each matrix block is Lipschitz continuous with respect to θ, i.e., $\|A_{i,j}(\theta) - A_{i,j}(\theta')\| \le L_{i,j} \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d, 1 \le i, j \le N$. Let $L = \max\{L_{i,j} : 1 \le i, j \le N\}$. Then, the norm of $A(\theta)$ is also Lipschitz, i.e.,

$$\|A(\theta) - A(\theta')\| \le NL \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d.$$

Proof. Let $x \in \mathbb{R}^d$ be a vector such that $x = \begin{bmatrix} x_1^\top & x_2^\top & \dots & x_N^\top \end{bmatrix}^\top$ and $\|x\|^2 = \sum_{i=1}^N \|x_i\|^2 = 1$, where $x_i \in \mathbb{R}^{d_i}, \forall 1 \le i \le N$. We have

$$
\begin{aligned}
\|(A(\theta) - A(\theta'))x\|^2 =& \left\| \begin{bmatrix} \sum_{j=1}^N (A_{1,j}(\theta) - A_{1,j}(\theta'))\, x_j \\ \vdots \\ \sum_{j=1}^N (A_{N,j}(\theta) - A_{N,j}(\theta'))\, x_j \end{bmatrix} \right\|^2 \\
=& \sum_{i=1}^N \left\| \sum_{j=1}^N (A_{i,j}(\theta) - A_{i,j}(\theta'))\, x_j \right\|^2 \\
\le& N \sum_{i=1}^N \sum_{j=1}^N \|(A_{i,j}(\theta) - A_{i,j}(\theta'))\, x_j\|^2 \\
\le& N \sum_{i=1}^N \sum_{j=1}^N \|A_{i,j}(\theta) - A_{i,j}(\theta')\|^2 \|x_j\|^2,
\end{aligned}
$$

where the first inequality follows from the Cauchy-Schwarz inequality, and the last step is due to the definition of the matrix norm. Applying the Lipschitz continuity of each matrix block $\|A_{i,j}(\theta) - A_{i,j}(\theta')\| \le L_{i,j} \|\theta - \theta'\|$ yields

$$
\begin{aligned}
\|(A(\theta) - A(\theta'))x\|^2 \le& N \sum_{i=1}^N \sum_{j=1}^N \|A_{i,j}(\theta) - A_{i,j}(\theta')\|^2 \|x_j\|^2 \\
\le& N \sum_{i=1}^N \sum_{j=1}^N L_{i,j}^2 \|\theta - \theta'\|^2 \|x_j\|^2 \\
\le& N^2 L^2 \|\theta - \theta'\|^2,
\end{aligned}
$$

where the last step uses the facts that $L_{i,j} \le L, \forall 1 \le i,j \le N$ and $\sum_{j=1}^N \|x_j\|^2 = 1$. Since the above condition holds for any vector x with $\|x\| = 1$, we know from the definition of the matrix norm that

$$
\|A(\theta) - A(\theta')\| \le NL \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d.
$$

This concludes the proof for the Lipschitz continuity of $A(\theta)$. □

B Proofs for Section 3

B.1 Proof of Theorem 1

We introduce one more notation before presenting the proof. For each iteration $t \in [T]$ and step $h \in [H]$, define the Q-function estimation error as

$$
\delta_h^t := \|Q_h^{\tau(t)} - Q_h^\star\|_\infty.
$$

Note that since Algorithm 1 performs stage-based value updates, the value estimation error δ_h^t does not change within a stage $\tau(t)$; that is, δ_h^t takes the same value for all $t \in [t_\tau^{\text{start}}, t_\tau^{\text{end}}]$. For this reason, we will sometimes abuse the notation and simply use δ_h^τ to denote the estimation error for a stage τ. In the rest of this paper, we will write δ_h^τ and δ_h^t interchangeably since one of them will be more convenient than the other in certain contexts.

Further, recall that for any $(\tau, h, s) \in [\bar{\tau}] \times [H] \times \mathcal{S}$, the per-state regrets for the two players are defined as

$$
\begin{aligned}
\mathrm{reg}_{h,1}^\tau(s) :=& \max_{\mu_h^{\tau,\dagger} \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^\tau \nu_h^j \right\rangle (s), \\
\mathrm{reg}_{h,2}^\tau(s) :=& \max_{\nu_h^{\tau,\dagger} \in \Delta(\mathcal{B})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \nu_h^j - \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j \right\rangle (s).
\end{aligned} \tag{14}
$$

Note that the best response policies $\mu_h^{\tau,\dagger}(\cdot|s)$ and $\nu_h^{\tau,\dagger}(\cdot|s)$ should be state-dependent, but we will oftentimes omit the dependence on s for notational convenience. This leads us to the initialization-dependent convergence rate of Algorithm 1, which we re-state and prove as follows.

Theorem 1. If we run Algorithm 1 on a two-player zero-sum Markov game for T iterations with a learning rate $\eta \leq 1/(8H^2)$, the output policy pair $(\bar{\mu}, \bar{\nu})$ satisfies:

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{192H^3}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \max_s \left(D_R(\mu_h^{\tau,\dagger}(\cdot|s), \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}(\cdot|s), \tilde{\nu}_h^\tau(\cdot|s)) \right).$$

In addition, if we initialize the players' policies to be uniform policies, i.e., $\tilde{\mu}_h^\tau(\cdot|s) = \mathbf{1}/A$ and $\tilde{\nu}_h^\tau(\cdot|s) = \mathbf{1}/B, \forall s \in \mathcal{S}, \tau \in [\bar{\tau}], h \in [H]$, we further have

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{768H^5 \log T \log(AB)}{T}.$$

Proof. The proof of the theorem follows from a series of lemmas, which we state and prove in the next few subsections. In particular, we first show in Lemma 7 that upper bounding the NE-gap breaks down to controlling the per-state regrets $\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s)$ and the value estimation errors δ_h^τ, in a similar fashion as in the analysis of [72]. For this purpose, Lemma 8 provides an upper bound on the per-state regrets, while Lemma 9 and Lemma 10 together bound the value estimation error via a recursive argument. The rest of the proof follows by putting all the aforementioned results together.

Specifically, for $\eta \leq 1/(8H^2)$, by plugging in the results of Lemma 8 and Lemma 9 to Lemma 7, we obtain that

$$\begin{aligned}
\text{NE-gap}(\bar{\mu}, \bar{\nu}) &\leq \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \max_{s\in\mathcal{S}} \left(\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s) \right) + \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \delta_h^\tau \\
\leq& \frac{16H^2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \max_{s\in\mathcal{S}} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s)) \right) \\
&+ \frac{192H^2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \sum_{h'=h+1}^{H} \max_{s\in\mathcal{S}} \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right) \\
\leq& \frac{192H^2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \sum_{h'=h}^{H} \max_{s\in\mathcal{S}} \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right) \\
\leq& \frac{192H^3}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \max_s \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s)) \right),
\end{aligned} \tag{15}$$

where the last step is by switching the order of counting. This proves the first claim in the Theorem.

We now proceed to establish the second statement. Recall that we chose the negative entropy as the regularizer R. In this case, the Bregman divergence $D_R(\cdot,\cdot)$ reduces to the Kullback–Leibler divergence. Since $\mu_h^{\tau,\dagger}$ lies in the simplex, when we initialize $\tilde{\mu}_h^\tau(\cdot|s) = \mathbf{1}/A$ to be a uniform distribution, we naturally have $D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) \leq \log A, \forall s \in \mathcal{S}, h \in [H]$. A similar result holds for $D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s))$. We can hence obtain that

$$\max_s \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s)) \right) \leq \log(AB). \tag{16}$$

To prove the statement, it remains to upper bound the total number of stages $\bar{\tau}$. Recall that we have defined the lengths of the stages to increase exponentially with $L_{\tau+1} = \lfloor (1+1/H)L_\tau \rfloor$. Since the $\bar{\tau}$ stages sum up to T iterations in total, by taking the sum of a geometric series, it suffices to find a value of $\bar{\tau}$ such that $(1+1/H)^{\bar{\tau}} \geq T/H$. Using the Taylor series expansion, one can show that $(1+\frac{1}{H})^H \geq e - \frac{e}{2H}$. Hence, it reduces to finding a minimum $\bar{\tau}$ such that

$$\left(e - \frac{e}{2H} \right)^{\bar{\tau}/H} \geq \frac{T}{H}. \tag{17}$$

One can easily see that any $\bar{\tau} \geq \frac{H \log T}{\log(e/2)}$ satisfies the condition. Together with (15) and (16), we obtain that

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{768 H^5 \log T}{T} \log(AB).$$

This completes the proof of the theorem. $\square$

B.2 Supporting Lemmas for Section 3

Before presenting the supporting lemmas of the section, we remark that we will reload the notations μ_h^t and ν_h^t with some slight abuse of notations. Specifically, when t is the last iteration of a stage, μ_h^t can be used to denote not only the policy at iteration t, but also the initial policy of the next stage (see Line 10 of Algorithm 1). In the following proofs, it should be clear from the context which specific policy μ_h^t refers to. A similar rule applies to ν_h^t.

Lemma 7. *Let $(\bar{\mu}, \bar{\nu})$ be the output policies of Algorithm 1. Then,*

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \leq \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \max_{s \in \mathcal{S}} \left(\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s) \right) + \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \delta_h^\tau.$$

Proof. From Lemma C.1 in [77], we know that

$$\begin{aligned}
&\text{NE-gap}(\bar{\mu}, \bar{\nu}) \\
=& V_1^{\dagger, \bar{\nu}}(s_1) - V_1^\star(s_1) + V_1^\star(s_1) - V_1^{\bar{\mu}, \dagger}(s_1) \\
\leq& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \left[\langle \mu_h^\dagger, Q_h^\star \bar{\nu}_h \rangle - \langle \nu_h^\dagger, (Q_h^\star)^\top \bar{\mu}_h \rangle \right](s) \right\} \\
=& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^\star \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^\star)^\top \mu_h^t \rangle \right](s) \right\} \\
\leq& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^{\tau(t)} \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^{\tau(t)})^\top \mu_h^t \rangle \right](s) \right\} + \frac{2}{T} \sum_{h=1}^{H} \sum_{t=1}^{T} \delta_h^t, \qquad (18)
\end{aligned}$$

where the last step is by adding and subtracting the estimated values $Q_h^{\tau(t)}$, and invoking the definition that $\delta_h^t = \left\| Q_h^{\tau(t)} - Q_h^\star \right\|_\infty$. To further bound the first term in (18), notice that

$$\begin{aligned}
&\max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^{\tau(t)} \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^{\tau(t)})^\top \mu_h^t \rangle \right](s) \right\} \\
\leq& \frac{1}{T} \sum_{\tau=1}^{\bar{\tau}} \max_s \left\{ \max_{\mu_h^{\tau,\dagger}, \nu_h^{\tau,\dagger}} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left[\langle \mu_h^{\tau,\dagger}, Q_h^\tau \nu_h^j \rangle - \langle \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j \rangle \right](s) \right\} \\
\leq& \frac{1}{T} \sum_{\tau=1}^{\bar{\tau}} L_\tau \max_s \left(\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s) \right). \qquad (19)
\end{aligned}$$

The first step holds because the LHS uses a fixed pair of best responses $(\mu_h^\dagger, \nu_h^\dagger)$ for the entire T iterations, while the RHS uses a separate best response pair $(\mu_h^{\tau,\dagger}, \nu_h^{\tau,\dagger})$ for each individual stage τ and then puts them together. The RHS clearly upper bounds the LHS as the RHS maximizes over each stage separately. The last step in (19) holds due to the definitions of $\text{reg}_{h,1}^\tau(s)$ and $\text{reg}_{h,2}^\tau(s)$ that

$$\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s) = \max_{\mu_h^{\tau,\dagger}, \nu_h^{\tau,\dagger}} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left[\langle \mu_h^{\tau,\dagger}, Q_h^\tau \nu_h^j \rangle - \langle \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j \rangle \right](s).$$

To control the second term in (18), we use the fact that with stage-based value updates, the value estimation error δ_h^t does not change within a stage. Therefore,

$$\frac{2}{T} \sum_{h=1}^{H} \sum_{t=1}^{T} \delta_h^t = \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \delta_h^j = \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \delta_h^\tau. \qquad (20)$$

Finally, substituting (19) and (20) back to (18) completes the proof. $\square$

Lemma 8. *For every stage $\tau \in \mathbb{N}_+$, every step $h \in [H]$ and every state $s \in \mathcal{S}$, the per-state average regret is bounded by:*

$$\begin{aligned}\operatorname{reg}_{h,1}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2 \\ & - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2, \end{aligned} \tag{21}$$

$$\begin{aligned}\operatorname{reg}_{h,2}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2 \\ & - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2. \end{aligned} \tag{22}$$

In particular, for $\eta \leq 1/(8H^2)$, we further have

$$\begin{aligned}\operatorname{reg}_{h,1}^{\tau}(s) + \operatorname{reg}_{h,2}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) \right) \\ & - \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \frac{4\eta H^3}{L_\tau} \left(\|\nu_h^j(\cdot|s) - \nu_h^{j-1}(\cdot|s)\|_1^2 + \|\mu_h^j(\cdot|s) - \mu_h^{j-1}(\cdot|s)\|_1^2 \right). \end{aligned} \tag{23}$$

Proof. We prove the regret bound for the max-player, i.e., $\operatorname{reg}_{h,1}^{\tau}(s)$. The bound for the min-player holds analogously. Notice that the policy update steps in Algorithm 1 are exactly the same as the optimistic online mirror descent algorithm [55, 62], with the loss vector $g^t = [Q_h^{\tau}\nu_h^t](s,\cdot)$ and the recency bias $M^t = [Q_h^{\tau}\nu_h^{t-1}](s,\cdot)$. Since our stage-based value updates assign equal weights to each iteration, we end up with a classic no-(average-)regret learning problem instead of a no-(weighed-)regret learning problem as in [72, 77]. This allows us to directly apply the standard optimistic OMD results (e.g., Lemma 1 in [55] and Proposition 5 in [62]) to obtain

$$\begin{aligned}\operatorname{reg}_{h,1}^{\tau}(s) = & \max_{\mu_h^{\tau,\dagger} \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^{\tau}\nu_h^j \right\rangle (s) \\ \leq & \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{\eta}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\| [Q_h^{\tau}\nu_h^j - Q_h^{\tau}\nu_h^{j-1}](s,\cdot) \right\|_\infty^2 \end{aligned} \tag{24}$$

$$- \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2. \tag{25}$$

To further upper bound the term in (24), notice that

$$\left\| \left[Q_h^{\tau}\nu_h^j - Q_h^{\tau}\nu_h^{j-1} \right](s,\cdot) \right\|_\infty^2 \leq 2H^2 \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2,$$

where we used the Hölder's inequality and the fact that $\|Q_h^{\tau}(s,\cdot)\|_\infty \leq H$. Substituting the above result back to (25) yields

$$\begin{aligned}\operatorname{reg}_{h,1}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{\eta}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} 2H^2 \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2 \\ & - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2. \end{aligned}$$

This completes the proof of (21). The regret bound in (22) can be shown via symmetry.

Combining (21) and (22) leads to

$$
\begin{aligned}
&\mathrm{reg}_{h,1}^{\tau}(s)+\mathrm{reg}_{h,2}^{\tau}(s)\\
\leq&\frac{1}{\eta L_\tau}\left(D_R(\mu_h^{\tau,\dagger},\tilde{\mu}_h^{\tau}(\cdot|s))+D_R(\nu_h^{\tau,\dagger},\tilde{\nu}_h^{\tau}(\cdot|s))\right)\\
&+\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\left(\frac{2H^2\eta}{L_\tau}-\frac{1}{8\eta L_\tau}\right)\left(\|\nu_h^j(\cdot|s)-\nu_h^{j-1}(\cdot|s)\|_1^2+\|\mu_h^j(\cdot|s)-\mu_h^{j-1}(\cdot|s)\|_1^2\right).
\end{aligned}
$$

When $\eta\leq 1/(8H^2)$, we further have

$$
\begin{aligned}
\mathrm{reg}_{h,1}^{\tau}(s)+\mathrm{reg}_{h,2}^{\tau}(s)\leq&\frac{1}{\eta L_\tau}\left(D_R(\mu_h^{\tau,\dagger},\tilde{\mu}_h^{\tau}(\cdot|s))+D_R(\nu_h^{\tau,\dagger},\tilde{\nu}_h^{\tau}(\cdot|s))\right)\\
&-\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\frac{4\eta H^3}{L_\tau}\left(\|\nu_h^j(\cdot|s)-\nu_h^{j-1}(\cdot|s)\|_1^2+\|\mu_h^j(\cdot|s)-\mu_h^{j-1}(\cdot|s)\|_1^2\right).
\end{aligned}
$$

This completes the proof of the lemma. □

Lemma 9. *With $\eta\leq 1/(8H^2)$, for any iteration $t\in[T]$ and any step $h\in[H]$, we have that*

$$
\delta_h^t\leq\frac{12}{\eta L_{\tau(t)}}\sum_{h'=h+1}^{H}\max_s\left(D_R(\mu_{h'}^{\tau(t)-h'+h,\dagger},\tilde{\mu}_{h'}^{\tau(t)-h'+h}(\cdot|s))+D_R(\nu_{h'}^{\tau(t)-h'+h,\dagger},\tilde{\nu}_{h'}^{\tau(t)-h'+h}(\cdot|s))\right).
$$

Proof. In the following, when we consider a fixed iteration $t\in[T]$, we drop the notational dependence on t and simply use τ (instead of $\tau(t)$) to denote the stage that iteration t belongs to. For any $h\in[H-1]$, we can use Lemma 10 (similar to Lemma C.2 of [77]) to establish the following recursion for the value estimation error:

$$
\delta_h^t\leq\delta_{h+1}^{\tau-1}+\mathrm{reg}_{h+1}^{\tau-1},\tag{26}
$$

where recall that $\mathrm{reg}_h^{\tau}=\max_{s\in\mathcal{S}}\{\mathrm{reg}_{h,1}^{\tau}(s),\mathrm{reg}_{h,2}^{\tau}(s)\}$. Using Lemma 8, we can upper bound the individual regrets $\mathrm{reg}_{h,1}^{\tau}(s)$ and $\mathrm{reg}_{h,2}^{\tau}(s)$ by

$$
\mathrm{reg}_{h,1}^{\tau}(s)\leq\frac{1}{\eta L_\tau}D_R(\mu_h^{\tau,\dagger},\tilde{\mu}_h^{\tau}(\cdot|s))+\frac{2\eta H^2}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\left\|\nu_h^j(\cdot\mid s)-\nu_h^{j-1}(\cdot\mid s)\right\|_1^2,\tag{27}
$$

$$
\mathrm{reg}_{h,2}^{\tau}(s)\leq\frac{1}{\eta L_\tau}D_R(\nu_h^{\tau,\dagger},\tilde{\nu}_h^{\tau}(\cdot|s))+\frac{2\eta H^2}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\left\|\mu_h^j(\cdot\mid s)-\mu_h^{j-1}(\cdot\mid s)\right\|_1^2.\tag{28}
$$

where we have dropped the negative terms in (21) and (22). Following a similar approximate non-negativity argument as in Lemma 5 of [72] (reproduced in Lemma 11 for our stage-based approach), we obtain that

$$
\mathrm{reg}_{h,1}^{\tau}(s)+\mathrm{reg}_{h,2}^{\tau}(s)\geq-2\delta_h^{\tau}.
$$

Together with (23) in Lemma 8, we obtain that

$$
\begin{aligned}
&\frac{2\eta H^2}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\left(\|\nu_h^j(\cdot|s)-\nu_h^{j-1}(\cdot|s)\|_1^2+\|\mu_h^j(\cdot|s)-\mu_h^{j-1}(\cdot|s)\|_1^2\right)\\
\leq&\frac{\delta_h^{\tau}}{H}+\frac{1}{2H\eta L_\tau}\left(D_R(\mu_h^{\tau,\dagger},\tilde{\mu}_h^{\tau}(\cdot|s))+D_R(\nu_h^{\tau,\dagger},\tilde{\nu}_h^{\tau}(\cdot|s))\right)
\end{aligned}
$$

Since the above inequality holds for any state $s\in\mathcal{S}$, substituting it back to (27) and (28) yields

$$
\mathrm{reg}_h^{\tau}\leq\max_s\frac{3}{2\eta L_\tau}\left(D_R(\mu_h^{\tau,\dagger},\tilde{\mu}_h^{\tau}(\cdot|s))+D_R(\nu_h^{\tau,\dagger},\tilde{\nu}_h^{\tau}(\cdot|s))\right)+\frac{\delta_h^{\tau}}{H}.\tag{29}
$$

We can further substitute the regret bound above back to the recursion 26 to get that

$$\delta_h^\tau \leq \frac{3}{2\eta L_{\tau-1}} \max_s \Big(D_R(\mu_{h+1}^{\tau-1,\dagger}, \tilde{\mu}_{h+1}^{\tau-1}(\cdot|s)) + D_R(\nu_{h+1}^{\tau-1,\dagger}, \tilde{\nu}_{h+1}^{\tau-1}(\cdot|s)) \Big) + (1 + \frac{1}{H})\delta_{h+1}^{\tau-1}, \quad (30)$$

where we used the fact that the value estimation error δ_h^t does not change within a stage τ since we perform stage-based value updates. Using a backward inductive argument (starting from the induction basis that $\delta_H^\tau = 0, \forall \tau$), the above recursion in (30) leads us to the following result:

$$\begin{aligned}
\delta_h^\tau \leq & \sum_{h'=h+1}^{H} \frac{3}{2\eta L_{\tau-h'+h}} \left(1 + \frac{1}{H}\right)^{h'-h-1} \max_s \Big(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \Big) \\
\leq & \frac{3}{2\eta L_\tau} \sum_{h'=h+1}^{H} \left(1 + \frac{1}{H}\right)^{2(h'-h)-1} \max_s \Big(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \Big) \\
\leq & \frac{3}{2\eta L_\tau} \sum_{h'=h+1}^{H} \left(1 + \frac{1}{H}\right)^{2H} \max_s \Big(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \Big) \\
\leq & \frac{12}{\eta L_\tau} \sum_{h'=h+1}^{H} \max_s \Big(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \Big), \quad (31)
\end{aligned}$$

where the second step uses our choice of the stage lengths that $L_{\tau+1} = \lfloor (1 + 1/H)L_\tau \rfloor$, which further implies that

$$\frac{1}{L_{\tau-h'+h}} \leq \frac{1}{L_\tau} \left(1 + \frac{1}{H}\right)^{h'-h}.$$

The last step in (31) is due to the fact that $(1 + 1/H)^H \leq e \approx 2.71828$. This completes the proof of the lemma. $\square$

Lemma 10. *(Value estimation error recursion) For any iteration $t \in [T]$ and any step $h \in [H]$, we have the following recursion for the value estimation error δ_h^t:*

$$\delta_h^t \leq \delta_{h+1}^{\tau(t)-1} + \mathrm{reg}_{h+1}^{\tau(t)-1}.$$

Proof. The proof essentially follows a similar procedure as that of Lemma C.2 in [77]. Let $\tau = \tau(t)$. For any $(h, s, a, b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$, we know from the definition of $Q_h^\star$ that

$$\begin{aligned}
Q_h^\star(s,a,b) =& r_h(s,a,b) + \max_{\mu_{h+1} \in \Delta(\mathcal{A})} \min_{\nu_{h+1} \in \Delta(\mathcal{B})} P_h \left[\mu_{h+1}^\top Q_{h+1}^\star \nu_{h+1} \right](s,a,b) \\
\leq & r_h(s,a,b) + \max_{\mu_{h+1}} P_h \left[\mu_{h+1}^\top Q_{h+1}^\star \left(\frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \nu_{h+1}^j \right) \right](s,a,b) \\
\leq & r_h(s,a,b) + \max_{\mu_{h+1} \in} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} P_h \left[\mu_{h+1}^\top Q_{h+1}^\star \nu_{h+1}^j \right](s,a,b) \\
\leq & r_h(s,a,b) + \max_{\mu_{h+1} \in} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left(P_h \left[\mu_{h+1}^\top Q_{h+1}^{\tau-1} \nu_{h+1}^j \right](s,a,b) + \left\| Q_{h+1}^\star - Q_{h+1}^{\tau-1} \right\|_\infty \right),
\end{aligned}$$

where the second step holds because $\frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \nu_{h+1}^j(\cdot|s) \in \Delta(\mathcal{B})$. Using the definitions of $\mathrm{reg}_{h+1}^{\tau-1}$ and $\delta_{h+1}^{\tau-1}$, the above inequality further leads to

$$\begin{aligned}
Q_h^\star(s,a,b) \leq & r_h(s,a,b) + \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} P_h \left[(\mu_{h+1}^j)^\top Q_{h+1}^{\tau-1} \nu_{h+1}^j \right](s,a,b) + \delta_{h+1}^{\tau-1} + \mathrm{reg}_{h+1}^{\tau-1} \\
\leq & Q_h^\tau(s,a,b) + \delta_{h+1}^{\tau-1} + \mathrm{reg}_{h+1}^{\tau-1}
\end{aligned}$$

where the last step is due to the value update rule in Algorithm 1. This implies that

$$Q_h^\star(s,a,b) - Q_h^\tau(s,a,b) \le \delta_{h+1}^{\tau-1} + \mathrm{reg}_{h+1}^{\tau-1}.$$

Using a similar argument, we can show a symmetric result for the min-player:

$$Q_h^\tau(s,a,b) - Q_h^\star(s,a,b) \le \delta_{h+1}^{\tau-1} + \mathrm{reg}_{h+1}^{\tau-1}.$$

Combining both directions yields the desired result. □

Lemma 11. *(Approximate non-negativity) For any $\tau \in [\bar\tau]$ and $h \in [H]$, we have that*

$$\mathrm{reg}_{h,1}^\tau(s) + \mathrm{reg}_{h,2}^\tau(s) \ge -2\delta_h^\tau.$$

Proof. This lemma can be considered as a stage-based variant of Lemma 5 in [72]. From the definitions of $\mathrm{reg}_{h,1}^\tau(s)$ and $\mathrm{reg}_{h,2}^\tau(s)$, we have that

$$\begin{aligned}
&\mathrm{reg}_{h,1}^\tau(s) + \mathrm{reg}_{h,2}^\tau(s)\\
=&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}} \frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle \mu_h^{\tau,\dagger}, Q_h^\tau \nu_h^j\right\rangle - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j\right\rangle\right)(s)\\
=&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}} \frac{1}{L_\tau}\left[\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle \mu_h^{\tau,\dagger}, Q_h^\star \nu_h^j\right\rangle - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\star)^\top \mu_h^j\right\rangle\right)(s)\right.\\
&\left.+\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle \mu_h^{\tau,\dagger}, (Q_h^\tau - Q_h^\star) \nu_h^j\right\rangle - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\tau - Q_h^\star)^\top \mu_h^j\right\rangle\right)(s)\right]\\
\ge&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}} \frac{1}{L_\tau}\left[\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle \mu_h^{\tau,\dagger}, Q_h^\star \nu_h^j\right\rangle - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\star)^\top \mu_h^j\right\rangle\right)(s)\right] - 2\delta_h^\tau,
\end{aligned}\tag{32}$$

where the second step is by adding and subtracting the same term, and the last step uses the definition that $\delta_h^\tau = \|Q_h^\tau - Q_h^\star\|_\infty$. Since both $\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \mu_h^j(\cdot|s)$ and $\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \nu_h^j(\cdot|s)$ are valid probability distributions over the action spaces, the first term in (32) is always non-negative:

$$\begin{aligned}
&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}} \frac{1}{L_\tau}\left[\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle \mu_h^{\tau,\dagger}, Q_h^\star \nu_h^j\right\rangle - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\star)^\top \mu_h^j\right\rangle\right)(s)\right]\\
=&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}}\left[\left\langle \mu_h^{\tau,\dagger}, Q_h^\star\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\nu_h^j\right)\right\rangle(s) - \left\langle \nu_h^{\tau,\dagger}, (Q_h^\star)^\top\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\mu_h^j\right)\right\rangle(s)\right]\\
\ge&\left\langle\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\mu_h^j\right), Q_h^\star\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\nu_h^j\right)\right\rangle(s) - \left\langle\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\nu_h^j\right), (Q_h^\star)^\top\left(\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\mu_h^j\right)\right\rangle(s)\\
=&0.
\end{aligned}$$

Plugging the above inequality back into (32) completes the proof. □

B.3 Proof of Theorem 2

Proof. First, recall the definitions of $(\tilde\mu^k,\tilde\nu^k)$, $(\bar\mu^k,\bar\nu^k)$ and $(\mu^{k,\dagger},\nu^{k,\dagger})$. Since we use a negative entropy regularizer R, the Bregman divergence $D_R(\cdot,\cdot)$ reduces to the Kullback–Leibler divergence. Using these notations, our convergence results of learning in an individual zero-sum game $\mathbb{G}^k$ (Theorem 1) can be written more succinctly as

$$\text{NE-gap}(\bar\mu^k,\bar\nu^k) \le \frac{192H^3}{T}\left(\mathrm{KL}\left(\mu^{k,\dagger}\|\tilde\mu^k\right) + \mathrm{KL}\left(\nu^{k,\dagger}\|\tilde\nu^k\right)\right),$$

where for ease of notations, we write

$$\mathrm{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right) := \sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}} \max_s \mathrm{KL}\left(\mu_h^{k,\tau,\dagger}(\cdot|s)\|\tilde{\mu}_h^k(\cdot|s)\right).$$

Here, $\mu_h^{k,\tau,\dagger}(\cdot|s)$ represents the value of $\mu_h^{\tau,\dagger}(\cdot|s)$ in game $\mathbb{G}^k$. The notation $D_R(\nu^{k,\dagger},\tilde{\nu}^k)$ can be decomposed in a similar manner. By running Algorithm 1 on a sequence of K games, we have that

$$\frac{1}{K}\sum_{k=1}^{K}\text{NE-gap}(\bar{\mu}^k,\bar{\nu}^k) \le \frac{192H^3}{KT}\sum_{k=1}^{K}\left(\mathrm{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right)+\mathrm{KL}\left(\nu^{k,\dagger}\|\tilde{\nu}^k\right)\right). \tag{33}$$

In the following, we will focus on the term for the maximizing player in (33). The results for the minimizing player's term can be obtained via symmetry.

Recall the notation that $[\mathbf{x}]_\alpha = (1-\alpha)\mathbf{x} + \frac{\alpha}{d}\mathbf{1}$ for $\mathbf{x}\in\mathbb{R}^d$. By applying this notation entry-wise to each probability distribution in $\mu^{k,\dagger}$ and invoking Lemma 1, we obtain that

$$\frac{1}{K}\sum_{k=1}^{K}\mathrm{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right) \le \frac{1}{K}\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) + 4H\bar{\tau}\alpha\ln\frac{A}{\alpha}. \tag{34}$$

Notice that the conditions of Lemma 1 are satisfied here because we select our initial policies to be $\tilde{\mu}^k = \frac{1}{k-1}\sum_{k'=1}^{k-1}[\mu^{k',\dagger}]_\alpha$, which assigns a probability of at least $\alpha\mathbf{1}/A$ to each action. Adding and subtracting the same term leads to

$$\begin{aligned}\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) =& \min_\mu\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right) + \min_\mu\sum_{k=1}^{K}\left(\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) - \mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right)\right)\\ \le& \min_\mu\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right) + \frac{8A(1+\ln K)}{\alpha},\end{aligned} \tag{35}$$

where the minimum μ is taken over all policies of the form of $\mu : [\bar{\tau}]\times[H]\times\mathcal{S}\to\Delta(\mathcal{A})$. We now turn to establish the second step in (35), which reduces to bounding the following regret where the loss functions are given by the Bregman divergences:

$$\mathrm{reg} = \min_\mu\sum_{k=1}^{K}\left(\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) - \mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right)\right).$$

It is known that the unique minimum of $\sum_{k'=1}^{k}\mathrm{KL}([\mu^{k',\dagger}]_\alpha\|\cdot)$ is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\mu^{k',\dagger}]_\alpha$ (see Proposition 1 of [5] for a proof of this claim). Therefore, by letting $\tilde{\mu}^k = \frac{1}{k-1}\sum_{k'=1}^{k-1}[\mu^{k',\dagger}]_\alpha$, we are essentially running the follow-the-leader (FTL) algorithm (separately for each entry $(\tau,h,s)\in[\bar{\tau}]\times[H]\times\mathcal{S}$) on the sequence of losses defined by $\sum_{k=1}^{K}\mathrm{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$. We can then invoke the logarithmic regret guarantee of FTL with respect to Bregman divergences, which was established in [35] and was reproduced as Lemma 2 in Appendix A for completeness. To show that Lemma 2 is applicable, we remark that the Kullback–Leibler divergence is not Lipschitz continuous near the boundary of the probability simplex, which breaks condition required by Lemma 2. However, by restricting to policies of the form $[\mu]_\alpha = (1-\alpha)\mu + \frac{\alpha}{A}\mathbf{1}$, which is at least $\frac{\alpha}{A}$-distance away from the simplex boundary, the Kullback–Leibler divergence is indeed Lipschitz continuous within this $\frac{\alpha}{A}$-restricted domain. One can show that the Lipschitz constant of each entry of $\mathrm{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$ is $\frac{2A}{\alpha}$ within the $\frac{\alpha}{A}$-restricted domain. This allows us to apply Lemma 2 to obtain the result in (35).

Moving forward from (35), we again apply the property that the unique minimum of $\sum_{k=1}^{K}\mathrm{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$ is attained at $\mu = \frac{1}{K}\sum_{k=1}^{K}[\mu^{k,\dagger}]_\alpha$, which leads to

$$\begin{aligned}\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) \le& \min_\mu\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right) + \frac{8A(1+\ln K)}{\alpha}\\ =& \sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|[\mu^\star]_\alpha\right) + \frac{8A(1+\ln K)}{\alpha}\\ \le& (1-\alpha)\sum_{k=1}^{K}\mathrm{KL}\left(\mu^{k,\dagger}\|\mu^\star\right) + \frac{8A(1+\ln K)}{\alpha},\end{aligned} \tag{36}$$

where the second step uses the definition that $\mu^\star = \frac{1}{K}\sum_{k=1}^K \mu^{k,\dagger}$, and the last step is by the (joint) convexity of the Kullback–Leibler divergence. Substituting (36) to (34) yields

$$\frac{1}{K}\sum_{k=1}^K \mathrm{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right) \le \frac{1}{K}\sum_{k=1}^K \mathrm{KL}\left(\mu^{k,\dagger}\|\mu^\star\right) + \frac{8A(1+\ln K)}{K\alpha} + 4H\bar{\tau}\alpha\ln\frac{A}{\alpha}.$$

By a similar argument, we can show an analogous result for the minimizing player:

$$\frac{1}{K}\sum_{k=1}^K \mathrm{KL}\left(\nu^{k,\dagger}\|\tilde{\nu}^k\right) \le \frac{1}{K}\sum_{k=1}^K \mathrm{KL}\left(\nu^{k,\dagger}\|\nu^\star\right) + \frac{8B(1+\ln K)}{K\alpha} + 4H\bar{\tau}\alpha\ln\frac{B}{\alpha}$$

Substituting the above results back into (33) and using the definition

$$\Delta_{\mu,\nu} = \sum_{k=1}^K \left(\mathrm{KL}\left(\mu^{k,\dagger}\|\mu^\star\right) + \mathrm{KL}\left(\nu^{k,\dagger}\|\nu^\star\right)\right),$$

we obtain that

$$\frac{1}{K}\sum_{k=1}^K \text{NE-gap}(\bar{\mu}^k,\bar{\nu}^k) \le \frac{192H^3}{KT}\left(\Delta_{\mu,\nu} + \frac{10(A+B)\ln K}{\alpha} + 4KH\bar{\tau}\alpha\ln\frac{AB}{\alpha^2}\right)$$

Further using the conditions that $\alpha = 1/\sqrt{K}$ and $\bar{\tau} \le 4H\log T$ (see (17) for a proof) yields

$$\frac{1}{K}\sum_{k=1}^K \text{NE-gap}(\bar{\mu}^k,\bar{\nu}^k) \le \frac{192H^3}{T}\left(\frac{\Delta_{\mu,\nu}}{K} + \frac{10(A+B)\log K}{\sqrt{K}} + \frac{16H^2\log T\log(ABK)}{\sqrt{K}}\right).$$

This completes the proof of the theorem. □

C Infinite-Horizon Discounted Markov Potential Game

To be consistent with existing results in the literature, we consider an infinite-horizon γ-discounted reward setting for MPGs [43, 39, 76, 15]. An N-player, infinite-horizon, discounted stochastic (or Markov) game $\mathbb{G}$ is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, P, \{r_i\}_{i=1}^N, \gamma, \rho)$, where (1) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of players (or agents); (2) $\mathcal{S}$ is the finite state space; (3) $\mathcal{A}_i$ is the finite action space for agent $i \in \mathcal{N}$; (4) $P : \mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, where $\mathcal{A} = \times_{i=1}^N \mathcal{A}_i$ is the joint action space, and $P(\cdot|s,a) \in \Delta(\mathcal{S})$ denotes the distribution over the next state for $a \in \mathcal{A}$; (5) $r_i : \mathcal{S}\times\mathcal{A} \to [-1, 1]$ is the reward function for agent i; (6) $\gamma \in [0, 1)$ denotes the discount factor; and (7) $\rho \in \Delta(\mathcal{S})$ is the initial state distribution. Both the reward function and the state transition function depend on the joint actions of all the agents. We use $a_i \in \mathcal{A}_i$ to denote the individual action of agent $i \in \mathcal{N}$. The subscript $-i$ to denotes the set of agents excluding agent i, i.e., $\mathcal{N}\backslash\{i\}$. We can rewrite $a = (a_i, a_{-i})$ using this convention. Let $S = |\mathcal{S}|$, $A_i = |\mathcal{A}_i|, \forall i \in \mathcal{N}$, and $A_{\max} = \max_{i\in\mathcal{N}} A_i$.

A (Markov) policy $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$ for agent $i \in \mathcal{N}$ is a mapping from the state space to a distribution over the action space. We let agent i's policy be parameterized by $\theta_i = \{\theta_i(a_i|s) \in \mathbb{R}\}_{s\in\mathcal{S},a_i\in\mathcal{A}_i}$, and denote the policy by π_{θ_i} to emphasize such parameterization. Important examples include direct policy parameterization $\pi_{\theta_i}(a_i|s) = \theta_i(a_i|s)$ and softmax parameterization $\pi_{\theta_i}(a_i|s) = \exp(\theta_i(a_i|s))/\sum_{a_i'\in\mathcal{A}_i}\exp(\theta_i(a_i'|s)), \forall s \in \mathcal{S}, a_i \in \mathcal{A}_i$. Let Θ_i denote the parameterization-dependent[1] space where θ_i takes values from, and let $\Theta = \times_{i=1}^N \Theta_i$. A joint (product) policy $\pi_\theta = (\pi_{\theta_1}, \ldots, \pi_{\theta_N})$ induces a probability measure over the sequence of states and joint actions. When the policy parameterization scheme is fixed, we sometimes denote a policy π_θ (resp. π_{θ_i}) simply by its parameter θ (resp. θ_i). For a joint policy $\theta = (\theta_1, \ldots, \theta_N)$, and for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we define the value function and the state-action value function (or Q-function) for agent i as follows:

$$V_i^s(\theta;\mathbb{G}) := \mathbb{E}_{\theta,\mathbb{G}}\Big[\sum_{t=0}^\infty \gamma^t r_i(s^t, a^t) \mid s^0 = s\Big], \tag{37}$$

$$Q_i^{s,a}(\theta;\mathbb{G}) := \mathbb{E}_{\theta,\mathbb{G}}\Big[\sum_{t=0}^\infty \gamma^t r_i(s^t, a^t) \mid s^0 = s, a^0 = a\Big].$$

[1] For example, direct parameterization requires that $\theta_{s,a_i} \ge 0$ and $\sum_{a_i\in\mathcal{A}_i}\theta_{s,a_i} = 1, \forall s \in \mathcal{S}, a_i \in \mathcal{A}_i$, while softmax parameterization allows for $\Theta_i = \mathbb{R}^{|\mathcal{S}||\mathcal{A}_i|}$.

For each agent i, by averaging over the other agents' policies, we define the averaged Q-function $\bar{Q}_i^{s,a_i}$ of a joint policy $\theta = (\theta_i, \theta_{-i})$ for any $s \in \mathcal{S}, a_i \in \mathcal{A}_i$ as:

$$\bar{Q}_i^{s,a_i}(\theta; \mathbb{G}) := \sum_{a_{-i} \in \mathcal{A}_{-i}} \theta_{-i}(a_{-i}|s) Q_i^{s,(a_i,a_{-i})}(\theta; \mathbb{G}).$$

With a slight abuse of notation, we write $V_i^\rho(\theta; \mathbb{G}) := \mathbb{E}_{s\sim\rho}[V_i^s(\theta; \mathbb{G})]$ for a state distribution $\rho \in \Delta(\mathcal{S})$. We sometimes also suppress the notation of $\mathbb{G}$ when it is clear from context.

Each agent seeks to find a policy that maximizes its own cumulative reward. The notion of Nash equilibrium in such an infinite-horizon discounted reward setting is defined as follows.

Definition 1. *(Nash Equilibrium). For any $\varepsilon \geq 0$, a joint (product) policy $\theta^\star = (\theta_i^\star, \theta_{-i}^\star)$ is an ε-approximate (Markov perfect) Nash equilibrium of a game $\mathbb{G}$ if*

$$V_i^s(\theta_i^\star, \theta_{-i}^\star; \mathbb{G}) \geq V_i^s(\theta_i, \theta_{-i}^\star; \mathbb{G}) - \varepsilon, \forall i \in \mathcal{N}, \theta_i \in \Theta_i, s \in \mathcal{S}.$$

In the infinite-horizon setting, a Markov game $\mathbb{G}$ is a Markov potential game (MPG) if there exists a global potential function $\Phi : \Theta \times \mathcal{S} \to \mathbb{R}$, such that for any state $s \in \mathcal{S}$, any $i \in \mathcal{N}$, and any $\theta_i, \theta_i' \in \Theta_i, \theta_{-i} \in \Theta_{-i}$:

$$\Phi_s(\theta_i, \theta_{-i}; \mathbb{G}) - \Phi_s(\theta_i', \theta_{-i}; \mathbb{G}) = V_i^s(\theta_i, \theta_{-i}; \mathbb{G}) - V_i^s(\theta_i', \theta_{-i}; \mathbb{G}). \tag{38}$$

Intuitively, MPGs capture the variations of the agents' individual values by a single global potential function. MPGs cover Markov teams [36] as a special case, a cooperative setting where all agents share the same reward function $r = r_i, \forall i \in \mathcal{N}$. We also write $\Phi(\theta; \mathbb{G}) := \mathbb{E}_{s\sim\rho}[\Phi_s(\theta; \mathbb{G})]$ for the initial state distribution $\rho \in \Delta(\mathcal{S})$. By linearity of expectation, $\Phi(\theta_i, \theta_{-i}; \mathbb{G}) - \Phi(\theta_i', \theta_{-i}; \mathbb{G}) = V_i^\rho(\theta_i, \theta_{-i}; \mathbb{G}) - V_i^\rho(\theta_i', \theta_{-i}; \mathbb{G})$. One can easily show that there exists a constant $\Phi_{\max} \in [0, \frac{2N}{1-\gamma}]$, such that $|\Phi(\theta; \mathbb{G}) - \Phi(\theta'; \mathbb{G})| \leq \Phi_{\max}, \forall \theta, \theta' \in \Theta$. Finally, we define the discounted state visitation distribution of policy θ on game $\mathbb{G}$ as

$$d_\rho^\theta(s; \mathbb{G}) = (1-\gamma)\mathbb{E}_{s^0\sim\rho} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{\theta,\mathbb{G}}(s^t = s|s_0).$$

Subsequently, the distribution mismatch coefficient of game $\mathbb{G}$ is defined as $\kappa(\mathbb{G}) = \sup_{\theta\in\Theta} \|d_\theta^\rho(\cdot\,; \mathbb{G})/\rho\|_\infty$. For a set $\mathcal{G}$ of games, we let $\kappa = \sup_{\mathbb{G}\in\mathcal{G}} \kappa(\mathbb{G})$.

D Supplementary Material for Section 4

D.1 Proof of Theorem 3

Proof. Proposition 1 implies that if the agents run projected Q-descent on the Markov potential game $\mathbb{G}^k$ for T iterations, we have

$$\sum_{t=0}^{T-1} \max_{i\in\mathcal{N}} \left(\max_{\theta_i'\in\Theta_i} V_i^\rho(\theta_i', \theta_{-i}^{k,t}; \mathbb{G}^k) - V_i^\rho(\theta_i^{k,t}, \theta_{-i}^{k,t}; \mathbb{G}^k) \right) \leq \sqrt{\frac{\kappa(\mathbb{G}^k)T(\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k,0}, \mathbb{G}^k))}{\alpha(1-\gamma)^2}}. \tag{39}$$

From the Cauchy-Schwarz inequality, we have that

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K} \sqrt{\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k,0}; \mathbb{G}^k)} &\leq \sqrt{\frac{1}{K}\sum_{k=1}^{K} \left(\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k,0}; \mathbb{G}^k)\right)} \\
&\leq \sqrt{\frac{1}{K}\left(2\Phi_{\max} + \sum_{k=1}^{K-1} \left(\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k+1,0}; \mathbb{G}^{k+1})\right)\right)} \\
&\leq \sqrt{\frac{1}{K}\left(2\Phi_{\max} + \sum_{k=1}^{K-1} \left(\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k,T}; \mathbb{G}^{k+1})\right)\right)} \\
&\leq \sqrt{\frac{1}{K}\left(2\Phi_{\max} + \Delta_\Phi\right)}
\end{aligned}$$

where the third inequality uses the outer stage update rule that $\theta^{k+1,0}=\theta^{k,T}$, and the last inequality follows from the definition of the similarity metric Δ_Φ. Plugging the above result into (39), we have that

$$\frac{1}{K}\frac{1}{T}\sum_{k=1}^{K}\sum_{t=0}^{T-1}\max_{i\in\mathcal{N}}\left(\max_{\theta_i'\in\Theta_i}V_i^\rho(\theta_i',\theta_{-i}^{k,t};\mathbb{G}^k)-V_i^\rho(\theta_i^{k,t},\theta_{-i}^{k,t};\mathbb{G}^k)\right)$$
$$\le\sqrt{\frac{\kappa(2\Phi_{\max}+\Delta_\Phi)}{\alpha(1-\gamma)^2KT}}\le\sqrt{\frac{8\kappa^4NA_{\max}(2\Phi_{\max}+\Delta_\Phi)}{(1-\gamma)^6KT}},$$

where in the second inequality we set the learning rate as $\alpha=\frac{(1-\gamma)^4}{8\kappa^3NA_{\max}}$. Therefore, for an average game, $T=O\left(\frac{NA_{\max}\kappa^4(\Phi_{\max}+\Delta_\Phi)}{K(1-\gamma)^6\varepsilon^2}\right)$ steps in the inner stage suffice to find an ε-approximate Nash equilibrium. □

D.2 Model-Agnostic Meta-Learning in Markov Potential Games

In what follows, we study meta-learning in MPG under the same formulation as MAML [20, 17, 30]. Let $\mathcal{G}=\{\mathbb{G}^j\}$ be a set of different infinite-horizon discounted reward Markov potential games. The games are drawn from a fixed distribution p that we can sample from. Each game is defined by a tuple $\mathbb{G}^j=(\mathcal{N},\mathcal{S},\{\mathcal{A}_i\}_{i=1}^N,P^j,\{r_i^j\}_{i=1}^N,\gamma,\rho^j)$, where we assume without loss of generality that the games share the same agent set, state & action spaces and discount factor, but can have different transition and reward functions and initial state distributions. MAML tries to learn a good initialization from which running one or a few steps of gradient descents/ascents with respect to a new task lead to well-performing model parameters. In the case of multi-agent meta-reinforcement learning with one gradient ascent step, the problem can be formulated as

$$\max_{\theta\in\Theta}F_1(\theta):=\mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\theta+\alpha\nabla\Phi(\theta;\mathbb{G});\mathbb{G}\right)\right], \tag{40}$$

where $\alpha>0$ is the step size of the policy gradient update. Such a formulation can also be extended to multiple steps of policy gradients. Let $\zeta(\cdot\,;\mathbb{G})$ denote the operator of performing one step of policy gradient update on game $\mathbb{G}$, i.e., $\zeta(\theta;\mathbb{G}):=\theta+\alpha\nabla\Phi(\theta;\mathbb{G})$. The T-step extension of the objective (40) can be written as

$$\max_{\theta\in\Theta}F_T(\theta):=\mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\zeta(\ldots(\zeta(\theta;\mathbb{G}))\ldots);\mathbb{G}\right)\right], \tag{41}$$

where the operator $\zeta(\cdot\,;\mathbb{G})$ is applied T times.

Optimizing the multi-step MAML objective typically involves two nested stages: The inner stage (or base algorithm) runs multiple steps of gradient ascents for each individual task, while the outer stage (or meta-algorithm) is an iterative process that updates the meta-parameter θ over different tasks. Specifically, suppose the outer stage runs for K iterations. Let θ^k denote the value of θ at the beginning of the k-th iteration of the outer stage. In each iteration, we sample games from the set $\mathcal{G}$ according to the distribution p. For each individual game $\mathbb{G}\in\mathcal{G}$ encountered during iteration k, the inner stage runs T steps of gradient ascent (or its variants) on it:

$$\theta^{k,t+1}(\mathbb{G})\leftarrow\psi(\theta^{k,t}(\mathbb{G});\mathbb{G}),\ \text{for } 0\le t\le T-1, \tag{42}$$

where $\theta^{k,0}(\mathbb{G})=\theta^k,\forall\mathbb{G}\in\mathcal{G}$. We often suppress the notation of $\mathbb{G}$ in $\theta^{k,t}(\mathbb{G})$ when there is no ambiguity. Finally, the outer stage updates the meta-parameter by

$$\theta^{k+1}\leftarrow\Psi(\theta^k,\mathcal{G}), \tag{43}$$

using a certain update rule Ψ. The meta-parameter θ^{k+1} is then used as the initialization $\theta^{k+1,0}$ for iteration $k+1$. For simplicity of presentation, we present our results in the same setting as in [66] where $\mathcal{G}$ consists of a finite set of M games and p is a uniform distribution. Our results can be easily extended to the settings where there is an infinite number of games and p is a generic probability distribution, as has been done in existing works [17, 18, 30].

In the following, we develop a meta-learning procedure (ψ,Ψ) that finds a stationary point of the meta-objective (41) while at the same time converging to an approximate Nash equilibrium for each individual game encountered, assuming a sufficient number of policy gradient steps are taken in each game. We focus on softmax parameterization where each agent's policy is given by

$\pi_{\theta_i}(a_i|s) = \exp(\theta_i(a_i|s))/\sum_{a_i' \in \mathcal{A}_i} \exp(\theta_i(a_i'|s)), \forall s \in \mathcal{S}, a_i \in \mathcal{A}_i$. In the inner stage, each agent independently runs gradient ascents with respect to its own value functions to update its parameters. Specifically, on each game $\mathbb{G} \in \mathcal{G}$ encountered during the k-th outer iteration, agent i updates its policy parameter θ_i by

$$\theta_i^{k,t+1}(\mathbb{G}) \leftarrow \theta_i^{k,t}(\mathbb{G}) + \alpha \nabla_{\theta_i} V_i^\rho(\theta^{k,t}(\mathbb{G}); \mathbb{G}), \forall 0 \leq t \leq T-1. \tag{44}$$

We sometimes omit the dependence of $\theta_i^{k,t}(\mathbb{G})$ on $\mathbb{G}$ when the game is clear from the context. Using (the multi-agent extension of) the policy gradient theorem [60, 76], the gradient $\nabla_{\theta_i} V_i^\rho(\theta; \mathbb{G})$ can be calculated as

$$\frac{\partial V_i^\rho(\theta; \mathbb{G})}{\partial \theta_i(a_i|s)} = \frac{1}{1-\gamma} d_\rho^\theta(s; \mathbb{G}) \pi_{\theta_i}(a_i|s) \bar{A}_i^{s,a_i}(\theta; \mathbb{G}), \tag{45}$$

where $d_\rho^\theta(s; \mathbb{G}) = (1-\gamma)\mathbb{E}_{s^0 \sim \rho} \sum_{t=0}^\infty \gamma^t \mathbb{P}_{\theta,\mathbb{G}}(s^t = s|s_0)$ is the discounted state visitation distribution, and $\bar{A}_i^{s,a_i}(\theta; \mathbb{G})$ is the averaged advantage function. Unbiased estimators of the policy gradient can be constructed by using the sampler from [1]. For simplicity, we assume that the exact policy gradients are given. It follows from the definition of the potential function (38) that $\nabla_{\theta_i} V_i^\rho(\theta; \mathbb{G}) = \nabla_{\theta_i} \Phi(\theta; \mathbb{G})$, which indicates that independent policy gradient updates with individual value functions (44) is equivalent to running centralized gradient ascents with respect to the potential function (42). Hence, the base algorithm for each individual game can be executed in a decentralized way. Finally, we invoke Theorem 5 of [78] to show that under mild assumptions, our policy gradient updates with softmax parameterization (44) find an approximate Nash equilibrium of each individual game. Specifically, for any $\varepsilon > 0$, if we run the inner stage for sufficient number of steps $T = O(1/\varepsilon^2)$, our method will find an ε-approximate NE for each individual game.

Our outer stage follows the MAML algorithm by running gradient ascent with respect to the meta-objective F_T from (40). The gradient of F_T can be written as

$$\nabla F_T(\theta) = \mathbb{E}_{\mathbb{G} \sim p(\mathcal{G})} \left[\Big(\prod_{t=0}^{T-1} \big(I + \alpha \nabla^2 \Phi(\theta^{(t)}(\mathbb{G}); \mathbb{G}) \big) \Big) \nabla \Phi(\theta^{(T)}(\mathbb{G}); \mathbb{G}) \right], \tag{46}$$

where $\theta^{(0)}(\mathbb{G}) = \theta$ and $\theta^{(t+1)}(\mathbb{G}) = \Psi(\theta^{(t)}(\mathbb{G}); \mathbb{G})$. Accordingly, we instantiate the outer stage update (43) as

$$\theta^{k+1} \leftarrow \theta^k + \frac{\eta}{|\mathcal{G}|} \sum_{\mathbb{G} \in \mathcal{G}} \Big(\prod_{t=0}^{T-1} \big(I + \alpha \nabla^2 \Phi(\theta^{k,t}(\mathbb{G}); \mathbb{G}) \big) \Big) \nabla \Phi(\theta^{k,T}(\mathbb{G}); \mathbb{G}), \tag{47}$$

where $\eta > 0$ is the learning rate of the outer stage. We assume for simplicity that the exact values of the policy gradient $\nabla \Phi(\theta^{k,T}(\mathbb{G}); \mathbb{G})$ and the policy Hessian $\nabla^2 \Phi(\theta^{k,t}(\mathbb{G}); \mathbb{G})$ are given. In practice, one can construct unbiased estimators of the policy gradient from samples, as the policy gradient and policy Hessian can be written explicitly in a closed form that is compatible with samplers (Lemma 15). We remark that the policy Hessian depends on the cross terms of the agents' policy parameters, which can only be calculated in a centralized way. Our inner stage, though, can still be executed in a decentralized manner. Our algorithm hence falls into in the regime of centralized (meta-)training with decentralized (meta-)execution [42], a popular strategy used for training MARL algorithms.

In order to establish the convergence of (47) to the stationary point of the meta-objective (40), we first show the smoothness of the meta-objective through the following sequence of lemmas.

Lemma 12. *Under softmax parameterization, for any policy parameter $\theta \in \Theta$, any state $s \in \mathcal{S}$ and any joint action $a \in \mathcal{A}$, we have (i) $\|\nabla_\theta \log \pi_\theta(a|s)\| \leq \sqrt{2N}$, and (ii) $\left\|\nabla_\theta^2 \log \pi(a|s)\right\| \leq 2$. Furthermore, for any policy parameters $\theta, \theta' \in \Theta$, we have (iii) $\left\|\nabla_\theta^2 \log \pi_\theta(a|s) - \nabla_\theta^2 \log \pi_{\theta'}(a|s)\right\| \leq 12 \|\theta - \theta'\|$.*

Lemma 13. *Under softmax parameterization, for any Markov potential game $\mathbb{G} \in \mathcal{G}$, any policy parameters $\theta, \theta' \in \Theta$, any state $s \in \mathcal{S}$ and any joint action $a \in \mathcal{A}$, the potential function Φ satisfies the following properties:*

(i) Bounded policy gradient: $\|\nabla \Phi(\theta; \mathbb{G})\| \leq B_G := \frac{\sqrt{2N}}{(1-\gamma)^2}$;

(ii) Bounded policy Hessian: $\left\|\nabla^2 \Phi(\theta; \mathbb{G})\right\| \leq L_G := \frac{6N}{(1-\gamma)^3}$;

(iii) Lipschitz policy Hessian: $\left\|\nabla^2 \Phi(\theta; \mathbb{G}) - \nabla^2 \Phi(\theta'; \mathbb{G})\right\| \leq L_H \|\theta - \theta'\|$, where $L_H := \frac{56 N^{3/2}}{(1-\gamma)^4}$.

Lemma 14. *(Meta-objective smoothness). Consider running* (44) *with softmax parameterization and* $\alpha = \frac{(1-\gamma)^3}{2N\gamma A_{\max}}$ *as the inner stage and running* (47) *as the outer stage. Then, the meta-objective* (41) *is L_F-smooth for $L_F = (\alpha T B_G L_H + L_G)2^{2T}$.*

The smoothness constant L_F has an exponential dependence on the number of inner stage update steps T, which seems unavoidable even in supervised meta-learning. Based on the smoothness property, we can show that our method finds a stationary point of the meta-objective (Theorem 4).

D.3 Proof of Lemma 12

Proof. For agent $i \in \mathcal{N}$, for any state $s \in \mathcal{S}$ and action $a_i \in \mathcal{A}_i$, the softmax policy with parameter θ_i can be written as

$$\pi_{\theta_i}(a_i|s) = \frac{\exp(\mathbf{1}_{s,a_i}^\top \theta_i)}{\sum_{a_i' \in \mathcal{A}_i} \exp(\mathbf{1}_{s,a_i'}^\top \theta_i)},$$

where $\theta_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}_i|}$, and $\mathbf{1}_{s,a_i}$ is an $|\mathcal{S}||\mathcal{A}_i|$-dimensional one-hot vector that has a 1 at index (s, a_i) and 0s at all the other indices. It is known that (see, e.g., [1])

$$\frac{\partial \log \pi_{\theta_i}(a_i|s)}{\partial \theta_i(a_i'|s')} = \mathbb{1}[s = s'](\mathbb{1}[a = a'] - \pi_{\theta_i}(a'|s)),$$

where $\mathbb{1}[\cdot]$ is the indicator function. Hence, we have

$$\|\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|s)\| \leq \sqrt{2}. \tag{48}$$

Since we consider product policies, for any joint action $a = (a_1, \ldots, a_N)$, we have $\pi_\theta(a|s) = \prod_{i=1}^N \pi_{\theta_i}(a_i|s)$. Therefore, it holds that

$$\|\nabla_\theta \log \pi_\theta(a|s)\|^2 \leq \sum_{i=1}^N \|\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|s)\|^2 \leq 2N.$$

We can hence conclude that $\|\nabla_\theta \log \pi_\theta(a|s)\| \leq \sqrt{2N}$. This completes the proof of result (i). Next, to show result (ii), we first write the Hessian $\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s)$ as (see, e.g., [18] for a proof)

$$\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s) = -\mathbb{E}_{a_i' \sim \pi_{\theta_i}(a_i'|s)} \left[\left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]\right) \left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]\right)^\top \right].$$

To find the upper bound and Lipschitz constant of $\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s)$, we will rely on two technical lemmas from [18], reproduced as Lemmas 3 and 4 in Appendix A. Since $\|\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|s)\| \leq 2$, from Lemma 4, we know that $\mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]$ is Lipschitz continuous with constant 2. By the definition of $\mathbf{1}_{s,a_i}$, we have $\left\|\mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]\right\| \leq 1$. Since for any matrix A, a sub-multiplicative matrix norm $\|\cdot\|$ satisfies $\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty$, we can conclude that

$$\left\| \left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]\right) \left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}]\right)^\top \right\| \leq 2. \tag{49}$$

Further, by Lemma 3, the term in (49) is Lipschitz continuous with constant 8. By applying Lemma 4 one more time, we know that

$$\left\|\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s)\right\| \leq 2, \text{ and } \left\|\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s) - \nabla^2_{\theta_i'} \log \pi_{\theta_i'}(a_i|s)\right\| \leq 12 \|\theta_i - \theta_i'\|. \tag{50}$$

Since $\nabla^2_\theta \log \pi_\theta(a|s)$ is a block diagonal matrix, we apply the result on the block diagonal matrix norm in Lemma 5 to show that

$$\left\|\nabla^2_\theta \log \pi_\theta(a|s)\right\| \leq \max_{i \in \mathcal{N}} \left\|\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s)\right\| \leq 2.$$

This completes the proof of result (ii). To show result (iii), we again apply Lemma 5 to conclude that

$$\left\|\nabla^2_\theta \log \pi_\theta(a|s) - \nabla^2_\theta \log \pi_{\theta'}(a|s)\right\| \leq \max_{i \in \mathcal{N}} \left\|\nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s) - \nabla^2_{\theta_i'} \log \pi_{\theta_i'}(a_i|s)\right\| \leq 12 \|\theta - \theta'\|,$$

where the last step is by (50). This completes the proof of the lemma. □

D.4 Proof of Lemma 13

In the following, since there is no possibility of ambiguity, we drop the dependence on $\mathbb{G}$ and simply write $\nabla\Phi(\theta;\mathbb{G})$ and $V_i^{\rho}(\theta;\mathbb{G})$ as $\nabla\Phi(\theta)$ and $V_i^{\rho}(\theta)$, respectively.

To establish Lemma 13, we first derive an explicit formula for the policy Hessian $\nabla^2\Phi(\theta)$. Notice that $\nabla^2\Phi(\theta)$ can be written as a block matrix with $N \times N$ blocks:

$$\nabla^2\Phi(\theta) = \begin{bmatrix} \nabla^2_{1,1}\Phi(\theta) & \dots & \nabla^2_{1,N}\Phi(\theta) \\ \vdots & \ddots & \vdots \\ \nabla^2_{N,1}\Phi(\theta) & \dots & \nabla^2_{N,N}\Phi(\theta) \end{bmatrix}, \tag{51}$$

where in each block $\nabla^2_{i,j}\Phi(\theta) \in \mathbb{R}^{|\mathcal{A}_i|\times|\mathcal{A}_j|}$ we first take the gradient of Φ with respect to agent i's policy parameters θ_i and then take the gradient with respect to agent j's parameters θ_j, i.e., $\nabla^2_{i,j}\Phi(\theta) = \frac{\partial^2\Phi}{\partial\theta_i\partial\theta_j}, \forall i,j \in \mathcal{N}$. The following lemma states that each $\nabla^2_{i,j}\Phi(\theta)$ block can be written in an explicit form. This lemma can be considered as a multi-agent extension of Theorem 3 in [24]. For clarity of presentation, we defer its proof to Appendix D.5.

Lemma 15. *Each matrix block $\nabla^2_{i,j}\Phi(\theta)$ in the policy Hessian matrix* (51) *takes the form*

$$\nabla^2_{i,j}\Phi(\theta) = \mathcal{H}_1^{i,j}(\theta) + \mathcal{H}_2^{i,j}(\theta) + \mathcal{H}_{12}^{i,j}(\theta) + (\mathcal{H}_{12}^{i,j})^\top(\theta).$$

The matrices $\mathcal{H}_1^{i,j}(\theta), \mathcal{H}_2^{i,j}(\theta)$, and $\mathcal{H}_{12}^{i,j}(\theta)$ can be written as

$$\begin{aligned}
\mathcal{H}_1^{i,j}(\theta) &= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_\rho^\theta(s,a) Q_i^{s,a}(\theta)\nabla_{\theta_i}\log\pi_\theta(a|s)\nabla_{\theta_j}^\top\log\pi_\theta(a|s), \\
\mathcal{H}_2^{i,j}(\theta) &= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_\rho^\theta(s,a) Q_i^{s,a}(\theta)\nabla^2_{\theta_i\theta_j}\log\pi_\theta(a|s), \\
\mathcal{H}_{12}^{i,j}(\theta) &= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_\rho^\theta(s,a) \nabla_{\theta_i}\log\pi_\theta(a|s)\nabla_{\theta_j}^\top Q_i^{s,a}(\theta),
\end{aligned}$$

where we define $d_\rho^\theta(s,a) := d_\rho^\theta(s)\cdot\pi_\theta(a|s)$ for $d_\rho^\theta(s) = (1-\gamma)\mathbb{E}_{s^0\sim\rho}\sum_{t=0}^\infty\gamma^t\mathbb{P}_\theta(s^t = s|s_0)$.

The next lemma states that each matrix block $\nabla^2_{i,j}\Phi(\theta)$ is Lipschitz continuous with respect to θ. The proof is deferred to Appendix D.6.

Lemma 16. *Each matrix block $\nabla^2_{i,j}\Phi(\theta)$ in the policy Hessian matrix* (51) *is Lipschitz continuous:*

$$\left\|\nabla^2_{i,j}\Phi(\theta) - \nabla^2_{i,j}\Phi(\theta')\right\| \le L_{ij}\left\|\theta-\theta'\right\|, \forall i,j\in\mathcal{N},$$

where the Lipschitz constant satisfies $L_{ij} \le \frac{56\sqrt{N}}{(1-\gamma)^4}$.

Equipped with the results from Lemma 15 and Lemma 16, we are now ready to prove Lemma 13.

Proof (of Lemma 13).

Proof of (i): From the definition of the potential function (38), we know that $\nabla_{\theta_i}\Phi(\theta) = \nabla_{\theta_i}V_i^\rho(\theta)$, and hence $\nabla\Phi(\theta) = (\nabla_{\theta_1}V_1^\rho(\theta),\dots,\nabla_{\theta_N}V_N^\rho(\theta))$. For each agent i, the policy gradient theorem states that

$$\nabla_{\theta_i}V_i^\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\theta, a_i\sim\pi_{\theta_i}(\cdot|s)}\left[\nabla_{\theta_i}\log\pi_{\theta_i}(a_i|s)\bar{Q}_i^{s,a_i}(\theta)\right].$$

Since (48) from Lemma 13 suggests that $\|\nabla_{\theta_i}\log\pi_{\theta_i}(a_i|s)\| \le \sqrt{2}$, we obtain $\|\nabla_{\theta_i}V_i^\rho(\theta)\| \le \frac{\sqrt{2}}{(1-\gamma)^2}$. Hence, $\|\nabla\Phi(\theta)\| \le \frac{\sqrt{2N}}{(1-\gamma)^2}$.

Proof of (ii): See Lemma 29 of [78].

Proof of (iii): From the above reasoning, we know that $\nabla^2\Phi(\theta)$ can be written as a block matrix $\nabla^2\Phi(\theta) = [\nabla^2_{i,j}\Phi(\theta)]_{1\le i,j\le N}$, and Lemma 16 implies that each such block is Lipschitz continuous

$$\left\|\nabla^2_{i,j}\Phi(\theta) - \nabla^2_{i,j}\Phi(\theta')\right\| \le L_{ij}\left\|\theta-\theta'\right\|, \forall i,j\in\mathcal{N},$$

with $L_{ij} \leq \frac{56\sqrt{N}}{(1-\gamma)^4}$. We can then use Lemma 6 to conclude that $\nabla^2\Phi(\theta)$ is also Lipschitz

$$\left\|\nabla^2\Phi(\theta) - \nabla^2\Phi(\theta')\right\| \leq \frac{56N^{3/2}}{(1-\gamma)^4}\left\|\theta - \theta'\right\|.$$

This completes the proof of Lemma 13. □

D.5 Proof of Lemma 15

Proof. The proof follows steps similar to those used in the proof of Theorem 3 in [24]. We first introduce a few notations. Let $s^{0:t}$ denote the sequence of states $(s^0, \ldots, s^t)$, and let $a^{0:t} := (a^0, \ldots, a^t)$, where $a^t = (a_1^t, \ldots, a_N^t)$ is the joint action at time step t. Further, let

$$p_\theta(s^{0:t}, a^{0:t} \mid \rho) := \mathbb{P}_\theta(s^{0:t}, a^{0:t}|s^0 \sim \rho) = \rho(s^0)\prod_{\tau=0}^{t-1}\left(\pi_\theta(a^\tau|s^\tau)P(s^{\tau+1}|s^\tau, a^\tau)\right)\pi_\theta(a^t|s^t). \tag{52}$$

From the definition in (37), we have

$$V_i^\rho(\theta) = \mathbb{E}_\theta\left[\sum_{t=0}^{\infty}\gamma^t r_i(s^t, a^t) \mid s^0 \sim \rho\right] = \sum_{t=0}^{\infty}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t}, a^{0:t}|\rho) r_i(s^t, a^t).$$

Using the definition of the potential function (38), we know that

$$\nabla_{\theta_i}\Phi(\theta) = \nabla_{\theta_i}V_i^\rho(\theta) = \sum_{t=0}^{\infty}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t}, a^{0:t} \mid \rho)\nabla_{\theta_i}\log p_\theta(s^{0:t}, a^{0:t} \mid \rho) r_i(s^t, a^t),$$

where we used the fact that $\nabla p_\theta = p_\theta \nabla \log p_\theta$. The second-order partial derivative can hence be written as

$$\begin{aligned}\nabla^2_{i,j}\Phi(\theta) =& \underbrace{\sum_{t=0}^{\infty}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t}, a^{0:t} \mid \rho)\nabla^2_{\theta_i\theta_j}\log p_\theta(s^{0:t}, a^{0:t}) r_i(s^t, a^t)}_{①} \\ &+ \underbrace{\sum_{t=0}^{\infty}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t}, a^{0:t} \mid \rho)\nabla_{\theta_i}\log p_\theta(s^{0:t}, a^{0:t} \mid \rho)\nabla^\top_{\theta_j}\log p_\theta(s^{0:t}, a^{0:t} \mid \rho) r_i(s^t, a^t)}_{②}\end{aligned}$$

From (52), we can see that $\nabla^2_{\theta_i\theta_j}\log p_\theta(s^{0:t}, a^{0:t} \mid \rho) = \sum_{\tau=0}^t \nabla^2_{\theta_i\theta_j}\log \pi_\theta(a^\tau|s^\tau)$. Hence, the first term in the above equation can be written as

$$\begin{aligned}① =& \sum_{t=0}^{\infty}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t}, a^{0:t} \mid \rho)\sum_{\tau=0}^{t}\nabla^2_{\theta_i\theta_j}\log\pi_\theta(a^\tau|s^\tau) r_i(s^t, a^t) \\ =& \sum_{\tau=0}^{\infty}\gamma^\tau\sum_{s^\tau}\sum_{a^\tau}p_\theta(s^\tau, a^\tau \mid \rho)\nabla^2_{\theta_i\theta_j}\log\pi_\theta(a^\tau|s^\tau)\sum_{t=\tau}^{\infty}\gamma^{t-\tau}\sum_{s^t}\sum_{a^t}\mathbb{P}_\theta(s^t, a^t|s^\tau, a^\tau) r^i(s^t, a^t) \\ =& \sum_{\tau=0}^{\infty}\gamma^\tau\sum_{s^\tau}\sum_{a^\tau}p_\theta(s^\tau, a^\tau \mid \rho)\nabla^2_{\theta_i\theta_j}\log\pi_\theta(a^\tau|s^\tau) Q_i^{s^\tau, a^\tau}(\theta) \\ =& \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}d_\rho^\theta(s, a) Q_i^{s,a}(\theta)\nabla^2_{\theta_i\theta_j}\log\pi_\theta(a|s) \\ =& \mathcal{H}_2^{i,j}(\theta).\end{aligned}$$

The second term can be written as

$$
\begin{aligned}
② = &\sum_{t=0}^{\infty}\sum_{\tau=0}^{t}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^\tau|s^\tau)\nabla_{\theta_j}^\top\log\pi_\theta(a^\tau|s^\tau)r_i(s^t,a^t)\\
&+\sum_{t=0}^{\infty}\sum_{\tau_2=0}^{t}\sum_{\tau_1=0}^{\tau_2-1}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^{\tau_1}|s^{\tau_1})\nabla_{\theta_j}^\top\log\pi_\theta(a^{\tau_2}|s^{\tau_2})r_i(s^t,a^t)\\
&+\sum_{t=0}^{\infty}\sum_{\tau_1=0}^{t}\sum_{\tau_2=0}^{\tau_1-1}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^{\tau_1}|s^{\tau_1})\nabla_{\theta_j}^\top\log\pi_\theta(a^{\tau_2}|s^{\tau_2})r_i(s^t,a^t).
\end{aligned}
\tag{53}
$$

By switching the order of summations and following a similar procedure as in the derivation of ①, we can show that the first term on the RHS of (53) is equal to $\mathcal{H}_1^{i,j}(\theta)$. The second and third terms on the RHS of (53) can be shown to be $\mathcal{H}_{12}^{i,j}(\theta)$ and $(\mathcal{H}_{12}^{i,j})^\top(\theta)$, respectively. We skip the rest of the proof as it follows the same procedure as in the proof of Theorem 3 in [24]. $\square$

D.6 Proof of Lemma 16

Proof. Recall from Lemma 15 that

$$\nabla_{i,j}^2\Phi(\theta)=\mathcal{H}_1^{i,j}(\theta)+\mathcal{H}_2^{i,j}(\theta)+\mathcal{H}_{12}^{i,j}(\theta)+(\mathcal{H}_{12}^{i,j})^\top(\theta).$$

For any (s,a), we write

$$
\begin{aligned}
h_1^{i,j}(\theta)=&Q_i^{s,a}(\theta)\nabla_{\theta_i}\log\pi_\theta(a|s)\nabla_{\theta_j}^\top\log\pi_\theta(a|s),\\
h_2^{i,j}(\theta)=&Q_i^{s,a}(\theta)\nabla_{\theta_i\theta_j}^2\log\pi_\theta(a|s),\\
h_{12}^{i,j}(\theta)=&\nabla_{\theta_i}\log\pi_\theta(a|s)\nabla_{\theta_j}^\top Q_i^{s,a}(\theta),
\end{aligned}
$$

and hence $\nabla_{i,j}^2\Phi(\theta)$ can be rewritten as

$$\nabla_{i,j}^2\Phi(\theta)=\frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}d_\rho^\theta(s,a)\left(h_1^{i,j}(\theta)+h_2^{i,j}(\theta)+h_{12}^{i,j}(\theta)+(h_{12}^{i,j})^\top(\theta)\right).$$

In the following, we proceed by showing that each of the three terms $h_1^{i,j}(\theta)$, $h_2^{i,j}(\theta)$, and $h_{12}^{i,j}(\theta)$ is bounded and Lipschitz.

(i) Analysis of $h_1^{i,j}(\theta)$: First, notice that $|Q_i^{s,a}(\theta)|\le\frac{1}{1-\gamma}$. From the Bellman equation $Q_i^{s,a}(\theta)=r_i(s,a)+\gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V_i^{s'}(\theta)]$, we have $\nabla Q_i^{s,a}(\theta)=\gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[\nabla V_i^{s'}(\theta)]$. The policy gradient theorem states that

$$\nabla_{\theta_i}V_i^\rho(\theta)=\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\theta,a_i\sim\pi_{\theta_i}(\cdot|s)}\left[\nabla_{\theta_i}\log\pi_{\theta_i}(a_i|s)\bar{Q}_i^{s,a_i}(\theta)\right].$$

Since (48) from Lemma 12 suggests $\|\nabla_{\theta_i}\log\pi_{\theta_i}(a_i|s)\|\le\sqrt{2}$, we obtain $\|\nabla_{\theta_i}V_i^\rho(\theta)\|\le\frac{\sqrt{2}}{(1-\gamma)^2}$. Hence, $\|\nabla Q_i^{s,a}(\theta)\|\le\frac{\sqrt{2}\gamma}{(1-\gamma)^2}$, and $Q_i^{s,a}(\theta)$ is Lipschitz continuous with constant $\frac{\sqrt{2}\gamma}{(1-\gamma)^2}$. In addition, the proof of Lemma 12 implies that $\nabla_{\theta_i}\log\pi_\theta(a|s)$ is bounded by $\sqrt{2}$ and is 2-Lipschitz continuous. Further using Lemma 3, we can conclude that

$$\left\|h_1^{i,j}(\theta)\right\|\le\frac{2}{1-\gamma}\text{ and }\left\|h_1^{i,j}(\theta)-h_1^{i,j}(\theta')\right\|\le\frac{2\sqrt{2}(2-\gamma)}{(1-\gamma)^2}\left\|\theta-\theta'\right\|. \tag{54}$$

(ii) Analysis of $h_2^{i,j}(\theta)$: From step (i) of the proof, we know that $Q_i^{s,a}(\theta)$ is bounded by $\frac{1}{1-\gamma}$ and is $\frac{\sqrt{2}\gamma}{(1-\gamma)^2}$-Lipschitz continuous. Since π_θ is a product policy, for $i\ne j$, we simply have $\nabla_{\theta_i\theta_j}^2\log\pi_\theta(a|s)=0$. For $i=j$, we know from (50) that $\|\nabla_{\theta_i\theta_j}^2\log\pi_\theta(a|s)\|\le 2$, and $\|\nabla_{\theta_i\theta_j}^2\log\pi_\theta(a|s)-\nabla_{\theta_i\theta_j}^2\log\pi_{\theta'}(a|s)\|\le 12\left\|\theta_i-\theta_i'\right\|$. Therefore, we obtain from Lemma 3 that

$$h_2^{i,j}(\theta)=0,\text{ if } i\ne j;\text{ and }\left\|h_2^{i,j}(\theta)\right\|\le\frac{2}{1-\gamma},\left\|h_2^{i,j}(\theta)-h_2^{i,j}(\theta')\right\|\le\frac{8(2-\gamma)}{(1-\gamma)^2}\left\|\theta-\theta'\right\|,\text{ if } i=j. \tag{55}$$

(iii) Analysis of $h_{12}^{i,j}(\theta)$: In the following, we first establish the Lipschitz continuity of $\nabla_{\theta_j} Q_i^{s,a}(\theta)$, which can be shown in a similar manner as in Lemma A.2 of [74] and is reproduced below for completeness. Let

$$p_\theta(s^{0:t}, a^{0:t} \mid s, a) := \mathbb{P}_\theta(s^{0:t}, a^{0:t} | s^0 = s, a^0 = a) = \prod_{\tau=0}^{t-1} \pi_\theta(a^{\tau+1}|s^{\tau+1}) P(s^{\tau+1}|s^\tau, a^\tau).$$

By the definition of the Q-function (37),

$$Q_i^{s,a}(\theta) = \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_\theta(s^{0:t}, a^{0:t} \mid s, a) r_i(s^t, a^t)$$

The gradient of $Q_i^{s,a}(\theta)$ can hence be written as

$$\begin{aligned}
\nabla_{\theta_j} Q_i^{s,a}(\theta) &= \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_\theta(s^{0:t}, a^{0:t} \mid s, a) \nabla_{\theta_j} \log p_\theta(s^{0:t}, a^{0:t} \mid s, a) r_i(s^t, a^t) \\
&= \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_\theta(s^{0:t}, a^{0:t} \mid s, a) \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) r_i(s^t, a^t).
\end{aligned}$$

To show the Lipschitz continuity of $Q_i^{s,a}(\theta)$, we first write

$$\begin{aligned}
&\left|\nabla_{\theta_j} Q_i^{s,a}(\theta) - \nabla_{\theta_j} Q_i^{s,a}(\theta')\right| \\
\leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t \Big| p_\theta(s^{0:t}, a^{0:t}|s, a) \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau) \Big| \\
\leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t \left| p_\theta(s^{0:t}, a^{0:t}|s, a) - p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \right| \left\| \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) \right\| \qquad (56) \\
&+ \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \left\| \sum_{\tau=1}^{t} \left(\nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau) \right) \right\|. \qquad (57)
\end{aligned}$$

In the following, we upper bound each of the two terms above separately. To analyze (56), we first apply the mean-value theorem to the function $\prod_{\tau=1}^{t} \pi_\theta(a^\tau|s^\tau)$ of θ and obtain

$$\begin{aligned}
\left| \prod_{\tau=1}^{t} \pi_\theta(a^\tau|s^\tau) - \prod_{\tau=1}^{t} \pi_{\theta'}(a^\tau|s^\tau) \right| &= \left| (\theta - \theta')^\top \left[\sum_{m=1}^{t} \nabla \pi_{\tilde{\theta}}(a^m|s^m) \prod_{\tau \neq m, \tau=1}^{t} \pi_{\tilde{\theta}}(a^\tau|s^\tau) \right] \right| \\
&\leq \|\theta - \theta'\| \cdot \sum_{m=1}^{t} \left\| \nabla \log \pi_{\tilde{\theta}}(a^m|s^m) \right\| \cdot \prod_{\tau=1}^{t} \pi_{\tilde{\theta}}(a^\tau|s^\tau) \\
&\leq \sqrt{2N} t \, \|\theta - \theta'\| \cdot \prod_{\tau=1}^{t} \pi_{\tilde{\theta}}(a^\tau|s^\tau),
\end{aligned}$$

where $\tilde{\theta} = \lambda\theta + (1-\lambda)\theta'$ for some $\lambda \in [0, 1]$, the first inequality uses the fact that $\nabla \pi_{\tilde{\theta}}(a^m|s^m) = \pi_{\tilde{\theta}}(a^m|s^m) \nabla \log \pi_{\tilde{\theta}}(a^m|s^m)$, and the second inequality is due to Lemma 12 (i). Using the above property, we obtain

$$\begin{aligned}
&\left| p_\theta(s^{0:t}, a^{0:t}|s, a) - p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \right| \\
=& \left| \prod_{\tau=0}^{t-1} \pi_\theta(a^{\tau+1}|s^{\tau+1}) P(s^{\tau+1}|s^\tau, a^\tau) - \prod_{\tau=0}^{t-1} \pi_{\theta'}(a^{\tau+1}|s^{\tau+1}) P(s^{\tau+1}|s^\tau, a^\tau) \right| \\
\leq& \prod_{\tau=0}^{t-1} P(s^{\tau+1}|s^\tau, a^\tau) \cdot \sqrt{2N} t \, \|\theta - \theta'\| \cdot \prod_{\tau=1}^{t} \pi_{\tilde{\theta}}(a^\tau|s^\tau) \\
=& p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s, a) \cdot \sqrt{2N} t \, \|\theta - \theta'\|.
\end{aligned}$$

Substituting the above equation back into (56) yields

$$(56) \leq \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \sqrt{2N} t \gamma^t p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s, a) \cdot \left\| \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) \right\| \cdot \|\theta - \theta'\|$$
$$\leq \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} 2\sqrt{N} t^2 \gamma^t p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s, a) \|\theta - \theta'\|,$$

where the second step uses (48) from Lemma 12 and the fact that π_θ is a product policy.

To upper bound (57), we apply Lemma 12 (ii) and obtain

$$(57) \leq \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \sum_{\tau=1}^{t} \left\| \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau) \right\|$$
$$\leq \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} 2t \gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \|\theta - \theta'\|.$$

Substituting the above upper bounds back into (56) and (57), we have

$$\begin{aligned}
&\left| \nabla_{\theta_j} Q_i^{s,a}(\theta) - \nabla_{\theta_j} Q_i^{s,a}(\theta') \right| \\
\leq &\sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t \left(2\sqrt{N} t^2 p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s, a) + 2t p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \right) \|\theta - \theta'\| \\
= &\sum_{t=0}^{\infty} \gamma^t \left(2\sqrt{N} t^2 + 2t \right) \|\theta - \theta'\| \\
\leq &\frac{4\sqrt{N}\gamma(1+\gamma)}{(1-\gamma)^3} \|\theta - \theta'\|,
\end{aligned}$$

where the second step holds because $\sum_{a^{0:t}} \sum_{s^{0:t}} p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s, a) = 1$. The last step uses the facts that $2t \leq 2\sqrt{N} t^2$, and that

$$\sum_{t=1}^{\infty} \gamma^t \cdot t^2 = \frac{1}{1-\gamma} \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \cdot t^2 = \frac{1}{1-\gamma} \cdot \mathbb{E}[T^2] = \frac{1}{1-\gamma} \cdot \frac{\gamma(1+\gamma)}{(1-\gamma)^2},$$

where T is a random variable following a geometric distribution. We have hence derived that $\nabla_{\theta_j} Q_i^{s,a}(\theta)$ is Lipschitz continuous with constant $\frac{4\sqrt{N}\gamma(1+\gamma)}{(1-\gamma)^3}$.

Following the same reasoning as in step (i), we obtain that $\nabla_{\theta_i} \log \pi_\theta(a|s)$ is bounded by $\sqrt{2}$ and is 2-Lipschitz continuous. Similar to step (i), we can also use the Bellman equation and the policy gradient theorem to show that $\|\nabla_{\theta_j}^\top Q_i^{s,a}(\theta)\| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}$. Again, by applying Lemma 3, we can conclude that

$$\left\| h_{12}^{i,j}(\theta) \right\| \leq \frac{2\gamma}{(1-\gamma)^2} \text{ and } \left\| h_{12}^{i,j}(\theta) - h_{12}^{i,j}(\theta') \right\| \leq \frac{6\sqrt{2N}\gamma(1+\gamma)}{(1-\gamma)^3}. \tag{58}$$

(iv) Putting everything together: Let $h(\theta) := h_1^{i,j}(\theta) + h_2^{i,j}(\theta) + h_{12}^{i,j}(\theta) + (h_{12}^{i,j})^\top(\theta)$. Using the simple observation that the sum of two Lipschitz continuous functions is also Lipschitz continuous, we obtain from (54), (55), and (58) that

$$\|h(\theta)\| \leq \frac{4}{(1-\gamma)^2}, \text{ and } \|h(\theta) - h(\theta')\| \leq \frac{50\sqrt{N}}{(1-\gamma)^3} \|\theta - \theta'\|. \tag{59}$$

Recall from Lemma 15 that

$$\nabla_{i,j}^2 \Phi(\theta) = \frac{1}{1-\gamma} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_\rho^\theta(s, a) h(\theta).$$

By adding and subtracting the same value,

$$
\begin{aligned}
&\left\|\nabla^2_{i,j}\Phi(\theta)-\nabla^2_{i,j}\Phi(\theta')\right\|\\
\leq&\frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left\|d^{\theta}_{\rho}(s,a)h(\theta)-d^{\theta'}_{\rho}(s,a)h(\theta')\right\|\\
\leq&\frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left(\left|d^{\theta}_{\rho}(s,a)-d^{\theta'}_{\rho}(s,a)\right|\|h(\theta)\|+d^{\theta'}_{\rho}(s,a)\|h(\theta)-h(\theta')\|\right)\\
\leq&\frac{4}{(1-\gamma)^3}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left|d^{\theta}_{\rho}(s,a)-d^{\theta'}_{\rho}(s,a)\right|+\frac{50\sqrt{N}}{(1-\gamma)^4}\|\theta-\theta'\|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}d^{\theta'}_{\rho}(s,a)\\
\leq&\frac{56\sqrt{N}}{(1-\gamma)^4}\|\theta-\theta'\|.
\end{aligned}
$$

The third step uses the upper bounds from (59). The fourth step can be derived by using the following result from Equation (A.67) of [74]:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left|d^{\theta}_{\rho}(s,a)-d^{\theta'}_{\rho}(s,a)\right|\leq\frac{\sqrt{2N}}{1-\gamma}\|\theta-\theta'\|.$$

This completes the proof of the Lipschitz continuity that $\left\|\nabla^2_{i,j}\Phi(\theta)-\nabla^2_{i,j}\Phi(\theta')\right\|\leq L_{ij}\|\theta-\theta'\|,\forall i,j\in\mathcal{N}$ for $L_{ij}=\frac{56\sqrt{N}}{(1-\gamma)^4}$. ☐

D.7 Proof of Lemma 14

Proof. Recall from (46) that the gradient of the meta-objective can be written as

$$\nabla F_T(\theta)=\mathbb{E}_{\mathbb{G}\sim\text{Unif}(\mathcal{G})}\left[\Big(\prod_{t=0}^{T-1}\big(I+\alpha\nabla^2\Phi(\theta^{(t)}(\mathbb{G});\mathbb{G})\big)\Big)\nabla\Phi(\theta^{(T)}(\mathbb{G});\mathbb{G})\right],$$

where $\theta^{(0)}(\mathbb{G})=\theta$ and $\theta^{(t+1)}(\mathbb{G})=\Psi(\theta^{(t)}(\mathbb{G});\mathbb{G})$. It suffices to show that for each individual game $\mathbb{G}\in\mathcal{G}$, the term

$$\Big(\prod_{t=0}^{T-1}\big(I+\alpha\nabla^2\Phi(\theta^{(t)}(\mathbb{G});\mathbb{G})\big)\Big)\nabla\Phi(\theta^{(T)}(\mathbb{G});\mathbb{G})\tag{60}$$

is Lipschitz continuous. In the following, we drop the dependence on $\mathbb{G}$ and simply write $\theta^{(t)}(\mathbb{G})$ and $\nabla\Phi(\theta^{(t)}(\mathbb{G});\mathbb{G})$ as $\theta^{(t)}$ and $\nabla\Phi(\theta^{(t)})$, respectively.

We proceed by finding the upper bound and Lipschitz constant of each individual term in (60). First, from Lemma 13(ii), we know that $\left\|I+\alpha\nabla^2\Phi(\theta^{(t)})\right\|\leq 1+\alpha L_G,\forall 0\leq t\leq T-1$. By using the chain rule, we also know that

$$\nabla_\theta\theta^{(t)}=\prod_{t'=0}^{t-1}(I+\alpha\nabla^2\Phi(\theta^{(t')})).$$

Hence, since $\left\|I+\alpha\nabla^2\Phi(\theta^{(t)})\right\|\leq 1+\alpha L_G,\forall 0\leq t\leq T-1$, we know that $\theta^{(t)}$ is Lipschitz continuous with constant $(1+\alpha L_G)^t$. Further, combining Lemma 13 (iii) with the fact that the Lipschitz constant of a composite function is equal to the product of the Lipschitz constants of the base functions, we conclude that $I+\alpha\nabla^2\Phi(\theta^{(t)})$ is Lipschitz (with respect to θ) with constant $\alpha L_H(1+\alpha L_G)^t$. For the case of $T\geq 2$, Lemma 3 thus implies that the $\prod_{t=0}^{T-1}(I+\alpha\nabla^2\Phi(\theta^{(t)}))$ factor from (60) is Lipschitz with constant $\alpha T L_H(1+\alpha L_G)^{2T-1}$, while for $T=1$, the Lipschitz constant is simply αL_H.

For the $\nabla\Phi(\theta^{(T)})$ factor in (60), we know from Lemma 13(i) that it is bounded by B_G. Using Lemma 13(iii) and the Lipschitzness of a composite function, we also know that $\nabla\Phi(\theta^{(T)})$ is $L_G(1+\alpha L_G)^T$-Lipschitz continuous. Finally, along with the results that the $\prod_{t=0}^{T-1}(I+\alpha\nabla^2\Phi(\theta^{(t)}))$ factor is bounded by $(1+\alpha L_G)^T$ and Lipschitz with constant $\alpha T L_H(1+\alpha L_G)^{2T-1}$, we again apply Lemma 3 to obtain that (60) is Lipschitz continuous with constant $\alpha T B_G L_H(1+\alpha L_G)^{2T-1}+L_G(1+\alpha L_G)^{2T}$. Using the fact that $\alpha\in(0,1/L_G]$, we can conclude that the meta-objective $F_T(\theta)$ is L_F-smooth with $L_F=(\alpha T B_G L_H+L_G)2^{2T}$. ☐

D.8 Proof of Theorem 4

Proof. Based on the aforementioned series of lemmas, we are now ready to establish Theorem 4. The proof follows from standard analysis in non-convex optimization. Since the meta-objective function is L_F-smooth (Lemma 14), the smoothness property implies that

$$F_T(\theta^{k+1}) \geq F_T(\theta^k) + \nabla F_T(\theta^k)^\top(\theta^{k+1} - \theta^k) - \frac{L_F}{2}\left\|\theta^{k+1} - \theta^k\right\|^2.$$

Using the outer stage update rule (47) that

$$\theta^{k+1} = \theta^k + \eta\nabla F_T(\theta^k),$$

we obtain

$$F_T(\theta^{k+1}) \geq F_T(\theta^k) + \eta\left\|\nabla F_T(\theta^k)\right\|^2 - \frac{L_F\eta^2}{2}\left\|\nabla F_T(\theta^k)\right\|^2 \geq F_T(\theta^k) + \frac{1}{2L_F}\left\|\nabla F_T(\theta^k)\right\|^2,$$

where the last step uses $\eta = 1/L_F$. Summing the above inequality over k and rearranging the terms lead to

$$\sum_{k=0}^{K-1}\left\|\nabla F_T(\theta^k)\right\|^2 \leq 2L_F\sum_{k=0}^{K-1}(F_T(\theta^{k+1}) - F_T(\theta^k)) = 2L_F(F_T(\theta^K) - F_T(\theta^0)) \leq \frac{4NL_F}{1-\gamma},$$

where the last step holds because $|\Phi(\theta;\mathbb{G}) - \Phi(\theta';\mathbb{G})| \leq \Phi_{\max} \leq \frac{2N}{1-\gamma}, \forall\theta,\theta' \in \Theta, \mathbb{G} \in \mathcal{G}$. Therefore, for $K \geq \frac{4NL_F}{(1-\gamma)\varepsilon^2}$, we have

$$\min_{0\leq k\leq K-1}\left\|\nabla F_T(\theta^k)\right\|^2 \leq \frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla F_T(\theta^k)\right\|^2 \leq \frac{4NL_F}{K(1-\gamma)} \leq \varepsilon^2.$$

This completes the proof of the theorem. □

E Supplementary Material for Section 5

E.1 Base Algorithm

In this appendix, we first describe our base algorithm for learning CCE in a general-sum Markov game, which was omitted in the main text due to space limitations. The optimistic online mirror descent algorithm for learning CCE in a general-sum Markov game is presented in Algorithm 2. Similar to Algorithm 1 for zero-sum Markov games, Algorithm 2 performs optimistic online mirror descent [55, 62] for policy updates in order to establish initialization-dependent convergence. Algorithm 2 also utilizes stage-based value updates to avoid the need for a complicated no-weighted-regret analysis. Different from Algorithm 1, the output policy $\bar{\pi}$ of Algorithm 2 is no longer a state-wise average policy but rather a correlated policy. The construction of $\bar{\pi}$, similar to the construction of the "certified policies" in the literature, is described in Algorithm 3.

We further introduce a few notations similar to the zero-sum game setting. For any (τ, h, s), we define the per-state regret for player $i \in \mathcal{N}$ as

$$\mathrm{reg}_{h,i}^\tau(s) := \max_{\pi_{h,i}^{\tau,\dagger}(\cdot|s)\in\Delta(\mathcal{A}_i)} \frac{1}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\left\langle \pi_{h,i}^{\tau,\dagger} - \pi_{h,i}^j, Q_{h,i}^\tau \pi_{h,-i}^j\right\rangle(s). \tag{61}$$

We define the maximal regret (over the states and all the players) as

$$\mathrm{reg}_h^\tau := \max_{s\in\mathcal{S}}\max_{i\in\mathcal{N}}\{\mathrm{reg}_{h,i}^\tau(s)\}.$$

Lemma 17 provides an upper bound of the per-state regret (61), which further leads us to the following initialization-dependent convergence guarantee of Algorithm 2. We finally define $\delta_{h,i}^t := \max_{s\in\mathcal{S}}(V_{h+1,i}^{\dagger,\bar{\pi}_{h,-i}^t} - V_{h,i}^{\bar{\pi}_h^t})(s)$, and let $\delta_h^t := \max_{i\in\mathcal{N}}\delta_{h,i}^t$.

Algorithm 2: Optimistic Online Mirror Descent for CCE in General-Sum Markov Game

Input: Initial policies $\tilde{\pi} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A}_{\text{all}})$;
Set stage index $\tau \leftarrow 1$, $t^{\text{start}}_{\tau} \leftarrow 1$, and $L_\tau \leftarrow H$;
Initialize: $\pi^0_h = \hat{\pi}^0_h \leftarrow \tilde{\pi}^1_h$, and $Q^\tau_h \leftarrow \mathbf{0}, \forall h \in [H]$;
for *iteration* $t \leftarrow 1$ *to* T **do**
  Auxiliary policy update: for each player $i \in \mathcal{N}$, step $h \in [H]$ and state $s \in \mathcal{S}$:

$$\hat{\pi}^t_{h,i}(\cdot|s) \leftarrow \operatorname*{argmax}_{\mu \in \Delta(\mathcal{A}_i)} \eta \left\langle \mu, [Q^\tau_{h,i}\pi^{t-1}_{h,-i}](s,\cdot) \right\rangle - D_R(\mu, \hat{\pi}^{t-1}_{h,i}(\cdot|s));$$

  Policy update: for each player $i \in \mathcal{N}$, step $h \in [H]$ and state $s \in \mathcal{S}$:

$$\pi^t_{h,i}(\cdot|s) \leftarrow \operatorname*{argmax}_{\mu \in \Delta(\mathcal{A}_i)} \eta \left\langle \mu, [Q^\tau_{h,i}\pi^{t-1}_{h,-i}](s,\cdot) \right\rangle - D_R(\mu, \hat{\pi}^{t}_{h,i}(\cdot|s));$$

  if $t - t^{start}_{\tau} + 1 \geq L_\tau$ **then**
    $t^{\text{end}}_\tau \leftarrow t, t^{\text{start}}_{\tau+1} \leftarrow t+1, L_{\tau+1} \leftarrow \lfloor (1 + 1/H) L_\tau \rfloor$;
    Value update: for each $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}_{\text{all}}, i \in \mathcal{N}$:

$$Q^{\tau+1}_{h,i}(s,\boldsymbol{a}) \leftarrow \frac{1}{L_\tau} \sum_{t'=t^{\text{start}}_\tau}^{t^{\text{end}}_\tau} \left(r_{h,i} + P_h[Q^\tau_{h+1,i}\pi^{t'}_{h+1}] \right)(s,\boldsymbol{a});$$

    $\tau \leftarrow \tau + 1$; $\pi^t_h = \hat{\pi}^t_h \leftarrow \tilde{\pi}^\tau_h, \forall h \in [H]$;
Output policy: Sample $t \sim \mathrm{Unif}([T])$. Output $\bar{\pi} := \bar{\pi}^t_1$ as defined in Algorithm 3.

Algorithm 3: Construction of $\bar{\pi}^t_h$

Input: Policy trajectory $\{\pi^t_h\}_{h \in [H], t \in [T]}$ of Algorithm 2;
for *step* $h' \leftarrow h$ *to* H **do**
  Uniformly sample j from $\{t^{\text{start}}_{\tau(t)-1}, t^{\text{start}}_{\tau(t)-1} + 1, \ldots, t^{\text{end}}_{\tau(t)-1}\}$;
  Execute policy π^j_h for step h;
  Set $t \leftarrow j$;

E.2 Proof of Theorem 5

Proof. From the construction of $\bar{\pi}$ (Algorithm 3) and the definition of CCE-gap, we have

$$\begin{aligned}
\text{CCE-gap}(\bar{\pi}) &= \max_{i \in \mathcal{N}} V^{\dagger, \bar{\pi}_{-i}}_{1,i}(s_1) - V^{\bar{\pi}}_{1,i}(s_1) \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \max_{i \in \mathcal{N}} \max_{s \in \mathcal{S}} \left(V^{\dagger, \bar{\pi}^t_{1,-i}}_{1,i}(s) - V^{\bar{\pi}^t_1}_{1,i}(s) \right) \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \delta^t_1.
\end{aligned}$$

Using Lemma 19, the above term can be further bounded by

$$
\begin{aligned}
\text{CCE-gap}(\bar{\pi}) &\leq \frac{1}{T}\sum_{t=1}^{T}\delta_1^t \\
&\leq \sum_{t=1}^{T}\frac{3}{\eta T L_{\tau(t)}}\sum_{h=1}^{H}\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}} D_R(\pi_{h,i}^{\tau(t)-h,\dagger},\tilde{\pi}_{h',i}^{\tau(t)-h}(\cdot|s)) + 36(N-1)^2\eta^2H^4 \\
&= \frac{3}{\eta T}\sum_{\tau=1}^{\bar{\tau}}\sum_{h=1}^{H}\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}} D_R(\pi_{h,i}^{\tau-h,\dagger},\tilde{\pi}_{h,i}^{\tau-h}(\cdot|s)) + 36(N-1)^2\eta^2H^4 \\
&\leq \frac{3}{\eta T}\sum_{\tau=1}^{\bar{\tau}}\sum_{h=1}^{H}\max_{i\in\mathcal{N},s\in\mathcal{S}} D_R(\pi_{h,i}^{\tau,\dagger},\tilde{\pi}_{h,i}^{\tau}(\cdot|s)) + 36(N-1)^2\eta^2H^4,
\end{aligned}
$$

where the last step is simply by changing the counting method. This completes the proof for the first claim in the Theorem.

We now proceed to establish the second statement, which follows a similar argument as in the proof of Theorem 1 for the two-player zero-sum game setting. We repeat the proof below for completeness. Recall that we chose the negative entropy as the regularizer R. The Bregman divergence $D_R(\cdot,\cdot)$ reduces to the Kullback–Leibler divergence. Since $\pi_{h,i}^{\tau,\dagger}$ lies in the simplex, when we initialize $\tilde{\pi}_{h,i}^{\tau}(\cdot|s) = \mathbf{1}/A_i$ to be a uniform distribution, we naturally have $D_R(\pi_{h,i}^{\tau,\dagger},\tilde{\pi}_{h,i}^{\tau}(\cdot|s)) \leq \log A_i, \forall i \in \mathcal{N}, s \in \mathcal{S}$, and $h \in [H]$.

It remains to upper bound the total number of stages $\bar{\tau}$. Recall that we have defined the lengths of the stages to increase exponentially with $L_{\tau+1} = \lfloor (1+1/H)L_\tau \rfloor$. Since the $\bar{\tau}$ stages sum up to T iterations in total, by taking the sum of a geometric series, it suffices to find a value of $\bar{\tau}$ such that $(1+1/H)^{\bar{\tau}} \geq T/H$. Using the Taylor series expansion, one can show that $(1+\frac{1}{H})^H \geq e - \frac{e}{2H}$. Hence, it reduces to finding a minimum $\bar{\tau}$ such that

$$
\left(e - \frac{e}{2H}\right)^{\bar{\tau}/H} \geq \frac{T}{H}. \tag{62}
$$

One can easily see that any $\bar{\tau} \geq \frac{H\log T}{\log(e/2)}$ satisfies the condition. Summarizing the above results, we can conclude that

$$
\text{CCE-gap}(\bar{\pi}) \leq \frac{12H^2\log T}{\eta T}\log A_{\max} + 36(N-1)^2\eta^2H^4.
$$

Choosing $\eta = H^{-2/3}T^{-1/3}(N-1)^{-2/3}$ yields the second claim in the Theorem. □

E.3 Supporting Lemmas for Section 5

Lemma 17. *For every stage $\tau \in \mathbb{N}_+$, every step $h \in [H]$ and every state $s \in \mathcal{S}$, the per-state average regret of player $i \in \mathcal{N}$ is bounded by:*

$$
\text{reg}_{h,i}^{\tau}(s) \leq \frac{1}{\eta L_\tau} D_R(\pi_{h,i}^{\tau,\dagger},\tilde{\pi}_{h,i}^{\tau}(\cdot|s)) + 36(N-1)^2\eta^2H^3. \tag{63}
$$

Proof. Notice that the policy update steps in Algorithm 2 are exactly the same as the optimistic online mirror descent algorithm [55, 62], with the loss vector $g^t = [Q_{h,i}^{\tau}\pi_{h,-i}^{t}](s,\cdot)$ and the recency bias $M^t = [Q_{h,i}^{\tau}\pi_{h,-i}^{t-1}](s,\cdot)$. Since our stage-based value updates assign equal weights to each iteration, we end up with a classic no-(average-)regret learning problem instead of a no-(weighed-)regret learning problem as in [72, 77]. This allows us to directly apply the standard optimistic OMD results (e.g., Lemma 1 in [55] and Proposition 5 in [62]) to obtain

$$\begin{aligned}
\mathrm{reg}_{h,i}^{\tau}(s) = & \max_{\pi_{h,i}^{\tau,\dagger}\in\Delta(\mathcal{A}_i)} \frac{1}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \pi_{h,i}^{\tau,\dagger} - \pi_{h,i}^{j}, Q_{h,i}^{\tau}\pi_{h,-i}^{j}\right\rangle(s) \\
\le & \frac{1}{\eta L_\tau} D_R(\pi_{h,i}^{\tau,\dagger}, \tilde{\pi}_{h,i}^{\tau}(\cdot|s)) + \frac{\eta}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\| [Q_{h,i}^{\tau}\pi_{h,-i}^{j} - Q_{h,i}^{\tau}\pi_{h,-i}^{j-1}](s,\cdot)\right\|_\infty^2 \\
& - \frac{1}{8\eta L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\|\pi_{h,i}^{j}(\cdot\mid s) - \pi_{h,i}^{j-1}(\cdot\mid s)\right\|_1^2 \\
\le & \frac{1}{\eta L_\tau} D_R(\pi_{h,i}^{\tau,\dagger}, \tilde{\pi}_{h,i}^{\tau}(\cdot|s)) + \frac{\eta}{L_\tau}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} 2H^2\left\|\pi_{h,-i}^{j}(\cdot\mid s) - \pi_{h,-i}^{j-1}(\cdot\mid s)\right\|_1^2, \qquad (64)
\end{aligned}$$

where in the last step we used the Hölder's inequality and the fact that $\|Q_h^{\tau}(s,\cdot)\|_\infty \le H$. To further upper bound (64), we apply Lemma 18 to obtain that for any $t \in [t_\tau^{\text{start}}, t_\tau^{\text{end}}]$,

$$\left\|\pi_{h,-i}^{t}(\cdot\mid s) - \pi_{h,-i}^{t-1}(\cdot\mid s)\right\|_1^2 \le 18(N-1)^2\eta H. \qquad (65)$$

We remark that the policy stability condition above has a slightly worse dependence on η than those of the optimistic FTRL algorithms. In particular, Lemma G.4 of [77] has shown a $\left\|\pi_{h,-i}^{t}(\cdot\mid s) - \pi_{h,-i}^{t-1}(\cdot\mid s)\right\|_1^2 \le 16(N-1)^2\eta^2H^2$ condition for optimistic FTRL. This is because unlike optimistic FTRL, optimistic OMD lacks a smoothness condition that directly connects the stability of policies to the stability of utility functions (e.g., Lemma A.5 of [77]). Plugging (65) back into (64) leads to the desired result. □

Lemma 18. *For a fixed τ and any $t \in [t_\tau^{start}, t_\tau^{end}], i \in \mathcal{N}, h \in [H], s \in \mathcal{S}$, the optimistic online mirror descent policy updates in Algorithm 2 satisfy:*

$$\left\|\pi_{h,i}^{t}(\cdot\mid s) - \pi_{h,i}^{t-1}(\cdot\mid s)\right\|_1^2 \le 18\eta H.$$

Consequently,

$$\left\|\pi_{h,-i}^{t}(\cdot\mid s) - \pi_{h,-i}^{t-1}(\cdot\mid s)\right\|_1^2 \le 18(N-1)^2\eta H.$$

Proof. In this proof, since we focus on a fixed $(s,h) \to \mathcal{S}\times[H]$, we will drop the dependence on (s,h) for notational convenience. To prove the first claim in the lemma, we first use the triangle inequality to obtain that

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \le \left\|\pi_i^t - \hat{\pi}_i^t\right\|_1 + \left\|\hat{\pi}_i^t - \hat{\pi}_i^{t-1}\right\|_1 + \left\|\hat{\pi}_i^{t-1} - \pi_i^{t-1}\right\|_1. \qquad (66)$$

In the following, we derive an upper bound for the first term on the RHS of the above inequality. The other two terms on the RHS can be bounded in a similar way.

We know from the Pinsker's inequality that

$$\left\|\pi_i^t - \hat{\pi}_i^t\right\|_1 \le \sqrt{2\,\mathrm{KL}\left(\pi_i^t \| \hat{\pi}_i^t\right)}. \qquad (67)$$

In the following, it suffices to find an upper bound of $\mathrm{KL}\left(\hat{\pi}_i^t \| \pi_i^t\right)$. Recall that Algorithm 2 updates the policies as

$$\pi_i^t = \operatorname*{argmax}_{\mu\in\Delta(\mathcal{A}_i)} \eta\left\langle \mu, [Q_i^{\tau}\pi_{-i}^{t-1}]\right\rangle - D_R(\mu, \hat{\pi}_i^t).$$

Since we chose the negative entropy as the regularizer R, the policy update rule above is known (see Section 5.4.2 of [28]) to be equivalent to the following multiplicative weights update:

$$\pi_i^t(a) = \frac{\hat{\pi}_i^t(a)\exp(\eta[Q_i^{\tau}\pi_{-i}^{t-1}](a))}{\sum_{a'}\hat{\pi}_i^t(a')\exp(\eta[Q_i^{\tau}\pi_{-i}^{t-1}](a'))}, \forall a \in \mathcal{A}_i.$$

Hence, we have that

$$\begin{aligned}
\mathrm{KL}\left(\pi_i^t \| \hat{\pi}_i^t\right) &= \sum_{a\in\mathcal{A}_i} \pi_i^t(a) \ln \frac{\pi_i^t(a)}{\hat{\pi}_i^t(a)} \\
&= \sum_{a\in\mathcal{A}_i} \pi_i^t(a) \ln \frac{\exp(\eta[Q_i^\tau \pi_{-i}^{t-1}](a))}{\sum_{a'} \hat{\pi}_i^t(a') \exp(\eta[Q_i^\tau \pi_{-i}^{t-1}](a'))} \\
&\le \sum_{a\in\mathcal{A}_i} \pi_i^t(a) \ln \frac{\exp(\eta H)}{\sum_{a'} \hat{\pi}_i^t(a')} \\
&= \eta H,
\end{aligned}$$

where the inequality uses the facts that $Q_i^\tau \ge 0$ and $\|Q_i^\tau\|_1 \le H$. Substituting the above result back to (67) leads to

$$\left\|\pi_i^t - \hat{\pi}_i^t\right\|_1 \le \sqrt{2\,\mathrm{KL}\left(\pi_i^t \| \hat{\pi}_i^t\right)} \le \sqrt{2\eta H}.$$

Similar results also hold for the other two terms on the RHS of (66). Therefore, we can conclude that $\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \le 3\sqrt{2\eta H}$ and

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1^2 \le 18\eta H.$$

This proves the first claim in the lemma. To establish the second claim, we use the following simple fact for product distributions:

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

Applying Jensen's inequality yields

$$\left\|\pi_{-i}^t - \pi_{-i}^{t-1}\right\|_1^2 \le \left(\sum_{j\ne i} \left\|\pi_j^t - \pi_j^{t-1}\right\|_1\right)^2 \le (N-1)\sum_{j\ne i} \left\|\pi_j^t - \pi_j^{t-1}\right\|_1^2 \le 18(N-1)^2\eta H.$$

This proves the second claim in the lemma. □

Lemma 19. *For any iteration $t \in [T]$ and any step $h \in [H]$, we have that*

$$\delta_h^t \le \frac{3}{\eta L_{\tau(t)}} \sum_{h'=h}^{H} \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} D_R(\pi_{h',i}^{\tau(t)-h'+h-1,\dagger}, \tilde{\pi}_{h',i}^{\tau(t)-h'+h-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^4.$$

Proof. In the following, when we consider a fixed iteration $t \in [T]$, we drop the notational dependence on t and simply use τ (instead of $\tau(t)$) to denote the stage that iteration t belongs to. For any $h \in [H-1]$, using a similar argument as in Lemma 10 for the zero-sum game setting, one can establish the following recursion for the value estimation error:

$$\delta_h^t \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \delta_{h+1}^j + \mathrm{reg}_h^{\tau-1}, \tag{68}$$

where we recall that $\mathrm{reg}_h^\tau := \max_{s\in\mathcal{S}} \max_{i\in\mathcal{N}} \{\mathrm{reg}_{h,i}^\tau(s)\}$. Using Lemma 17, we can upper bound the regret by

$$\mathrm{reg}_h^\tau \le \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \frac{1}{\eta L_\tau} D_R(\pi_{h,i}^{\tau,\dagger}, \tilde{\pi}_{h,i}^\tau(\cdot|s)) + 36(N-1)^2\eta^2 H^3.$$

We substitute the regret bound above back into the recursion 68 to get that

$$\delta_h^t \le \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \frac{1}{\eta L_{\tau-1}} D_R(\pi_{h,i}^{\tau-1,\dagger}, \tilde{\pi}_{h,i}^{\tau-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^3 + \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \delta_{h+1}^j. \tag{69}$$

Notice that according to the definition in Algorithm 3, the behavior of the policy $\bar{\pi}_h^t$ does not change with t within the same stage τ as it always uniformly sample a time index from the previous stage and execute the corresponding history policy. Consequently, the δ_{h+1}^j term is also unchanged within a stage. Hence, we have

$$\frac{1}{L_{\tau-1}}\sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}}\delta_{h+1}^j=\delta_{h+1}^{\tau-1}.$$

The recursion in (69) can hence be rewritten more succinctly as

$$\delta_h^t\le\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}}\frac{1}{\eta L_{\tau-1}}D_R(\pi_{h,i}^{\tau-1,\dagger},\tilde{\pi}_{h,i}^{\tau-1}(\cdot|s))+36(N-1)^2\eta^2H^3+\delta_{h+1}^{\tau-1}.$$

Applying the above inequality recursively over h leads to

$$\begin{aligned}\delta_h^t\le&\sum_{h'=h}^{H}\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}}\frac{1}{\eta L_{\tau-h'+h-1}}D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger},\tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s))+36(N-1)^2\eta^2H^3(H-h+1)\\ \le&\sum_{h'=h}^{H}\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}}\frac{1}{\eta L_\tau}\left(1+\frac{1}{H}\right)^{h'-h+1}D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger},\tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s))+36(N-1)^2\eta^2H^4\\ \le&\frac{3}{\eta L_\tau}\sum_{h'=h}^{H}\max_{i\in\mathcal{N}}\max_{s\in\mathcal{S}}D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger},\tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s))+36(N-1)^2\eta^2H^4,\end{aligned}\tag{70}$$

where the second step uses our choice of the stage lengths that $L_{\tau+1}=\lfloor(1+1/H)L_\tau\rfloor$, which further implies that

$$\frac{1}{L_{\tau-h'+h-1}}\le\frac{1}{L_\tau}\left(1+\frac{1}{H}\right)^{h'-h+1}.$$

The last step in (70) is due to the fact that $(1+1/H)^H\le e\approx2.71828$. □

E.4 Proof of Theorem 6

Proof. First, recall the definitions of $\tilde{\pi}^k$, $\bar{\pi}^k$ and $\pi_i^{k,\dagger}$. Since we use a negative entropy regularizer R, the Bregman divergence $D_R(\cdot,\cdot)$ reduces to the Kullback–Leibler divergence. Using these notations, our convergence results of learning CCE in an individual game $\mathbb{G}^k$ (Theorem 5) can be written more succinctly as

$$\text{CCE-gap}(\bar{\pi}^k)\le\frac{3}{\eta T}\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right)+36N^2\eta^2H^4.$$

where for ease of notations, we write

$$\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right):=\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}}\sum_{i=1}^{N}\max_{s\in\mathcal{S}}\text{KL}\left(\pi_{h,i}^{k,\tau,\dagger}(\cdot|s)\|\tilde{\pi}_{h,i}^k(\cdot|s)\right).$$

Here, $\pi_{h,i}^{k,\tau,\dagger}(\cdot|s)$ represents the value of $\pi_{h,i}^{\tau,\dagger}(\cdot|s)$ in game $\mathbb{G}^k$. By running Algorithm 2 on a sequence of K games, we have that

$$\frac{1}{K}\sum_{k=1}^{K}\text{CCE-gap}(\bar{\pi}^k)\le\frac{3}{\eta KT}\sum_{k=1}^{K}\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right)+36N^2\eta^2H^4.\tag{71}$$

Recall the notation that $[\mathbf{x}]_\alpha=(1-\alpha)\mathbf{x}+\frac{\alpha}{d}\mathbf{1}$ for $\mathbf{x}\in\mathbb{R}^d$. By applying this notation entry-wise to each probability distribution in $\pi^{k,\dagger}$ and invoking Lemma 1, we obtain that

$$\frac{1}{K}\sum_{k=1}^{K}\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right)\le\frac{1}{K}\sum_{k=1}^{K}\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right)+4H\bar{\tau}\alpha\ln\frac{A_{\max}}{\alpha}.\tag{72}$$

Notice that the conditions of Lemma 1 are satisfied here because we select our initial policies to be $\tilde{\pi}_i^k=\frac{1}{k-1}\sum_{k'=1}^{k-1}[\pi_i^{k',\dagger}]_\alpha,\forall i\in\mathcal{N}$, which assigns a probability of at least $\alpha\mathbf{1}/A_i$ to each action.

Adding and subtracting the same term leads to

$$\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right)=\min_\pi\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)+\min_\pi\sum_{k=1}^{K}\left(\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right)-\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)\right)$$
$$\leq\min_\pi\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)+\frac{8A_{\max}(1+\ln K)}{\alpha},\tag{73}$$

where the minimum π is taken over all policies of the form of $\pi=(\pi_1,\ldots,\pi_N)$ such that $\pi_i:[\bar{\tau}]\times[H]\times\mathcal{S}\to\Delta(\mathcal{A}_i)$. We now turn to establish the second step in (73), which reduces to bounding the following regret where the loss functions are given by the Bregman divergences:

$$\mathrm{reg}=\min_\pi\sum_{k=1}^{K}\left(\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right)-\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)\right).$$

It is known that the unique minimum of $\sum_{k'=1}^{k}\mathrm{KL}([\pi^{k',\dagger}]_\alpha\|\cdot)$ is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\pi^{k',\dagger}]_\alpha$ (see Proposition 1 of [5] for a proof of this claim). Therefore, by letting $\tilde{\pi}_i^k=\frac{1}{k-1}\sum_{k'=1}^{k-1}[\pi_i^{k',\dagger}]_\alpha$, we are essentially running the follow the leader (FTL) algorithm (separately for each entry $(\tau,h,s)\in[\bar{\tau}]\times[H]\times\mathcal{S}$) on the sequence of losses defined by $\sum_{k=1}^{K}\mathrm{KL}([\pi^{k,\dagger}]_\alpha\|\cdot)$. We can then invoke the logarithmic regret guarantee of FTL with respect to Bregman divergences, which was established in [35] and is reproduced as Lemma 2 in Appendix A for completeness.

To show that Lemma 2 is applicable, we remark that the Kullback–Leibler divergence is not Lipschitz continuous near the boundary of the probability simplex, which breaks condition required by Lemma 2. However, by restricting to policies of the form $[\pi_i]_\alpha=(1-\alpha)\pi_i+\frac{\alpha}{A_i}\mathbf{1}$, which is at least $\frac{\alpha}{A_i}$-distance away from the simplex boundary, the Kullback–Leibler divergence is indeed Lipschitz continuous within this $\frac{\alpha}{A_i}$-restricted domain. One can show that the Lipschitz constant of each entry of $\mathrm{KL}([\pi_i^{k,\dagger}]_\alpha\|\cdot)$ is $\frac{2A_{\max}}{\alpha}$ within the $\frac{\alpha}{A_{\max}}$-restricted domain. This allows us to apply Lemma 2 to obtain the result in (73).

Moving forward from (73), we again apply the property that the unique minimum of $\sum_{k'=1}^{k}\mathrm{KL}([\pi^{k',\dagger}]_\alpha\|\cdot)$ is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\pi^{k',\dagger}]_\alpha$, which leads to

$$\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right)\leq\min_\pi\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)+\frac{8A_{\max}(1+\ln K)}{\alpha}$$
$$=\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|[\pi^\star]_\alpha\right)+\frac{8A_{\max}(1+\ln K)}{\alpha}$$
$$\leq(1-\alpha)\sum_{k=1}^{K}\mathrm{KL}\left(\pi^{k,\dagger}\|\pi^\star\right)+\frac{8A_{\max}(1+\ln K)}{\alpha},\tag{74}$$

where the second step uses the definition that $\pi_i^\star=\frac{1}{K}\sum_{k=1}^{K}\pi_i^{k,\dagger}$, and the last step is by the (joint) convexity of the Kullback–Leibler divergence. Substituting (74) to (72) yields

$$\frac{1}{K}\sum_{k=1}^{K}\mathrm{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right)\leq\frac{1}{K}\sum_{k=1}^{K}\mathrm{KL}\left(\pi^{k,\dagger}\|\pi^\star\right)+\frac{8A_{\max}(1+\ln K)}{K\alpha}+4H\bar{\tau}\alpha\ln\frac{A_{\max}}{\alpha}.$$

Further substituting the above result back into (71) and using the definition

$$\Delta_\pi=\sum_{k=1}^{K}\sum_{i=1}^{N}\mathrm{KL}\left(\pi_i^{k,\dagger}\|\pi_i^\star\right),$$

we obtain that

$$\frac{1}{K}\sum_{k=1}^{K}\text{CCE-gap}(\bar{\pi}^k)\leq\frac{3}{\eta KT}\left(\Delta_\pi+\frac{8A_{\max}(1+\ln K)}{\alpha}+4KH\bar{\tau}\alpha\ln\frac{A_{\max}}{\alpha}\right)+36N^2\eta^2H^4.$$

Finally, using the conditions that $\alpha = 1/\sqrt{K}$, $\eta = K^{-1/6}H^{-2/3}T^{-1/3}N^{-2/3}$, and $\bar{\tau} \le 4H \log T$ (see (62) for a proof) yields

$$\frac{1}{K}\sum_{k=1}^{K} \text{CCE-gap}(\bar{\pi}^k) \le \left(\frac{HN}{T}\right)^{\frac{2}{3}} \left(\frac{\Delta_\pi}{K^{5/6}} + \frac{10A_{\max} \ln K}{K^{1/3}} + \frac{52H^2 \ln T \log(A_{\max}K)}{K^{1/3}}\right).$$

This completes the proof of the theorem. $\square$

F Simulations

In this appendix, we provide detailed discussions of our simulation results. We first evaluate our algorithms on a sequence of handcrafted two-player zero-sum Markov games (Appendix F.1) and Markov potential games (Appendix F.2). Then, in Appendix F.3, we further demonstrate the scalability of our methods by considering larger-scale tasks, including a simplified version of the Poker endgame considered in [27] and a 1D linear-quadratic tracking task [37].

F.1 Zero-Sum Markov Games

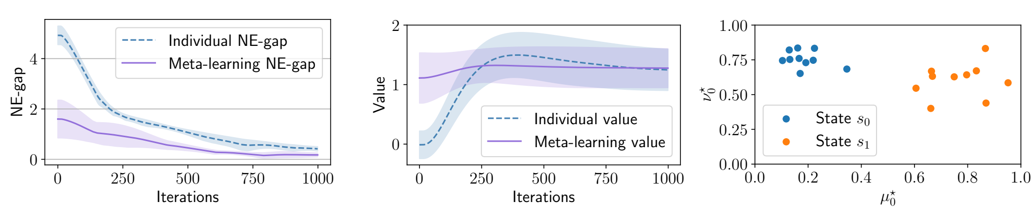
We first evaluate our meta-learning procedure presented in Section 3 on a sequence of $K = 10$ two-player zero-sum Markov games. We generate a sequence of K similar games by first specifying a "base game" and then adding random perturbations to its reward function to get K slightly different games. For our base game, we consider a simple zero-sum game with two states $\mathcal{S} = \{s_0, s_1\}$, where each player has two candidate actions $\mathcal{A} = \{a_0, a_1\}$ and $\mathcal{B} = \{b_0, b_1\}$, respectively. The reward matrices for the max-player at the two states are given in Table 1. We add independent $\mathcal{N}(0, 0.1)$ Gaussian perturbation to each entry of the reward matrix to generate $K = 10$ slightly different games.

s_0	b_0	b_1
a_0	0.5	0
a_1	-1	0.5

s_1	b_0	b_1
a_0	0.5	0
a_1	0.2	1

Table 1: Reward matrices for the max-player in the base game.

To better visualize the similarity level of these games, we plot the NE policies of the two perturbed matrix games in each of the $K = 10$ games. In particular, let $\mu^\star = (\mu_0^\star, \mu_1^\star) \in [0,1]^2$ and $\nu^\star = (\nu_0^\star, \nu_1^\star) \in [0,1]^2$ denote the NE policies of the two players in a certain game. Since $\mu_0^\star + \mu_1^\star = 1$ and $\nu_0^\star + \nu_1^\star = 1$, it suffices to simply use the two values $\mu_0^\star \in [0,1]$ and $\nu_0^\star \in [0,1]$ to characterize the NE policies. Figure 2 (c) plots the relative position of the $(\mu_0^\star, \nu_0^\star)$ pairs of the $K \times 2$ games in the space of $[0,1] \times [0,1]$ to illustrate their closeness, where the $[0,1] \times [0,1]$ space is large enough to cover all possible zero-sum games of the same form. We note that Figure 2 (c) only plots the NE pairs with respect to the perturbed matrix games as defined in Table 1. Due to the existence of the state transitions, the NE policies with respect to the stage Q-functions can be more diversified. In this sense, we can see that our similarity assumption of the games is not too stringent as it allows the games to have relatively diverse NE policies.



(a) Zero-sum game NE-gap (b) Zero-sum game value (c) NE visualization

Figure 2: Average (a) NE-gaps and (b) values of the policies output by individual learning and meta-learning in zero-sum Markov games. Shaded areas denote the standard deviations. (c) visualizes the NE policies of the K games in the normalized space $[0, 1] \times [0, 1]$ to illustrate their closeness.

The state transition function is defined as follows: In both states s_0 and s_1, if the two players take matching actions (namely (a_0, b_0) or (a_1, b_1)), the system stays at the current state with probability 0.9, and transitions to the other state with probability 0.1. On the other hand, if the two players take opposite actions (namely (a_0, b_1) or (a_1, b_0)), the environment will stay at the current state with probability 0.1, and will transition to the other state with probability 0.9.

Each of the K games lasts for $H = 10$ steps, and we run our algorithm for $T = 1000$ iterations on each game. We use a learning rate of $\eta = 0.02$ for Algorithm 1. We evaluate the convergences of the algorithms in terms of $\text{NE-gap}(\mu, \nu) := V_1^{\dagger,\nu}(s_1) - V_1^{\mu,\dagger}(s_1)$, which measures the distances from the output policies to each agent's best response policy. Figure 2 (a) compares the average NE-gap over the K games between individual learning and meta-learning. Figure 2 (b) further compares the average values achieved by the two methods. All results are obtained on a laptop with an Intel Core i5-1240P CPU. We see that compared to learning each task individually, meta-learning can utilize knowledge from previous tasks to attain better policy initialization in a new task and converges to an approximate NE policy (and value) using much fewer iterations.

F.2 Markov Potential Games

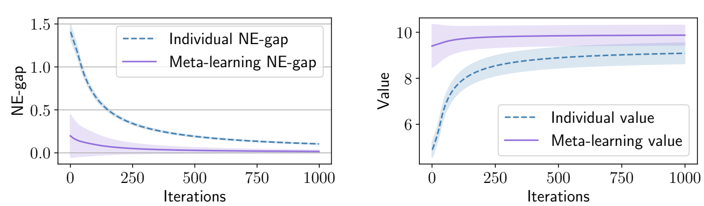
We now evaluate our meta-learning algorithm from Section 4 on a sequence of Markov potential games. We illustrate our algorithm in cooperative games, an important class of MPGs where the agents share the same rewards. We again generate a sequence of K similar games by first specifying a base game and then adding random perturbations to its reward function to get K slightly different games. Our base game has two states $\mathcal{S} = \{s_0, s_1\}$ and each player has two candidate actions $\mathcal{A} = \{a_0, a_1\}$ and $\mathcal{B} = \{b_0, b_1\}$. The shared reward matrices for both players at the two states are given in Table 2. We add independent $\mathcal{N}(0, 0.1)$ Gaussian perturbation to each entry of the reward matrix to generate $K = 10$ slightly different games.

s_0	b_0	b_1
a_0	0.1	0.5
a_1	0.5	1

s_1	b_0	b_1
a_0	0.8	0.2
a_1	0.2	0.8

Table 2: Reward matrices for both players in the base game.

The state transition function is defined in the same way as in Appendix F.1: In both states s_0 and s_1, if the two players take matching actions (namely (a_0, b_0) or (a_1, b_1)), the system stays at the current state with probability 0.9, and transitions to the other state with probability 0.1. On the other hand, if the two players take opposite actions (namely (a_0, b_1) or (a_1, b_0)), the environment will stay at the current state with probability 0.1, and will transition to the other state with probability 0.9.



(a) Potential game NE-gap (b) Potential game value

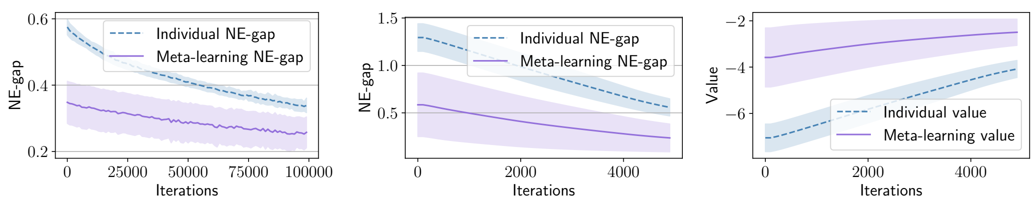
Figure 3: Average (a) NE-gaps and (b) values of the policies output by individual learning and meta-learning in Markov potential games. Shaded areas denote the standard deviations.

Each of the K games lasts for $H = 10$ steps, and we run our algorithm for $T = 1000$ iterations on each game. We use a learning rate of $\alpha = 0.05$ for the independent projected Q-descent algorithm (7). We evaluate the convergences of the algorithms in terms of $\text{NE-gap}(\mu, \nu) := \frac{1}{2}(V_1^{\dagger,\nu}(s_1) + V_1^{\mu,\dagger}(s_1)) - V_1^{\mu,\nu}(s_1)$, which measures the distances from the algorithm's output policies to each agent's best response policy. Figure 3 (a) compares the average NE-gap over the K games between individual learning and meta-learning. Figure 3 (b) further compares the average values achieved by the two methods. Again, we see that meta-learning finds better policy initialization in a new task and converges to an approximate NE policy (and value) using much fewer iterations.

F.3 Scalability

To demonstrate the scalability of our algorithms, we further provide simulation results on some larger-scale tasks including a Poker endgame and a 1D linear-quadratic tracking task.

The Poker endgame that we consider here is a simplified version of the one used in [27]. We use a public River endgame ("Endgame A" of [27]) that was released in the Brains vs AI competition [6]. This task is a zero-sum game with 2 players and roughly 1.7 million states. We simplify the game setup by restricting to 2 actions (namely calling and folding) for each player. Poker is a partially observable game, but we found that our algorithm still performs well if each agent simply uses its local observation as the state. We generate a sequence of $K = 10$ similar games by adding $\mathcal{N}(0, 0.5)$ perturbations to the normalized stack amounts of the players, which essentially perturbs the reward functions. The convergence of the average $\mathrm{NE\text{-}gap}$ over the K games in Figure 4(a) shows that our method can handle such a large state space, and our meta-learning method can converge to an approximate NE policy faster than individual learning.



(a) Poker endgame $\mathrm{NE\text{-}gap}$ (b) LQ tracking $\mathrm{NE\text{-}gap}$ (c) LQ tracking value

Figure 4: Average $\mathrm{NE\text{-}gaps}$ and values of the policies output by individual learning and meta-learning in the Poker endgame and linear-quadratic tracking task. Shaded areas denote the standard deviations.

In the 1D linear-quadratic tracking problem, each agent tries to track the positions of the other agents and stay close to them. We adopt the discrete setting as has been utilized in a few recent works [52, 37, 46], which is an approximation of the classic continuous linear-quadratic formulations. This task has primarily been formulated as a mean-field game, but we consider a finite-agent variant of it in our simulations. Specifically, the task we consider can be modeled as a Markov potential game with 4 players, 625 states, and a joint action space of size 81. For each agent i, let $s_{t,i} \in \mathcal{S}_i$ and $a_{t,i} \in \mathcal{A}_i$, respectively, denote its local state (i.e., position) and local action at time step t, and we write $s_t = (s_{t,1}, \ldots, s_{t,4})$ and $a_t = (a_{t,1}, \ldots, a_{t,4})$. Each agent has 3 candidate actions $\mathcal{A}_i = \{-1, 0, 1\}$ and can stay at 5 different positions $\mathcal{S} = \{-2, -1, 0, 1, 2\}$. The state transition of agent i is given by $s_{t+1,i} = s_{t,i} + a_{t,i}\Delta_t + \sigma\varepsilon_t\sqrt{\Delta_t}$, where Δ_t is the time duration, and ε_t is the i.i.d. noise taking values from $\{-2, -1, 0, 1, 2\}$ following a normal distribution. Let μ_t denote the empirical mean of all the agents' positions at time t, i.e., $\mu_t = \frac{1}{4}\sum_{i=1}^{4} s_{t,i}$. The reward function for agent i is specified as $r_i(s, a) = (-\frac{1}{2}a_{t,i}^2 - \frac{\kappa}{2}(\mu_t - s_{t,i})^2)\Delta_t$. Intuitively, this reward function incentivizes agents to track and stay close to the population (despite the random drift ε_t), but discourages agents from taking large-magnitude actions. We do not consider terminal costs in our simulations. The parameters are set as $\Delta_t = 1, \sigma = 1$, and $\kappa = 0.5$. We generate a sequence of similar games by adding $\mathcal{N}(0, 0.5)$ perturbations to the local state transition drift magnitudes. Figures 4(b) and 4(c) demonstrate that our meta-learning method achieves faster NE-gap and value convergences than individual learning in the linear-quadratic tracking task.