MULTI-AGENT REINFORCEMENT LEARNING FOR NONZERO-SUM MARKOV GAMES

BY

WEICHAO MAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Emeritus Professor Tamer Başar, Chair
Professor Ravishankar K. Iyer
Professor Rayadurgam Srikant
Professor Maxim Raginsky

Abstract

In recent years, multi-agent reinforcement learning (MARL) has shown remarkable capabilities in addressing sequential decision-making problems that involve the strategic interactions of more than one decision-maker. Motivated by the empirical successes, many research efforts have been devoted to lay the theoretical foundations of MARL. In this dissertation, we contribute to this line of theoretical research by developing MARL algorithms with convergence and sample complexity guarantees in nonzero-sum Markov games, a regime that has been barely touched on in prior research. First, we design sample-efficient MARL algorithms for learning (coarse) correlated equilibria in general-sum Markov games. Our algorithms integrate variants of optimistic Q-learning for efficient exploration with uncoupled no-regret learning for policy updates. These algorithms are decentralized in the sense that each agent makes decisions based on only its local information with no need of communication or central coordination. We theoretically establish the sample complexity guarantees for our algorithms, which appear to be the first for decentralized MARL in general-sum Markov games. Second, we study reinforcement learning (RL) under environmental non-stationarity, a major challenge faced by MARL agents. When both the reward functions and the state transition distributions may vary over time, we propose a simple but effective restart-based algorithm particularly tailored to such non-stationary environments. We analyze the dynamic regret of our algorithm and show that it is near-optimal by establishing an almost matching information-theoretical lower bound. We demonstrate that our non-stationary RL method can be readily applied to learning the team-optimal policies in a specific category of cooperative games with slowly-changing opponents. Third, we propose to use meta-learning to transfer useful information across multiple MARL tasks so as to learn related tasks collectively and more efficiently. 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. Numerical results are provided to corroborate our theoretical findings. Finally, we conclude this dissertation and discuss future research directions.

To XZX, for her love and support.

Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Professor Tamer Başar, for his continued support, patience, and encouragement. My research has benefited in countless ways from his knowledgeability and sharp insights. None of my achievements in this dissertation would ever be possible without his help. His kindness, open-mindedness, and strong work ethic have shaped my personality as a researcher and will continue to guide me for the rest of my life.

I am also grateful to the other members of my dissertation committee: Professor Ravishankar K. Iyer, Professor Rayadurgam Srikant, and Professor Maxim Raginsky. They have provided many invaluable comments and suggestions that helped significantly enhance the quality of this dissertation.

This dissertation has greatly benefited from the collaboration with many beautiful minds. I would like to thank my long-time collaborator Haoran Qiu for the inspiring and fruitful discussions that led to multiple research projects related to this dissertation. I would also like to express my gratitude to Kaiqing Zhang and Erik Miehling, who introduced me to the fascinating field of multi-agent reinforcement learning and provided valuable guidance in the early stage of my doctoral research. I am very thankful to Xiangyuan Zhang, Ruihao Zhu, Chen Wang, Hubertus Franke, and Professor Zbigniew Kalbarczyk for sharing their knowledge and expertise in numerous discussions.

Gratitude goes to my fellow labmates: Muhammad Aneeq uz Zaman, Raj Kiriti Velicheti, Shubham Aggarwal, Melih Bastopcu, Abdullah Alawad, Erkan Bayram, and Arda Guclu. I appreciate the friendly, collaborative and fun environment they created together in our group.

I would like to take this opportunity to thank all the fortunate friendships that I have made during my stay at Urbana-Champaign. A special thanks goes to Dawei Sun, Jiaqi Guan, Mengchao Zhang, and Yifan Zhu for being tremendously fun and caring. I would also like to thank my parents for their unconditional support. Their encouragement helps me stay optimistic in the face of uncertainties in life.

Table of contents

List of Commonly Used Acronyms

CCE	Coarse correlated equilibrium.
CE	Correlated equilibrium.
FTRL	Follow-the-regularizer-leader.
MAML	Model-agnostic meta-learning.
MARL	Multi-agent reinforcement learning.
MDP	Markov decision process.
MPG	Markov potential game.
NE	Nash equilibrium.
NFG	Normal-form game.
OFTRL	Optimistic follow-the-regularizer-leader.
OMD	Online mirror descent.
RL	Reinforcement learning.

Chapter 1

Introduction

Reinforcement learning (RL) has achieved tremendous successes in recent years and has led to major breakthroughs in artificial intelligence [1]–[4]. In RL, a learning agent tries to learn an optimal decision-making policy by sequentially interacting with an unknown environment and maximizing its cumulative rewards along the way [5]. Due to its natural and universal formulation, RL draws great interests from many disciplines where optimal decision-making is concerned, including control theory, management science, operations research, and multi-agent systems.

One such discipline that is particularly relevant to this dissertation is game theory [6]. Many real-world sequential decision-making problems involve the strategic interactions of more than one agent in a shared environment. Game-theoretical thinking naturally arises in resolving such complex systems with multiple self-interested agents. These multi-agent decision-making problems are usually modeled under the mathematical framework of stochastic games [7] (also known as Markov games), and oftentimes addressed with multi-agent reinforcement learning (MARL) [8]. Well-known application scenarios of MARL include playing the game of Go [2], Poker [3], real-time strategy games [9], autonomous driving [10], and robotics [11].

Despite the encouraging empirical successes, rigorous theoretical understandings of MARL still leave a lot to be desired, and MARL algorithms with provable convergence and sample complexity guarantees are relatively lacking. In practice, training MARL algorithms with deep neural-networks as function approximators is known to be notoriously hard. Deep MARL agents often exhibit oscillating behaviors during training due to the strong coupling of the agents' running policies. When the number of agents is large, many MARL methods may also require a significant number of training samples to thoroughly explore the state-action space. This is due to the well-known *curse of multiagents* [12]: The joint action space in a MARL problem in general amounts to the Cartesian product of all the agents' individual action spaces, which scales exponentially in the number of agents. These undesirable aspects of empirical MARL solutions pressingly call for the development of sample-efficient MARL algorithms with provable convergence guarantees. In the following, we identify some key limitations in existing theoretical results of MARL research and seek to make improvements along these directions.

First, prior theoretical efforts in MARL have been primarily focused on simplified game settings with special reward structures, such as fully competitive or cooperative settings. One prevalent setting is MARL in two-player zero-sum Markov games [13], [14], where the two agents have exactly opposite objectives. This is mainly due to the fundamental computational difficulty in more general scenarios, since calculating a Nash equilibrium (NE) in a generic general-sum game is known to be PPAD-complete [15]. Consequently,

many prior works often fail to justify the empirical successes of MARL to application scenarios beyond these simplified settings. A broad spectrum of games in the generic form (such as general-sum Markov games) are still left largely open.

Second, a major challenge faced by MARL is non-stationarity of the environment, yet a rigorous treatment of RL in non-stationary environments is relatively lacking in earlier works. Specifically, in MARL, the state transitions and rewards depend on the collective actions of all the agents. As a result, the environment often looks non-stationary from each agent's own perspective when the agents learn and update their local policies simultaneously, because the environment may be altered by the unobserved behavior of the other agents. Conventional RL results no longer apply to such a non-stationary environment, as these results are mostly established under the assumption of a stationary Markov decision process (MDP) where the state transition and reward functions are fixed. To address the challenge of non-stationarity, one needs to specifically design and analyze RL algorithms suitable for non-stationary environments, and to establish their connections to the non-stationarity faced by MARL agents.

Third, prior research in MARL focuses on solving an individual task in isolation but often neglects the potential connections between multiple related tasks. In many practical scenarios of MARL, the environment is dynamically evolving, and hence a MARL algorithm needs to not only solve a single task alone but instead to collectively resolve a set of related tasks. By exploiting the knowledge obtained from other tasks, a sample-efficient MARL algorithm should ideally be able to solve an unseen task using much fewer training samples than learning from scratch, especially when the tasks share some inherent similarities. Such a practical consideration poses the important question of designing a MARL method that can exploit the connections across multiple related tasks and use its prior knowledge to expedite the learning process on a new task.

Our goal in this dissertation is to develop theoretically well-founded RL algorithms that can address the aforementioned limitations in existing MARL research. Our results in this regard are summarized in the following subsections.

1.1 MARL in General-Sum Markov Games

To address the first limitation above, we propose multiple sample-efficient MARL algorithms for general-sum Markov games with no specialized reward structure assumptions [16], [17]. Given the fundamental difficulty of calculating a Nash equilibrium (NE), we aim at two weaker solution concepts, namely coarse correlated equilibrium (CCE) and correlated equilibrium (CE). Both CCE and CE are standard game-theoretical notions that generalize NE by allowing possible correlations among the agents' strategies.

To avoid the exponential sample complexity implied by the curse of multiagents, our algorithms are specifically designed to be *decentralized*. In our algorithms, each agent makes decisions based on only its local information. Neither communication nor centralized coordination is required during learning. In fact, each agent can be completely oblivious to the presence of others. This way, each agent optimizes its policy in its own action space instead of the Cartesian product action space of all agents. Hence, our algorithms can readily scale up to a large number of agents, without suffering from the exponential dependence on the number of agents.

Though seemingly restrictive, we show that decentralized learning dynamics suffice to efficiently find (coarse) correlated equilibria in general-sum Markov games. Specifically, we propose multiple V-learning-based algorithms, where each agent independently runs optimistic V-learning (a variant of Q-learning) to efficiently

explore the unknown environment, while using a no-regret learning subroutine for policy updates. In episodic general-sum Markov games, we show that our algorithms can learn an ε-approximate CCE in $\widetilde{O}(H^5SA_{\max}/\varepsilon^2)$ episodes, and an ε-approximate CE in $\widetilde{O}(H^5SA_{\max}^2/\varepsilon^2)$ episodes, where S is the number of states, $A_{\max}$ is the size of the largest individual action space, and H is the length of an episode. Our results appear to be the first sample complexity guarantees for decentralized MARL in generic general-sum Markov games. In addition, we extend our results to the full-information feedback setting where each agent can observe the complete reward vector. By exploiting the "self-play" structure, we develop new algorithms that converge to CCE or CE in general-sum Markov games at a fast rate of $\widetilde{O}(T^{-1})$ within T iterations of policy updates when the same algorithms are run by all the agents. Numerical simulations are also provided to corroborate our theoretical findings.

1.2 Non-Stationary RL and Cooperative Markov Games

In view of the second limitation identified above, we present an RL algorithm for *non-stationary* MDPs, and demonstrate its connection to an important subclass of MARL problems named cooperative Markov games. In non-stationary MDPs, both the reward functions and the state transition distributions are allowed to vary over time, either gradually or abruptly, as long as their cumulative variation magnitude does not exceed certain budgets. In addition to its close connection to MARL, non-stationary RL is an interesting topic on its own, as it can also capture time-varying environments in a wide range of intriguing sequential decision-making problems such as online advertisement auctions [18], [19], dynamic pricing [20], and inventory control [21], [22].

We propose an RL algorithm named Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB) [23] that is particularly tailored to non-stationary environments. Our algorithm adopts a simple but effective restarting strategy that resets the memory of the agent according to a predefined schedule. The restarting strategy ensures that our algorithm only refers to the most up-to-date experience in the time-varying environment for decision-making. Compared to conventional RL algorithms in stationary MDPs, RestartQ-UCB also utilizes an extra optimism term (in addition to the standard Hoeffding/Freedman-based bonus) to encourage additional exploration in the non-stationary environment.

Our analysis shows that RestartQ-UCB outperforms existing non-stationary RL solutions in terms of dynamic regret, a notion commonly utilized to measure the performance of online learning algorithms in non-stationary environments. In particular, RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret bound of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where S and A are the numbers of states and actions, respectively, $\Delta > 0$ is the total variation magnitude, H is the number of time steps per episode, and T is the total number of time steps. We further show that our algorithm is nearly optimal by establishing an information-theoretical lower bound of $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$, which is the first impossibility result that characterizes the fundamental limits in non-stationary RL. We also illustrate how our non-stationary RL algorithm is connected to the non-stationarity issue inherent in MARL. Specifically, we show that RestartQ-UCB can be readily applied to learning the team-optimal policies in cooperative smooth games against a slowly-changing opponent.

1.3 Meta-Learning in Markov Games

In response to the third aforementioned issue, we propose to use meta-learning to learn multiple related Markov games collectively. Meta-learning [24]–[27] studies the use of data samples from existing tasks to learn

useful representations that enable quick adaptation to new tasks. We focus on the classic model-agnostic meta-learning (MAML) [28] type of algorithms that aim to learn a good initialization such that running a few steps of gradient descent from this initialization quickly leads to a desirable policy on any new task. To study the convergence of MAML, an important prerequisite is to understand how the convergence of MARL algorithms depends on the quality of policy initialization, but such a result is missing in the literature.

We make an initial attempt toward characterizing some of the central theoretical properties of meta-learning in a wide range of fundamental MARL settings, and, along the way, we develop multiple MARL algorithms with initialization-dependent convergence guarantees [29]. First, for learning Nash equilibria (NE) in two-player zero-sum Markov games, we first propose an optimistic online mirror descent algorithm with a refined convergence analysis that explicitly characterizes the dependence on policy initialization. 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. Second, we consider learning NE in Markov potential games (MPGs), an important subclass of MARL tasks where the agents are largely cooperative with objectives aligned by a global potential function. For MPGs, we show that a simple refinement of an existing policy gradient ascent algorithm suffices to provide initialization-dependent guarantees. We establish sharper convergence rates of meta-learning when the potential functions of the MPGs have small deviations. In addition, with a properly chosen policy update rule, we prove non-asymptotic convergence of the exact MAML algorithm in MPGs, despite the convoluted learning dynamics of multiple loosely-coupled agents. Third, for learning coarse correlated equilibria in general-sum Markov games, we analogously design an initialization-dependent MARL algorithm, and then establish the sharper convergence rate of meta-learning under natural similarity metrics. Finally, we provide numerical results to illustrate the expedited convergence and scalability of our algorithms. Our work appears to be the first to investigate the theoretical properties of meta-learning in MARL and provide reliable justifications of its benefits.

1.4 Outline

The rest of this dissertation is organized as follows. In Chapter 2, we present decentralized MARL algorithms for learning (coarse) correlated equilibria in general-sum Markov games and analyze their sample complexities or convergence rates. In Chapter 3, we study RL in non-stationary environments and illustrate its connection to cooperative Markov games. In Chapter 4, we present our meta-learning method that achieves faster convergence by learning multiple Markov games collectively. Finally, in Chapter 5, we conclude this dissertation and suggest future research directions.

Chapter 2

Learning (Coarse) Correlated Equilibria in General-Sum Markov Games

Multi-agent reinforcement learning (MARL) algorithms often suffer from an exponential sample complexity dependence on the number of agents, a phenomenon known as the curse of multiagents. In this chapter, we address this challenge by investigating decentralized sample-efficient MARL algorithms that efficiently learn equilibria in general-sum Markov games. Given the fundamental difficulty of calculating a Nash equilibrium (NE), we aim at learning a coarse correlated equilibrium (CCE) or correlated equilibrium (CE), two solution concepts that generalize NE by allowing possible correlations among the agents' strategies.

We first propose the V-learning OMD algorithm, where each agent independently runs optimistic V-learning (a variant of Q-learning) to efficiently explore the unknown environment, while using a stabilized online mirror descent (OMD) subroutine for policy updates. In episodic general-sum Markov games, we show that the agents can find an ε-approximate CCE in at most $\widetilde{O}(H^6SA_{\max}/\varepsilon^2)$ episodes, where S is the number of states, $A_{\max}$ is the size of the largest individual action space, and H is the length of an episode. This appears to be the first sample complexity result for decentralized MARL in generic general-sum Markov games. Our results rely on a novel investigation of an anytime high-probability regret bound for OMD with a dynamic learning rate and weighted regret, which would be of independent interest.

One key feature of the V-learning OMD algorithm is that it is *decentralized*, where each agent can make decisions based on only its local information. Neither communication nor centralized coordination is required during learning. In fact, each agent can be completely oblivious to the presence of others. This way, this algorithm can readily scale up to a large number of agents, without suffering from the exponential dependence on the number of agents.

We further generalize and improve V-learning OMD in multiple different aspects. First, we propose *stage-based* V-learning algorithms that significantly simplify the algorithmic design and analysis of V-learning OMD, and circumvent a rather complicated no-*weighted*-regret bandit subroutine. We also show that stage-based V-learning improves the sample complexity of V-learning OMD for learning (coarse) correlated equilibria in general-sum Markov games. In particular, stage-based V-learning can learn an ε-approximate CCE in $\widetilde{O}(H^5SA_{\max}/\varepsilon^2)$ episodes, and an ε-approximate CE in $\widetilde{O}(H^5SA_{\max}^2/\varepsilon^2)$ episodes. Second, we extend the

V-learning framework to the full-information feedback setting where each agent can observe the expected rewards it would have received had it played any candidate action. We develop no-regret learning algorithms with accompanying value update procedures and establish their fast $\widetilde{O}(T^{-1})$ convergence to CCE or CE in full-information general-sum Markov games when the same algorithms are adopted by all the players. Numerical simulations are provided to corroborate these theoretical findings.

2.1 Introduction

Reinforcement learning (RL) has recently shown the capability to solve many challenging sequential decision-making problems, ranging from the game of Go [2], Poker [3], and real-time strategy games [9], to autonomous driving [10], and robotics [11]. Many of the RL applications involve the interaction of multiple agents, which are modeled systematically within the framework of multi-agent reinforcement learning (MARL). These success stories have inspired a remarkable line of studies on the theoretical aspects of MARL.

Most of the theoretical efforts in MARL, however, have been devoted to Markov games with special reward structures, such as fully competitive or cooperative games. One prevalent setting is MARL in two-player zero-sum Markov games [13], [14], where the two agents have exactly opposite objectives. Such prevalence is mostly due to the fundamental computational difficulty in more general scenarios: Finding a Nash equilibrium (NE) is known to be PPAD-complete both for two-player general-sum games [15] and zero-sum games with more than two players [30]. Given the daunting impossibility results, convergence to NE in generic games with no special structure seems hopeless in general. As a result, many important problems in the multi-player general-sum settings, which can model broader and more practical interactive behaviors of decision makers, have been left relatively open.

In this chapter, we make an initial attempt toward understanding some of the theoretical aspects of MARL in decentralized general-sum Markov games. Given the inherent challenges for computing Nash equilibria, we need to target a slightly weaker solution concept than NE. One reasonable alternative is to find a coarse correlated equilibrium (CCE) [31], [32] of the game. Unlike NE, CCE can always be found in polynomial time for general-sum games [33], and due to its tractability, calculating CCE has also been commonly used as an important subroutine toward finding Nash equilibria in two-player zero-sum Markov games [14], [34].

Our interest in CCE is mostly motivated by the following folklore result for learning in normal-form games: When the agents independently run no-regret learning algorithms in general-sum normal-form games, their empirical frequency of plays converges to the set of CCE of the game [35], [36]. In no-regret learning, each agent independently adapts its policy to minimize the cumulative regret based on only its local information, irrespective of the actions or rewards of the other agents. Well-known examples of no-regret learning algorithms include multiplicative weights update (MWU) [37] and online gradient descent [38]. Such a folk result hence suggests that CCE is a natural outcome of the simple and *uncoupled* learning dynamics of the agents. A natural question to ask is whether a similar result also holds for Markov games. Specifically, in this chapter, we ask the following questions: Can we find CCE in general-sum Markov games using decentralized/uncoupled learning dynamics? If so, can we achieve such a result efficiently, by showing an explicit sample complexity upper bound?

Before answering these questions, we would like to remark that MARL in general-sum games can be highly challenging due to the well-known *curse of multiagents* [12]: The joint action space in a MARL problem is equal to the Cartesian product of the individual action spaces of all agents, which scales exponentially in the number of agents. Typical algorithms that easily fail at this challenge are those using centralized/joint

learning [39], [40]. Specifically, centralized learning assumes the existence of a single coordinator who can access the local information of all the agents, and learns policies jointly for all of them. This centralized training (though possibly decentralized execution) approach has become a common practice in empirical MARL [41]–[46]. Centralized learning essentially reduces the multi-agent problem to a single-agent one, but unfortunately suffers from the exponential dependence as it usually needs to exhaustively search the joint action space. Such a computation bottleneck can be partially resolved by allowing communications among the agents and hence distributing the workload to each of them [47]–[49]. However, communication-based methods instead suffer from the additional communication overheads, which can be unrealistic in some real-world scenarios where communication may be expensive and/or unreliable, such as in unmanned aerial vehicle (UAV) field coverage [50].

Given the aforementioned limitations, in this thesis, we are interested in a more practical setting: *decentralized* learning[1]. We focus on solutions where each agent can make decisions based on only its local information (e.g., local actions and rewards), and need not communicate with its opponents or be coordinated by any central controller during learning. In fact, in our algorithms, the agents can be completely oblivious to the presence of other agents. Under such weak assumptions, decentralized algorithms are suitable for many practical MARL scenarios [59], and do not suffer from the exponential sample & computation complexity. Such algorithms are naturally model-free, as they do not maintain explicit estimates of the transition functions. Compared with model-based algorithms, model-free ones typically enjoy higher time- and space-efficiency, and are more compatible with the modern deep RL architectures [60], [61].

In decentralized learning, since both the reward and the transition are affected by the other agents, the environment becomes *non-stationary* from each agent's own perspective, especially when the agents learn and update their policies simultaneously. Hence, an agent needs to efficiently explore the unknown environment while bearing in mind that the information it gathered a while ago might no longer be accurate. This makes many successful single-agent RL solutions, which assume that the agent is learning in a stationary Markovian environment, inapplicable. Furthermore, compared with RL in two-player zero-sum games, an additional challenge in general-sum games is *equilibrium selection*. In zero-sum games, all NE have the same value [62], and there is no ambiguity in defining the sub-optimality of a policy. However, in general-sum games, multiple equilibria can have different values. We hence need to first identify which equilibrium to compare with when we are trying to measure the performance of a policy.

Contributions. Despite the challenges identified above, we answer both of the aforementioned questions affirmatively, by presenting an algorithm in which the agents can find a CCE in general-sum Markov games efficiently through decentralized learning. In the first part of this chapter (Sections 2.4-2.6), we study provably efficient exploration in decentralized general-sum Markov games. We propose an algorithm named Optimistic V-learning with Stabilized Online Mirror Descent (V-learning OMD), where V-learning [34] is a simple variant of Q-learning. In V-learning OMD, each agent independently runs an optimistic V-learning algorithm to explore the unknown environment, while using an online mirror descent procedure for policy updates. Following the learning process, the CCE can be extracted by simply letting the agents randomly repeat their previous strategies using a common random seed. We also show that if all agents in the game run the V-learning OMD algorithm, they can find an ε-approximate coarse correlated equilibrium in at most $\widetilde{O}(SA_{\max}H^6/\varepsilon^2)$ episodes, where S is the number of states, $A_{\max}$ is the size of the largest action space among the agents, and H is the length of an episode. Our result complements its counterpart in normal-form games

[1]This setting has been studied under various names in the literature, including individual learning [51], decentralized learning [52], agnostic learning [53], [54], and independent learning [40], [55]. It also belongs to a more general category of teams/games with decentralized information structure [56]–[58].

that uncoupled no-regret learning dynamics lead to CCE. We further show that our sample complexity is nearly-optimal in that it matches all the parameter dependences in the information-theoretical lower bound, except the horizon length H. As an important building block of our analysis, we conduct a novel investigation of a high-probability regret bound for OMD with a dynamic learning rate and weighted regret, which might be of independent interest. We emphasize that due to the decentralization property, our algorithm readily generalizes to a large number of agents without suffering from the exponential dependence on the number of agents. Our work appears to be the first to provide non-asymptotic guarantees for MARL in generic general-sum Markov games with efficient exploration, with an additional appealing feature of being decentralized.

Despite being the first decentralized algorithm for learning CCE in general-sum Markov games, an undesirable aspect of the V-learning OMD algorithm is the need of the complicated no-*weighted*-regret bandit analysis. This turns out to be a routine yet painful procedure that many existing V-learning-based methods [12], [16], [63] need to go through. In the second part of this chapter (Section 2.7), we provide a solution to this problem and improve the V-learning OMD algorithm in multiple different aspects. Specifically, we propose two variants of V-learning OMD that use a *stage-based* V-learning method. We show that stage-based V-learning helps significantly simplify the algorithmic design and analysis of V-learning OMD, and circumvent the rather complicated no-weighted-regret bandit subroutine. We also demonstrate that stage-based V-learning can be combined with any off-the-shelf no(-average)-regret learning algorithm to improve the sample complexity of V-learning OMD. In particular, stage-based V-learning can learn an ε-approximate CCE in $\widetilde{O}(H^5SA_{\max}/\varepsilon^2)$ episodes, and an ε-approximate CE in $\widetilde{O}(H^5SA_{\max}^2/\varepsilon^2)$ episodes.

The $\widetilde{O}(1/\varepsilon^2)$ sample complexities of our V-learning-based algorithms rely on establishing an $O(\sqrt{T})$ regret bound for an adversarial bandit procedure. Such an $O(\sqrt{T})$ regret is unimprovable against an adversarial environment, but it need not be the case for learning equilibria in games because each player in a game is interacting with other learning players who may not necessarily act adversarially. In the third part of this chapter (Section 2.8), we exploit this structure and seek to establish faster convergence to CCE/CE in full-information general-sum Markov games [64]. For CE, we consider the optimistic follow-the-regularizer-leader (OFTRL) algorithm with a log-barrier regularizer and integrate it with the celebrated external-to-swap-regret reduction [65] and smooth value updates. For CCE, we consider OFTRL with negative entropy regularization and combine it with a stage-based value update scheme. We show that our algorithms converge to CCE or CE in full-information general-sum Markov games at a fast convergence rate of $\widetilde{O}(T^{-1})$, matching the best-known results in normal-form games.

Outline. The rest of the chapter is organized as follows: We start with a literature review in Section 2.2. In Section 2.3, we introduce the mathematical model of our problem and necessary preliminaries. In Section 2.4, we present our V-learning OMD algorithm for learning coarse correlated equilibria in general-sum Markov games. A sample complexity analysis of V-learning OMD is given in Section 2.5. In Section 2.6, we analyze a specific adversarial multi-armed bandit problem, which plays a central role in our analysis of the V-learning OMD algorithm. In Section 2.7, we improve the V-learning OMD algorithm by using a stage-based V-learning method, and analyze its sample complexities for learning CCE and CE. In Section 2.8, we extend our results to the full-information setting and establish the fast $\widetilde{O}(T^{-1})$ convergence rates to CCE/CE. For clarity of presentations, most proofs are deferred to Sections 2.9-2.14. Finally, we conclude this chapter in Section 2.15.

2.2 Related Work

A common mathematical framework of multi-agent RL is stochastic games [7], which are also referred to as Markov games. Given the PPAD completeness of finding a Nash equilibrium in generic games [15], [30], convergence to NE has mostly been studied in games with special structures, such as two-player zero-sum games or cooperative games. Early attempts to learn Nash equilibria in Markov games include [8], [66]–[68], but they either assume the transition kernel and rewards are known, or only yield asymptotic guarantees. In particular, [8] has proposed a Q-learning based algorithm named minimax-Q, whose asymptotic convergence guarantee has later been established in [69].

More recently, various sample efficient methods have been proposed [13], [14], [34], [70]–[73], mostly for learning in two-player zero-sum Markov games. Most notably, several works have investigated two-player zero-sum games in a *decentralized* environment similar to ours: [55] has shown non-asymptotic convergence guarantees for independent policy gradient methods when the learning rates of the two agents follow a two-timescale rule. [53] has studied online learning when the actions of the opponents are not observable, and have achieved the first sub-linear regret $\widetilde{O}(K^{\frac{3}{4}})$ in the decentralized setting for K episodes. More recently, [54] has proposed an Optimistic Gradient Descent Ascent algorithm with a slowly-learning critic, and have shown a strong finite-time last-iterate convergence result in the decentralized/agnostic environment. Overall, these works have mainly focused on two-player zero-sum games. These results do not carry over in any way to general-sum games or MPGs that we consider in this thesis.

MARL has also been studied in teams or cooperative games. Without enforcing a decentralized environment, [39] has proposed to coordinate the agents by letting them take actions in a lexicographic order. In a similar setting, [74] has studied optimal adaptive learning that converges to the optimal NE in Markov teams. [75] has presented an independent learning algorithm that achieves a Pareto optimal NE in common interest games with limited communication. These methods critically relied on communications among the agents (beforehand) or observing the teammates' actions. In contrast, the distributed Q-learning algorithm in [76] is decentralized and coordination-free, which, however, only works for deterministic tasks, and has no non-asymptotic guarantees. More recently, [52] has shown that decentralized Q-learning can converge to NE in weakly acyclic games, which cover Markov teams and potential games as important special cases. Later, [77] has further improved [52] and achieved convergence to the team-optimal equilibrium.

A few works have considered games beyond the zero-sum or cooperative settings: [67], [66], and [78] have established convergence guarantees under the assumptions that either a saddle point equilibrium or a coordination equilibrium exists. [79] has bypassed the computation of NE in general-sum games by targeting correlated equilibria instead, but no theoretical convergence result has been given. Other approaches for finding NE in general-sum games include minimizing the Bellman-like residuals learned from offline/batch data [80], or using a two-timescale algorithm to learn the policy of each player from an optimization perspective [81]. Nevertheless, none of these works has considered sample-efficient exploration in a decentralized environment, a more challenging objective that we pursue in this thesis. More recently, [82] has studied the non-asymptotic properties of learning CCE in general-sum Markov games, but their sample complexity bound scales exponentially in the number of agents as a consequence of using a centralized learning approach.

In general-sum normal-form games, a folklore result is that when the agents independently run no-regret learning algorithms, their empirical frequency of plays converges to the set of coarse correlated equilibria (CCE) of the game [35]. However, a CCE may suggest that the agents play obviously non-rational strategies. For example, [83] has constructed an example where a CCE assigns positive probabilities only to strictly

dominated strategies. On the other hand, given the PPAD completeness of finding a Nash equilibrium, convergence to NE seems hopeless in general. An impossibility result [84] has shown that uncoupled no-regret learning does not converge to Nash equilibrium in general, due to the informational constraint that the adjustment in an agent's strategy does not depend on the reward functions of the others. Hence, convergence to Nash equilibria is guaranteed mostly in games with special reward structures, such as two-player zero-sum games [85] and potential games [86], [87].

For learning in general-sum Markov games, [88] has shown a sample complexity lower bound for NE that is exponential in the number of agents. Recently, [72] has presented a line of results on learning NE, CE, or CCE, but their algorithm is model-based, and suffers from such exponential dependence. Since the publication of the first part of this chapter [16], a few closely related works [12], [63] have also used V-learning based methods for learning CCE and/or CE, and avoid the exponential dependence. Our methods in the second part of this chapter significantly simplify the algorithmic design and analysis in these related works, by introducing a stage-based V-learning update rule that circumvents their rather complicated no-weighted-regret bandit subroutine.

Another line of research has considered RL in Markov potential games (MPGs) [89]–[91]. [52] has shown that decentralized Q-learning style algorithms can converge to NE in weakly acyclic games, which cover MPGs as an important special case. Their decentralized setting is similar to ours in that each agent is completely oblivious to the presence of the others. Later, such a method has been improved in [77] to achieve team-optimality. However, both of them require a coordinated exploration phase, and only yield asymptotic guarantees. Decentralized learning has also been studied in single-stage weakly acyclic games [92] or potential games [87], [93]. [94] has shown that independent Natural Policy Gradient also converges to NE in MPGs, though only asymptotic convergence has been established. Finally, MPGs have also been studied in [63], but their model-based method is not decentralized, and requires the agents to take turns to learn the policies.

Efficient exploration has also been widely studied in the literature of single-agent RL, see, e.g., [60], [95]–[97]. For the tabular episodic setting, various methods [61], [97], [98] have achieved the sample complexity of $\widetilde{O}(H^3SA/\varepsilon^2)$, which matches the information-theoretical lower bound. When reduced to the bandit case, decentralized MARL is also related to the cooperative multi-armed bandit (MAB) problem [99], [100], originated from the literature of cognitive radio networks. The difference is that, in cooperative MAB, each agent is essentially interacting with an individual copy of the bandit, with an extra caution of action collisions; in the MARL formulation, the reward function is defined on the Cartesian product of the action spaces, which allows the agents to be coupled in more general forms. [101] has studied cooperative multi-player multi-armed bandits with information asymmetry. Nevertheless, [101] requires stronger conditions than our decentralized setting as their algorithm relies on playing a predetermined sequence of actions.

2.3 Preliminaries

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

$A_{\max} = \max_{i \in \mathcal{N}} A_i$.

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

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

$$V_{h,i}^{\pi}(s) := \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \boldsymbol{a}_{h'}) \mid s_h = s\right], \tag{2.1}$$

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

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

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

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

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

For non-Markov correlated policies, we can still define their value functions at step $h = 1$ in a sense similar to (2.1). A best response $\pi_i^\star$ with respect to the non-Markov policies π_{-i} is a policy (independent of the randomness of π_{-i}) that maximizes agent i's value at step 1, i.e., $V_{1,i}^{\pi_i^\star, \pi_{-i}}(s_1) = \sup_{\pi_i} V_{1,i}^{\pi_i, \pi_{-i}}(s_1)$. The best response to the non-Markov policies of the opponents is not necessarily Markov.

(Coarse) correlated equilibrium. Given the PPAD-hardness of calculating Nash equilibria in general-sum games [30], we introduce two relaxed solution concepts, namely coarse correlated equilibrium (CCE) and correlated equilibrium (CE). A CCE states that no agent has the incentive to deviate from a correlated policy π by playing a different independent policy.

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

CCE relaxes NE by allowing possible correlations in the policies. For illustrative purposes, let us compare the definitions of NE and CCE in a simple normal-form game named Hawk-Dove (with no state transitions). There are two players in this game. The row player has the action space $\mathcal{A} = \{a_1, a_2\}$, and the column player's action space is $\mathcal{B} = \{b_1, b_2\}$. The reward matrix of the Hawk-Dove game is described in Table 2.1. There are three Nash equilibria in this game: (a_1, b_2) and (a_2, b_1) are two pure strategy NE, and $((0.5, 0.5), (0.5, 0.5))$ is a NE in mixed strategies. Table 2.2 gives a CCE distribution of the Hawk-Dove game, which assigns equal probabilities to three action pairs: (a_1, b_1), (a_1, b_2), and (a_2, b_1). We can see that NE defines for each player an *independent* probability distribution over a player's own action space; in contrast, a CCE is a probability distribution over the joint action space of the players. In this sense, CCE generalizes NE by allowing possible correlations among the strategies of the agents. In our proposed algorithm, such correlation is implicitly achieved by letting the players use a common random seed.

Table 2.1: The Hawk-Dove game.

	b_1	b_2
a_1	4,4	1,5
a_2	5,1	0,0

Table 2.2: A CCE in the Hawk-Dove game.

	b_1	b_2
a_1	1/3	1/3
a_2	1/3	0

Before introducing the definition of CE, we need to first specify the concept of a strategy modification.

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

Given a strategy modification ψ_i, for any policy π, step h and state s, if π selects the joint action $\boldsymbol{a}_h = (a_{h,1}, \ldots, a_{h,N})$, then the modified policy $\psi_i \diamond \pi$ will select $(a_{h,1}, \ldots, a_{h,i-1}, \psi_{h,i}^s(a_{h,i}), a_{h,i+1}, \ldots, a_{h,N})$. Let Ψ_i denote the set of all possible strategy modifications for agent i. A CE is a distribution where no agent has the incentive to deviate from a correlated policy π by using any strategy modification. It is known that {NE}⊂{CE}⊂{CCE} in general-sum games [102].

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

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

To better illustrate the difference between CCE and CE, it is helpful to consider the equivalent forms of their definitions in normal-form games. Specifically, in an N-player normal-form game, let $\mathcal{A}_i$ denote the action space for agent i and let $\mathcal{A} = \times_{i=1}^N \mathcal{A}_i$. Let $u_i : \mathcal{A} \to \mathbb{R}$ denote the utility function for agent i. The equivalent forms of Definitions 2 and 4 in normal-form games are as follows, respectively:

Definition 5. *(Coarse correlated equilibrium, normal-form game). A probability distribution $\sigma \in \Delta(\mathcal{A})$ is an*

ε-approximate coarse correlated equilibrium for a normal-form game if

$$\mathbb{E}_{\boldsymbol{a}\sim\sigma}\left[u_i(a_i^\star, a_{-i})\right] - \mathbb{E}_{\boldsymbol{a}\sim\sigma}\left[u_i(\boldsymbol{a})\right] \le \varepsilon, \forall a_i^\star \in \mathcal{A}_i, i \in \mathcal{N}.$$

Definition 6. *(Correlated equilibrium, normal-form game). A probability distribution $\sigma' \in \Delta(\mathcal{A})$ is an ε-approximate correlated equilibrium for a normal-form game if*

$$\mathbb{E}_{\boldsymbol{a}\sim\sigma'}\left[u_i(a_i^\star, a_{-i}) \mid a_i\right] - \mathbb{E}_{\boldsymbol{a}\sim\sigma'}\left[u_i(\boldsymbol{a}) \mid a_i\right] \le \varepsilon, \forall a_i^\star \in \mathcal{A}_i, i \in \mathcal{N}.$$

Intuitively, in normal-form games, a CE is a probability distribution $\sigma' \in \Delta(\mathcal{A})$ such that after a joint action $\boldsymbol{a} = (a_i, a_{-i})$ is drawn from σ', playing a_i is a best strategy for agent i conditioned on seeing a_i, given that all the other players will play according to a_{-i}. A CCE $\sigma \in \Delta(\mathcal{A})$ is different in the sense that playing the recommended action a_i when $\boldsymbol{a}$ is drawn from σ is player i's best strategy *in expectation*, before agent i sees a_i. CCE is suitable for the scenarios when each agent is committed to following the recommended action up front and is not able to deviate from the recommended action after seeing it.

For notational convenience, for the first part of this chapter (Sections 2.4-2.6), we illustrate our V-learning OMD algorithm and its results for the special case of two-player general-sum games, i.e., $N = 2$. It is straightforward to extend such results to the general N-player games as we defined above. With two players, we use $\mathcal{A}$ and $\mathcal{B}$ to denote the action spaces of players 1 and 2, respectively. Let $S = |\mathcal{S}|$, $A = |\mathcal{A}|$ and $B = |\mathcal{B}|$. We also rewrite the correlated policies (π_1, π_2) as (μ, ν). In the second part of this chapter (Section 2.7), we will present the results in the generic N-player general-sum games.

2.4 V-Learning OMD

In this section, we introduce our algorithm Optimistic V-learning with Stabilized Online Mirror Descent (V-learning OMD) for learning coarse correlated equilibria in general-sum Markov games.

V-learning OMD naturally integrates the idea of optimistic V-learning in single-agent RL [60] with Online Mirror Descent (OMD) [38], [103] in online convex optimization. First, our algorithm uses optimistic V-learning to efficiently explore the unknown environment, as in single-agent RL. Second, each agent selects its actions following a no-regret OMD algorithm in order to achieve a CCE. The intuition of using no-regret learning here is to defend against the unobserved behavior of the opponents, by presuming that the opponents' behavior will impair the reward sequence arbitrarily. Seemingly conservative, we will show that this suffices to find the CCE. The use of no-regret learning is also reminiscent of the well-known result in normal-form games that if all agents run a no-regret learning algorithm, the empirical frequency of their actions converge to a CCE [36]. These components also make our algorithm decentralized, which can be implemented individually using only the local rewards received and the local actions executed, without any communication among the agents.

The algorithm run by agent 1 (with action space $\mathcal{A}$) is presented in Algorithm 1. The algorithm for agent 2 (or other agents in the setting with more than two agents) is symmetric, by simply replacing the action space $\mathcal{A}$ with the agent's own action space. We thus omit the index of an agent in the notations for clarity. We use $\theta_h(a \mid s_h)$ to denote the probability of taking action a at state s_h and step h, where $\theta_h(\cdot \mid s_h) \in \Delta(\mathcal{A})$. At each step h of an episode, the agent first takes an action a_h according to a policy $\theta_h(\cdot \mid s_h)$ for the current state s_h, and observes the reward r_h and the next state s_{h+1}. It also counts the number of times $t := N_h(s_h)$ that state s_h has been visited, and constructs a bonus term $\beta_t = c\sqrt{\frac{H^4 A\iota}{t}}$ (c is some absolute constant and

Algorithm 1: Optimistic V-learning with Stabilized Online Mirror Descent (V-learning OMD)

1 **Define:** $F(\theta) = \sum_{a=1}^{A}(\theta(a)\log(\theta(a)) - \theta(a))$ for $\theta \in \mathbb{R}_+^A$, $D_F(u,v) = F(u) - F(v) - \langle u - v, \nabla F(v)\rangle$ for $u, v \in \mathbb{R}_+^A$.

2 **Initialize:** $\overline{V}_h(s) = V_h(s) \leftarrow H - h + 1, N_h(s) \leftarrow 0, \theta_h(a \mid s) \leftarrow 1/A$, $\forall h \in [H+1], s \in \mathcal{S}, a \in \mathcal{A}$.

3 **for** *episode* $k \leftarrow 1$ *to* K **do**

4 Receive s_1;

5 **for** *step* $h \leftarrow 1$ *to* H **do**

6 Take action $a_h \sim \theta_h(\cdot \mid s_h)$;

7 Observe reward r_h and next state s_{h+1};

8 $N_h(s_h) \leftarrow N_h(s_h) + 1, t \leftarrow N_h(s_h)$;

9 $\alpha_t \leftarrow \frac{H+1}{H+t}, \beta_t \leftarrow c\sqrt{\frac{H^4 A\iota}{t}}, \gamma_t \leftarrow \sqrt{\frac{\log A}{At}}, \eta_t \leftarrow \sqrt{\frac{\log A}{At}}$;

10 $V_h(s_h) \leftarrow (1-\alpha_t)V_h(s_h) + \alpha_t\left(r_h + \overline{V}_{h+1}(s_{h+1}) + \beta_t\right)$;

11 $\overline{V}_h(s_h) \leftarrow \min\{V_h(s_h), H - h + 1\}$;

12 **for** *action* $a \in \mathcal{A}$ **do**

13 $\hat{l}_h(s_h, a) \leftarrow (H - r_h - \overline{V}_{h+1}(s_{h+1}))\mathbb{1}\{a_h = a\} / (\theta_h(a \mid s_h) + \gamma_t)$;

14 $\theta' \leftarrow \arg\min_{\theta \in \Delta(\mathcal{A})}\left\{\eta_t\left\langle\theta, \hat{l}_h(s_h, \cdot)\right\rangle + D_F(\theta, \theta_h(\cdot \mid s_h))\right\}$;

15 $\theta_h(\cdot \mid s_h) \leftarrow \lambda_t\theta' + (1-\lambda_t)\mathbf{1}/A$, where $\lambda_t = \frac{\eta_{t+1}\alpha_t(1-\alpha_{t+1})}{\eta_t\alpha_{t+1}}$;

ι is a log factor to be defined later) that is used to upper bound the state value function. The agent then updates the optimistic state value functions by:

$$V_h(s_h) \leftarrow (1-\alpha_t)V_h(s_h) + \alpha_t\left(r_h + \overline{V}_{h+1}(s_{h+1}) + \beta_t\right), \tag{2.2}$$

where the learning rate is $\alpha_t = (H+1)/(H+t)$. This update rule essentially follows the optimistic Q-learning algorithm [60] in the single-agent scenario, except that instead of estimating the Q-functions, we maintain optimistic estimates of the state value functions. This is because the definition of $Q(s,a)$ explicitly depends on the joint actions of all the agents, which cannot be observed in a decentralized environment. Such an argument is also consistent with the Optimistic Nash V-learning [34] and the V-OL [53] algorithms for RL in two-player zero-sum games.

Unlike RL in the single-agent problem where the agent takes an action with the largest optimistic Q-function, in a multi-agent environment, the agent proceeds more conservatively by running an adversarial bandit algorithm to account for the unobserved effects of other agents' policy changes. At each step $h \in [H]$ and each state $s_h \in \mathcal{S}$, we use a variant of online mirror descent with bandit feedback to compute a policy $\theta_h(\cdot \mid s_h)$. OMD is an iterative process that computes the current policy by carrying out a simple gradient update in the dual space, where the dual space is defined by a mirror map (or a regularizer) F. In our algorithm, we use a standard unnormalized negentropy regularizer $F(\theta) = \sum_{a=1}^{A}(\theta(a)\log(\theta(a)) - \theta(a))$ for $\theta \in \mathbb{R}_+^A$. Given a mirror map F, the F-induced Bregman divergence is defined as $D_F(u,v) = F(u) - F(v) - \langle u - v, \nabla F(v)\rangle$. Given the (bandit-feedback) loss vector $\hat{l}_h(s_h, \cdot)$ at step h and state s_h, the OMD update rule is given by (Line 13 of Algorithm 1):

$$\theta^{\text{new}}(\cdot \mid s_h) \leftarrow \arg\min_{\theta \in \Delta(\mathcal{A})}\left\{\eta_t\left\langle\theta, \hat{l}_h(s_h, \cdot)\right\rangle + D_F(\theta, \theta^{\text{old}}(\cdot \mid s_h))\right\},$$

where $\eta_t = \sqrt{\log A/(At)}$ is the learning rate. We remark that OMD itself is a well-developed algorithmic framework with a rich literature. But in our case, to be consistent with the changing learning rate in the

V-learning part and the high-probability nature of the sample complexity bounds, we additionally require an OMD algorithm to have (1) a dynamic learning rate and (2) a high probability regret bound, with respect to (3) a weighted definition of regret. Such a result is absent in the literature, as far as we know. Interestingly, incorporating OMD with a dynamic learning rate is an active and challenging sub-area per se: An impossibility result [104] has shown that standard OMD with an $\eta_t \propto \sqrt{1/t}$ learning rate can incur linear regret when the Bregman divergence is unbounded, which actually covers our choice of D_F. A stabilization technique [105] was later introduced to resolve this problem, by replacing the policy at each step with a convex combination of this policy and the initial policy. This stabilization technique is also helpful in our method (Line 14 in Algorithm 1), although the design of the convex combination is a little more involved, due to the weighted regret. We provide a more detailed description of the bandit subroutine and an analysis of our OMD algorithm in Section 2.6.

2.5 Theoretical Analyses

In this section, we present our main results on the sample complexity upper bound of V-learning OMD, and characterize the fundamental limits of the problem by providing a lower bound.

We first introduce a few notations to facilitate the analysis. For a given step $h \in [H]$ of episode $k \in [K]$, we denote by s_h^k the state that the agents observe at this step. Let $\mu_h^k : \mathcal{S} \to \Delta(\mathcal{A})$ and $\nu_h^k : \mathcal{S} \to \Delta(\mathcal{B})$ be the (interim) strategies at step h of episode k specified by θ_h in Algorithm 1 to agents 1 and 2, respectively. Let $a_h^k \in \mathcal{A}$ and $b_h^k \in \mathcal{B}$ be the actual actions taken by the two agents. Let $\overline{V}_h^k(s_h^k), V_h^k(s_h^k)$, and $N_h^k(s_h^k)$, respectively, be the values of $\overline{V}_h(s_h), V_h(s_h)$, and $N_h(s_h)$ in Algorithm 1 calculated by agent 1 at the *beginning* of the k-th episode. Symmetrically, define $\widetilde{V}_h^k(s_h^k)$ to be the value of $\overline{V}_h(s_h)$ calculated by agent 2, which does not necessarily take the same value as $\overline{V}_h^k(s_h^k)$. For notational convenience, we often suppress the sub/super-scripts (h, k) when there is no possibility of any ambiguity. When the state s_h^k is clear from the context, we also sometimes abbreviate $N_h^k(s_h^k)$ as n_h^k or even simply as t. For a fixed state $s \in \mathcal{S}$, let $t = N_h^k(s)$, and suppose that s was visited at episodes $k^1 < k^2 < \cdots < k^t$ at the h-th step before the k-th episodes. If we further define $\alpha_t^0 := \prod_{j=1}^t (1-\alpha_j)$ and $\alpha_t^i := \alpha_i \prod_{j=i+1}^t (1-\alpha_j)$, one can show that the update rule in (2.2) can be equivalently expressed as

$$V_h^k(s) = \alpha_t^0 (H-h+1) + \sum_{i=1}^t \alpha_t^i \left[r_h\left(s, a_h^{k^i}, b_h^{k^i}\right) + \overline{V}_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right) + \beta_i \right]. \tag{2.3}$$

This update rule follows the standard optimistic Q-learning algorithm [60] in single-agent RL, and has also appeared in RL for two-player zero-sum games [34]. In the following lemma, we recall several properties of α_t^i that are useful in our analysis.

Lemma 1. *(Properties for α_t^i, Lemma 4.1 in [60]).*

1. $\sum_{i=1}^t \alpha_t^i = 1$ *and* $\alpha_t^0 = 0$ *for* $t \geq 1$.
2. $\sum_{i=1}^t \alpha_t^i = 0$ *and* $\alpha_t^0 = 1$ *for* $t = 0$.
3. $\frac{1}{\sqrt{t}} \leq \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} \leq \frac{2}{\sqrt{t}}$ *for every* $t \geq 1$.
4. $\max_{i \in [t]} \alpha_t^i \leq \frac{2H}{t}$ *and* $\sum_{i=1}^t \left(\alpha_t^i\right)^2 \leq \frac{2H}{t}$ *for every* $t \geq 1$.
5. $\sum_{t=i}^\infty \alpha_t^i = 1 + \frac{1}{H}$ *for every* $i \geq 1$.

Algorithm 2: Construction of $(\bar{\mu}_h^k, \bar{\nu}_h^k)$

Require: A common random seed shared by both agents.
Input: The strategy trajectories $\{(\mu_h^k, \nu_h^k)\}_{h=1,k=1}^{H,K}$ specified by Algorithm 1.
for *step* $h' \leftarrow h$ *to* H **do**
    Receive $s_{h'}$;
    $t \leftarrow N_{h'}^k(s_{h'})$;
    Sample m from $[t]$ with $\mathbb{P}(m = i) = \alpha_t^i$ using the common random seed;
    Let k be the index of the episode in which $s_{h'}$ was visited for the m-th time during the execution of Algorithm 1;
    Execute the strategy pair $(\mu_{h'}^k(\cdot \mid s_{h'}), \nu_{h'}^k(\cdot \mid s_{h'}))$;

Based on the strategy trajectories $\{(\mu_h^k, \nu_h^k)\}_{h=1,k=1}^{H,K}$ of the two agents specified by Algorithm 1, we construct an auxiliary pair of correlated policies $(\bar{\mu}_h^k, \bar{\nu}_h^k)$ for each $(h,k) \in [H] \times [K]$. The construction of such correlated policies, largely inspired by the construction of the "certified policies" in [34], is formally defined in Algorithm 2. Such auxiliary correlated policies will play a significant role throughout our analysis, and are closely related to the CCE correlated policy that we will construct later. In words, $(\bar{\mu}_h^k, \bar{\nu}_h^k)$ proceeds as follows: It first observes the current state s_h, and let $t = N_h^k(s_h)$. Then, it randomly samples an episode index k^j from $\{k^1, k^2, \ldots, k^t\}$, the set of episodes in which the state s_h was previously visited during the execution of the first k episodes of Algorithm 1. Each index k^i has a probability of α_t^i to be selected. It is easy to verify that $\sum_{i=1}^t \alpha_t^i = 1$, and, hence, we have specified a well-defined probability distribution over the episode index set. Finally, $(\bar{\mu}_h^k, \bar{\nu}_h^k)$ executes the sampled strategy $(\mu_h^k(\cdot \mid s_h), \nu_h^k(\cdot \mid s_h))$ at step h, and then repeats a similar procedure using $(\bar{\mu}_{h+1}^{k^j}, \bar{\nu}_{h+1}^{k^j})$ at step $h+1$, and so on.

From the collection of such auxiliary correlated policies $\{(\bar{\mu}_h^k, \bar{\nu}_h^k)\}_{h=1,k=1}^{H,K}$, we finally construct a correlated policy $(\bar{\mu}, \bar{\nu})$, which we will show later is a CCE. A detailed description of the construction of $(\bar{\mu}, \bar{\nu})$ is presented in Algorithm 3. By construction, $(\bar{\mu}, \bar{\nu})$ first uniformly samples an index k from $[K]$ using a common random seed, and then proceeds by following the auxiliary correlated policy $(\bar{\mu}_1^k, \bar{\nu}_1^k)$. One can see that the notations we have defined are related through the following equation: $V_1^{\bar{\mu},\bar{\nu}}(s_1) = \frac{1}{K}\sum_{k=1}^K V_1^{\bar{\mu}_1^k,\bar{\nu}_1^k}(s_1)$. We also remark that the common random seed used in Algorithms 2 and 3 implicitly plays the role of the "trusted coordinator" typically used in the language of correlated equilibria.

For notational convenience, we further introduce the operator $\mathbb{P}_h V(s,a,b) = \mathbb{E}_{s' \sim P_h(\cdot|s,a,b)} V(s')$ for any value function V, and $\mathbb{D}_{\mu_h \times \nu_h} Q(s) = \mathbb{E}_{(a,b)\sim(\mu_h \times \nu_h)} Q(s,a,b)$ for any strategy pair (μ_h, ν_h) and any state-action value function Q. With these notations, for any $(s,a,b,h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H]$ and for any policy pair (μ, ν), the Bellman equations can be rewritten more succinctly as $Q_h^{\mu,\nu}(s,a,b) = \left(r_h + \mathbb{P}_h V_{h+1}^{\mu,\nu}\right)(s,a,b)$, and $V_h^{\mu,\nu}(s) = \left(\mathbb{D}_{\mu_h \times \nu_h} Q_h^{\mu,\nu}\right)(s)$. Recalling the definitions of the best responses, we further define $V_{k,H+1}^{\star,\bar{\nu}_{H+1}^k}(s) = 0, \forall k \in [K], s \in \mathcal{S}$. Then, we know that for each $(k,h,s) \in [K] \times [H] \times \mathcal{S}$,

$$V_{k,h}^{\star,\bar{\nu}_h^k}(s) \le \max_{\mu_h} \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{k^i,h+1}^{\star,\bar{\nu}_{h+1}^k}\right)(s), \tag{2.4}$$

In what follows, we will simply write $V_{k,h}^{\star,\bar{\nu}_h^k}$ as $V_{k,h}^{\star,\bar{\nu}}$ for notational convenience, because the time step (h,k) is always clear from the subscripts. We can also define $V_{k,h}^{\bar{\mu},\star}(s)$ analogously.

Remark 1. *Our definition of the correlated policy is inspired by the "certified policies" [34] for learning in two-player zero-sum Markov games, but with additional challenges to address: In the zero-sum setting, the*

Algorithm 3: Construction of the Correlated Policy $(\bar{\mu}, \bar{\nu})$

Require: A common random seed shared by both agents.
Input: The strategy trajectories $\{(\mu_h^k, \nu_h^k)\}_{h=1,k=1}^{H,K}$ specified by Algorithm 1.
Uniformly sample k from $[K]$ using the common random seed.
for *step* $h \leftarrow 1$ *to* H **do**
    Receive s_h;
    $t \leftarrow N_h^k(s_h)$;
    Sample m from $[t]$ with $\mathbb{P}(m = i) = \alpha_t^i$ using the common random seed;
    Let k be the index of the episode in which s_h was visited for the m-th time during the execution of Algorithm 1;
    Execute the strategy pair $(\mu_h^k(\cdot \mid s_h), \nu_h^k(\cdot \mid s_h))$;

Nash equilibrium value is always unique, and the regret with respect to the equilibrium value can be easily defined a priori (by means of the "duality gap"). But in general-sum games, the equilibrium value is not necessarily unique. We hence need to first specify an equilibrium before we are able to define the regret. In our analysis, the equilibrium value we choose is the one associated with the correlated policy $(\bar{\mu}, \bar{\nu})$. In addition, we also emphasize that the correlated policy is only used for analytical purposes; the actual strategies adopted by the agents during the execution of Algorithm 1 are still $\{(\mu_h^k, \nu_h^k)\}$.

We start with an intermediate result, which states that the optimistic $\overline{V}_h^k(s)$ and $\widetilde{V}_h^k(s)$ values are indeed high-probability upper bounds of $V_{k,h}^{\star,\bar{\nu}}(s)$ and $V_{k,h}^{\bar{\mu},\star}(s)$, respectively. The proof is deferred to Section 2.9 for clarity of presentation. It relies on a delicate investigation of a high-probability regret bound for OMD with a dynamic learning rate, which we will elaborate on in Section 2.6.

Lemma 2. *For any $p \in (0,1]$, let $\iota = \log(2S\max\{A,B\}T/p)$. It holds with probability at least $1-p$ that $\overline{V}_h^k(s) \geq V_{k,h}^{\star,\bar{\nu}}(s)$ and $\widetilde{V}_h^k(s) \geq V_{k,h}^{\bar{\mu},\star}(s)$, for all $(s,h,k) \in \mathcal{S} \times [H] \times [K]$.*

By construction of the auxiliary correlated policies $(\bar{\mu}_h^k, \bar{\nu}_h^k)$, we know that for any $(s,h,k) \in \mathcal{S} \times [H] \times [K]$, the corresponding value function can be written recursively as follows:

$$V_{k,h}^{\bar{\mu},\bar{\nu}}(s) = \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{k^i,h+1}^{\bar{\mu},\bar{\nu}} \right)(s),$$

and $V_{k,H+1}^{\bar{\mu},\bar{\nu}}(s) = 0$ for any $k \in [K], s \in \mathcal{S}$, where again notice that we have dropped the dependence on (h,k). The following result shows that, on average, the agents have no incentive to deviate from the correlated policies, up to a regret term of the order $\widetilde{O}(\sqrt{H^6SA/K})$.

Theorem 1. *For any $p \in (0,1]$, let $\iota = \log(2S\max\{A,B\}T/p)$. With probability at least $1-p$,*

$$\frac{1}{K}\sum_{k=1}^{K} \left(V_{k,1}^{\star,\bar{\nu}}(s_1) - V_{k,1}^{\bar{\mu},\bar{\nu}}(s_1) \right) \leq O(\sqrt{H^6SA\iota/K}), \text{ and}$$
$$\frac{1}{K}\sum_{k=1}^{K} \left(V_{k,1}^{\bar{\mu},\star}(s_1) - V_{k,1}^{\bar{\mu},\bar{\nu}}(s_1) \right) \leq O(\sqrt{H^6SB\iota/K}).$$

The proof of Theorem 1 can be found in Section 2.9. From the relationship between $(\bar{\mu}, \bar{\nu})$ and $(\bar{\mu}_1^k, \bar{\nu}_1^k)$, and that $V_1^{\bar{\mu},\bar{\nu}}(s_1) = \frac{1}{K}\sum_{k=1}^{K} V_1^{\bar{\mu}_1^k,\bar{\nu}_1^k}(s_1)$, we can immediately conclude from Theorem 1 that the correlated policy $(\bar{\mu}, \bar{\nu})$ constitutes an approximate CCE.

Corollary 1. *(Sample complexity of V-learning OMD). For any $p \in (0,1]$, set $\iota = \log(2S\max\{A,B\}T/p)$, and let the two agents run Algorithm 1 for K episodes with $K = \Omega(H^6 S\max\{A,B\}\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the two agents can obtain an ε-approximate coarse correlated equilibrium using a common random seed.*

Finally, to obtain a sample complexity lower bound for the problem, one simple way is to consider a Markov game instance where either $\mathcal{A}$ or $\mathcal{B}$ is a singleton, i.e., $A = 1$ or $B = 1$. In this case, there is no need to correlate the actions of the agents, and hence a CCE in such a game reduces to a NE. In addition, learning a NE against an opponent with a fixed policy is equivalent to learning an optimal policy in a fixed environment. Hence, we have reduced the problem of learning a CEE in a Markov game to a single-agent RL problem either for agent 2 or for agent 1. Applying the regret lower bound of single-agent RL yields the following result for RL in Markov games.

Corollary 2. *(Corollary of Theorem 5 in [96]). For any algorithm, the sample complexity on achieving an ε-approximate CCE in two-player general-sum Markov games is at least $\Omega(H^3 S\max\{A,B\}/\varepsilon^2)$.*

Comparing Corollaries 1 and 2, we see that the sample complexity of Algorithm 1 matches the information-theoretical lower bound in terms of the dependences on S, A, B and ε, leaving a gap only in the dependence of H. Notably, the tight dependence on $\max\{A,B\}$ is a natural benefit from decentralized learning, which would not have been achieved by centralized approaches.

2.6 Adversarial Bandits with Weighted Regret

In this section, we close the gap in the proof of Lemma 2 by formally presenting a bandit regret bound that we used in (2.12). Specifically, we consider an adversarial multi-armed bandit problem, and propose an online mirror descent based algorithm for this problem, which also serves as an important subroutine in Algorithm 1. Our OMD algorithm achieves an anytime high probability bound with respect to a weighted definition of regret. Such a result complements the Follow the Regularized Leader based algorithm in [34] and might be of independent interest.

Specifically, we consider an A-armed bandit problem, i.e., the action space is $\mathcal{A} = \{1, 2, \ldots, A\}$. The arms are associated with an adversarial sequence of loss vectors $(l_t)_{t=1}^T$, where $l_t \in [0,1]^A$. The bandit proceeds for T rounds. At each round t, the player specifies a distribution $\theta_t \in \Delta(\mathcal{A})$ over the actions, and takes an action a_t sampled from this distribution. We consider bandit feedback, where the player only observes the loss associated with the chosen action $l_t(a_t)$. The player's objective is to minimize the weighted regret with respect to the best fixed policy in hindsight for any time step $t \in [T]$:

$$\mathrm{Reg}_t(\theta^\star) := \sum_{i=1}^{t} w_i \mathbb{E}_{a\sim\theta^\star}\left[l_i(a_i) - l_i\left(a\right) \mid \mathcal{F}_i\right] = \sum_{i=1}^{t} w_i \left\langle \theta_i - \theta^*, l_i \right\rangle,$$

where $\theta^\star \in \Delta(\mathcal{A})$ is an arbitrary but fixed policy, $0 \le w_i \le 1$ is the weight of the regret for round i, and $\mathcal{F}_i$ is the σ-algebra generated by the events up to and including round $i-1$. We can check that such a problem formulation indeed captures the adversarial bandit subroutine with weighted regret used in the analysis of Lemma 2.

We present our OMD-based algorithm in Algorithm 4. We again use the unnormalized negentropy regularizer $F(\theta) = \sum_{a=1}^{A}(\theta(a)\log(\theta(a)) - \theta(a))$ with domain $\mathcal{D} = \mathrm{dom}(F)$. Direct calculation shows that the

Algorithm 4: Stabilized Online Mirror Descent with Weighted Regret

1 **Input:** The weight of the regret $w_t \in [0,1]$ for each round t.
2 **Initialize:** $\theta_1 \leftarrow \mathbf{1}/A := (\frac{1}{A}, \ldots, \frac{1}{A})$.
3 **for** $t \leftarrow 1$ *to* T **do**
4 Take action $a_t \sim \theta_t$, and observe loss $\tilde{l}_t(a_t)$;
5 $\hat{l}_t(a) \leftarrow \tilde{l}_t(a)\mathbb{1}\{a_t = a\} / (\theta_t(a) + \gamma_t)$ for all $a \in \mathcal{A}$, where $\gamma_t = \sqrt{\frac{\log A}{At}}$;
6 $\tilde{\theta}_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{D}} \left\{ \eta_t \left\langle \theta, \hat{l}_t \right\rangle + D_F(\theta, \theta_t) \right\}$, where $\eta_t \leftarrow \sqrt{\frac{\log A}{At}}$;
7 $\theta'_{t+1} \leftarrow \arg\min_{\theta \in \Delta(\mathcal{A})} D_F(\theta, \tilde{\theta}_{t+1})$;
8 $\theta_{t+1} \leftarrow \beta_t \theta'_{t+1} + (1-\beta_t)\theta_1$, where $\beta_t \leftarrow \frac{\eta_{t+1} w_t}{\eta_t w_{t+1}}$;

Bregman divergence with respect to F is

$$D_F(u,v) = F(u) - F(v) - \langle u - v, \nabla F(v) \rangle = \sum_{a=1}^{A} u(a) \log(u(a)/v(a)),$$

which coincides with the Kullback–Leibler divergence when u and v are defined on the simplex.

The structure of Algorithm 4 essentially follows the well-developed OMD framework, but with the following two critical refinements in order to achieve an anytime high-probability regret bound: First, to establish high-probability regret guarantees, we use an *implicit exploration* technique [106], and deliberately maintain a biased estimate of the true losses as

$$\hat{l}_t(a) \leftarrow \frac{\tilde{l}_t(a)}{\theta_t(a) + \gamma_t} \mathbb{1}\{a_t = a\}.$$

We can show (in Lemma 7 below) that with an appropriately chosen $\gamma_t > 0$, loss estimates of this form constitute a lower confidence bound of the true losses, and hence are critical in establishing high-probability regret guarantees of the bandit problem. Second, to achieve an anytime regret bound, we use a stabilization technique [105] by replacing the policy at each step with a convex combination of this policy and the initial policy (Line 8). It has been shown in [104] that standard OMD with an $\eta_t \propto \sqrt{1/t}$ learning rate can incur linear regret when the Bregman divergence is unbounded. To resolve this unboundedness, the stabilization technique mixes a small fraction of θ_1 into each iterate θ_t. In this sense, every iterate θ_t remains somewhat close (with respect to the Bregman divergence) to the point θ_1. Since the distance between θ_1 and any other point in $\Delta(\mathcal{A})$ is small (due to our initialization of θ_1), we know that each iterate θ_t is also not too far from all the other points in $\Delta(\mathcal{A})$. This hence ensures that the Bregman divergences involved with the iterates are always bounded. In the original stabilization technique [105], a $(1 - \frac{\eta_{t+1}}{\eta_t})$-fraction of θ_1 is mixed into each iterate θ_t; while in our algorithm, this fraction is set to $1 - \frac{\eta_{t+1} w_t}{\eta_t w_{t+1}}$ because we need to additionally address the weighted regret. The following theorem presents the regret guarantee of Algorithm 4.

Theorem 2. *For any $p \in (0,1]$, let $\iota = \log(AT/p)$. For any $t \in [T]$, suppose $\eta_i \le 2\gamma_i$, $0 \le w_i \le 1$, $\beta_i \in (0,1], \forall i \in [t]$, and γ_i is non-increasing in i. Then, with probability at least $1 - 3p$, the weighted regret of Algorithm 4 is upper bounded by:*

$$Reg_t(\theta^\star) \le 2 \max_{i \le t} w_i \sqrt{At\iota} + \frac{3\sqrt{A\iota}}{2} \sum_{i=1}^{t} \frac{w_i}{\sqrt{i}} + \frac{1}{2} \max_{i \le t} w_i \iota + \sqrt{2\iota \sum_{i=1}^{t} w_i^2}.$$

We can verify that our choices of the parameter values in Algorithm 1 indeed satisfy the requirements in Theorem 2, that is, $\eta_i \le 2\gamma_i$, $0 \le w_i \le 1$, $\beta_i \in (0,1], \forall i \in [t]$, and γ_i is non-increasing in i. Therefore, the regret bound in Theorem 2 can be applied to the proof of Lemma 2. The only caution is that in this section we have assumed for simplicity that the loss function is bounded in $[0,1]$, while the actual losses in Section 2.5 are bounded in $[0,H]$. Hence, multiplying the regret bound in Theorem 2 by a factor of H leads to the result in (2.12).

A final remark is that Algorithm 4 assumes that the weights of the regret w_i for $1 \le i \le t$ are given a priori; but when Algorithm 4 is utilized as a subroutine in Algorithm 1, the weight w_i at round i actually corresponds to α_t^i, which cannot be pre-computed when t is not given. To address this subtlety, we specifically design Algorithm 4 in a way such that the weights w_i influence the algorithm only through $\beta_i \leftarrow \frac{\eta_{i+1} w_i}{\eta_i w_{i+1}}$ (Line 8). By the definition of α_t^i, we see that $\frac{w_i}{w_{i+1}} = \frac{\alpha_t^i}{\alpha_t^{i+1}} = \frac{\alpha_i(1-\alpha_{i+1})}{\alpha_{i+1}}$ can be calculated even when the value of t is unknown. In this way, Algorithm 1 has bypassed the subtlety that the weights of the regret should be given beforehand, as required in Algorithm 4.

Proof. (of Theorem 2). The weighted regret $\mathrm{Reg}_t(\theta^\star)$ can be decomposed into three terms:

$$\begin{aligned}\mathrm{Reg}_t(\theta^\star) =& \sum_{i=1}^t w_i \langle \theta_i - \theta^\star, l_i \rangle \\ =& \underbrace{\sum_{i=1}^t w_i \left\langle \theta_i - \theta^\star, \hat{l}_i \right\rangle}_{\text{(A)}} + \underbrace{\sum_{i=1}^t w_i \left\langle \theta_i, l_i - \hat{l}_i \right\rangle}_{\text{(B)}} + \underbrace{\sum_{i=1}^t w_i \left\langle \theta^\star, \hat{l}_i - l_i \right\rangle}_{\text{(C)}}.\end{aligned}$$

We bound each of the three terms (A), (B) and (C) in Lemmas 9, 10, and 11 of Section 2.10, respectively. By setting $\eta_t = \gamma_t = \sqrt{\frac{\log A}{At}}$, we can verify that the conditions in Lemma 9 and Lemma 11 are satisfied. We specifically define $\eta_{t+1} = \eta_t$ and $w_{t+1} = w_t$. One can verify that these two parameters influence the algorithm only through β_t, and the results stated in the lemmas still hold. Plugging back the results and taking a union bound, it holds with probability at least $1 - 3p$:

$$\begin{aligned}\mathrm{Reg}_t(\theta^\star) \le& \frac{w_{t+1}\log A}{\eta_{t+1}} + \frac{A}{2}\sum_{i=1}^t \eta_i w_i + \frac{1}{2}\max_{i\le t} w_i \iota \\ &+ A\sum_{i=1}^t \gamma_i w_i + \sqrt{2\iota \sum_{i=1}^t w_i^2} + \max_{i\le t} w_i \iota/\gamma_t \\ =& 2\max_{i\le t} w_i \sqrt{At\iota} + \frac{3\sqrt{A\iota}}{2}\sum_{i=1}^t \frac{w_i}{\sqrt{i}} + \frac{1}{2}\max_{i\le t} w_i\iota + \sqrt{2\iota\sum_{i=1}^t w_i^2}.\end{aligned}$$

This completes the proof of Theorem 2. □

2.7 Stage-Based V-Learning for General-Sum Markov Games

So far, we have presented V-learning OMD, the first decentralized MARL algorithm for learning CCE in general-sum Markov games. However, V-learning OMD involves a complicated no-*weighted*-regret bandit analysis (Section 2.6), which turns out to be an undesirable routine that appears in the analysis of many prior V-learning-based methods [12], [16], [63].

In this section, we improve V-learning OMD by presenting a *stage-based* V-learning method. We show that stage-based V-learning helps significantly simplify the algorithmic design and analysis of V-learning OMD, and circumvent the rather complicated no-weighted-regret bandit subroutine. We demonstrate that stage-based V-learning can be combined with any off-the-shelf no-regret learning algorithm to improve the sample complexity of V-learning OMD. We also show that, when combined with a no-swap-regret learning algorithm, stage-based V-learning can be used to learn a correlated equilibrium (CE) in general-sum Markov games, a stronger solution concept than CCE.

Algorithm 5: Stage-Based V-Learning for CCE (agent i)

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

2.7.1 Learning CCE

The Stage-Based V-Learning for CCE algorithm run by a generic agent $i \in \mathcal{N}$ is presented in Algorithm 5. The agent maintains upper confidence bounds on the value functions to actively explore the unknown environment and uses a stage-based rule to independently update the value estimates.

For each step-state pair $(h,s) \in [H] \times \mathcal{S}$, we divide the visitations to this pair into multiple *stages*, where the lengths of the stages increase exponentially at a rate of $(1+1/H)$ [61]. Specifically, we let $e_1 = H$, and $e_{i+1} = \lfloor (1+1/H)e_i \rfloor, i \geq 1$ denote the lengths of the stages, and let the partial sums $\mathcal{L} := \{\sum_{i=1}^{j} e_i \mid j = 1,2,3,\dots\}$ denote the set of ending times of the stages. For each (h,s) pair, we update our optimistic estimates $\overline{V}_h(s_h)$ of the value function at the end of each stage (i.e., when the total number of visitations to (s,h) lies in the set $\mathcal{L}$), using samples only from this single stage (Lines 11-16). This way, our stage-based V-learning ensures that only the most recent $O(1/H)$ fraction of the collected samples are used to calculate $\overline{V}_h(s_h)$, while the first $1-O(1/H)$ fraction is forgotten. Such a stage-based update framework in some sense mimics the celebrated optimistic Q-learning algorithm with a learning rate of $\alpha_t = \frac{H+1}{H+t}$ [60], which also roughly uses the last $O(1/H)$ fraction of samples for value updates. Stage-based value updates also create a stage-wise stationary environment for the agents, thereby partly alleviating the well-known challenge of *non-stationarity* in MARL. As a side remark, stage-based Q-learning has also achieved near-optimal regret

Algorithm 6: Construction of the Output Policy $\bar{\pi}$

Input: The distribution trajectory specified by Algorithm 5: $\{\mu_{h,i}^k : i \in \mathcal{N}, h \in [H], k \in [K]\}$;
Uniformly sample k from $[K]$;
for *step* $h \leftarrow 1$ *to* H **do**
    Receive s_h;
    Take joint action $\boldsymbol{a}_h \sim \times_{i=1}^N \mu_{h,i}^k(\cdot \mid s_h)$;
    Uniformly sample j from $\{1, 2, \ldots, \check{N}_h^k(s_h)\}$;
    Set $k \leftarrow \check{l}_{h,j}^{k'}$, where $\check{l}_{h,j}^k$ is the index of the episode such that state s_h was visited the j-th time (among the total $\check{N}_h^k(s_h)$ times) in the last stage;

bounds in single-agent RL [61].

At each time step h and state s_h, agent i selects its action $a_{h,i}$ by following a distribution $\mu_{h,i}(\cdot \mid s_h)$, where $\mu_{h,i}(\cdot \mid s_h)$ is updated using an adversarial bandit subroutine (Lines 9-10). This is consistent with the recent works under the V-learning framework [12], [16], [63], but with a vital improvement: Existing works using the celebrated $\alpha_t = \frac{H+1}{H+t}$ learning rate for V-learning inevitably entail a no-*weighted*-regret bandit problem, because such a time-varying learning rate assigns different weights to each step in the history. A few methods such as weighted follow-the-regularized-leader [12], [63] and stabilized online mirror descent [16] have been recently proposed to address such a challenge, by simultaneously dealing with a changing step size, a weighted regret, and a high-probability guarantee, at the cost of less natural algorithms and more sophisticated analyses. In contrast, our stage-based V-learning assigns uniform weights to each step in the previous stage, and hence leads to a standard no(-average)-regret bandit problem. This allows us to directly plug in any off-the-shelf adversarial bandit algorithm and its analysis to our problem. For example, Algorithm 5 utilizes a simple Exp3 [107] subroutine for policy updates, and a standard implicit exploration technique [106] to achieve high-probability guarantees. We provide a more detailed discussion on such an improvement in Remark 2 of Section 2.11.

Based on the policy trajectories from Algorithm 5, we construct an output policy profile $\bar{\pi}$ that we will show is a CCE. For any step $h \in [H]$ of an episode $k \in [K]$ and any state $s \in \mathcal{S}$, we let $\mu_{h,i}^k(\cdot \mid s) \in \Delta(\mathcal{A}_i)$ be the distribution prescribed by Algorithm 5 at this step. Let $\check{N}_h^k(s)$ denote the value of $\check{N}_h(s)$ at the *beginning* of the k-th episode. Our construction of the output policy is presented in Algorithm 6, which follows the "certified policies" introduced in [34]. We further let the agents sample the episode indices using a common source of randomness, and hence the output policy is correlated by nature. Such common randomness is also termed a correlation device, and is standard in decentralized learning [108]–[110]. In practice, this can be achieved by letting the agents agree on a common random seed at the very beginning of the game, which only requires exchanging a single scalar value. Note that the correlation device is never used during the learning process to coordinate the exploration, but is simply used to synchronize the selection of the policies after they have been generated. A common random seed is generally considered as a mild assumption and does not break the decentralized paradigm. It is also worth remarking that our stage-based update rule simplifies the generating procedure of the output policy: In the original construction of [34], the certified policy plays a weighted mixture of $\{\mu_{h,i}^k(\cdot \mid s) : k \in [K]\}$, while in Algorithm 6, we only need to uniformly sample an episode index from the previous stage.

The following theorem presents the sample complexity guarantee of Algorithm 5 for learning CCE in general-sum Markov games. Our sample complexity bound improves over [16] and matches those established in [12], [63], while significantly simplifying their algorithmic design and analysis. The proof is deferred to

Algorithm 7: Stage-Based V-Learning for CE (agent i)

Initialize: $\overline{V}_{h,i}(s) \leftarrow H-h+1, \tilde{V}_{h,i}(s) \leftarrow H-h+1, N_h(s) \leftarrow 0, \check{N}_h(s) \leftarrow 0, \check{r}_{h,i}(s) \leftarrow 0, \check{v}_{h,i}(s) \leftarrow 0, \check{T}_h(s) \leftarrow H, p_{h,i}(a \mid s) \leftarrow 1/A_i$, $L^s_{h,i}(a' \mid a) \leftarrow 0$, $\forall h \in [H], s \in \mathcal{S}, a, a' \in \mathcal{A}_i$.

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

Section 2.11 for the clarity of presentation.

Theorem 3. *(Sample complexity of learning CCE). For any $p \in (0,1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 5 for K episodes with $K = O(SA_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ of Algorithm 6 is an ε-approximate CCE.*

2.7.2 Learning CE

In this subsection, we aim at learning a more strict solution concept named correlated equilibrium. Our algorithm for learning CE, formally presented in Algorithm 7, also relies on stage-based V-learning, but replaces the no-regret learning subroutine in Algorithm 5 with a no-swap-regret learning algorithm. Our no-swap-regret algorithm follows the generic reduction introduced in [65], and converts a follow-the-regularized-leader (FTRL) algorithm with sublinear external regret to a no-swap-regret algorithm [12]. A detailed description of such a no-swap-regret FTRL subroutine, as well as its regret analysis, is presented in Section 2.12. Again, due to the stage-based update rule, we can avoid the additional complication of dealing with a weighted swap regret as faced by recent works [12], [63]. The construction of the output policy $\bar{\pi}$ is the same as Algorithm 6 and thus omitted. The following theorem shows that our sample complexity guarantee for learning CE improves over [63] and matches the best known result in the literature [12]. The proof of the theorem can also be found in Section 2.12.

Theorem 4. *(Sample complexity of learning CE). For any $p \in (0,1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 7 for K episodes with $K = O(SA^2_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ is an ε-approximate CE.*

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

2.7.3 Simulations

In this section, we demonstrate the empirical performances of our algorithm, and compare their performances with various benchmarks. We evaluate Algorithm 5 on two Markov games, namely GoodState and BoxPushing [111].

The GoodState task is a simple Markov team problem inspired by [77]. It has two states $\mathcal{S} = \{s_0, s_1\}$, where s_0 is the "good state" and s_1 is the "bad state". Each agent has two candidate actions $\mathcal{A}_1 = \{a_0, a_1\}$ and $\mathcal{A}_2 = \{b_0, b_1\}$. The reward function at each state is presented in Table 2.3. Specifically, at state s_1, both agents get a reward of 0 no matter what actions they select, while at state s_0, they will obtain a strictly positive reward if they either take the joint action (a_0, b_1) or the one (a_1, b_0). The state transition function is defined as follows:

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

and all the other transitions happen with probability ε. Intuitively, no matter which state the agents are in, they will transition to the good state s_0 with a high probability $1 - \varepsilon$ at the next step as long as they select the action pair (a_0, b_1). All the other joint actions will lead to the bad state s_1 with a high probability $1 - \varepsilon$. The task hence rewards the agents who learn to consistently play the action pair (a_0, b_1). We run Algorithm 5 on this example for $K = 50000$ episodes, each episode containing $H = 10$ steps. We set the transition probability $\varepsilon = 0.1$. For Algorithm 5, the step size is set to be $\eta_i = \frac{1}{5\sqrt{A_i \check{T}_h(s_h)}}$, and the implicit exploration parameter is $\gamma_i = \eta_i/2$.

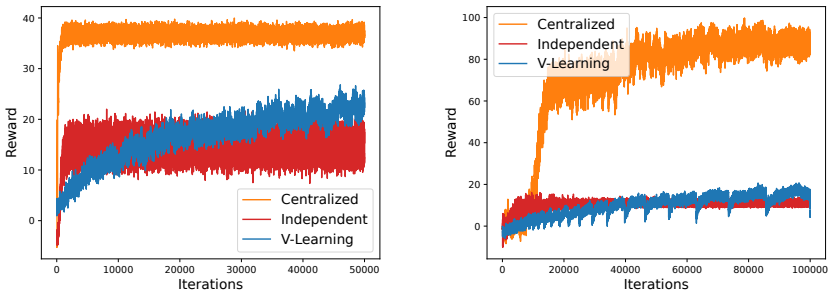
s_0	b_0	b_1
a_0	-2	5
a_1	2	-2

s_1	b_0	b_1
a_0	0	0
a_1	0	0

Table 2.3: Reward tables for GoodState.

The BoxPushing task [111] is a classic DecPOMDP problem with with ∼100 states. It has two 2 agents, where each agent has 4 candidate actions. In the original BoxPushing problem, each agent only has a partial observation of the state. We make proper modifications to the task so that the agents can fully observe the state information and fit in our problem formulation. For Algorithm 5 on this task, the step size is set to be $\eta_i = \frac{1}{20\sqrt{A_i \check{T}_h(s_h)}}$, and the implicit exploration parameter is $\gamma_i = \eta_i/2$.

We compare Algorithm 5 with two meaningful benchmarks. The first benchmark is a "Centralized" oracle. This oracle acts as a centralized coordinator that can control the actions of both agents. Such an oracle essentially converts the multi-agent task into a single-agent RL problem. In our simulations, we implement "Centralized" by using a Hoeffding-based variant of a state-of-the-art single-agent RL algorithm UCB-ADVANTAGE [112]. This algorithm has achieved a tight sample complexity bound for single-agent RL in theory, and has also demonstrated remarkable empirical performances in practice [23]. Such an algorithm



(a) GoodState (b) BoxPushing

Figure 2.1: Rewards of Algorithm 5 on the (a) GoodState and (b) BoxPusing tasks. "V-Learning" denotes the policies at the current iterate t of Algorithms 5. "Centralized" is an oracle that can control the actions of the agents in a centralized way. In "Independent", each agent runs a naïve single-agent Q-learning algorithm independently, by taking greedy actions with respect to its local Q-function estimates. All results are averaged over 20 runs.

could provide a strong performance upper bound in our task. The second benchmark we consider is the naïve "Independent" Q-learning. Specifically, we let each agent run a single-agent Q-learning algorithm independently, without being aware of the existence of the other agent or the structure of the game. Each agent maintains an local optimistic Q-function, and takes greedy actions with respect to such optimistic estimates, without taking into account the other agents' actions. Since the agents update their policies simultaneously, the stationarity assumption of the environment in single-agent RL quickly collapses, and the theoretical guarantees for single-agent Q-learning no longer hold. This is also reminiscent of the "independent learner" approach proposed in an early work [40] for learning in Markov teams. We believe that such a benchmark could provide meaningful intuitions about the consequences of not taking care of the multi-agent structure in decentralized methods. In our simulations, we implement such a benchmark by letting each agent running a variant of the single-agent UCB-ADVANTAGE [112] algorithm independently.

Figure 2.1 illustrates the performances of our Algorithm 5 and the two benchmark methods in terms of the collected rewards, where "V-Learning" denotes the policy at the current iterate t of Algorithms 5. Notice that the *actual* policy trajectories of the algorithm numerically converge and achieve high rewards. This is more encouraging than our theoretical guarantees, because for Algorithm 5, our Theorem 3 only holds for a "certified" output policy but not the last-iterate policy. Further, Algorithm 5 outperforms the "Independent" learning benchmark on the two tasks. On the other hand, the "Independent" benchmark converges, albeit faster, to a clearly suboptimal value. This reiterates that the naïve idea of independent learning does not work well for MARL in general, and a careful treatment of the game structure (like our adversarial bandit subroutine) is necessary. Finally, the implemented algorithm takes much fewer samples to converge than our theoretical results suggested. This indicates that the theoretical bounds might be overly conservative, and our algorithm could converge much faster in practice.

Table 2.4: No-regret learning convergence rates in NFGs and Markov games.

Learning objective	Normal-form games	Markov games
Nash equilibrium (two-player zero-sum)	$\widetilde{O}(T^{-1})$ [113]	$O(T^{-1})$ [122]
Correlated equilibrium	$\widetilde{O}(T^{-1})$ [120]	$\widetilde{O}(T^{-1/4})$ [123] $\widetilde{O}(T^{-1})$ (Theorem 5)
Coarse correlated equilibrium	$\widetilde{O}(T^{-1})$ [118]	$\widetilde{O}(T^{-3/4})$ [121] $\widetilde{O}(T^{-1})$ (Theorem 6)

2.8 $\widetilde{O}(T^{-1})$ Convergence in Full-Information Markov Games

So far, we have primarily considered MARL algorithms that run an adversarial bandit procedure at each state. We have shown that such algorithms achieve an $\widetilde{O}(\sqrt{T})$ regret[2] with respect to an arbitrary reward sequence after T iterations, which directly implies an $\widetilde{O}(1/\sqrt{T})$ convergence rate to a (coarse) correlated equilibrium. While the $O(\sqrt{T})$ regret is unimprovable against an adversarial environment, it need not be the case for learning equilibria in games because each player in a repeated game is not facing adversarial payoffs, but instead is interacting with other players who also exhibit certain learning behavior. In this section, we exploit this structure and seek to establish a faster $\widetilde{O}(T^{-1})$ convergence to CCE/CE in full-information general-sum Markov games.

Faster convergences than $\widetilde{O}(1/\sqrt{T})$ are indeed shown to be possible for certain scenarios of learning in games. For learning Nash equilibria (NE) in two-player zero-sum normal-form games (NFGs), the seminal work [113] developed an algorithm based on the Nesterov's excessive gap technique and established its $\widetilde{O}(T^{-1})$ convergence when the algorithm is adopted by both players. Recent works [114]–[120] significantly strengthened this line of results by devising other no-regret learning dynamics that find different equilibrium solutions at a faster rate than $O(1/\sqrt{T})$. Notably, [115] showed that if all the players in a general-sum NFG employ an *optimistic* version of follow-the-regularizer-leader (henceforth OFTRL), the players' strategies converge to the set of CCE at a fast rate of $O(T^{-3/4})$; such a rate was later improved to $\widetilde{O}(T^{-1})$ by [118]. More recently, the $\widetilde{O}(T^{-1})$ rate was established for swap regrets and CE in NFGs [119], [120]. Despite the encouraging fast convergence results in NFGs, very few results are known for the more challenging regime of Markov games. The only exceptions include [121] and [122], who established the $\widetilde{O}(T^{-1})$ convergence of OFTRL (together with smooth value updates) to NE in two-player zero-sum full-information Markov games, matching the best rates in NFGs. As for general-sum Markov games, the best known results for CCE and CE are $\widetilde{O}(T^{-3/4})$ [121] and $\widetilde{O}(T^{-1/4})$ [123], respectively, which largely lag behind their $\widetilde{O}(T^{-1})$ counterparts in NFGs. In fact, establishing $\widetilde{O}(T^{-1})$ convergence to CCE or CE in general-sum Markov games has been raised as an important open question by [122].

In this section, we close this gap by developing no-regret learning algorithms with accompanying value update procedures and establishing their fast $\widetilde{O}(T^{-1})$ convergence to CCE or CE in general-sum Markov games. For CE (Section 2.8.1), we consider the OFTRL algorithm with a log-barrier regularizer, and integrate it with the celebrated external-to-swap-regret reduction [65] and smooth value updates. Our $\widetilde{O}(T^{-1})$ convergence analysis builds on a Regret bounded by Variation in Utilities (RVU) property [115] for the weighted swap regret at each state. We make a seemingly trivial observation that swap regrets are always non-negative and use it to easily bound the second-order path lengths of the learning dynamics. For CCE

[2]In this section, we use $\widetilde{O}(\cdot)$ to suppress the poly-logarithmic dependence on T.

Algorithm 8: Optimistic follow-the-regularized-leader for correlated equilibria (agent i)

Initialize: $Q^0_{h,i}(s,\boldsymbol{a}) \leftarrow 0, \pi^0_{h,i}(s,a_i) \leftarrow 1/A_i, \forall s \in \mathcal{S}, h \in [H], a_i, a_i' \in \mathcal{A}_i, \boldsymbol{a} \in \mathcal{A}_{\text{all}}$;
for *iteration* $t \leftarrow 1$ *to* T **do**
  Policy update:
  for *action* $a_i \in \mathcal{A}_i$ **do**
    $\ell^{t,a_i}_{h,i}(s,a_i') \leftarrow \sum_{j=1}^{t-1} w_j \pi^j_{h,i}(s,a_i)[Q^j_{h,i}\pi^j_{h,-i}](s,a_i') + w_t \pi^{t-1}_{h,i}(s,a_i)[Q^{t-1}_{h,i}\pi^{t-1}_{h,-i}](s,a_i')$;
    $q^{t,a_i}_{h,i}(s,\cdot) \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\Delta(\mathcal{A}_i)} \left(\langle \boldsymbol{x}, \eta \ell^{t,a_i}_{h,i}(s,\cdot)/w_t\rangle - \mathcal{R}(\boldsymbol{x})\right), \forall s \in \mathcal{S}, h \in [H]$;
  Find $\pi^t_{h,i}$ such that $\pi^t_{h,i}(s,\cdot) = \sum_{a_i\in\mathcal{A}_i} \pi^t_{h,i}(s,a_i) q^{t,a_i}_{h,i}(s,\cdot), \forall s \in \mathcal{S}, h \in [H], a_i \in \mathcal{A}_i$;
  Value update:
  for $h \leftarrow H$ *to* 1 **do**
    $Q^t_{h,i}(s,\boldsymbol{a}) \leftarrow (1-\alpha_t) Q^{t-1}_{h,i}(s,\boldsymbol{a}) + \alpha_t \left(r_{h,i} + P_h[Q^t_{h+1,i}\pi^t_{h+1}]\right)(s,\boldsymbol{a}), \forall s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}_{\text{all}}$;
Output policy: $\bar{\pi} = \bar{\pi}^T_1$, where $\bar{\pi}^t_h$ is defined in Algorithm 9.

(Section 2.8.2), we consider standard OFTRL with negative entropy regularization but combine it with a stage-based value update scheme. We show that this algorithm induces a no-average-regret problem within each stage, which allows us to apply existing analysis for the *individual* regret of the players [118]. Table 2.4 compares our results with the best-known convergence rates of no-regret learning in NFGs and Markov games. We further provide numerical results (Section 2.8.3) to validate the $\widetilde{O}(T^{-1})$ convergence behavior of our algorithms.

Notations. For notational convenience, for any value function $V : \mathcal{S} \to \mathbb{R}$ (as defined in Section 2.3), we define $[P_h V](s,\boldsymbol{a}) := \mathbb{E}_{s'\sim P_h(\cdot|s,\boldsymbol{a})}[V(s')]$. For an arbitrary Q-function $Q_{h,i} : \mathcal{S} \times \mathcal{A}_{\text{all}} \to \mathbb{R}$, we write $[Q_{h,i}\pi_h](s) := \langle Q_{h,i}(s,\cdot), \pi_h(s,\cdot)\rangle$ and $[Q_{h,i}\pi_{h,-i}](s,a_i) := \langle Q_{h,i}(s,a_i,\cdot), \pi_{h,-i}(s,\cdot)\rangle$ for short.

Full-information feedback. Following [54], [121]–[123], we consider the full-information feedback setting where each agent can observe the expected rewards it would have received had it played any candidate action. In our formulation, this can be interpreted as an oracle from which each agent i can query $[Q_{h,i}\pi_{h,-i}](s,a_i)$ for each candidate action $a_i \in \mathcal{A}_i$ at any state $s \in \mathcal{S}$.

2.8.1 Convergence to Correlated Equilibria

In this subsection, we present our optimistic follow-the-regularized-leader (OFTRL) algorithm for learning correlated equilibria in general-sum Markov games and then establish its $\widetilde{O}(T^{-1})$ convergence.

Algorithm 8 describes the OFTRL procedure run by agent $i \in \mathcal{N}$. Since the algorithms run by all the agents are exactly symmetric, in the following, we only illustrate our algorithm using a single agent i as an example. Algorithm 8 consists of three major components: The policy update step that computes the strategy for each matrix game, the value update step that updates the (Q-)value functions, and the policy output step that generates a CE policy.

Policy update. At each fixed $(s,h) \in \mathcal{S} \times [H]$, the agents are essentially faced with a sequence of matrix games, where the payoff matrix for agent i in the t-th matrix game is given by the estimated Q-function $Q^t_{h,i}(s,\cdot)$ at the corresponding iteration t. For learning CE in matrix games, a folklore result suggests that each agent should employ a no-swap-regret learning algorithm. Specifically, suppose that each agent employs a no-swap-regret algorithm such that the cumulative swap regret up to time $T \in \mathbb{N}_+$ is upper bounded by SwapReg^T; then, the empirical distribution of the joint actions played by the players is an $(\text{SwapReg}^T/T)$-approximate CE [35].

For a fixed matrix game at $(s,h)\times\mathcal{S}\times[H]$, we follow the generic reduction introduced in [65] to obtain a no-swap-regret learning algorithm $\mathscr{A}_{\text{swap}}$ from a no-(external-)regret base algorithm $\mathscr{A}$. Specifically, [65] maintain a separate no-regret algorithm $\mathscr{A}_a$ for each candidate action $a\in\mathcal{A}_i$ of the agent. $\mathscr{A}_{\text{swap}}$ computes a strategy by combining the strategies of the A_i base algorithms. At time step $t\in[T]$, each base algorithm $\mathscr{A}_a$ outputs a distribution $q^{t,a}(\cdot)\in\Delta(\mathcal{A}_i)$, where $q^{t,a}(a')$ is the probability that it selects $a'\in\mathcal{A}_i$. Then, a (row) stochastic matrix $q^t\in\mathbb{R}^{A_i\times A_i}$ is constructed, where the a-th row of q^t is equal to the $q^{t,a}$ vector. $\mathscr{A}_{\text{swap}}$ obtains the action selection strategy by computing a stationary distribution[3] $\pi^t\in\Delta(\mathcal{A}_i)$ of q^t such that $(q^t)^\top\pi^t=\pi^t$. Upon receiving the payoff vector $\boldsymbol{u}^t\in\mathbb{R}^{A_i}$ (in the case of Algorithm 8, $\boldsymbol{u}^t=[Q^j_{h,i}\pi^j_{h,-i}](s,\cdot)$ for agent i) from the environment, $\mathscr{A}_{\text{swap}}$ returns to each $\mathscr{A}_a$ base algorithm a $\pi^t(a)$ fraction of the received utility, so that $\mathscr{A}_a$ is updated with a utility vector of $\pi^t(a)\boldsymbol{u}^t\in\mathbb{R}^{A_i}$. It is shown that $\mathscr{A}_{\text{swap}}$ guarantees no-swap-regret as long as each base algorithm $\mathscr{A}_a$ has sublinear (external) regret in T.

In Algorithm 8, we use weighted OFTRL as the no-regret base algorithm $\mathscr{A}$. OFTRL [115] extends the standard FTRL paradigm by maintaining a prediction sequence $\boldsymbol{m}^t$ of the utilities. Given a utility sequence $(\boldsymbol{u}^1,\dots,\boldsymbol{u}^T)$, OFTRL computes the strategies by

$$\boldsymbol{x}^t:=\operatorname*{argmax}_{\boldsymbol{x}\in\Delta(\mathcal{A}_i)}\left\{\eta\left\langle\boldsymbol{x},\boldsymbol{m}^t+\sum_{j=1}^{t-1}\boldsymbol{u}^j\right\rangle-\mathcal{R}(\boldsymbol{x})\right\},\tag{2.5}$$

where $\eta>0$ is the learning rate, and $\mathcal{R}$ is the regularizer. In Algorithm 8, we instantiate (2.5) with $\boldsymbol{m}^t=\boldsymbol{u}^{t-1}$ and the log-barrier regularizer $\mathcal{R}(\boldsymbol{x})=-\sum_{a_i\in\mathcal{A}_i}\log(\boldsymbol{x}[a_i])$. Such a log-barrier regularizer satisfies the self-concordant condition in [120], which is used to establish the Regret bounded by Variation in Utilities (RVU) property [115] of the swap regret. Due to the time-varying learning rates in the value update step (to be discussed momentarily), we additionally use a weighted variant of OFTRL that considers a weighted sum over the utility sequence. The choice of the weights $\{w_j\}_{j\in[t]}$ will also be defined shortly. Combining the OFTRL base algorithm, the utility weights and the external-to-swap-regret reduction, we arrive at the policy update rule as presented in Algorithm 8. With the [65] reduction, we name our no-swap-regret algorithm BM-OFTRL.

Value update. For any $(h,s,\boldsymbol{a})$, we update the Q-value estimates at each iteration in a Bellman manner using a weighted average of previous estimates. We perform incremental updates using the classic step size $\alpha_t=(H+1)/(H+t)$ proposed by [60]. With this step size, the value update rule in Algorithm 8 effectively becomes:

$$Q^t_{h,i}(s,\boldsymbol{a})=\sum_{j=1}^{t}\alpha_t^j\left(r_{h,i}+P_h[Q^j_{h+1,i}\pi^j_{h+1}]\right)(s,\boldsymbol{a}),\forall s\in\mathcal{S},\boldsymbol{a}\in\mathcal{A}_{\text{all}},\tag{2.6}$$

where $\alpha_t^j:=\alpha_j\prod_{j'=j+1}^{t}(1-\alpha_{j'})$ and $\alpha_t^t:=\alpha_t$. One can verify that $\sum_{j=1}^{t}\alpha_t^j=1$. Given the time-varying weights α_t^j, to ensure that our policy update step is no-swap-regret in the matrix games defined by the Q-value estimates, we define the weights of our weighted OFTRL procedure in Algorithm 8 to be $w_j:=\alpha_t^j/\alpha_t^1$ for any fixed $t\in[T]$.

Policy output. Our output policy $\bar{\pi}$ is a state-wise weighted average of the history policies, where the weights are again related to the step sizes α_t^j. The construction of $\bar{\pi}$ is formally defined in Algorithm 9, which is closely related to the "certified policies" from [34]. Specifically, Algorithm 9 takes the policy trajectory $\{\pi_h^t\}_{h\in[H],t\in[T]}$ of Algorithm 8 as input. For each step $h\in[H]$, Algorithm 9 randomly samples a joint policy from the policy trajectory using the sampling probabilities α_t^j and let all the agents play this joint policy at

[3] It is known that such a distribution π^t exists and is computationally efficient.

Algorithm 9: Policy $\bar{\pi}_h^t$

Input: Policy trajectory $\{\pi_h^t\}_{h\in[H],t\in[T]}$ of Algorithm 8;
for *step* $h' \leftarrow h$ *to* H **do**
    Sample $\tau \in [t]$ with probability $\mathbb{P}(\tau = j) = \alpha_t^j$;
    Play policy $\pi_{h'}^\tau$ at step h';
    Set $t \leftarrow \tau$.

the given step. The constructed policy $\bar{\pi}$ is a correlated policy because the agents implicitly use a common source of randomness to select the same history iteration. We will show that the output policy constitutes an approximate CE.

Analysis. In the following, we present the analysis of Algorithm 8. We use the following notion of CE-Gap to measure the distance of a correlated policy to a CE:

$$\text{CE-Gap}(\pi) := \max_{i\in\mathcal{N}} \max_{\phi_i\in\Phi_i} \left(V_{1,i}^{\phi_i\diamond\pi}(s_1) - V_{1,i}^{\pi}(s_1)\right),$$

where recall that Φ_i is the set of strategy modifications for agent i. The following theorem states that Algorithm 8 finds an $\widetilde{O}(T^{-1})$-approximate CE in T iterations.

Theorem 5. *If Algorithm 8 is run on an N-player episodic Markov game for T iterations with a learning rate $\eta = \frac{1}{256NH\sqrt{HA_{\max}}}$, the output policy $\bar{\pi}$ satisfies:*

$$\text{CE-Gap}(\bar{\pi}) \le \frac{2048NH^{\frac{7}{2}}A_{\max}^{\frac{5}{2}}\log T}{T}.$$

Theorem 5 improves the existing $\widetilde{O}(T^{-1/4})$ rate [123] of no-regret learning to CE in full-information Markov games. The parameter dependences in Theorem 5 also match the best known rate for normal-form games [120], except that Theorem 5 introduces an additional $O(H^{\frac{7}{2}})$ dependence on the Markov game episode length. We remark that we make no effort to improve the constant factors in the bounds, which can certainly be tightened.

The proof structure of Theorem 5 is conceptually similar to those for learning Nash equilibria in two-player zero-sum Markov games [121], [122] . We first introduce a few notations to facilitate the proof. For any $(s,h) \in \mathcal{S}\times[H]$, we define the per-state weighted swap regret up to iteration $t \in [T]$ in the corresponding matrix game as

$$\text{SwapReg}_{h,i}^t(s) := \max_{\phi_{h,i}^s:\mathcal{A}_i\to\mathcal{A}_i} \sum_{j=1}^t \alpha_t^j \left\langle \phi_{h,i}^s \diamond \pi_{h,i}^j(s,\cdot) - \pi_{h,i}^j(s,\cdot), [Q_{h,i}^j\pi_{h,-i}^j](s,\cdot)\right\rangle,$$
$$\text{SwapReg}_h^t := \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \text{SwapReg}_{h,i}^t(s).$$

For any $(h,t) \in [H]\times[T]$, we further define the best response CE value gap as

$$\delta_h^t := \max_{i\in\mathcal{N}} \max_{\phi_i} \max_{s\in\mathcal{S}} \left(V_{h,i}^{\phi_i\diamond\bar{\pi}_h^t}(s) - V_{h,i}^{\bar{\pi}_h^t}(s)\right),$$

where $\bar{\pi}_h^t$ is defined in Algorithm 9 and we slightly abuse the notation ϕ_i to denote a strategy modification that is only effective starting from step h. By the definition of δ_h^t and $\bar{\pi}$, one can easily see that $\text{CE-Gap}(\bar{\pi}) = \text{CE-Gap}(\bar{\pi}_1^T) \le \delta_1^T$. To control δ_1^T, we first use the following lemma to establish the recursive relationship of

the best response CE value gaps between two consecutive steps h and $h+1$:

Lemma 3. *(Recursion of best response CE value gaps) For any fixed $(h,t) \in [H] \times [T]$, we have*

$$\delta_h^t \le \sum_{j=1}^{t} \alpha_t^j \delta_{h+1}^j + \mathrm{SwapReg}_h^t \,. \tag{2.7}$$

Therefore, upper bounding CE-Gap($\bar{\pi}$) breaks down to controlling the per-state weighted swap regrets for every $(s,h) \in \mathcal{S} \times [H]$. We can further establish the upper bound of $\mathrm{SwapReg}_{h,i}^t(s)$ in the next lemma. The proof of this lemma relies on an RVU bound for the swap regret of BM-OFTRL under time-varying learning rates in normal-form games.

Lemma 4. *(Per-state weighted swap regret bounds) For any $t \in [T], h \in [H], s \in \mathcal{S}$ and $i \in \mathcal{N}$, Algorithm 8 ensures that*

$$\mathrm{SwapReg}_{h,i}^t(s) \le \frac{4A_i^2 H \log t}{\eta t} + \frac{32\eta H^3 N^2}{t} + 8\eta N H^2 \sum_{j=2}^{t} \sum_{k \neq i} \alpha_t^j \left\| \pi_{h,k}^j(s,\cdot) - \pi_{h,k}^{j-1}(s,\cdot) \right\|_1^2 . \tag{2.8}$$

If $\eta \le \frac{1}{256NH\sqrt{HA_{\max}}}$, we further have

$$\begin{aligned} \sum_{i=1}^{N} \mathrm{SwapReg}_{h,i}^t(s) \le & \frac{4NA_{\max}^2 H \log t}{\eta t} + \frac{32\eta N H^2 (N^2+H)}{t} \\ & - \frac{1}{2048\eta H} \sum_{i=1}^{N} \sum_{j=2}^{t} \frac{\alpha_t^j}{A_i} \left\| \pi_{h,i}^j(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot) \right\|_1^2 . \end{aligned} \tag{2.9}$$

We note that there is a discrepancy between (2.7) and (2.9). Specifically, (2.7) requires an upper bound for the *maximum* of the swap regrets over the agents while (2.9) controls the *sum* of them. This poses some additional challenges for learning NE (in zero-sum Markov games) or CCE in existing works [121], [122], because some players may experience negative regret [124] and the sum of regrets in general does not upper bound the maximum individual regret of the players. For CE, however, we can take advantage of a seemingly trivial property that the swap regret is always non-negative. This is in sharp contrast to the (external) regret and one can easily verify this property by letting all the strategy modifications $\phi_{h,i}^s$ in $\mathrm{SwapReg}_{h,i}^t(s)$ be identity mappings. In this case, the discrepancy will not impede us, as we can easily upper bound the maximum (2.7) by the sum (2.9), which already yields an $\widetilde{O}(t^{-1})$ convergence rate. Our proof of Theorem 5 instead follows a different route that upper bounds the second-order path lengths of the learning dynamics, which leads to an improved rate in terms of the dependence on N.

2.8.2 Convergence to Coarse Correlated Equilibria

In this subsection, we turn to learning coarse correlated equilibria. We start by introducing our stage-based OFTRL algorithm followed by presenting its analysis.

Algorithm 10 describes the stage-based OFTRL procedure run by agent $i \in \mathcal{N}$ for learning CCE. Similar to Section 2.8.1, Algorithm 10 also consists of three components: policy update, value update, and policy output. The policy update step is standard OFTRL with a negative entropy regularizer, which is also known as the optimistic Hedge (see e.g., [117]). Our policy output step, formally described in Algorithm 11, is also conceptually similar to Algorithm 9 for CE.

Algorithm 10: Stage-based OFTRL for coarse correlated equilibria (agent i)

Initialize: $Q^1_{h,i}(s,\boldsymbol{a}) \leftarrow 0, \pi^0_{h,i}(s,a_i) \leftarrow 1/A_i, \forall s \in \mathcal{S}, h \in [H], a_i, a_i' \in \mathcal{A}_i, \boldsymbol{a} \in \mathcal{A}_{\text{all}}$;

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

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

  Policy update: For all $s \in \mathcal{S}, h \in [H]$, and $a_i \in \mathcal{A}_i$,

$$\ell^t_{h,i}(s,a_i) \leftarrow \sum_{t'=t_\tau^{\text{start}}}^{t-1} [Q^\tau_{h,i}\pi^{t'}_{h,-i}](s,a_i) + [Q^\tau_{h,i}\pi^{t-1}_{h,-i}](s,a_i);$$

$$\pi^t_{h,i}(s,\cdot) \leftarrow \operatorname*{argmax}_{\boldsymbol{x}\in\Delta(\mathcal{A}_i)} \left(\langle \boldsymbol{x}, \eta_\tau \ell^t_{h,i}(s,\cdot)/H\rangle - \mathcal{R}(\boldsymbol{x})\right);$$

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

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

    Value update: For each $h \in [H], s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}_{\text{all}}, i \in \mathcal{N}$:

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

    $\tau \leftarrow \tau + 1$; $\pi^t_{h,i}(s,a_i) \leftarrow 1/A_i, \forall s \in \mathcal{S}, h \in [H], a_i \in \mathcal{A}_i$;

Output policy: Sample $t \sim \text{Unif}([T])$. Output $\bar{\pi} := \bar{\pi}^t_1$ where $\bar{\pi}^t_h$ is defined in Algorithm 11.

Algorithm 11: Policy $\bar{\pi}^t_h$ for stage-based OFTRL

Input: Policy trajectory $\{\pi^t_h\}_{h\in[H],t\in[T]}$ of Algorithm 10;

for *step* $h' \leftarrow h$ *to* H **do**

  Uniformly sample j from $\{t^{\text{start}}_{\tau(t)-1}, t^{\text{start}}_{\tau(t)-1}+1, \ldots, t^{\text{end}}_{\tau(t)-1}\}$;

  Play policy $\pi^j_{h'}$ for step h';

  Set $t \leftarrow j$.

The value update step here is substantially different from that of Section 2.8.1. Rather than performing incremental updates as in Algorithm 8, we instead employ *stage-based* value updates by dividing the total T iterations into multiple stages and only updating the value estimates at the end of a stage. We use $\tau \in \mathbb{N}_+$ to index the stages and use L_τ to denote the length (i.e., number of iterations) of the τ-th stage. We set the lengths of the stages to grow exponentially at a rate of $(1+1/H)$ so that $L_{\tau+1} = \lfloor (1+1/H)L_\tau \rfloor$. 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. Such a mechanism was initially proposed in single-agent RL [61] and has later been advocated for creating a piece-wise stationary environment in MARL [17]. The benefit of using stage-based value updates here is that we only need to bound the per-state *average* regret in the corresponding matrix games (in contrast to the weighted regret as in Section 2.8.1), which allows us to easily apply existing regret analysis results for normal-form games.

Analysis. We use the following notion of CCE-Gap to measure the distance of a correlated policy to a CCE:

$$\text{CCE-Gap}(\pi) := \max_{i\in\mathcal{N}} \left(V^{\dagger,\pi_{-i}}_{1,i}(s_1) - V^\pi_{1,i}(s_1)\right).$$

In the following theorem, we show that Algorithm 10 finds an $\widetilde{O}(T^{-1})$-approximate CCE in T iterations.

Theorem 6. *If Algorithm 10 is run on an N-player episodic Markov game for T iterations with a learning*

Table 2.5: Reward matrices for Player 1.

s_0	b_0	b_1
a_0	0.8	0.2
a_1	0.0	1.0

s_1	b_0	b_1
a_0	1.0	0.2
a_1	0.5	0.8

Table 2.6: Reward matrices for Player 2.

s_0	b_0	b_1
a_0	0.2	1.0
a_1	0.5	0.0

s_1	b_0	b_1
a_0	0.5	1.0
a_1	1.0	0.2

rate $\eta_\tau = \Theta(\frac{1}{N \log^4 L_\tau})$ *in each stage* τ*, then the output policy* $\bar{\pi}$ *satisfies:*

$$\text{CCE-Gap}(\bar{\pi}) = O\left(\frac{NH^3 \log A_{\max} \cdot \log^5 T}{T}\right).$$

Theorem 6 improves the best-known rate of $\widetilde{O}(T^{-3/4})$ [121] for OFTRL in general-sum Markov games. Since CCE reduces to NE in two-player zero-sum games [34], Theorem 6 additionally suggests that a simple variant of Algorithm 10 leads to an $\widetilde{O}(T^{-1})$ convergence to NE in two-player zero-sum Markov games, which can further improve the existing $O(H^5 \log A_{\max}/T)$ result [122] when $\log T = O(H^{2/5})$. Compared to its counterpart $O(N \log A_{\max} \cdot \log^4 T/T)$ in normal-form games [118], Theorem 6 incurs an extra $O(\log T)$ factor due to the stage-based value estimates.

The proof of Theorem 6 starts by showing a recursive relationship of the best response CCE value gaps between two consecutive steps h and $h+1$. As a consequence of stage-based value updates, CCE-gap($\bar{\pi}$) breaks down to the sum of the per-state average regret over the stages, which allows us to apply each player's individual (average) regret bound in NFGs [118] for each stage. The proof is then completed by upper bounding the total number of stages. We defer the complete proof of Theorem 6 to Section 2.14 for clarity of presentation.

2.8.3 Numerical Results

In this subsection, we numerically evaluate Algorithm 8 (denoted by "Smooth OFTRL CE") and Algorithm 10 ("Stage-based OFTRL CCE") to validate our $\widetilde{O}(T^{-1})$ theoretical convergence guarantees. Our simulations additionally consider an OFTRL algorithm with incremental value updates similar to that of Algorithm 8 for learning CCE ("Smooth OFTRL CCE"). We did not prove the convergence of such an algorithm but would be interested to see its numerical performance given its intuitive form. Our numerical studies are conducted on a simple general-sum Markov game with 2 players, 2 states $\mathcal{S} = \{s_0, s_1\}$ and $H = 2$ steps per episode. Each player has 2 candidate actions $\mathcal{A} = \{a_0, a_1\}$ and $\mathcal{B} = \{b_0, b_1\}$, respectively. The reward matrices for Player 1 and Player 2 at the two states are given in Tables 2.5 and 2.6, respectively. 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 0.8, and transitions to the other state with probability 0.2. 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 0.2, and will transition to the other state with probability 0.8. We choose a constant learning rate $\eta = 0.2$ for all the three algorithms. We have also experimented with other choices of the transition and reward functions and have observed similar behavior, as shown in Figures 2.2 and 2.3.

Figure 2.2 illustrates the convergence of the three algorithms to their corresponding equilibrium solutions as the number of iterations increases. To clearly demonstrate their convergence rates, we further plot the behavior of CCE/CE-Gap($\bar{\pi}$) $\times T$ as T increases. We can observe from Figure 2.3 that for all three algorithms, CCE/CE-Gap($\bar{\pi}$) $\times T$ essentially become a constant for any reasonably large value of T. This indicates

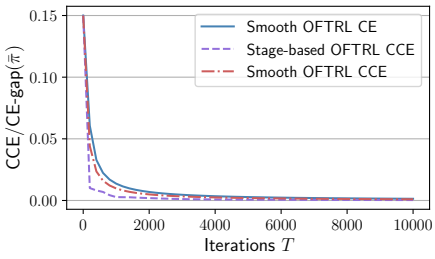


Figure 2.2: Convergence of CCE/CE-Gap($\bar{\pi}$)

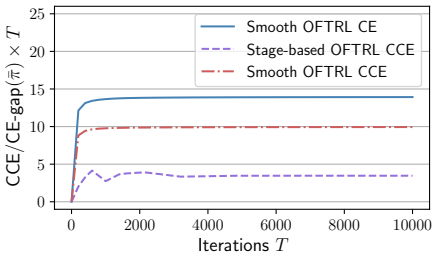


Figure 2.3: Convergence of CCE/CE-Gap($\bar{\pi}$) $\times T$

that our algorithms indeed converge at a rate of $\widetilde{O}(T^{-1})$ numerically. We also observe that OFTRL with stage-based value updates numerically converges faster than its incrementally-updated counterpart despite using the same learning rate, which advocates the use of stage-based value updates in Markov games.

2.9 Proofs for Section 2.5

2.9.1 Proof of Lemma 2

Proof. In the following, we provide a proof for the first inequality. The second inequality can be shown using a similar argument.

Notice that it suffices to show $V_h^k(s) \geq V_{k,h}^{\star,\bar{\nu}}(s)$, because $\overline{V}_h^k(s) = \min\{V_h^k(s), H-h+1\}$, and $V_{k,h}^{\star,\bar{\nu}}(s) \leq H-h+1$ always holds. Our proof relies on backward induction on $h \in [H]$. First, the claim holds for $h = H+1$ by the definition of $V_{H+1}^k(s)$. Now, suppose $V_{h+1}^k(s) \geq V_{k,h+1}^{\star,\bar{\nu}}(s)$ for all $s \in \mathcal{S}$. By the definition of $V_{k,h}^{\star,\bar{\nu}}(s)$ and the induction hypothesis,

$$\begin{aligned} V_{k,h}^{\star,\bar{\nu}}(s) &\leq \max_{\mu_h} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{k^i,h+1}^{\star,\bar{\nu}}\right)(s) \\ &\leq \max_{\mu_h} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i}\right)(s). \end{aligned} \tag{2.10}$$

Further, define

$$R_t = \max_{\mu_h} \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i}\right)(s) - \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i}\right)(s). \tag{2.11}$$

One may observe that from the perspective of player 1, R_t is the weighted sum of the differences between the actual value that player 1 collected for the first t times that state s is visited, and the value that could have been achieved using the best fixed policy in hindsight. R_t can hence be thought of as the weighted regret of an adversarial bandit problem, which we formally present and analyze in Section 2.6. Specifically, the loss function of the bandit problem is defined as

$$l_i(a) = \mathbb{E}_{b \sim \nu_h^{k^i}(s)} \left\{ H - h + 1 - r_h(s,a,b) - \mathbb{P}_h \overline{V}_{h+1}^{k^i}(s,a,b) \right\}.$$

The weight of the regret at round i is $w_i = \alpha_t^i$. If we define

$$\mu_h^\star := \arg\min_{\mu_h} \sum_{i=1}^t w_i \langle \mu_h, l_i \rangle = \arg\max_{\mu_h} \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i} \right)(s),$$

then, R_t can be equivalently rewritten as

$$R_t = \sum_{i=1}^t w_i \left\langle \mu_h^\star - \mu_h^{k^i}, l_i \right\rangle.$$

Later in Section 2.6, we will analyze an adversarial bandit problem in exactly the same form. Applying the regret bound (which is presented in Theorem 2 of Section 2.6) of this bandit problem, we obtain the following result with probability at least $1 - p/(2SHK)$:

$$\begin{aligned} R_t \leq & 2H\alpha_t^t \sqrt{At\iota} + \frac{3H\sqrt{A\iota}}{2} \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} + \frac{1}{2} H \alpha_t^t \iota + H \sqrt{2\iota \sum_{i=1}^t \left(\alpha_t^i\right)^2} \\ \leq & 4H^2 \sqrt{A\iota/t} + 3H\sqrt{A\iota/t} + H^2\iota/t + \sqrt{4H^3\iota/t} \\ \leq & 10H^2 \sqrt{A\iota/t}, \end{aligned} \tag{2.12}$$

where in the first step we have used the fact that w_i is increasing and $\max_{i \leq t} w_i = \alpha_t^t$, and the second step is due to Lemma 1.

Finally, let $\mathcal{F}_i$ be the σ-algebra generated by all the random variables before episode k^i. Then, we can see that $\{r_h(s, a_h^{k^i}, b_h^{k^i}) + \overline{V}_{h+1}^{k^i}(s_{h+1}^{k^i})\}_{i=1}^t$ is a martingale with respect to $\{\mathcal{F}_i\}_{i=1}^t$. From the Azuma-Hoeffding inequality, it holds with probability at least $1 - p/(2SHK)$ that

$$\begin{aligned} & \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i} \right)(s) \\ \leq & \sum_{i=1}^t \alpha_t^i \left[r_h \left(s, a_h^{k^i}, b_h^{k^i} \right) + \overline{V}_{h+1}^{k^i} \left(s_{h+1}^{k^i} \right) \right] + 2\sqrt{H^3\iota/t}, \end{aligned} \tag{2.13}$$

where ι suppresses logarithmic terms. Finally, combining the results in (2.10), (2.11), (2.12), (2.13), and applying a union bound, we obtain that

$$\begin{aligned} V_{k,h}^{\star,\bar{\nu}}(s) \leq & \max_{\mu_h} \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu_h \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i} \right)(s) \\ \leq & \sum_{i=1}^t \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \overline{V}_{h+1}^{k^i} \right)(s) + 10H^2\sqrt{A\iota/t} \\ \leq & \sum_{i=1}^t \alpha_t^i \left[r_h \left(s, a_h^{k^i}, b_h^{k^i} \right) + \overline{V}_{h+1}^{k^i} \left(s_{h+1}^{k^i} \right) \right] + 10H^2\sqrt{A\iota/t} + 2\sqrt{H^3\iota/t} \\ \leq & \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i \left[r_h \left(s, a_h^{k^i}, b_h^{k^i} \right) + \overline{V}_{h+1}^{k^i} \left(s_{h+1}^{k^i} \right) + \beta_i \right] \\ = & V_h^k(s), \end{aligned}$$

where the second to last step is by the definition of $\beta_t = c\sqrt{\frac{H^4 A\iota}{t}}$ for some large constant c, and Lemma 1. In the last step we used the formulation of $V_h^k(s)$ in (2.3). This completes the proof of the induction step. □

2.9.2 Proof of Theorem 1

Proof. We provide a proof for the first bound. The second one can be shown using a similar argument. For analytical purposes, we introduce two new notations $\underline{V}$ and $\underset{\sim}{V}$ that serve as lower confidence bounds of the value estimates for agent 1. Specifically, for any $(s,h,k) \in \mathcal{S} \times [H+1] \times [K]$, we define $\underline{V}_h^k(s) = \underset{\sim}{V}_h^k(s) = 0$ if $h = H+1$ or the (h,s) pair has not been visited before episode k, and otherwise define

$$\underset{\sim}{V}_h^k(s) = \sum_{i=1}^{t} \alpha_t^i \left[r_h\left(s, a_h^{k^i}, b_h^{k^i}\right) + \underline{V}_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right) - \beta_i \right], \text{ and } \underline{V}_h^k(s) = \max\{\underset{\sim}{V}_h^k(s), 0\}.$$

Notice that these two notations are only introduced for ease of analysis, and the agent does not need to explicitly maintain such values during the learning process. In the following, we show that $\underline{V}_h^k(s) \le V_{k,h}^{\bar{\mu},\bar{\nu}}(s)$, for all $(s,h,k) \in \mathcal{S} \times [H] \times [K]$. Again, it suffices to show that $\underset{\sim}{V}_h^k(s) \le V_{k,h}^{\bar{\mu},\bar{\nu}}(s)$, because $\underline{V}_h^k(s) = \max\{\underset{\sim}{V}_h^k(s), 0\}$, and $V_{k,h}^{\bar{\mu},\bar{\nu}}(s) \ge 0$ always holds. Our proof relies on backward induction on $h \in [H]$. The claim trivially holds for $h = H+1$. Suppose $\underset{\sim}{V}_{h+1}^k(s) \le V_{k,h+1}^{\bar{\mu},\bar{\nu}}(s)$ for all $s \in \mathcal{S}$. By the definition of $\underset{\sim}{V}_h^k(s)$,

$$\begin{aligned}
\underset{\sim}{V}_h^k(s) =& \sum_{i=1}^{t} \alpha_t^i \left[r_h\left(s, a_h^{k^i}, b_h^{k^i}\right) + \underline{V}_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right) - \beta_i \right] \\
\le& \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h \underline{V}_{h+1}^{k^i} \right)(s) \\
\le& \sum_{i=1}^{t} \alpha_t^i \mathbb{D}_{\mu_h^{k^i} \times \nu_h^{k^i}} \left(r_h + \mathbb{P}_h V_{k^i,h+1}^{\bar{\mu},\bar{\nu}} \right)(s) \\
=& V_{k,h}^{\bar{\mu},\bar{\nu}}(s).
\end{aligned}$$

where the second step uses the Azuma-Hoeffding inequality and the definition of β_i, and the third step is by the induction hypothesis. This completes the proof of the induction.

Together with Lemma 2, we know that

$$\sum_{k=1}^{K} \left(V_{k,1}^{\star,\bar{\nu}}(s_1) - V_{k,1}^{\bar{\mu},\bar{\nu}}(s_1) \right) \le \sum_{k=1}^{K} \left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \right),$$

and so we only need to find an upper bound for the RHS. Define $\delta_h^k := \overline{V}_h^k(s_h^k) - \underline{V}_h^k(s_h^k)$. The main idea of the proof is similar to optimistic Q-learning in the single-agent setting [60]: We seek to upper bound $\sum_{k=1}^K \delta_h^k$ by the next step $\sum_{k=1}^K \delta_{h+1}^k$, and then obtain a recursive formula.

By the definitions of $\overline{V}_h^k(s_h^k)$ and $\underline{V}_h^k(s_h^k)$, we know that

$$\begin{aligned}
\delta_h^k =& \overline{V}_h^k(s_h^k) - \underline{V}_h^k(s_h^k) \\
\leq& \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \left[\overline{V}_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right) - \underline{V}_{h+1}^{k^i}\left(s_{h+1}^{k^i}\right) + 2\beta_i \right] \\
=& \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \delta_{h+1}^{k^i} + 2\sum_{i=1}^{t} \alpha_t^i \beta_i \\
\leq& \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \delta_{h+1}^{k^i} + c\sqrt{AH^4\iota/t},
\end{aligned}$$

for some constant c, and the last step is due to Lemma 1. Summing over k, notice that

$$\sum_{k=1}^{K} \alpha_{n_h^k}^0 H = \sum_{k=1}^{K} H \mathbb{1}\left\{n_h^k = 0\right\} \leq HS,$$

because there are at most SH pairs of (s,h) to be visited. Further,

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k_h^i(s_h^k)} \leq& \sum_{k'=1}^{K} \delta_{h+1}^{k'} \sum_{i=n_h^{k'}+1}^{\infty} \alpha_i^{n_h^{k'}} \\
\leq& \left(1 + \frac{1}{H}\right) \sum_{k=1}^{K} \delta_{h+1}^k,
\end{aligned}$$

where the first step is by switching the order of summation, and the second uses the fact that $\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H}$ for every $i \geq 1$ from Lemma 1. Therefore,

$$\begin{aligned}
\sum_{k=1}^{K} \delta_h^k \leq& \sum_{k=1}^{K} \alpha_{n_h^k}^0 H + \sum_{k=1}^{K}\sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i \delta_{h+1}^{k_h^i(s_h^k)} + \sum_{k=1}^{K} c\sqrt{AH^4\iota/n_h^k} \\
\leq& HS + \left(1 + \frac{1}{H}\right) \sum_{k=1}^{K} \delta_{h+1}^k + \sum_{k=1}^{K} c\sqrt{AH^4\iota/n_h^k}. \qquad (2.14)
\end{aligned}$$

Applying this formula recursively for $h = H, H-1, \ldots, 1$ yields

$$\sum_{k=1}^{K} \delta_1^k \leq eSH^2 + ec\sum_{h=1}^{H}\sum_{k=1}^{K} \sqrt{AH^4\iota/n_h^k},$$

where we used the fact that $(1 + \frac{1}{H})^H \leq e$. Finally, for any $h \in [H]$,

$$\sum_{k=1}^{K} \sqrt{AH^4\iota/n_h^k} = \sum_{s\in\mathcal{S}} \sum_{n=1}^{N_h^K(s)} \sqrt{AH^4\iota/n} \leq O(\sqrt{H^4SAK\iota}),$$

where the last step holds because $\sum_{s\in\mathcal{S}} N_h^K(s) = K$, and the LHS is maximized when $N_h^K(s) = K/S$ for all

$s \in \mathcal{S}$. Summarizing the results above leads to the desired bound

$$\sum_{k=1}^{K} \left(V_{k,1}^{\star,\bar{\nu}}(s_1) - V_{k,1}^{\bar{\mu},\bar{\nu}}(s_1) \right) \leq \sum_{k=1}^{K} \delta_1^k \leq O(\sqrt{H^6 SAK\iota}).$$

□

2.10 Proofs for Section 2.6

In this section, we present some lemmas that were used in the proof of Theorem 2. We first recall the following two properties of the Bregman divergence that will be useful in our analysis.

Lemma 5. *(Pythagorean theorem for Bregman divergence, Lemma 4.1 in [125]). Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set, $y \in \mathbb{R}^n$, and $z = \arg\min_{u \in \mathcal{X}} D_F(u, y)$. Then, for any $x \in \mathcal{X}$,*

$$D_F(x, y) - D_F(z, y) \geq D_F(x, z).$$

Lemma 6. *(Convexity). Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the $(n-1)$-dimensional simplex, and let F be the unnormalized negentropy regularizer. For any $x, y \in \mathcal{X}$, the mapping $D_F(x, \cdot)$ is convex on $\mathcal{X}$.*

We start with the following technical result given in [34], which was in turn adapted from Lemma 1 in [106]. This lemma allows us to construct high probability regret bounds for Algorithm 4, rather than only regret bounds in expectation.

Lemma 7. *(Lemma 18 in [34]) For any sequence of coefficients $c_1, c_2, \ldots, c_t$ s.t. $c_i \in [0, 2\gamma_i]^A$ is $\mathcal{F}_i$ -measurable, we have with probability at least $1 - p/AT$,*

$$\sum_{i=1}^{t} w_i \left\langle c_i, \hat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \iota.$$

Lemma 8. *Suppose $\beta_i \in (0, 1], \forall i \in [t]$. For any fixed policy $\theta \in \Delta(\mathcal{A})$ and for any time step $t \in [T]$, the weighted regret of Algorithm 4 with respect to θ can be bounded by:*

$$\sum_{i=1}^{t} w_i \left\langle \theta_i - \theta, \hat{l}_i \right\rangle \leq \frac{w_{t+1} D_F(\theta, \theta_1)}{\eta_{t+1}} + \sum_{i=1}^{t} \frac{w_i D_F(\theta_i, \tilde{\theta}_{i+1})}{\eta_i}.$$

Proof. Since $\tilde{\theta}_{i+1} = \arg\min_{\theta \in \mathcal{D}} \left\{ \eta_i \left\langle \theta, \hat{l}_i \right\rangle + D_F(\theta, \theta_i) \right\}$, the first-order optimality condition implies that

$$\eta_i \hat{l}_i + \nabla F(\tilde{\theta}_{i+1}) - \nabla F(\theta_i) = 0.$$

Reordering and using the definition of the Bregman divergence,

$$\begin{aligned} \left\langle \theta_i - \theta, \hat{l}_i \right\rangle =& \frac{1}{\eta_i} \left\langle \theta_i - \theta, \nabla F(\theta_i) - \nabla F(\theta_{i+1}) \right\rangle \\ =& \frac{1}{\eta_i} (D_F(\theta, \theta_i) - D_F(\theta, \tilde{\theta}_{i+1}) + D_F(\theta_i, \tilde{\theta}_{i+1})). \end{aligned} \tag{2.15}$$

By the Pythagorean theorem for Bregman divergence (Lemma 5),

$$\begin{aligned}
&\beta_i(D_F(\theta,\tilde{\theta}_{i+1}) - D_F(\theta'_{i+1},\tilde{\theta}_{i+1})) + (1-\beta_i)D_F(\theta,\theta_1)\\
\geq&\beta_i D_F(\theta,\theta'_{i+1}) + (1-\beta_i)D(\theta,\theta_1)\\
\geq&D_F(\theta,\theta_{i+1}),
\end{aligned}$$

where the second step is by the convexity of $D_F(\theta,\cdot)$ (Lemma 6) and the fact that $\theta_{i+1} = \beta_t\theta'_{i+1} + (1-\beta_i)\theta_1$. Rearranging the terms yields

$$D_F(\theta,\tilde{\theta}_{i+1}) \geq \frac{1}{\beta_i}D_F(\theta,\theta_{i+1}) - \frac{1-\beta_i}{\beta_i}D_F(\theta,\theta_1) + D_F(\theta'_{i+1},\tilde{\theta}_{i+1}).$$

Plugging this into (2.15) and recalling the definition that $\beta_i = \eta_{i+1}/\eta_i$, we obtain

$$\begin{aligned}
&w_i\left\langle\theta_i-\theta,\hat{l}_i\right\rangle = \frac{w_i}{\eta_i}(D_F(\theta,\theta_i) - D_F(\theta,\tilde{\theta}_{i+1}) + D_F(\theta_i,\tilde{\theta}_{i+1}))\\
\leq&\frac{w_i}{\eta_i}(D_F(\theta,\theta_i) - \frac{1}{\beta_i}D_F(\theta,\theta_{i+1}) + \frac{1-\beta_i}{\beta_i}D_F(\theta,\theta_1) - D_F(\theta'_{i+1},\tilde{\theta}_{i+1}) + D_F(\theta_i,\tilde{\theta}_{i+1}))\\
=&\frac{w_iD_F(\theta,\theta_i)}{\eta_i} - \frac{w_{i+1}D_F(\theta,\theta_{i+1})}{\eta_{i+1}} + \left(\frac{w_{i+1}}{\eta_{i+1}} - \frac{w_i}{\eta_i}\right)D_F(\theta,\theta_1)\\
&\qquad - \frac{w_iD_F(\theta'_{i+1},\tilde{\theta}_{i+1})}{\eta_i} + \frac{w_iD_F(\theta_i,\tilde{\theta}_{i+1})}{\eta_i}.
\end{aligned}$$

Summing over i and telescoping leads to

$$\begin{aligned}
&\sum_{i=1}^t w_i\left\langle\theta_i-\theta,\hat{l}_i\right\rangle\\
\leq&\frac{w_1D_F(\theta,\theta_1)}{\eta_1} + \sum_{i=1}^t\left(\frac{w_{i+1}}{\eta_{i+1}} - \frac{w_i}{\eta_i}\right)D_F(\theta,\theta_1) + \sum_{i=1}^t\frac{w_i(D_F(\theta_i,\tilde{\theta}_{i+1}) - D_F(\theta'_{i+1},\tilde{\theta}_{i+1}))}{\eta_i}\\
=&\frac{w_{t+1}D_F(\theta,\theta_1)}{\eta_{t+1}} + \sum_{i=1}^t\frac{w_iD_F(\theta_i,\tilde{\theta}_{i+1})}{\eta_i},
\end{aligned}$$

where in the last step we used the fact that $D_F(\theta'_{i+1},\tilde{\theta}_{i+1}) \geq 0$ (by the convexity of F). □

Lemma 9. *If $\eta_i \leq 2\gamma_i$ and $0 \leq w_i \leq 1$ for all $i \leq t$, it holds with probability at least $1-p$ that*

$$\sum_{i=1}^t w_i\left\langle\theta_i-\theta^\star,\hat{l}_i\right\rangle \leq \frac{w_{t+1}\log A}{\eta_{t+1}} + \frac{A}{2}\sum_{i=1}^t\eta_iw_i + \frac{1}{2}\max_{i\leq t}w_i\iota.$$

Proof. Our proof relies on the following regret bound of OMD given in Lemma 8: For any $\theta \in \Delta(\mathcal{A})$ and any $t \in [T]$,

$$\sum_{i=1}^t w_i\left\langle\theta_i-\theta,\hat{l}_i\right\rangle \leq \frac{w_{t+1}D_F(\theta,\theta_1)}{\eta_{t+1}} + \sum_{i=1}^t\frac{w_iD_F(\theta_i,\tilde{\theta}_{i+1})}{\eta_i}. \tag{2.16}$$

Since $\tilde{\theta}_{i+1} = \arg\min_{\theta\in\mathcal{D}}\left\{\eta_i\left\langle\theta,\hat{l}_i\right\rangle + D_F(\theta,\theta_i)\right\}$, the minimum is achieved when $\eta_i\hat{l}_i + \nabla F(\tilde{\theta}_{i+1}) -$

$\nabla F(\theta_i) = 0$. Direct calculation shows that $\tilde{\theta}_{i+1}(a) = \theta_i(a)\exp(-\eta_i \hat{l}_i(a))$ for all $a \in \mathcal{A}$. Hence,

$$\begin{aligned} D_F(\theta_i, \tilde{\theta}_{i+1}) =& \sum_{a=1}^{A} \theta_i(a) \log\left(\frac{\theta_i(a)}{\tilde{\theta}_{i+1}(a)}\right) - \sum_{a=1}^{A} \theta_i(a) + \sum_{a=1}^{A} \tilde{\theta}_{i+1}(a) \\ =& \sum_{a=1}^{A} \theta_i(a)\left(\eta_i \hat{l}_i(a) - 1 + \exp(-\eta_i \hat{l}_i(a))\right) \\ \leq& \frac{\eta_i^2}{2} \sum_{a=1}^{A} \theta_i(a) \hat{l}_i(a)^2, \end{aligned}$$

where the last step holds because $\exp(x) \leq 1 + x + x^2/2$ for $x \leq 0$. Plugging this back to Equation (2.16), we have that

$$\begin{aligned} \sum_{i=1}^{t} w_i \left\langle \theta_i - \theta, \hat{l}_i \right\rangle \leq& \frac{w_{t+1} D_F(\theta, \theta_1)}{\eta_{t+1}} + \frac{1}{2} \sum_{i=1}^{t} \sum_{a=1}^{A} \eta_i w_i \theta_i(a) \hat{l}_i(a)^2 \\ \leq& \frac{w_{t+1} D_F(\theta, \theta_1)}{\eta_{t+1}} + \frac{1}{2} \sum_{i=1}^{t} \sum_{a=1}^{A} \eta_i w_i \hat{l}_i(a) \qquad (2.17) \\ \leq& \frac{w_{t+1} D_F(\theta, \theta_1)}{\eta_{t+1}} + \frac{1}{2} \sum_{i=1}^{t} \sum_{a=1}^{A} \eta_i w_i l_i(a) + \frac{1}{2} \max_{i \leq t} w_i \iota \qquad (2.18) \\ \leq& \frac{w_{t+1} \log A}{\eta_{t+1}} + \frac{A}{2} \sum_{i=1}^{t} \eta_i w_i + \frac{1}{2} \max_{i \leq t} w_i \iota, \end{aligned}$$

where (2.17) holds because $\hat{l}_i(a) \neq 0$ only if $\mathbb{1}\{a_i = a\} = 1$, and hence it follows that $\sum_{a=1}^{A} \theta_i(a) \hat{l}_i(a)^2 = \theta_i(a_i) \hat{l}_i(a_i)^2 = \theta_i(a_i) \frac{\tilde{l}_i(a_i)}{\theta_i(a_i) + \gamma_i} \hat{l}_i(a_i) \leq \hat{l}_i(a_i) = \sum_{a=1}^{A} \hat{l}_i(a)$. Step (2.18) is by applying Lemma 7, with $c_i(a) = \eta_i$ for all $1 \leq a \leq A$. The last step holds because $D_F(\theta, \theta_1) \leq \log A$ for $\theta_1 = \mathbf{1}/A$ and any $\theta \in \Delta(\mathcal{A})$. □

Lemma 10. *(Lemma 20 in [34]) With probability at least $1 - p$, for any $t \in [T]$,*

$$\sum_{i=1}^{t} w_i \left\langle \theta_i, l_i - \hat{l}_i \right\rangle \leq A \sum_{i=1}^{t} \gamma_i w_i + \sqrt{2\iota \sum_{i=1}^{t} w_i^2}.$$

Lemma 11. *(Lemma 21 in [34]) With probability at least $1 - p$, for any $t \in [T]$ and any $\theta^\star \in \Delta(\mathcal{A})$, if γ_i is non-increasing in i, then*

$$\sum_{i=1}^{t} w_i \left\langle \theta^*, \hat{l}_i - l_i \right\rangle \leq \max_{i \leq t} w_i \iota / \gamma_t.$$

2.11 Proofs for Section 2.7.1

We first introduce a few notations to facilitate the analysis. For a step $h \in [H]$ of an episode $k \in [K]$, we denote by s_h^k the state that the agents observe at this time step. For any state $s \in \mathcal{S}$, we let $\mu_{h,i}^k(\cdot \mid s) \in \Delta(\mathcal{A}_i)$ be the distribution prescribed by Algorithm 5 to agent i at this step. Notice that such notations are well-defined for every $s \in \mathcal{S}$, even if s might not be the state s_h^k that is actually visited at the given step. We further let $\mu_{h,i}^k = \{\mu_{h,i}^k(\cdot \mid s) : s \in \mathcal{S}\}$, and let $a_{h,i}^k \in \mathcal{A}_i$ be the actual action taken by agent i. For any $s \in \mathcal{S}$, let $N_h^k(s)$ and $\check{N}_h^k(s)$ denote, respectively, the values of $N_h(s)$ and $\check{N}_h(s)$ at the *beginning* of the k-th episode.

Note that it is proper to use the same notation to denote these values from all the agents' perspectives, because the agents maintain the same estimates of these terms as they can be calculated from the common observations (of the state-visitation). We also use $\overline{V}^k_{h,i}(s)$ and $\tilde{V}^k_{h,i}(s)$ to denote the values of $\overline{V}_{h,i}(s)$ and $\tilde{V}_{h,i}(s)$, respectively, at the beginning of the k-th episode from agent i's perspective.

Further, for a state s^k_h, let $\check{n}^k_h$ denote the number of times that state s^k_h has been visited (at the h-th step) in the stage right before the current stage, and let $\check{l}^k_{h,j}$ denote the index of the episode that this state was visited the j-th time among the $\check{n}^k_h$ times. For notational convenience, we use $\check{n}$ to denote $\check{n}^k_h$, and $\check{l}_j$ to denote $\check{l}^k_{h,j}$, whenever h and k are clear from the context. With the new notations, the update rule in Line 13 of Algorithm 5 can be equivalently expressed as

$$\tilde{V}_{h,i}(s_h) \leftarrow \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s_h, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}^{\check{l}_j}_{h+1,i}(s_{h+1}^{\check{l}_j}) \right) + b_{\check{n}}. \tag{2.19}$$

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

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

Lemma 12. *With probability at least $1 - \frac{p}{2}$, it holds for all $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ that*

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

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

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

Therefore, we only need to bound

$$R^\star_{\check{n}} := \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i} \times \mu^{\check{l}_j}_{h,-i}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{\check{l}_j}_{h+1,i} \right)(s) - \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}^{\check{l}_j}_{h+1,i} \right)(s). \tag{2.20}$$

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

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

Algorithm 5 applies the Exp3-IX algorithm [106], which ensures that with probability at least $1 - \frac{p}{4NHS}$, it holds for all $k \in [K]$ that

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

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

Remark 2. *We would like to discuss the alternative of using V-learning with the celebrated learning rate $\alpha_t = \frac{H+1}{H+t}$ [60] to update $\overline{V}_h$ instead of employing stage-based updates. This is the case for several recent works also under the V-learning formulation for MARL [12], [16], [34], [63]. Such a learning rate induces an update rule as follows:*

$$\overline{V}_{h,i}(s_h) \leftarrow (1-\alpha_t)\,\overline{V}_{h,i}(s_h) + \alpha_t\left(r_{h,i}(s_h, \boldsymbol{a}_h) + \overline{V}_{h+1,i}(s_{h+1}) + \beta_t\right), \tag{2.21}$$

where t is the number of times that s_h has been visited, and β_t is some bonus term. In this way, $\overline{V}_{h,i}(s_h)$ is updated every time the state s_h is visited. With such a learning rate, the update rule (2.21) of $\overline{V}_{h,i}$ can be equivalently expressed as

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

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

$$\alpha_t^0 = \prod_{j=1}^{t}(1-\alpha_j), \quad \text{and} \quad \alpha_t^j = \alpha_j \prod_{k=j+1}^{t}(1-\alpha_k), \forall 1 \leq j \leq t.$$

Compared with stage-based updates (2.20), we now need to upper bound a regret term of the following form:

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

Notice that the above definition of regret induces a adversarial bandit problem with a time-varying weighted regret, where the loss at time j is assigned a weight α_t^j. As t varies, the weight α_t^j assigned to the same step j also changes over time. These weights also cannot be pre-computed, because it relies on knowing the total number of times that a certain state s_h is visited during the entire horizon, which is impossible before seeing the output of the algorithm. To address such an additional challenge, [34] proposed a Follow-the-Regularized-Leader (FTRL) algorithm that simultaneously achieves with a changing step size, a weighted regret, and a high-probability guarantee, which inevitably leads to a more delicate analysis. In contrast, we have shown in (2.20) that our stage-based update rule leads to an adversarial bandit problem with a simple averaged regret. In our approach, it suffices to plug in any existing adversarial bandit solution with a high-probability regret bound, such as the Exp3-IX method that we used in Algorithm 5. Therefore, our stage-based update significantly simplifies both the algorithmic design and the analysis of V-learning in MARL.

Based on the trajectory of the distributions $\{\mu_{h,i}^k : i \in \mathcal{N}, h \in [H], k \in [K]\}$ specified by Algorithm 5, we construct a correlated policy $\bar{\pi}_h^k$ for each $(h, k) \in [H] \times [K]$. Our construction of the correlated policies,

Algorithm 12: Construction of the Correlated Policy $\bar{\pi}_h^k$

Input: The distribution trajectory $\{\mu_{h,i}^k : i \in \mathcal{N}, h \in [H], k \in [K]\}$ specified by Algorithm 5.
Initialize: $k' \leftarrow k$.
for *step* $h' \leftarrow h$ *to* H **do**
    Receive $s_{h'}$;
    Take joint action $\boldsymbol{a}_{h'} \sim \times_{i=1}^N \mu_{h',i}^{k'}(\cdot \mid s_{h'})$;
    Uniformly sample j from $\{1, 2, \ldots, \check{N}_{h'}^{k'}(s_{h'})\}$;
    Set $k' \leftarrow \check{l}_{h',j}^{k'}$, where $\check{l}_{h',j}^{k'}$ is the index of the episode such that state $s_{h'}$ was visited the j-th time (among the total $\check{N}_{h'}^{k'}(s_{h'})$ times) in the last stage;

largely inspired by the "certified policies" [34] for learning in two-player zero-sum games, is formally presented in Algorithm 12. We further define an output policy $\bar{\pi}$ that first uniformly samples an index k from $[K]$, and then proceed with $\bar{\pi}_1^k$. A more formal description of $\bar{\pi}$ has been given in Algorithm 6. By construction of the correlated policies $\bar{\pi}_h^k$, we know that for any $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H+1] \times [K]$, the corresponding value function can be written recursively as follows:

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

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

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

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

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

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

Lemma 13. *It holds with probability at least* $1-p$ *that for all* $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$,

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

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

We first prove the first inequality. It suffices to show that $\tilde{V}_{h,i}^k(s) \geq V_{h,i}^{\star, \bar{\pi}_{h,-i}^k}(s)$ because $\overline{V}_{h,i}^k(s) =$

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

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

$$\tilde{V}_{h,i}^k(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + b_{\check{n}}. \tag{2.22}$$

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

$$\begin{aligned}
V_{h,i}^{\star,\bar{\pi}_{h,-i}^k}(s) \leq & \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i} \times \mu_{h,-i}^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\star,\bar{\pi}_{h+1,-i}^{\check{l}_j}} \right)(s) \\
\leq & \max_{\mu_{h,i}} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\mu_{h,i} \times \mu_{h,-i}^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) \\
\leq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + 6\sqrt{H^2 A_i \iota / \check{n}} \\
\leq & \tilde{V}_{h,i}^k(s),
\end{aligned} \tag{2.23}$$

where the second step is by the induction hypothesis, the third step holds due to Lemma 12, and the last step is by the definition of $b_{\check{n}}$.

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

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

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

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

$$\underset{\sim}{V}_{h,i}^k(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) - b_{\check{n}}. \tag{2.24}$$

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

$$\begin{aligned}
V_{h,i}^{\bar{\pi}_h^k}(s) =& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}}\left(r_{h,i}+\mathbb{P}_h V_{h+1,i}^{\bar{\pi}_{h+1}^{\check{l}_j}}\right)(s) \\
\geq& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}}\left(r_{h,i}+\mathbb{P}_h \underset{\sim}{V}_{h+1,i}^{\check{l}_j}\right)(s) \\
\geq& \frac{1}{\check{n}}\sum_{j=1}^{\check{n}} \left(r_{h,i}(s,\boldsymbol{a}_h^{\check{l}_j})+\underset{\sim}{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right) - \sqrt{H^2\iota/\check{n}} \\
\geq& \underset{\sim}{V}_{h,i}^k(s), \qquad (2.25)
\end{aligned}$$

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

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

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

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

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

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

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

From Lemma 13, we know that

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

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

following notation:

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

The main idea of the subsequent proof is to upper bound $\sum_{k=1}^K \delta_h^k$ by the next step $\sum_{k=1}^K \delta_{h+1}^k$, and then obtain a recursive formula. From the update rule of $\overline{V}_{h,i}^k(s_h^k)$ in (2.19), we know that

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

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

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

$$\begin{aligned}
\delta_h^k \le & \mathbb{I}[n_h^k = 0]H + \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(\overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) - \underline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + 2b_{\check{n}} \\
\le & \mathbb{I}[n_h^k = 0]H + \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \delta_{h+1}^{\check{l}_j} + 2b_{\check{n}},
\end{aligned} \tag{2.26}$$

To find an upper bound of $\sum_{k=1}^K \delta_h^k$, we proceed to upper bound each term on the RHS of (2.26) separately. First, notice that $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right] \le SH$, because each fixed state-step pair (s,h) contributes at most 1 to $\sum_{k=1}^K \mathbb{I}\left[n_h^k = 0\right]$. Next, we turn to analyze the second term on the RHS of (2.26). Observe that

$$\begin{aligned}
\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} = & \sum_{k=1}^K \sum_{m=1}^K \frac{1}{\check{n}_h^k} \delta_{h+1}^m \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \\
= & \sum_{m=1}^K \delta_{h+1}^m \sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right].
\end{aligned} \tag{2.27}$$

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

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,j}^k = m\right] \le 1 + \frac{1}{H}. \tag{2.28}$$

Combining (2.27) and (2.28) leads to the following upper bound of the second term in (2.26):

$$\sum_{k=1}^K \frac{1}{\check{n}_h^k} \sum_{j=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,j}^k} \le (1 + \frac{1}{H}) \sum_{k=1}^K \delta_{h+1}^k. \tag{2.29}$$

So far, we have obtained the following upper bound:

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

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

$$\sum_{k=1}^{K}\delta_1^k \le O\left(SH^3 + \sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1} b_{\check{n}_h^k}\right), \tag{2.30}$$

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

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

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

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

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

$$\begin{aligned}
\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1} b_{\check{n}_h^k} =& O\left(\sqrt{H^4 A_i \iota}\sum_{s\in\mathcal{S}}\sum_{j\ge 1}(1+\frac{1}{H})^{h-1} w(s,j) e_j^{-\frac{1}{2}}\right) \\
\le& O\left(\sqrt{SA_i K H^5 \iota}\right).
\end{aligned} \tag{2.31}$$

Summarizing the results above leads to

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

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

the desired result:

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

This completes the proof of the theorem. □

An immediate corollary is that we obtain an ε-approximate CCE when $\sqrt{SA_{\max}H^5\iota/K} \le \varepsilon$, which is Theorem 3 in Section 2.7.

Theorem 3. (Sample complexity of learning CCE). For any $p \in (0,1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 5 for K episodes with $K = O(SA_{\max}H^5\iota/\varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ constitutes an ε-approximate coarse correlated equilibrium.

2.12 Proofs for Section 2.7.2

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

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

The swap regret, instead, measures the difference between the cumulative reward of an algorithm and the cumulative reward that could be achieved by swapping multiple pairs of actions of the algorithm. To be more specific, we define a strategy modification $F : \mathcal{A} \to \mathcal{A}$ to be a mapping from the action space to itself. For any action selection distribution p, we let $F \diamond p$ be the swapped distribution that takes action $a \in \mathcal{A}$ with probability $\sum_{a'\in\mathcal{A},F(a')=a} p(a')$. The swap regret[4] is then defined as

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

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

We follow the generic reduction introduced in [65], and convert a Follow-the-Regularized-Leader algorithm with sublinear external regret to a no-swap-regret algorithm [12]. The resulting algorithm is presented as Algorithm 13. The following lemma shows that Algorithm 13 is indeed a no-swap-regret learning algorithm.

Lemma 14. *[12, Theorem 26]. For any $T \in \mathbb{N}$ and $p \in (0,1)$, let $\iota = \log(A^2/p)$. With probability at least $1-3p$, it holds that*

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

[4] This is a modified version of the swap regret used in [65], which is defined as $R_{\text{swap}}(T) = \max_{F:\mathcal{A}\to\mathcal{A}}\sum_{t=1}^{T}\left(l_t(a_t) - l_t(F(a_t))\right)$.

Algorithm 13: No-swap-regret learning

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

It is worth noting that [12] presented a more general analysis with an anytime weighted swap regret guarantee. Such complication can be avoided in our algorithm, as our stage-based learning approach only entails a simple averaged swap regret analysis.

The complete Stage-Based V-Learning algorithm for CE is presented in Algorithm 7. In the following analysis, we follow the same notations as have been used in the CCE analysis. We again start with the following lemma that justifies our choice of the bonus term.

Lemma 15. *With probability at least* $1 - \frac{p}{2}$*, it holds for all* $(i, s, h, k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$ *that*

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

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

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

Therefore, we only need to bound

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

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

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

Such a problem can be addressed by a no-swap-regret learning algorithm as presented in Algorithm 13. Applying Lemma 14, we obtain that with probability at least $1 - \frac{p}{4NHS}$, it holds for all $k \in [K]$ that

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

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

We again define the notations $\bar{\pi}_h^k, \bar{\pi}, V_{h,i}^{\bar{\pi}_h^k}, \underline{V}_{h,i}^k$, and $\underset{\sim}{V}_{h,i}^k(s)$ in the same sense as in Section 2.11. The next lemma shows that $\overline{V}_{h,i}^k(s)$ and $\underline{V}_{h,i}^k(s)$ are valid upper and lower bounds.

Lemma 16. *It holds with probability at least $1-p$ that for all $(i,s,h,k) \in \mathcal{N} \times \mathcal{S} \times [H] \times [K]$,*

$$\overline{V}_{h,i}^k(s) \geq \max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s), \text{ and } \underline{V}_{h,i}^k(s) \leq V_{h,i}^{\bar{\pi}_h^k}(s).$$

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

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

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

$$\tilde{V}_{h,i}^k(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + b_{\check{n}}. \tag{2.32}$$

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

$$\begin{aligned}
\max_{\psi_i \in \Psi_i} V_{h,i}^{\psi_i \diamond \bar{\pi}_h^k}(s) \leq & \max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi_i \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \max_{\psi_i' \in \Psi_i} V_{h+1,i}^{\psi_i' \diamond \bar{\pi}_{h+1}^{\check{l}_j}} \right)(s) \\
\leq & \max_{\psi_i \in \Psi_i} \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\psi_i \diamond \boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \overline{V}_{h+1,i}^{\check{l}_j} \right)(s) \\
\leq & \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j}) \right) + 11\sqrt{H^2 A_i^2 \iota / \check{n}} \\
\leq & \tilde{V}_{h,i}^k(s),
\end{aligned} \tag{2.33}$$

where the second step is by the induction hypothesis, the third step holds due to Lemma 15, and the last step is by the definition of $b_{\check{n}}$.

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

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

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

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

$$\underset{\sim}{V}^k_{h,i}(s) = \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \underset{\sim}{V}^{\check{l}_j}_{h+1,i}(s^{\check{l}_j}_{h+1}) \right) - b_{\check{n}}. \tag{2.34}$$

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

$$\begin{aligned}
V^{\bar{\pi}^k_h}_{h,i}(s) =& \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h V^{\bar{\pi}^{\check{l}_j}_{h+1}}_{h+1,i} \right)(s) \\
\ge& \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \mathbb{D}_{\boldsymbol{\mu}_h^{\check{l}_j}} \left(r_{h,i} + \mathbb{P}_h \underset{\sim}{V}^{\check{l}_j}_{h+1,i} \right)(s) \\
\ge& \frac{1}{\check{n}} \sum_{j=1}^{\check{n}} \left(r_{h,i}(s, \boldsymbol{a}_h^{\check{l}_j}) + \underset{\sim}{V}^{\check{l}_j}_{h+1,i}(s^{\check{l}_j}_{h+1}) \right) - \sqrt{H^2\iota/\check{n}} \\
\ge& \underset{\sim}{V}^k_{h,i}(s),
\end{aligned} \tag{2.35}$$

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

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

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

Theorem 8. *For any $p \in (0, 1]$, let $\iota = \log(2NSA_{\max}KH/p)$. Suppose $K \ge \frac{SH}{A^2_{\max}\iota}$. With probability at least $1 - p$,*

$$\max_{\psi_i \in \Psi_i} V^{\psi_i \diamond \bar{\pi}}_{1,i}(s_1) - V^{\bar{\pi}}_{1,i}(s_1) \le O\left(\sqrt{H^5 S A^2_{\max} \iota / K} \right),$$

Proof. The proof follows a similar procedure as the proof of Theorem 7. From Lemma 16, we know that

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

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

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

The main idea of the subsequent proof is to upper bound $\sum_{k=1}^{K}\delta_h^k$ by the next step $\sum_{k=1}^{K}\delta_{h+1}^k$, and then obtain a recursive formula. From the update rule of $\overline{V}_{h,i}^{k}(s_h^k)$ in (2.19), we know that

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

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

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

$$\begin{aligned}\delta_h^k \le&\mathbb{I}[n_h^k=0]H+\frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\left(\overline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})-\underline{V}_{h+1,i}^{\check{l}_j}(s_{h+1}^{\check{l}_j})\right)+2b_{\check{n}}\\ \le&\mathbb{I}[n_h^k=0]H+\frac{1}{\check{n}}\sum_{j=1}^{\check{n}}\delta_{h+1}^{\check{l}_j}+2b_{\check{n}},\end{aligned}\tag{2.36}$$

To find an upper bound of $\sum_{k=1}^{K}\delta_h^k$, we proceed to upper bound each term on the RHS of (2.36) separately. First, notice that $\sum_{k=1}^{K}\mathbb{I}\left[n_h^k=0\right]\le SH$, because each fixed state-step pair (s,h) contributes at most 1 to $\sum_{k=1}^{K}\mathbb{I}\left[n_h^k=0\right]$. Next, we turn to analyze the second term on the RHS of (2.36). Observe that

$$\begin{aligned}\sum_{k=1}^{K}\frac{1}{\check{n}_h^k}\sum_{j=1}^{\check{n}_h^k}\delta_{h+1}^{\check{l}_{h,j}^k}=&\sum_{k=1}^{K}\sum_{m=1}^{K}\frac{1}{\check{n}_h^k}\delta_{h+1}^{m}\sum_{j=1}^{\check{n}_h^k}\mathbb{1}\left[\check{l}_{h,j}^k=m\right]\\ =&\sum_{m=1}^{K}\delta_{h+1}^{m}\sum_{k=1}^{K}\frac{1}{\check{n}_h^k}\sum_{j=1}^{\check{n}_h^k}\mathbb{1}\left[\check{l}_{h,j}^k=m\right].\end{aligned}\tag{2.37}$$

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

$$\sum_{k=1}^{K}\frac{1}{\check{n}_h^k}\sum_{j=1}^{\check{n}_h^k}\mathbb{1}\left[\check{l}_{h,j}^k=m\right]\le 1+\frac{1}{H}.\tag{2.38}$$

Combining (2.37) and (2.38) leads to the following upper bound of the second term in (2.36):

$$\sum_{k=1}^{K}\frac{1}{\check{n}_h^k}\sum_{j=1}^{\check{n}_h^k}\delta_{h+1}^{\check{l}_{h,j}^k}\le(1+\frac{1}{H})\sum_{k=1}^{K}\delta_{h+1}^k.\tag{2.39}$$

So far, we have obtained the following upper bound:

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

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

$$\sum_{k=1}^{K}\delta_1^k \le O\left(SH^3 + \sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1} b_{\check{n}_h^k}\right), \tag{2.40}$$

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

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

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

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

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

$$\begin{aligned}
\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1} b_{\check{n}_h^k} =& O\left(\sqrt{H^4A_i^2\iota}\sum_{s\in\mathcal{S}}\sum_{j\ge 1}(1+\frac{1}{H})^{h-1} w(s,j) e_j^{-\frac{1}{2}}\right) \\
\le& O\left(\sqrt{SA_i^2KH^5\iota}\right).
\end{aligned} \tag{2.41}$$

Summarizing the results above leads to

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

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

the desired result:

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

This completes the proof of the theorem. □

An immediate corollary is that we obtain an ε-approximate CE when $\sqrt{SA_{\max}^2 H^5 \iota / K} \le \varepsilon$, which is Theorem 4 in Section 2.7.

Theorem 4. (Sample complexity of learning CE). For any $p \in (0,1]$, set $\iota = \log(2NSA_{\max}KH/p)$, and let the agents run Algorithm 7 for K episodes with $K = O(SA_{\max}^2 H^5 \iota / \varepsilon^2)$. Then, with probability at least $1-p$, the output policy $\bar{\pi}$ constitutes an ε-approximate correlated equilibrium.

2.13 Proofs for Section 2.8.1

Lemma 17. *(Extension of Theorem 4.3 in [120] to time-varying learning rates) In a no-regret learning problem as defined in Section 2.8, suppose that BM-OFTRL (2.5) is run with log-barrier regularization and a time-varying learning rate* $\eta_t \le \frac{1}{128\sqrt{|\mathcal{A}|}}, \forall t \in [T]$. *Then, for any* $T \ge 2$, *the swap regret is bounded by*

$$\mathrm{SwapReg}^T \le \frac{2|\mathcal{A}|^2 \log T}{\eta_T} + 4\sum_{t=1}^{T} \eta_t \left\|\boldsymbol{u}^t - \boldsymbol{u}^{t-1}\right\|_\infty^2 - \frac{1}{2048|\mathcal{A}|} \sum_{t=1}^{T-1} \frac{1}{\eta_t} \left\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\right\|_1^2.$$

Proof sketch. The proof follows a similar procedure as that of Theorem 4.3 in [120], except that we need to re-derive their Theorems B.1 and 3.1 under a time-varying learning rate. We skip the proof here as such an extension is straightforward. □

Lemma 3. (Recursion of best response CE value gaps) For any fixed $(h,t) \in [H] \times [T]$, we have

$$\delta_h^t \le \sum_{j=1}^{t} \alpha_t^j \delta_{h+1}^j + \mathrm{SwapReg}_h^t.$$

Proof. For any fixed $i \in \mathcal{N}$ and $s \in \mathcal{S}$, we know from the definition of $\bar{\pi}_h^t$ from Algorithm 9 that

$$V_{h,i}^{\bar{\pi}_h^t}(s) = \sum_{j=1}^{t} \alpha_t^j \left\langle \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}]\right) \pi_{h,-i}^j\right](s,\cdot)\right\rangle. \tag{2.42}$$

For a fixed $\bar{\pi}_h^t$, we use $\phi_i^\star$ to denote the best response strategy modification that maximizes the value function starting from step h. In this case, we know from the definition of the value function that

$$\begin{aligned}
V_{h,i}^{\phi_i^\star \diamond \bar{\pi}_h^t}(s) &= \max_{\phi_{h,i}^s : \mathcal{A}_i \to \mathcal{A}_i} \sum_{j=1}^{t} \alpha_t^j \left\langle \phi_{h,i}^s \diamond \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j}]\right) \pi_{h,-i}^j\right](s,\cdot)\right\rangle \\
&= \max_{\phi_{h,i}^s} \sum_{j=1}^{t} \alpha_t^j \left\langle \phi_{h,i}^s \diamond \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] + [P_h V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j}] - [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}]\right) \pi_{h,-i}^j\right](s,\cdot)\right\rangle \\
&\le \max_{\phi_{h,i}^s} \sum_{j=1}^{t} \alpha_t^j \left\langle \phi_{h,i}^s \diamond \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}]\right) \pi_{h,-i}^j\right](s,\cdot)\right\rangle + \sum_{j=1}^{t} \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j} - V_{h+1,i}^{\bar{\pi}_{h+1}^j}\right)(s').
\end{aligned}$$

Subtracting (2.42) from the above equation leads to:

$$V_{h,i}^{\phi_i^\star \diamond \bar{\pi}_h^t}(s) - V_{h,i}^{\bar{\pi}_h^t}(s) \leq \sum_{j=1}^{t} \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j}(s') - V_{h+1,i}^{\bar{\pi}_{h+1}^j}(s') \right) + \max_{\phi_{h,i}^s} \sum_{j=1}^{t} \alpha_t^j \left\langle \phi_{h,i}^s \diamond \pi_{h,i}^j(s,\cdot) - \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] \right) \pi_{h,-i}^j \right](s,\cdot) \right\rangle. \tag{2.43}$$

In the following, we will show that (2.43) is equal to $\mathrm{SwapReg}_{h,i}^t(s)$. It suffices to show that $Q_{h,i}^t(s,\boldsymbol{a}) = \left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^t}]\right)(s,\boldsymbol{a}), \forall t \in [T], \boldsymbol{a} \in \mathcal{A}_{\mathrm{all}}$. We prove this claim by backward induction over $h \in [H]$. Notice that the claim trivially holds for $h = H$ as $Q_{H,i}^t(s,\boldsymbol{a}) = r_{H,i}(s,\boldsymbol{a}), \forall t \in [T], \boldsymbol{a} \in \mathcal{A}_{\mathrm{all}}$. Suppose that the claim holds for h; then, for step $h-1$, we have that

$$\begin{aligned}
Q_{h-1,i}^t(s,\boldsymbol{a}) =& \sum_{j=1}^{t} \alpha_t^j \left(r_{h-1,i} + P_{h-1}[Q_{h,i}^j \pi_h^j] \right)(s,\boldsymbol{a}) \\
=& r_{h-1,i}(s,\boldsymbol{a}) + P_{h-1}\left[\sum_{j=1}^{t} \alpha_t^j Q_{h,i}^j \pi_h^j \right](s,\boldsymbol{a}) \\
=& r_{h-1,i}(s,\boldsymbol{a}) + P_{h-1}\left[\sum_{j=1}^{t} \alpha_t^j \left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] \right) \pi_h^j \right](s,\boldsymbol{a}) \\
=& r_{h-1,i}(s,\boldsymbol{a}) + \left[P_{h-1} V_{h,i}^{\bar{\pi}_h^t} \right](s,\boldsymbol{a}),
\end{aligned}$$

where the first step is by (2.6), the second step changes the order of summation, the third step uses the induction hypothesis, and the last step is due to (2.42). This completes the proof of $Q_{h,i}^t(s,\boldsymbol{a}) = \left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^t}]\right)(s,\boldsymbol{a})$. Substituting it back to (2.43), we obtain that

$$V_{h,i}^{\phi_i^\star \diamond \bar{\pi}_h^t}(s) - V_{h,i}^{\bar{\pi}_h^t}(s) \leq \sum_{j=1}^{t} \alpha_t^j \max_{s' \in \mathcal{S}} \left(V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j}(s') - V_{h+1,i}^{\bar{\pi}_{h+1}^j}(s') \right) + \mathrm{SwapReg}_{h,i}^t(s).$$

Since the above inequality holds for any $i \in \mathcal{N}$ and $s \in \mathcal{S}$, and since $V_{h+1,i}^{\phi_i^\star \diamond \bar{\pi}_{h+1}^j}(s') \leq \max_{\phi_i} V_{h+1,i}^{\phi_i \diamond \bar{\pi}_{h+1}^j}(s')$ at step $h+1$, we can conclude that

$$\delta_h^t \leq \sum_{j=1}^{t} \alpha_t^j \delta_{h+1}^j + \mathrm{SwapReg}_h^t,$$

This completes the proof of the recursive relationship of best response CE value gaps. □

Lemma 4. (Per-state weighted swap regret bounds) For any $t \in [T], h \in [H], s \in \mathcal{S}$ and $i \in \mathcal{N}$, Algorithm 8 ensures that

$$\begin{aligned}
\mathrm{SwapReg}_{h,i}^t(s) \leq & \frac{4A_i^2 H \log t}{\eta t} + \frac{32\eta H^3 N^2}{t} + 8\eta N H^2 \sum_{j=2}^{t} \sum_{k \in \mathcal{N}, k \neq i} \alpha_t^j \left\| \pi_{h,k}^j(s,\cdot) - \pi_{h,k}^{j-1}(s,\cdot) \right\|_1^2 \\
& - \frac{1}{2048\eta A_i} \sum_{j=2}^{t} \alpha_t^{j-1} \left\| \pi_{h,i}^j(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot) \right\|_1^2.
\end{aligned}$$

Consequently, if $\eta \le \frac{1}{256NH\sqrt{HA_{\max}}}$, we further have

$$\sum_{i=1}^{N} \mathrm{SwapReg}_{h,i}^{t}(s) \le \frac{4NA_{\max}^2 H \log t}{\eta t} + \frac{32\eta N H^2(N^2+H)}{t} - \frac{1}{2048\eta H} \sum_{i=1}^{N} \sum_{j=2}^{t} \frac{\alpha_t^j}{A_i} \left\| \pi_{h,i}^{j}(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot) \right\|_1^2.$$

Proof. At each fixed $(s,h) \in \mathcal{S} \times [H]$, the agents essentially face a no-swap-regret learning problem in a matrix game, where the payoff matrix of agent i is $Q_{h,i}^{t}(s,\cdot)$ at iteration t. We can apply the weighted swap regret bound (Lemma 17) of OFTRL under the Blum-Mansour reduction in normal-form games to obtain:

$$\begin{aligned}
\mathrm{SwapReg}_{h,i}^{t}(s) =& \max_{\phi_{h,i}^{s}:\mathcal{A}_i \to \mathcal{A}_i} \sum_{j=1}^{t} \alpha_t^j \left\langle \phi_{h,i}^{s} \diamond \pi_{h,i}^{j}(s,\cdot) - \pi_{h,i}^{j}(s,\cdot), [Q_{h,i}^{j}\pi_{h,-i}^{j}](s,\cdot) \right\rangle \\
=& \alpha_t^1 \max_{\phi_{h,i}^{s}:\mathcal{A}_i \to \mathcal{A}_i} \sum_{j=1}^{t} \left\langle \phi_{h,i}^{s} \diamond \pi_{h,i}^{j}(s,\cdot) - \pi_{h,i}^{j}(s,\cdot), w_j [Q_{h,i}^{j}\pi_{h,-i}^{j}](s,\cdot) \right\rangle \qquad (2.44) \\
\le& \frac{2A_i^2 \alpha_t \log t}{\eta} + 4 \sum_{j=1}^{t} \frac{\eta \alpha_t^1}{w_j} \left\| w_j [Q_{h,i}^{j}\pi_{h,-i}^{j}](s,\cdot) - w_j [Q_{h,i}^{j-1}\pi_{h,-i}^{j-1}](s,\cdot) \right\|_\infty^2 \\
& - \frac{\alpha_t^1}{2048\eta A_i} \sum_{j=2}^{t} w_{j-1} \left\| \pi_{h,i}^{j}(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot) \right\|_1^2, \qquad (2.45)
\end{aligned}$$

where (2.44) is due to the choice of the weights $w_j = \alpha_t^j / \alpha_t^1$. (2.45) uses Lemma 17, by instantiating $\boldsymbol{u}^j(\cdot)$ in Lemma 17 as $w_j [Q_{h,i}^{j}\pi_{h,-i}^{j}](s,\cdot)$, the prediction $\boldsymbol{m}^t = w_j [Q_{h,i}^{j-1}\pi_{h,-i}^{j-1}](s,\cdot)$, and the learning rate $\eta_j = \eta / w_j$. To further upper bound the above equation, notice that

$$\begin{aligned}
& \sum_{j=1}^{t} \frac{\eta \alpha_t^1}{w_j} \left\| w_j [Q_{h,i}^{j}\pi_{h,-i}^{j}](s,\cdot) - w_j [Q_{h,i}^{j-1}\pi_{h,-i}^{j-1}](s,\cdot) \right\|_\infty^2 \\
=& \sum_{j=1}^{t} \eta \alpha_t^1 w_j \left\| \left([Q_{h,i}^{j}\pi_{h,-i}^{j}] - [Q_{h,i}^{j-1}\pi_{h,-i}^{j}] + [Q_{h,i}^{j-1}\pi_{h,-i}^{j}] - [Q_{h,i}^{j-1}\pi_{h,-i}^{j-1}] \right)(s,\cdot) \right\|_\infty^2 \\
\le& 2 \sum_{j=1}^{t} \eta \alpha_t^1 w_j \left(\left\| Q_{h,i}^{j}(s,\cdot) - Q_{h,i}^{j-1}(s,\cdot) \right\|_\infty^2 + H^2 \left\| \pi_{h,-i}^{j}(s,\cdot) - \pi_{h,-i}^{j-1}(s,\cdot) \right\|_1^2 \right) \\
\le& 2 \sum_{j=1}^{t} \eta \alpha_t^1 w_j (\alpha_j)^2 H^2 + 2 \sum_{j=1}^{t} \eta \alpha_t^1 w_j H^2 \left\| \pi_{h,-i}^{j}(s,\cdot) - \pi_{h,-i}^{j-1}(s,\cdot) \right\|_1^2, \qquad (2.46)
\end{aligned}$$

where the second step uses the observation that $(a+b)^2 \le 2a^2 + 2b^2$, the Hölder's inequality, and the fact that $\|Q_{h,i}^{j-1}\|_\infty \le H$. The third step is due to our value update rule in Algorithm 8, which yields

$$\begin{aligned}
\left\| Q_{h,i}^{j}(s,\cdot) - Q_{h,i}^{j-1}(s,\cdot) \right\|_\infty &= \left\| -\alpha_j Q_{h,i}^{j-1}(s,\cdot) + \alpha_j \left(r_{h,i} + P_h [Q_{h+1,i}^{j}\pi_{h+1}^{j}] \right)(s,\cdot) \right\|_\infty \\
&\le \alpha_j \max \left\{ \left\| Q_{h,i}^{j-1}(s,\cdot) \right\|_\infty, \left\| \left(r_{h,i} + P_h [Q_{h+1,i}^{j}\pi_{h+1}^{j}] \right)(s,\cdot) \right\|_\infty \right\} \\
&\le \alpha_j H.
\end{aligned}$$

To continue from (2.46), we apply the properties that $w_j = \alpha_t^j / \alpha_t^1$ and $\sum_{j=1}^{t} \alpha_t^j (\alpha_j)^2 \le \sum_{j=1}^{t} (\alpha_j)^2 / t \le$

$(H+2)/t \le 3H/t$ (see Lemma 6 in [122] for a proof) to obtain:

$$\begin{aligned}
(2.46) =& 2\sum_{j=1}^{t} \eta\alpha_t^1 w_j (\alpha_j)^2 H^2 + 2\sum_{j=1}^{t} \eta\alpha_t^1 w_j H^2 \left\|\pi_{h,-i}^j(s,\cdot) - \pi_{h,-i}^{j-1}(s,\cdot)\right\|_1^2 \\
\le& \frac{6\eta H^3}{t} + 2\sum_{j=1}^{t} \eta\alpha_t^1 w_j H^2 \left\|\pi_{h,-i}^j(s,\cdot) - \pi_{h,-i}^{j-1}(s,\cdot)\right\|_1^2 \\
\le& \frac{6\eta H^3}{t} + 2\eta(N-1)H^2 \sum_{j=1}^{t} \alpha_t^j \sum_{k\in\mathcal{N},k\neq i} \left\|\pi_{h,k}^j(s,\cdot) - \pi_{h,k}^{j-1}(s,\cdot)\right\|_1^2. \qquad (2.47)
\end{aligned}$$

In the last step, we used that the total variation between two product distributions is bounded by the sum of the total variations of each marginal distribution (see e.g. [126]):

$$\begin{aligned}
\left\|\pi_{h,-i}^j(s,\cdot) - \pi_{h,-i}^{j-1}(s,\cdot)\right\|_1^2 =& \left(\sum_{\boldsymbol{a}_{-i}\in\mathcal{A}_{-i}} \left|\pi_{h,-i}^j(s,\boldsymbol{a}_{-i}) - \pi_{h,-i}^{j-1}(s,\boldsymbol{a}_{-i})\right|\right)^2 \\
=& \left(\sum_{\boldsymbol{a}_{-i}\in\mathcal{A}_{-i}} \left|\prod_{k\neq i}\pi_{h,k}^j(s,a_k) - \prod_{k\neq i}\pi_{h,k}^{j-1}(s,a_k)\right|\right)^2 \\
\le& \left(\sum_{k\neq i} \left\|\pi_{h,k}^j(s,a_k) - \pi_{h,k}^{j-1}(s,a_k)\right\|_1\right)^2 \\
\le& (N-1)\sum_{k\neq i} \left\|\pi_{h,k}^j(s,a_k) - \pi_{h,k}^{j-1}(s,a_k)\right\|_1^2,
\end{aligned}$$

and the last step is by the Cauchy–Schwarz inequality. Substituting (2.47) back to (2.45) leads to

$$\begin{aligned}
\mathrm{SwapReg}_{h,i}^t(s) \le& \frac{2A_i^2\alpha_t\log t}{\eta} + 8\eta(N-1)H^2\sum_{j=1}^{t}\alpha_t^j \sum_{k\in\mathcal{N},k\neq i}\left\|\pi_{h,k}^j(s,\cdot) - \pi_{h,k}^{j-1}(s,\cdot)\right\|_1^2 \\
&+ \frac{24\eta H^3}{t} - \frac{\alpha_t^1}{2048\eta A_i}\sum_{j=2}^{t} w_{j-1}\left\|\pi_{h,i}^j(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2 \\
\le& \frac{4A_i^2 H\log t}{\eta t} + 8\eta(N-1)H^2\sum_{j=2}^{t}\alpha_t^j \sum_{k\in\mathcal{N},k\neq i}\left\|\pi_{h,k}^j(s,\cdot) - \pi_{h,k}^{j-1}(s,\cdot)\right\|_1^2 \\
&+ \frac{32\eta H^2(H+N^2)}{t} - \frac{1}{2048\eta A_i}\sum_{j=2}^{t}\alpha_t^{j-1}\left\|\pi_{h,i}^j(s,\cdot) - \pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2, \qquad (2.48)
\end{aligned}$$

where the second inequality uses $\alpha_t = (H+1)/(H+t) \le 2H/t$. This step also takes out the term for $j=1$ and upper bounds it by

$$8\eta(N-1)H^2\alpha_t^1 \sum_{k\in\mathcal{N},k\neq i}\left\|\pi_{h,k}^1(s,\cdot) - \pi_{h,k}^0(s,\cdot)\right\|_1^2 \le \frac{32\eta(N-1)^2H^2}{t},$$

using the fact that $\alpha_t^1 \le 1/t$ (Lemma 6 in [122]). This proves the first claim in the lemma. To further

establish the second statement, we sum over (2.48) to obtain

$$\begin{aligned}\sum_{i=1}^{N}\mathrm{SwapReg}_{h,i}^{t}(s)\leq&\frac{4NA_i^2H\log t}{\eta t}+8\eta(N-1)^2H^2\sum_{i=1}^{N}\sum_{j=2}^{t}\alpha_t^j\left\|\pi_{h,i}^{j}(s,\cdot)-\pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2\\&+\frac{32\eta NH^2(H+N^2)}{t}-\frac{1}{2048\eta}\sum_{i=1}^{N}\sum_{j=2}^{t}\frac{\alpha_t^{j-1}}{A_i}\left\|\pi_{h,i}^{j}(s,\cdot)-\pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2\\\leq&\frac{4NA_i^2H\log t}{\eta t}+\frac{32\eta NH^2(H+N^2)}{t}\\&+\sum_{i=1}^{N}\sum_{j=2}^{t}\left(8\eta(N-1)^2H^2-\frac{1}{2048\eta HA_i}\right)\alpha_t^j\left\|\pi_{h,i}^{j}(s,\cdot)-\pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2\\\leq&\frac{4NA_i^2H\log t}{\eta t}+\frac{32\eta NH^2(H+N^2)}{t}\\&-\frac{1}{2048\eta H}\sum_{i=1}^{N}\sum_{j=2}^{t}\frac{\alpha_t^j}{A_i}\left\|\pi_{h,i}^{j}(s,\cdot)-\pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2,\end{aligned}$$

where the second step uses the fact that $\alpha_t^{j-1}/\alpha_t^j=(j-1)/(H+j-1)\geq 1/H$, and the last step is due to the condition that $\eta\leq\frac{1}{256NH\sqrt{HA_{\max}}}$. □

Theorem 5. If Algorithm 8 is run on an N-player episodic Markov game for T iterations with a learning rate $\eta=\frac{1}{256NH\sqrt{HA_{\max}}}$, the output policy $\bar{\pi}$ satisfies:

$$\text{CE-Gap}(\bar{\pi})\leq\frac{2048NH^{\frac{7}{2}}A_{\max}^{\frac{5}{2}}\log T}{T}.$$

Proof. Using (2.9) from Lemma 4, we upper bound the second-order path lengths by

$$\begin{aligned}&8\eta NH^2\sum_{i=1}^{N}\sum_{j=2}^{t}\alpha_t^j\left\|\pi_{h,i}^{j}(s,\cdot)-\pi_{h,i}^{j-1}(s,\cdot)\right\|_1^2\\\leq&8\eta NH^2\cdot 2048\eta HA_{\max}\left(\frac{4NA_{\max}^2H\log t}{\eta t}+\frac{32\eta NH^2(N^2+H)}{t}\right),\end{aligned}$$

where we used the crucial fact that the swap regret is non-negative. Substituting the above equation back to (2.8) yields

$$\begin{aligned}\mathrm{SwapReg}_{h,i}^{t}(s)\leq&\frac{4A_i^2H\log t}{\eta t}+\frac{32\eta H^3N^2}{t}+\frac{2^{16}\eta N^2H^4A_{\max}^3\log t}{t}+\frac{2^{19}\eta^3N^4H^6}{t}\\\leq&\frac{2048NH^{\frac{5}{2}}A_{\max}^{\frac{5}{2}}\log t}{t},\end{aligned}\tag{2.49}$$

where the second step uses $\eta=\frac{1}{256NH\sqrt{HA_{\max}}}$. Since (2.49) holds for any $i\in\mathcal{N}$ and $s\in\mathcal{S}$, we can apply it back to the recursion of best response CE value gaps from Lemma 3 to obtain

$$\delta_h^t\leq\sum_{j=1}^{t}\alpha_t^j\delta_{h+1}^j+\frac{2048NH^{\frac{5}{2}}A_{\max}^{\frac{5}{2}}\log t}{t}.$$

Starting from $\delta_{H+1}^t = 0$, we can show via backward induction that for any $(h,t) \in [H] \times [T]$,

$$\delta_h^t \le \frac{2048 N A_{\max}^{\frac{5}{2}} (H-h+1) H^{\frac{5}{2}} \log t}{t}.$$

We conclude the proof of the theorem by referring to the property that $\text{CE-Gap}(\bar{\pi}) \le \delta_1^T$. □

2.14 Proofs for Section 2.8.2

Lemma 18. *(Theorem 3.1 from [118]) In a normal-form game with N players and A_i actions for player $i \in [N]$, suppose that all the players run OFTRL for T steps with negative entropy regularization and a learning rate $\eta = \Theta(\frac{1}{N \log^4 T})$. Then, there exists a constant $C > 1$ such that the regret of player i satisfies*

$$\text{Reg}_i^T \le C N \log A_i \cdot \log^4 T.$$

Lemma 19. *(Recursion of best response CCE value gaps) For any fixed $(h,t) \in [H] \times [T]$, let $\tau = \tau(t)$ denote the stage of t. Then, we have*

$$\zeta_h^t \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{start}}^{t_{\tau-1}^{end}} \zeta_{h+1}^j + \text{Reg}_h^{\tau-1}.$$

Proof. For any fixed $i \in \mathcal{N}$ and $s \in \mathcal{S}$, we know from the definition of $\bar{\pi}_h^t$ from Algorithm 11 that

$$V_{h,i}^{\bar{\pi}_h^t}(s) = \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left\langle \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] \right) \pi_{h,-i}^j \right](s,\cdot) \right\rangle. \tag{2.50}$$

From the definition of the best response value function,

$$\begin{aligned}
V_{h,i}^{\dagger,\bar{\pi}_{h,-i}^t}(s) &= \max_{\pi_i^\dagger(s,\cdot) \in \Delta(\mathcal{A}_i)} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left\langle \pi_i^\dagger(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\dagger,\bar{\pi}_{h+1,-i}^j}] \right) \pi_{h,-i}^j \right](s,\cdot) \right\rangle \\
&= \max_{\pi_i^\dagger} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left\langle \pi_i^\dagger(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] - [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] + [P_h V_{h+1,i}^{\dagger,\bar{\pi}_{h+1,-i}^j}] \right) \pi_{h,-i}^j \right](s,\cdot) \right\rangle \\
&\le \max_{\pi_i^\dagger} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left\langle \pi_i^\dagger(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}] \right) \pi_{h,-i}^j \right](s,\cdot) \right\rangle \\
&\quad + \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \max_{s' \in \mathcal{S}} \left(V_{h+1,i}^{\dagger,\bar{\pi}_{h+1,-i}^j} - V_{h+1,i}^{\bar{\pi}_{h+1}^j} \right)(s').
\end{aligned}$$

Subtracting (2.50) from the above equation leads to:

$$V_{h,i}^{\dagger,\bar{\pi}_{h,-i}^t}(s)-V_{h,i}^{\bar{\pi}_h^t}(s) \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \max_{s'\in\mathcal{S}} \left(V_{h+1,i}^{\dagger,\bar{\pi}_{h+1,-i}^j} - V_{h+1,i}^{\bar{\pi}_{h+1}^j}\right)(s')$$
$$+ \max_{\pi_i^\dagger} \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \left\langle \pi_i^\dagger(s,\cdot) - \pi_{h,i}^j(s,\cdot), \left[\left(r_{h,i} + [P_h V_{h+1,i}^{\bar{\pi}_{h+1}^j}]\right)\pi_{h,-i}^j\right](s,\cdot)\right\rangle. \tag{2.51}$$

Using a similar inductive argument as in the proof of Lemma 3, we can show that the term in (2.51) is equal to $\text{Reg}_{h,i}^{\tau-1}(s)$, which leads to

$$V_{h,i}^{\dagger,\bar{\pi}_{h,-i}^t}(s) - V_{h,i}^{\bar{\pi}_h^t}(s) \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \max_{s'\in\mathcal{S}} \left(V_{h+1,i}^{\dagger,\bar{\pi}_{h+1,-i}^j} - V_{h+1,i}^{\bar{\pi}_{h+1}^j}\right)(s') + \text{Reg}_{h,i}^{\tau-1}(s).$$

Since the above inequality holds for any $i \in \mathcal{N}$ and $s \in \mathcal{S}$, we can conclude that

$$\zeta_h^t \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \zeta_{h+1}^j + \text{Reg}_h^{\tau-1}.$$

This completes the proof of the recursive relationship of best response CCE value gaps. □

Theorem 6. If Algorithm 10 is run on an N-player episodic Markov game for T iterations with a learning rate $\eta_\tau = \Theta(\frac{1}{N\log^4 L_\tau})$ in each stage τ, the output policy $\bar{\pi}$ satisfies:

$$\text{CCE-Gap}(\bar{\pi}) = O\left(\frac{NH^3 \log A_{\max} \cdot \log^5 T}{T}\right).$$

Proof. We introduce a few more notations before presenting the proof. 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 (τ, h, s), we define the per-state (average) regret for player $i \in \mathcal{N}$ in the τ-th stage of the corresponding matrix game as

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

where $Q_{h,i}^\tau$ is player i's Q-function estimate at stage τ. For any $(h,t) \in [H]\times[T]$ and for the policy $\bar{\pi}_h^t$ as defined in Algorithm 11, we define the best response CCE value gap as

$$\zeta_h^t := \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \left(V_{h,i}^{\dagger,\bar{\pi}_{h,-i}^t}(s) - V_{h,i}^{\bar{\pi}_h^t}(s)\right).$$

By the definition of $\bar{\pi}$ and ζ_h^t, we have

$$\begin{aligned}\text{CCE-gap}(\bar{\pi}) =& \max_{i\in\mathcal{N}} \left(V_{1,i}^{\dagger,\bar{\pi}_{-i}}(s_1) - V_{1,i}^{\bar{\pi}}(s_1)\right) \\ \le& \frac{1}{T}\sum_{t=1}^{T} \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \left(V_{1,i}^{\dagger,\bar{\pi}_{1,-i}^t}(s) - V_{1,i}^{\bar{\pi}_1^t}(s)\right) \le \frac{1}{T}\sum_{t=1}^{T} \zeta_1^t. \end{aligned} \tag{2.52}$$

We use Lemma 19 to establish the following recursive relationship of the best response CCE value gaps between two consecutive steps h and $h+1$:

$$\zeta_h^t \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \zeta_{h+1}^j + \text{Reg}_h^{\tau-1}. \tag{2.53}$$

Hence, upper bounding CCE-gap$(\bar{\pi})$ breaks down to controlling the per-state regret in the corresponding matrix game for each $(\tau, s, h) \in [\bar{\tau}] \times \mathcal{S} \times [H]$. In our stage-based OFTRL, since the reward matrix $Q_{h,i}^\tau$ in each stage is fixed and $\text{Reg}_{h,i}^\tau(s)$ is the standard (average) regret, we can readily apply the individual regret bound of each player when running OFTRL in normal-form games [118]. Specifically, Theorem 3.1 from [118] (restated as Lemma 18) shows that with a learning rate $\eta_\tau = \Theta(\frac{1}{N\log^4 L_\tau})$, there exists a constant $C > 1$ such that for any $(i, \tau, s, h) \in \mathcal{N} \times [\bar{\tau}] \times \mathcal{S} \times [H]$,

$$\text{Reg}_{h,i}^\tau(s) \le \frac{CNH\log A_i \cdot \log^4 L_\tau}{L_\tau}. \tag{2.54}$$

Notice that we multiplied the regret bound by H because [118] assumes the rewards to be from $[0,1]$ but our rewards lie in $[0,H]$. According to the definition in Algorithm 11, the behavior of the policy $\bar{\pi}_h^t$ is unchanged for all t within the same stage τ as it always uniformly samples a time index from the previous stage and plays the corresponding history policy. Consequently, the value estimation error ζ_h^t does not change within a stage $\tau(t)$; that is, ζ_h^t takes the same value for all $t \in [t_\tau^{\text{start}}, t_\tau^{\text{end}}]$. We occasionally slightly abuse the notation and use ζ_h^τ to denote the estimation error for a stage τ. This immediately implies that $\frac{1}{L_{\tau-1}}\sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \zeta_{h+1}^j = \zeta_{h+1}^{\tau-1}$. Substituting (2.54) and the above equation back to the recursion (2.53), we obtain that

$$\begin{aligned}\zeta_h^t \le& \zeta_{h+1}^{\tau-1} + \frac{CNH\log A_{\max}\cdot\log^4 L_{\tau-1}}{L_{\tau-1}} \\ \le& \sum_{h'=h}^{H} \frac{CNH\log A_{\max}\cdot\log^4 T}{L_{\tau-h'+h-1}} \qquad (2.55) \\ \le& \frac{3CNH^2\log A_{\max}\cdot\log^4 T}{L_\tau}, \qquad (2.56)\end{aligned}$$

where the second step is by applying the inequality recursively over h, and the last step holds because our choice of the stage lengths $L_{\tau+1} = \lfloor(1+1/H)L_\tau\rfloor$ implies that

$$\frac{1}{L_{\tau-h'+h-1}} \le \frac{1}{L_\tau}\left(1+\frac{1}{H}\right)^{h'-h+1} \le \frac{1}{L_\tau}\left(1+\frac{1}{H}\right)^{H} \le \frac{3}{L_\tau}.$$

We then substitute (2.56) back to (2.52) and change the counting method to obtain

$$\begin{aligned}\text{CCE-gap}(\bar{\pi}) \leq \frac{1}{T}\sum_{t=1}^{T}\zeta_1^t \leq & \frac{1}{T}\sum_{\tau=1}^{\bar{\tau}}\sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}}\frac{3CNH^2\log A_{\max}\cdot\log^4 T}{L_\tau} \\ \leq & \frac{3CN\bar{\tau}H^2\log A_{\max}\cdot\log^4 T}{T}.\end{aligned}$$

It remains to bound the total number of stages $\bar{\tau}$. Since the lengths of the stages increase exponentially as $L_{\tau+1} = \lfloor(1+1/H)L_\tau\rfloor$ and the $\bar{\tau}$ stages sum up to T iterations, 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 $(1+\frac{1}{H})^H \geq e - \frac{e}{2H}$, and hence any $\bar{\tau} \geq \frac{H\log T}{\log(e/2)}$ satisfies the condition. This completes the proof of the theorem. □

2.15 Concluding Remarks

In this chapter, we have considered decentralized multi-agent reinforcement learning with efficient exploration in general-sum Markov games. We have proposed the V-learning OMD algorithm that provably finds an ε-approximate coarse correlated equilibrium in at most $\widetilde{O}(H^6SA/\varepsilon^2)$ episodes. As a useful side result, we have introduced an anytime online mirror descent algorithm with a dynamic learning rate and a high-probability regret bound.

We have also proposed stage-based V-learning algorithms that simplify the algorithmic design and analysis of V-learning OMD and achieve sharper sample complexity bounds. We have shown that stage-based V-learning can learn an ε-approximate CCE in $\widetilde{O}(H^5SA_{\max}/\varepsilon^2)$ episodes, and an ε-approximate CE in $\widetilde{O}(H^5SA_{\max}^2/\varepsilon^2)$ episodes. Our algorithms are decentralized and can readily scale up to a large number of agents without suffering from the exponential dependence. Furthermore, we have extended the V-learning framework to learning CCE/CE in full-information general-sum Markov games and established near-optimal $\widetilde{O}(T^{-1})$ convergence of our methods.

An interesting future direction would be to further tighten the sample complexity upper and lower bounds established in this chapter. In addition, in this chapter, we have considered the fully observable setup, where the agents have full access to the state information. This is in contrast to the more general setting with partially observable information structures [56], [127], [128], such as those modeled by decentralized partially observable Markov decision processes (DecPOMDPs) [58], [129], [130], where each agent has only a private partial observation of the state. Learning or even computing a NE in the latter case is much more challenging and would be an interesting future direction.

Chapter 3

Non-Stationary RL and Cooperative Markov Games

Reinforcement learning (RL) studies the problem where an agent maximizes its cumulative reward through sequential interactions with an initially unknown environment, usually modeled by a Markov Decision Process (MDP). The classical RL literature typically assumes that the state transition functions and the reward functions of the MDP are time-invariant. Such a stationary model, however, cannot capture the dynamic nature of many sequential decision-making problems in practice, especially those scenarios where multiple agents are involved.

In this chapter, we consider the problem of reinforcement learning in *non-stationary* MDPs. In our setting, both the reward functions and the state transition distributions are allowed to vary over time, either gradually or abruptly, as long as their cumulative variation magnitude does not exceed certain budgets. We propose an algorithm, named Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB), for this setting, which adopts a simple restarting strategy and an extra optimism term. We theoretically show that RestartQ-UCB outperforms existing solutions in terms of dynamic regret, a notion commonly utilized to measure the performance of an online learning algorithm in a non-stationary environment. Specifically, RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret bound of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where S and A are the numbers of states and actions, respectively, $\Delta > 0$ is the variation budget, H is the number of time steps per episode, and T is the total number of time steps. We further show that our algorithm is nearly optimal by establishing an information-theoretical lower bound of $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$, which appears to be the first impossibility result that characterizes the fundamental limits of non-stationary RL in general.

We further demonstrate the power of our results in the context of multi-agent RL, where non-stationarity is a key challenge. We show that RestartQ-UCB can be readily applied to learning team-optimal policies in cooperative smooth games against a slowly-changing opponent. To the best of our knowledge, RestartQ-UCB is the first model-free algorithm for non-stationary RL. Compared with model-based solutions, our algorithm is more time- and space-efficient, flexible, and compatible with the model deep RL architectures. We empirically evaluate RestartQ-UCB on RL tasks with both abrupt and gradual types of non-stationarity. Simulation results validate the advantages of RestartQ-UCB in terms of cumulative rewards and computational efficiency.

3.1 Introduction

Reinforcement learning (RL) focuses on the class of problems where an agent maximizes its cumulative reward through sequential interactions with an initially unknown but fixed environment, usually modeled by a Markov Decision Process (MDP). In classical RL problems, the state transition functions and the reward functions are assumed to be time-invariant, i.e., stationary. However, stationary models cannot capture the time-varying environments in a wide range of sequential decision-making problems, such as online advertisement auctions [18], [19], dynamic pricing [20], [131], traffic management [132], healthcare operations [133], multi-agent RL [8], and inventory control [21], [22]. Among the many intriguing applications, in the following, we specifically elaborate on two research areas, namely multi-agent RL and inventory control, that can significantly benefit from progresses on non-stationary RL.

- **Multi-agent RL:** In multi-agent RL, a set of agents either collaborate or compete by taking actions in a shared environment. This commonly occurs in many operational scenarios when multiple decision-makers interact with each other, such as ads auctions [134] and dynamic pricing [135]. In such scenarios, each agent faces a non-stationary environment, especially when the agents learn and update their policies simultaneously, as the actions of the other agents can alter the environment. We discuss this connection with more details in Section 3.8 through a concrete example, where we show that our non-stationary RL solution can be readily applied to a multi-agent RL problem against a slowly-changing opponent.
- **Inventory control across related but different products**: In conventional inventory control [21], [136], [137], the retailer typically focuses on managing the stock level of a single product. Nevertheless, the sequential launch of new related products (e.g., the line of iPhone) provides us with the opportunity to leverage experience from past products to inform inventory management for future products. In Section 3.9, we discuss how one can apply our non-stationary RL solutions to guide the inventory management not only for a single product but also across a sequence of related, but *different*, products.

Other areas that could benefit from non-stationary RL include sequential transfer in bandit [138] and RL [139] and multi-task RL [140], which in turn are conceptually related to continual RL [141] and life-long RL [142]. In the setting of sequential transfer/multi-task RL, the agent encounters a sequence of tasks over time with different system dynamics, and seeks to bootstrap learning by transferring knowledge from previously-solved tasks. Typical solutions in this area [139], [140], [143] need to assume that there are *finitely many* candidate tasks, and every task should be *sufficiently different* from the others[1]. Only under this assumption can the agent quickly identify the current task it is operating on, by essentially comparing the system dynamics it observes with the dynamics it has memorized for each candidate task. After identifying the current task with high confidence, the agent then invokes the policy that it learned through previous interactions with this specific task. This transfer learning paradigm in turn causes another problem—it "cold switches" between policies that are most likely very different, which might lead to unstable and inconsistent behaviors of the agent over time. Fortunately, non-stationary RL can help alleviate both the finite-task assumption and the cold-switching problem. First, non-stationary RL algorithms do not need the candidate tasks to be sufficiently different in order to correctly identify each of them, because the algorithm itself can tolerate some variations in the task environment. There will also be no need to assume the finiteness of the candidate task set anymore, and the candidate tasks can be drawn from a continuous space. Second, since we

[1]Needless to say, this assumption itself also to some extent contradicts the primary motivation of transfer learning. After all, we only want to transfer knowledge among tasks that are essentially similar to each other.

Table 3.1: Dynamic regret comparisons for RL in non-stationary MDPs. S and A are the numbers of states and actions, L is the number of abrupt changes, D is the maximum diameter, d is the dimension of the feature space for linear MDPs, H is the number of steps per episode, and T is the total number of steps. All upper bounds listed in the table are high-probability results that hold with probability at least $1-\delta$ for some $\delta \in (0,1)$, and $\widetilde{O}(\cdot)$ suppresses logarithmic dependence on S, A, T and $\frac{1}{\delta}$. Gray cells denote results from this thesis.

Setting	Algorithm	Regret	Model-Free	Comment
Undiscounted	[96]	$\widetilde{O}(S\ A^{\frac{1}{2}} L^{\frac{1}{3}} D\ T^{\frac{2}{3}})$	✗	only abrupt changes
	[145]	$\widetilde{O}(S^{\frac{2}{3}} A^{\frac{1}{3}} L^{\frac{1}{3}} D^{\frac{2}{3}} T^{\frac{2}{3}})$	✗	only abrupt changes
	[144]	$\widetilde{O}(S\ A^{\frac{1}{2}} \Delta^{\frac{1}{3}} D\ T^{\frac{2}{3}})$	✗	requires local variations
	[22]	$\widetilde{O}(S^{\frac{2}{3}} A^{\frac{1}{2}} \Delta^{\frac{1}{4}} D\ T^{\frac{3}{4}})$	✗	does not require Δ
	Lower bound	$\Omega(S^{\frac{1}{3}} A^{\frac{1}{3}} \Delta^{\frac{1}{3}} D^{\frac{2}{3}} T^{\frac{2}{3}})$		
Episodic	[146]	$\widetilde{O}(S\ A^{\frac{1}{2}} \Delta^{\frac{1}{3}} H^{\frac{4}{3}} T^{\frac{2}{3}})$	✗	also metric spaces
	RestartQ-UCB	$\widetilde{O}(S^{\frac{1}{3}} A^{\frac{1}{3}} \Delta^{\frac{1}{3}} H\ T^{\frac{2}{3}})$	✓	
	Double-Restart Q-UCB	$\widetilde{O}(S^{\frac{1}{3}} A^{\frac{1}{3}} \Delta^{\frac{1}{3}} H T^{\frac{2}{3}} + H^{\frac{3}{4}} T^{\frac{3}{4}})$	✓	does not require Δ
	Lower bound	$\Omega(S^{\frac{1}{3}} A^{\frac{1}{3}} \Delta^{\frac{1}{3}} H^{\frac{2}{3}} T^{\frac{2}{3}})$		
Linear MDPs	[147]	$\widetilde{O}(d^{\frac{4}{3}} \Delta^{\frac{1}{3}} H^{\frac{4}{3}} T^{\frac{2}{3}})$	✓	
	[148]	$\widetilde{O}(d^{\frac{5}{4}} \Delta^{\frac{1}{4}} H^{\frac{5}{4}} T^{\frac{3}{4}})$	✓	

are running the same non-stationary RL algorithm for a series of tasks, it improves its policy gradually over time, instead of cold-switching to a completely independent policy for each task. This could largely help with the unstable behavior issues.

RL in a non-stationary MDP is highly non-trivial due to the following challenges. First, similar to stationary RL, the agent faces the *exploration vs. exploitation* dilemma: it needs to explore the uncertain environment efficiently while maximizing its rewards along the way. In [96], the authors proposed to leverage the "optimism in the face of uncertain" principle to guide exploration. Another challenge, which is unique to non-stationary RL, is the trade-off between *remembering and forgetting*. On the one hand, since the underlying MDP varies over time, data samples collected in prior interactions can become obsolete. In fact, it has been shown that a standard stationary RL algorithm might incur a linear regret if the non-stationarity is not handled properly [144]. On the other hand, the agent needs to extract a sufficient amount of information from historical data to inform future decision-making.

To resolve the aforementioned challenges, [144] and [22] have proposed algorithms to guide learning in non-stationary MDPs. Although both model-based and model-free algorithms have been proposed for stationary RL, existing solutions for non-stationary RL are often built upon model-based methods. Nevertheless, it has been observed that model-based solutions often suffer from the following shortcomings:

• **Time- and space-inefficiency:** Model-based methods are in general more time- and space-consuming, and are less compatible with the design of modern deep RL architectures [60], [61].

• **Inefficient exploration:** In [22], [149], an example was given to show that under non-stationarity, the estimated model can incorrectly indicate that transitioning between states is very unlikely. This suggests that model-based methods, which try to estimate the latent model, might suffer "The Perils of Drift" [22].

• **Limited applicability:** In an important application of nonstationary RL — *decentralized* multi-agent RL, the agents cannot observe the actions taken by the other agents. This information structure precludes model-based methods, as the explicit estimation of the state transition functions is hardly possible without observing all the agents' actions.

These observations have thus motivated us to turn our attention to model-free methods, which, instead of maintaining estimates of the unknown underlying model, directly learn the Q-values.

Main Contributions. In this chapter, we focus on the problem of designing model-free algorithms with near-optimal performances for non-stationary RL. Our contributions are as follows:

1. We introduce an algorithm named Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB), which is the first model-free algorithm in the general setting of non-stationary RL. Our algorithm adopts a simple but effective restarting strategy [96], [150] that resets the memory of the agent according to a calculated schedule. The restarting strategy ensures that our algorithm only refers to the most up-to-date experience for decision-making. RestartQ-UCB also utilizes an extra optimism term (in addition to the standard Hoeffding/Freedman-based bonus) for exploration to counteract the non-stationarity of the MDP. This additional bonus term, depending on the local variation budgets (i.e., the environmental variation in each restarting interval), guarantees that our optimistic Q-value is still an upper bound of the optimal $Q^\star$-value even when the environment changes. Our analysis shows that RestartQ-UCB achieves the lowest dynamic regret bound when compared to existing works in the literature;
2. We present a variant of RestartQ-UCB that does not require knowledge of the local variation budget. Furthermore, we also show that our algorithm can completely remove the dependence on prior knowledge of the variation budget, a critical assumption commonly made in the literature [144], [147]. To accomplish that, we propose a parameter-free algorithm that leverages a "double restart" strategy to adaptively learn the variation budget [151];
3. We conduct simulations showing that RestartQ-UCB achieves highly competitive cumulative rewards against a state-of-the-art solution [147], while only taking 0.18% of its computation time;
4. We establish the first lower bounds in non-stationary RL, which suggest that our algorithm is optimal in all parameter dependences except for an $H^{\frac{1}{3}}$ factor, where H is the episode length;
5. To further showcase the flexibility and potential of non-stationary RL, we illustrate how it can be utilized to address the non-stationarity issue inherent in multi-agent RL. Specifically, we show that RestartQ-UCB can be readily applied to a multi-agent RL example against a slowly-changing opponent [152], [153]. The setting we consider is a more practical and general decentralized learning setting, which entails model-free solutions. We also discuss the application of our non-stationary RL algorithm in inventory control. Specifically, we demonstrate how to implement our RestartQ-UCB algorithm for the problem of inventory control across related, but *different* products with time-varying demands.

Related Works. Dynamic regret of non-stationary RL has been mostly studied using model-based solutions. [96] considers the setting where the MDP is allowed to change abruptly for L times. A sliding window approach is proposed in [145] under the same setting. [144] generalizes the previous setting by allowing the MDP to vary either abruptly or gradually at every step, subject to a total variation budget. [22] considers the same setting and introduce a Bandit-over-RL technique that adaptively tunes the algorithm without knowing the variation budget. Directly applying their method to our episodic setting will lead to a dynamic regret of $\widetilde{O}(S^{\frac{2}{3}}A^{\frac{1}{2}}\Delta^{\frac{1}{4}}HT^{\frac{3}{4}})$. Although it may be possible to further obtain an improved dependence on T, this is sub-optimal in terms of S and A. We remark that a recent (but later than ours) version of [22]

develops a lower bound tailored to the infinite horizon undiscounted non-stationary RL, but it is not directly applicable to our episodic non-stationary RL setting.

In a setting most similar to ours, [146] investigates non-stationary RL in the episodic setting, and propose a kernel-based approach when the state-action set forms a metric space. Their results can be reduced to an $\widetilde{O}(SA^{\frac{1}{2}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$ regret in the tabular case. [154] assumes stationary transitions and adversarial full-information rewards, and their setting is not directly comparable with ours. Two concurrent works [147] and [148] consider non-stationary RL in linear MDPs, but their regret bounds, $\widetilde{O}(S^{\frac{4}{3}}A^{\frac{4}{3}}\Delta^{\frac{1}{3}}H^{\frac{4}{3}}T^{\frac{2}{3}})$ and $\widetilde{O}(S^{\frac{5}{4}}A^{\frac{5}{4}}\Delta^{\frac{1}{4}}H^{\frac{5}{4}}T^{\frac{3}{4}})$ when reduced to the tabular RL setting, respectively, are less competitive than ours. After an earlier version of this work was made publicly available, [155] has proposed a black-box reduction procedure that turns an RL algorithm in a (nearly-)stationary environment to a non-stationary RL algorithm. In the episodic setting, [155] has achieved a strong dynamic regret bound of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$ (with or without knowledge of the degree of non-stationarity). However, their regret bound has a worse dependence on H when compared to ours, and it has been pointed out in [155] that such a sub-optimality cannot be improved upon by using a Freedman-style confidence bound as we do. Their compelling theoretical guarantees also come at the cost of a rather sophisticated and memory-inefficient algorithmic design, which needs to maintain many instances of the stationary subroutine, and constantly switch among them. Interested readers are referred to [156] for a comprehensive survey on RL in non-stationary environments. Table 3.1 compares our regret bounds with existing results that tackle similar settings as ours. It can be seen that our result is the first one that achieves the optimal dependence on S and A, and also establishes the tightest dependence on H/D and T among existing solutions in the literature, without relying on their assumptions.

Another related line of research studies online/adversarial MDPs [157]–[164], but they mostly only allow variations in reward functions, and use the static regret as a performance metric. In addition, RL with low switching cost [165] also shares a similar spirit as our restarting strategy since it also periodically forgets previous experiences. However, such algorithms do not address the non-stationarity of the environment, and their dynamic regret in terms of the variation budget is unclear.

Non-stationarity has also been considered in bandit problems [166]. Within different non-stationary multi-armed bandit (MAB) settings, various methods have been proposed, including decaying memory and sliding windows [167], [168], as well as restart-based strategies [107], [150], [169]. These methods largely inspired later research on non-stationary RL. A more recent line of work developed methods that do not require prior knowledge of the variation budget [170], [171] or the number of abrupt changes [172]. Other related settings considered in the literature include Markovian bandits [173]–[175], non-stationary contextual bandits [176], [177], linear bandits [178], [179], continuous-armed bandits [180], and learning with seasonal patterns [181].

Outline. The rest of this chapter is organized as follows: In Sections 3.2, we introduce the mathematical model of our problem and necessary preliminaries. In Section 3.3, we present our RestartQ-UCB algorithm. A dynamic regret analysis of RestartQ-UCB is provided in Section 3.4. In Section 3.5, we further propose a parameter-free algorithm that does not require prior knowledge of the variation budget. In Section 3.6, we establish information-theoretical lower bounds. Simulation results are presented in Section 3.7. In Sections 3.8 and 3.9, we discuss the applications of our method to two important scenarios: multi-agent RL and inventory control, respectively. For clarity of presentations, some supplementary material and the proofs of most results are deferred to Sections 3.10 to 3.15. Finally, we conclude this chapter in Section 3.16.

3.2 Preliminaries

We consider an episodic RL setting where an agent interacts with a non-stationary MDP for M episodes, with each episode containing H steps. We use a pair of integers (m, h) as a *time index* to denote the h-th step of the m-th episode. The environment can be denoted by a tuple $(\mathcal{S}, \mathcal{A}, H, P, r)$, where $\mathcal{S}$ is the finite set of states with $|\mathcal{S}| = S$, $\mathcal{A}$ is the finite set of actions with $|\mathcal{A}| = A$, H is the number of steps in one episode, $P = \{P_h^m\}_{m\in[M],h\in[H]}$ is the set of transition kernels, and $r = \{r_h^m\}_{m\in[M],h\in[H]}$ is the set of mean reward functions. Specifically, when the agent takes action $a_h^m \in \mathcal{A}$ in state $s_h^m \in \mathcal{S}$ at the time (m, h), it will receive a random reward $R_h^m(s_h^m, a_h^m) \in [0, 1]$ with expected value $r_h^m(s_h^m, a_h^m)$, and the environment transitions to a next state s_{h+1}^m following the distribution $P_h^m(\cdot \mid s_h^m, a_h^m)$. It is worth emphasizing that the transition kernel and the mean reward function depend both on m and h, and hence the environment is non-stationary over time. The episode ends when s_{H+1}^m is reached. We further denote $T = MH$ as the total number of steps.

A deterministic policy $\pi : [M] \times [H] \times \mathcal{S} \to \mathcal{A}$ is a mapping from the time index and state space to the action space, and we let $\pi_h^m(s)$ denote the action chosen in state s at time (m, h). Define $V_h^{m,\pi} : \mathcal{S} \to \mathbb{R}$ to be the value function under policy π at time (m, h), i.e.,

$$V_h^{m,\pi}(s) := \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}^m\left(s_{h'}, \pi_{h'}^m\left(s_{h'}\right)\right) \mid s_h = s\right],$$

where $s_{h'+1} \sim P_{h'}^m(\cdot \mid s_{h'}, a_{h'})$. Accordingly, the state-action value function $Q_h^{m,\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as:

$$Q_h^{m,\pi}(s, a) := r_h^m(s, a) + \mathbb{E}\left[\sum_{h'=h+1}^{H} r_{h'}^m\left(s_{h'}, \pi_{h'}^m\left(s_{h'}\right)\right) \mid s_h = s, a_h = a\right]$$

For simplicity of notation, we let $P_h^m V_{h+1}(s, a) := \mathbb{E}_{s'\sim P_h^m(\cdot|s,a)}\left[V_{h+1}(s')\right]$. Then, the Bellman equation gives $V_h^{m,\pi}(s) = Q_h^{m,\pi}(s, \pi_h^m(s))$ and $Q_h^{m,\pi}(s, a) = (r_h^m + P_h^m V_{h+1}^{m,\pi})(s, a)$, and we also have $V_{H+1}^{m,\pi}(s) = 0, \forall s \in \mathcal{S}$ by definition. Since the state space, the action space, and the length of each episode are all finite, there always exists an optimal policy $\pi^\star$ that gives the optimal value $V_h^{m,\star}(s) := V_h^{m,\pi^*}(s) = \sup_\pi V_h^{m,\pi}(s), \forall s \in \mathcal{S}, m \in [M], h \in [H]$. From the Bellman optimality equation, we have $V_h^{m,\star}(s) = \max_{a\in\mathcal{A}} Q_h^{m,\star}(s, a)$, where $Q_h^{m,\star}(s, a) := (r_h^m + P_h^m V_{h+1}^{m,\star})(s, a)$, and $V_{H+1}^{m,\star}(s) = 0, \forall s \in \mathcal{S}$.

Dynamic Regret: The agent aims to maximize the cumulative expected reward over the entire M episodes, by adopting some policy π. We measure the optimality of the policy π in terms of its *dynamic regret* [22], [146], which compares the agent's policy with the optimal policy of each individual episode in hindsight:

$$\mathcal{R}(\pi, M) := \sum_{m=1}^{M}\left(V_1^{m,\star}\left(s_1^m\right) - V_1^{m,\pi}\left(s_1^m\right)\right),$$

where the initial state s_1^m of each episode is chosen by an oblivious adversary [61]. Dynamic regret is a stronger measure than the standard (static) regret, which only considers the single policy that is optimal over all episodes combined.

Variation: We measure the non-stationarity of the MDP in terms of its *variation budget* in the mean reward function and transition kernels:

$$\Delta_r := \sum_{m=1}^{M-1}\sum_{h=1}^{H}\sup_{s,a}|r_h^m(s, a) - r_h^{m+1}(s, a)|, \quad \Delta_p := \sum_{m=1}^{M-1}\sum_{h=1}^{H}\sup_{s,a}\left\|P_h^m(\cdot \mid s, a) - P_h^{m+1}(\cdot \mid s, a)\right\|_1,$$

Algorithm 14: RestartQ-UCB (Hoeffding/Freedman)

for *epoch* $d \leftarrow 1$ *to* D **do**
  Initialize: $V_h(s) \leftarrow H-h+1, Q_h(s,a) \leftarrow H-h+1, N_h(s,a) \leftarrow 0, \check{N}_h(s,a) \leftarrow 0,$
  $\check{r}_h(s,a) \leftarrow 0, \check{v}_h(s,a) \leftarrow 0, \check{\mu}_h(s,a) \leftarrow 0, \check{\sigma}_h(s,a) \leftarrow 0, \mu_h^{\mathrm{ref}}(s,a) \leftarrow 0, \sigma_h^{\mathrm{ref}}(s,a) \leftarrow 0, V_h^{\mathrm{ref}}(s) \leftarrow H,$
  for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$;
  for *episode* $k \leftarrow (d-1)K+1$ *to* $\min\{dK, M\}$ **do**
    observe s_1;
    for *step* $h \leftarrow 1$ *to* H **do**
      Take action $a_h \leftarrow \arg\max_a Q_h(s_h,a)$, receive $R_h(s_h,a_h)$, and observe s_{h+1};
      $\check{r}_h(s_h,a_h) \leftarrow \check{r}_h(s_h,a_h) + R_h(s_h,a_h), \check{v}_h(s_h,a_h) \leftarrow \check{v}_h(s_h,a_h) + V_{h+1}(s_{h+1})$;
      $\check{\mu}(s_h,a_h) \leftarrow \check{\mu}(s_h,a_h) + V_{h+1}(s_{h+1}) - V_{h+1}^{\mathrm{ref}}(s_{h+1})$;
      $\check{\sigma}(s_h,a_h) \leftarrow \check{\sigma}(s_h,a_h) + \big(V_{h+1}(s_{h+1}) - V_{h+1}^{\mathrm{ref}}(s_{h+1})\big)^2$;
      $\mu^{\mathrm{ref}}(s_h,a_h) \leftarrow \mu^{\mathrm{ref}}(s_h,a_h) + V_{h+1}^{\mathrm{ref}}(s_{h+1}), \sigma^{\mathrm{ref}}(s_h,a_h) \leftarrow \sigma^{\mathrm{ref}}(s_h,a_h) + (V_{h+1}^{\mathrm{ref}}(s_{h+1}))^2$;
      $n := N_h(s_h,a_h) \leftarrow N_h(s_h,a_h)+1, \check{n} := \check{N}_h(s_h,a_h) \leftarrow \check{N}_h(s_h,a_h)+1$;
      if $N_h(s_h,a_h) \in \mathcal{L}$ **then**
        // Reaching the end of the stage
        $b_h \leftarrow \sqrt{\frac{H^2}{\check{n}}\iota} + \sqrt{\frac{1}{\check{n}}\iota},\ b_\Delta \leftarrow \Delta_r^{(d)} + H\Delta_p^{(d)}$;
        $\underline{b}_h \leftarrow 2\sqrt{\frac{\sigma^{\mathrm{ref}}/n - (\mu^{\mathrm{ref}}/n)^2}{n}\iota} + 2\sqrt{\frac{\check{\sigma}/\check{n} - (\check{\mu}/\check{n})^2}{\check{n}}\iota} + 5\big(\frac{H\iota}{n} + \frac{H\iota}{\check{n}} + \frac{H\iota^{3/4}}{n^{3/4}} + \frac{H\iota^{3/4}}{\check{n}^{3/4}}\big) + \sqrt{\frac{1}{\check{n}}\iota}$;
        $Q_h(s_h,a_h) \leftarrow \min\Big\{\frac{\check{r}}{\check{n}} + \frac{\check{v}}{\check{n}} + b_h + 2b_\Delta, \frac{\check{r}}{\check{n}} + \frac{\mu^{\mathrm{ref}}}{n} + \frac{\check{\mu}}{\check{n}} + 2\underline{b}_h + 4b_\Delta, Q_h(s_h,a_h)\Big\}$; (*)
        $V_h(s_h) \leftarrow \max_a Q_h(s_h,a)$;
        $\check{N}_h(s_h,a_h) \leftarrow 0, \check{r}_h(s_h,a_h) \leftarrow 0, \check{v}_h(s_h,a_h) \leftarrow 0, \check{\mu}_h(s_h,a_h) \leftarrow 0, \check{\sigma}_h(s_h,a_h) \leftarrow 0$;
      if $\sum_a N_h(s_h,a) = N_0$ **then** // Learn the reference value
        $V_h^{\mathrm{ref}}(s_h) \leftarrow V_h(s_h)$;

where $\|\cdot\|_1$ is the L^1-norm. Note that our definition of variation budgets only imposes restrictions on the summation of non-stationarity across two different episodes and does not put any restriction on the difference between two consecutive steps in the same episode; that is, $P_h^m(\cdot \mid s,a)$ and $P_{h+1}^m(\cdot \mid s,a)$ are allowed to be arbitrarily different. We further let $\Delta = \Delta_r + \Delta_p$, and assume $\Delta > 0$.

3.3 Algorithm: RestartQ-UCB

We present our algorithm Restarted Q-Learning with Hoeffding/Freedman Upper Confidence Bounds (RestartQ-UCB Hoeffding/Freedman) in Algorithm 14. For illustrative purposes, we start with a simpler RestartQ-UCB algorithm with Hoeffding-style bonus terms, which only executes the pseudocode colored in black in Algorithm 14. Further incorporating the gray parts in Algorithm 14 leads to the RestartQ-UCB algorithm with Freedman-style bonus terms and reference-advantage decomposition [61], which achieves a sharper dynamic regret bound at the cost of a slightly more involved analysis.

Common to both the Hoeffding and the Freedman bonus terms, RestartQ-UCB breaks the M episodes into D *epochs*, with each epoch containing $K = \lceil \frac{M}{D} \rceil$ episodes (except for the last epoch which possibly has less than K episodes). With a large value of D, Algorithm 14 restarts more frequently to adjust to the potential variations of the environment, at the cost of spending more time searching for new optimal policies. On the contrary, a small value of D would lead to running stable policies for long periods of time with less frequent restarts, but the resulting algorithm might not be able to adjust to the environmental variations rapidly enough. To strike a balance, we set the number of epochs to be $D = S^{-\frac{1}{3}} A^{-\frac{1}{3}} \Delta^{\frac{2}{3}} H^{-\frac{2}{3}} T^{\frac{1}{3}}$

so as to achieve the optimal dynamic regret bound, and such a choice will be justified later in our analysis. RestartQ-UCB periodically restarts a Q-learning algorithm with UCB exploration at the beginning of each epoch, thereby addressing the non-stationarity of the environment. For each $d \in [D]$, define $\Delta_r^{(d)}$ to be the *local variation budget* of the mean reward function within epoch d. By definition, we have $\sum_{d=1}^{D} \Delta_r^{(d)} \le \Delta_r$. Define the local variation budget of transitions $\Delta_p^{(d)}$ analogously.

Since our algorithm essentially invokes the same procedure for every epoch, in the following, we focus our analysis on what happens inside one epoch only (and without loss of generality, we focus on epoch 1, which contains episodes $1, 2, \ldots, K$). At the end of our analysis, we will merge the results across all epochs.

For each triple $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, we divide the visitations (within epoch 1) to the triple into multiple *stages*, where the length of the stages increases exponentially at a rate of $(1 + \frac{1}{H})$. Specifically, let $e_1 = H$, and $e_{i+1} = \lfloor (1 + \frac{1}{H}) e_i \rfloor, i \ge 1$ denote the lengths of the stages. Further, let the partial sums $\mathcal{L} := \{\sum_{i=1}^{j} e_i \mid j = 1, 2, 3, \ldots\}$ denote the set of the ending times of the stages. We remark that the stages are defined for each individual triple (s, a, h), and for different triples the starting and ending times of their stages do not necessarily align in time. Such a definition of stages is mostly motivated by the design of the learning rate $\alpha_t = \frac{H+1}{H+t}$ in [60]. It ensures that only the last $O(1/H)$ fraction of samples is given non-negligible weights when used to estimate the optimistic $Q_h(s, a)$ values, while the first $1 - O(1/H)$ fraction is forgotten [61]. We set $\iota := \log\left(\frac{2}{\delta}\right)$, where δ is an input parameter that can be set by us.

Recall that the time index (k, h) represents the h-th step of the k-th episode. At each step (k, h), we take the optimal action with respect to the optimistic $Q_h(s, a)$ value (Line 6 in Algorithm 14), which is designed as an optimistic estimate of the optimal $Q_h^{k,\star}(s, a)$ value of the corresponding episode. For each triple (s, a, h), we update the optimistic $Q_h(s, a)$ value at the end of each stage, using samples only from this latest stage that is about to end (Line 16 in Algorithm 14). The optimism in $Q_h(s, a)$ comes from two bonus terms $b_h/\underline{b}_h$ and b_Δ, where $b_h/\underline{b}_h$ is a standard Hoeffding/Freedman-based optimism that is commonly used in upper confidence bounds [60], [61], and b_Δ is the extra optimism that we need to take into account because of the non-stationarity of the environment. The definition of b_Δ requires knowledge of the local variation budget in each epoch, which is a rather strong assumption in practice. However, we can further show (later in Theorems 10 and 11) that if we simply replace Equation $(*)$ in Algorithm 14 with the following update rule:

$$Q_h(s_h, a_h) \leftarrow \min\left\{ \frac{\check{r}}{\check{n}} + \frac{\check{v}}{\check{n}} + b_h, \frac{\check{r}}{\check{n}} + \frac{\mu^{\mathrm{ref}}}{n} + \frac{\check{\mu}}{\check{n}} + 2\underline{b}_h, Q_h(s_h, a_h) \right\} \tag{3.1}$$

then our algorithm can achieve the same regret without assumptions on the local variation budget.

Compared with the Hoeffding-based algorithm, there are two major improvements in the Freedman-based one. The first improvement is the replacement of the Hoeffding-based bonus term b_h^k with a tighter term $\underline{b}_h^k$. The latter term takes into account the second moment information of the random variables, which allows sharper tail bounds that rely on second moments to come into use (in our case, the Freedman's inequality). The second improvement is a variance reduction technique, or more specifically, the reference-advantage decomposition as coined in [61]. The intuition is to first learn a reference value function V^{ref} that serves as a roughly accurate estimate of the optimal value function $V^\star$ in each epoch. The goal of learning the optimal value function $V^\star = V^{\mathrm{ref}} + (V^\star - V_{\mathrm{ref}})$ can hence be decomposed into estimating the two terms V^{ref} and $V^* - V_{\mathrm{ref}}$. The reference value V^{ref} is a fixed term, and can be accurately estimated using a large number of samples (in Algorithm 14, we estimate V^{ref} only when we have $N_0 = cSAH^6\iota$ samples for a large constant c). The advantage term $V^* - V^{\mathrm{ref}}$ can also be estimated more accurately due to the reduced variance.

3.4 Analysis

In this section, we present our main result—a dynamic regret analysis of the RestartQ-UCB algorithm. Our first result on RestartQ-UCB with Hoeffding-style bonus terms is summarized in the following theorem. Complete proofs of its supporting lemmas are given in Section 3.10.

Theorem 9. *(Hoeffding) For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Hoeffding bonuses is bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of S, A, T and $1/\delta$.*

Our proof relies on the following technical lemma, stating that for any triple (s,a,h), the difference of their optimal Q-values at two different episodes $1 \le k_1 < k_2 \le K$ is bounded by the variation of this epoch.

Lemma 20. *For any triple (s,a,h) and any $1 \le k_1 < k_2 \le K$, it holds that $|Q_h^{k_1,\star}(s,a) - Q_h^{k_2,\star}(s,a)| \le \Delta_r^{(1)} + H\Delta_p^{(1)}$.*

Let $Q_h^k(s,a)$ denote the value of $Q_h(s,a)$ at the *beginning* of the k-th episode in RestartQ-UCB Hoeffding. The following lemma states that the optimistic Q-value $Q_h^k(s,a)$ is an upper bound of the optimal Q-value $Q_h^{k,\star}(s,a)$ with high probability. Note that we only need to show that the event holds with probability $1 - poly(S,A,K,H)\delta$, because we can replace δ with $\delta/poly(S,A,K,H)$ in the end to get the desired high probability bound without affecting the polynomial part of the regret bound.

Lemma 21. *(Hoeffding) For $\delta \in (0,1)$, with probability at least $1-2KH\delta$, it holds that $Q_h^{k,\star}(s,a) \le Q_h^{k+1}(s,a) \le Q_h^k(s,a), \forall (s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$.*

Building upon Lemmas 20 and 21, a complete proof of Theorem 9 is given in Section 3.11. We remark that Algorithm 14 relies on the assumption that the local variations b_Δ are known a priori, which is a strong but commonly made assumption in the literature on non-stationary RL [144], [147]. To the best of our knowledge, existing restart-based solutions either crucially rely on this local variation assumption [144], or suffer a severe regret degeneration after removing this assumption [147]. Interestingly, in the following theorem, we show that this assumption can be safely removed in our approach without affecting the regret bound. The only modification to the algorithm is to replace the Q-value update rule in Equation $(*)$ of Algorithm 14 with the new update rule in Equation (3.1).

Theorem 10. *(Hoeffding, no local budgets) For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Hoeffding bonuses and no knowledge of local budgets is bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of S, A, T and $1/\delta$.*

To understand why this simple modification works, notice that in $(*)$ we add exactly the same value $2b_\Delta$ to the upper confidence bounds of all (s,a) pairs in the same epoch. Subtracting the same value from all optimistic Q-values simultaneously should not change the choice of actions in future steps. The only difference is that the new "optimistic" $Q_h^k(s_h,a_h)$ values would no longer be strict upper bounds of the optimal $Q_h^{k,\star}(s_h,a_h)$ anymore, but instead "upper bounds" subject to some error term induced by b_Δ. Specifically, since $Q_h(s_h,a_h)$ is updated using $V_{h+1}(s_{h+1})$, which, in turn, contains some error in terms of b_Δ, the error will propagate across the steps. By properly tracking such error terms, we can see that there are in total $H-h+1$ copies of the $2b_\Delta$ error accumulated from step H back to step h. This leads to the following variant of Lemma 21 that quantifies the error terms in the new "optimistic" bounds.

Lemma 22. *(Hoeffding, no local budgets) Suppose that we have no prior knowledge of the local variations and replace the update rule (∗) in RestartQ-UCB Hoeffding with Equation (3.1). For $\delta \in (0,1)$, with probability at least $1-2KH\delta$, it holds that $Q_h^{k,\star}(s,a)-2(H-h+1)b_\Delta \le Q_h^{k+1}(s,a) \le Q_h^k(s,a), \forall (s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$.*

Remark 3. *The easy removal of the local budget assumption is non-trivial in the design of the algorithm, and to the best of our knowledge is absent in the non-stationary RL literature with restarts. In fact, it has been shown in a concurrent work [147] that removing this assumption could lead to a much worse regret bound (cf. Corollary 2 and Corollary 3 therein).*

Replacing the Hoeffding-based upper confidence bound with a Freedman-style one will lead to a tighter regret bound, summarized in Theorem 11 below. To remove the local budget assumption, we also need to replace the update rule (∗) in Algorithm 14 with Equation (3.1). The proof of the theorem follows a similar procedure as in the proof of Theorem 10, and is given in Section 3.13. It relies on a reference-advantage decomposition technique for variance reduction as in [61].

Theorem 11. *(Freedman, no local budgets) For T greater than some polynomial of S, A, Δ and H, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of RestartQ-UCB with Freedman bonuses (Algorithm 14 including the gray parts) is upper bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of S, A, T and $1/\delta$.*

Remark 4 (From High Probability Regret Bound to Expected Regret Bound)**.** *We note that δ is an input parameter, and our high probability regret bounds can immediately imply expected regret bounds. In all the above theorems presented in this section, the dynamic regret depends on $1/\delta$ through logarithmic terms. Since the regret can at most be $O(T)$, by setting $\delta = 1/T$, one can retain the same regret bound in an expectation sense. For instance, in Theorem 11, by setting $\delta = 1/T$, we have that with probability at least $1-\delta$, the regret is $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, while with probability at most δ, the regret is $O(T)$. Hence, the expected regret of the algorithm is $(1-\delta)\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}}) + \delta O(T) = \widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$*

3.5 Unknown Variation Budgets

In Theorem 11, we have removed the assumption on the knowledge of "local" variation budgets $\Delta_r^{(d)}$ and $\Delta_p^{(d)}$ for $d \in [D]$, but the design of the algorithm still relies on knowledge of the "total" variation budget Δ. Specifically, to achieve the dynamic regret bound presented in Theorem 11, we need to set the number of epochs to $D^\star = S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}}$, which clearly requires to know Δ in advance. To further overcome such a limitation, in this section, we propose a parameter-free algorithm that adaptively learns the variation budget Δ when it is unknown a priori, while still achieving sublinear dynamic regret in T.

Our new algorithm, Double-Restart Q-UCB, for the unknown variation budget setting is presented in Algorithm 15. Inspired by the Bandit-over-Bandit algorithm [22], [171] that adaptively tunes the algorithm parameters in a linear bandit problem, we also use a multi-armed bandit algorithm as a master procedure to learn the optimal value $D^\star$ of D. Given a set $\mathcal{J}$ of candidate values for D, the idea of our algorithm is to first divide the time horizon T into multiple *phases*, and then in each phase we experiment with one candidate value from the set $\mathcal{J}$. If we choose values from $\mathcal{J}$ properly using a bandit algorithm, the cumulative reward we obtain through this experimentation procedure should be close to the performance of using the best fixed candidate from $\mathcal{J}$ in hindsight. Since the underlying environment need not drift according to any statistical pattern, we use an adversarial bandit algorithm Exp3.P [107] to defend against the possibly adversarial changes of the best D value in each phase.

Algorithm 15: Double-Restart Q-UCB

Input: Parameters $W, \mathcal{J}$, α, and γ as given in Equation (3.2) and (3.3).

Initialize: Weights of the bandit arms $s_1(j) = \exp\left(\frac{\alpha\gamma}{3}\sqrt{\frac{\lceil M/W \rceil}{J+1}}\right)$ for $j = 0, 1, \ldots, \lceil \ln W \rceil$.

for *phase* $i \leftarrow 1$ *to* $\lceil \frac{M}{W} \rceil$ **do**

  $p_i(j) \leftarrow (1-\gamma)\frac{s_i(j)}{\sum_{j'=0}^{J} s_i(j')} + \frac{\gamma}{J+1}, \forall j = 0, 1, \ldots, J;$

  Draw an arm A_i from $\{0, \ldots, J\}$ randomly according to the probabilities $p_i(0), \ldots, p_i(J)$;

  Set the estimated number of epochs $D_i \leftarrow \left\lfloor \frac{TW^{\frac{A_i}{J}}}{SAH^2W} \right\rfloor$;

  Run a new instance of Algorithm 14 (including gray parts) for W episodes with parameter value $D \leftarrow D_i$;

  Observe the cumulative reward R_i from the last W episodes;

  for *arm* $j \leftarrow 0, 1, \ldots, J$ **do**

    $\hat{R}_i(j) \leftarrow R_i \mathbb{I}\{j = A_i\} / (WHp_i(j));$

    $s_{i+1}(j) \leftarrow s_i(j) \exp\left(\frac{\gamma}{3(J+1)}\left(\hat{R}_i(j) + \frac{\alpha}{p_i(j)\sqrt{(J+1)\lceil M/W \rceil}}\right)\right);$

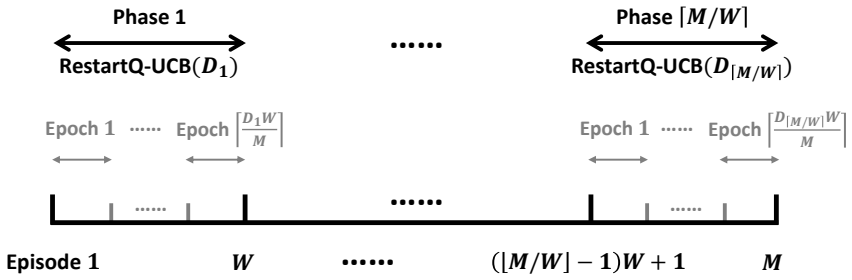


Figure 3.1: Structure of the Double-Restart Q-UCB algorithm.

We sketch the high-level structure of the Double-Restart Q-UCB algorithm in Figure 3.1 to help clarify any possible confusion regarding our definitions of "phases", "epochs", and "episodes". Concretely, we divide the overall M episodes into $\lceil \frac{M}{W} \rceil$ phases, each phase containing $W \in \mathbb{N}_+$ episodes (except that the last phase could have less than W episodes). At the beginning of each phase i, we start a new instance of Algorithm 14 (including gray parts) with a candidate value of $D_i \in \mathcal{J}$ to be experimented in this phase. Since Algorithm 14 itself is a restart-based process, it further sub-divides the W episodes in phase i into $\lceil \frac{D_i W}{M} \rceil$ epochs. To understand this value, suppose D_i is an appropriate value for D, such that dividing the overall horizon into D_i epochs leads to near-optimal dynamic regret. Then, since the overall horizon contains M episodes while each phase only contains W episodes, we should only divide each phase into $\lceil \frac{D_i W}{M} \rceil$ epochs to reflect the corresponding consequence of choosing D_i as the overall number of epochs. Since we restart Algorithm 14 in each phase and Algorithm 14 in turn restarts an optimistic Q-learning sub-routine in each epoch, our overall algorithm exhibits a double-loop restarting behavior, and hence the name Double-Restart Q-UCB.

In the following, we instantiate the choices of the set $\mathcal{J}$, the phase length W, as well as the parameter values used in the Exp3.P bandit algorithm. First, we define

$$W = \sqrt{HT}, J = \lceil \ln W \rceil, \text{ and } \mathcal{J} = \left\{ \left\lfloor \frac{T}{SAH^2W} \right\rfloor, \left\lfloor \frac{TW^{\frac{1}{J}}}{SAH^2W} \right\rfloor, \left\lfloor \frac{TW^{\frac{2}{J}}}{SAH^2W} \right\rfloor, \ldots, \left\lfloor \frac{TW}{SAH^2W} \right\rfloor \right\}, \quad (3.2)$$

where $\mathcal{J}$ is the set of candidate values for D and we can see that $|\mathcal{J}| = \lceil \ln W \rceil + 1 = J + 1$. Each candidate value in $\mathcal{J}$ is also called an "arm" in the language of bandits, and we use "arm j" to refer to the candidate value $\left\lfloor \frac{TW^{\frac{j}{J}}}{SAH^2W} \right\rfloor$ for $j = 0, 1, \ldots, J$. We initialize the weights of the bandit arms by $s_1(j) = \exp\left(\frac{\alpha\gamma}{3}\sqrt{\frac{\lceil M/W \rceil}{J+1}}\right)$ for $j = 0, 1, \ldots, J$, where as specified in [107],

$$\alpha = 2\sqrt{\ln\left(\lceil M/W \rceil (J+1)/\delta\right)}, \text{ and } \gamma = \min\left\{\frac{3}{5}, 2\sqrt{\frac{3}{5}\frac{(J+1)\ln(J+1)}{\lceil M/W \rceil}}\right\}, \tag{3.3}$$

for some failure probability $\delta > 0$. At the beginning of each phase $i \in \{1, 2, \ldots, \lceil \frac{M}{W} \rceil\}$, we randomly draw an arm j with probability $p_i(j)$ that is calculated from the weights

$$p_i(j) = (1-\gamma)\frac{s_i(j)}{\sum_{j'=0}^{J} s_i(j')} + \frac{\gamma}{J+1}, \forall j = 0, 1, \ldots, J.$$

We set our estimated parameter D_i to be the value associated with the selected arm j in the set $\mathcal{J}$. We then run Algorithm 14 for W episodes by setting the number of epochs to be $D = D_i$. To put it in another way, we execute a new instance of Algorithm 14 for $\lceil \frac{D_i W}{M} \rceil$ epochs, where each epoch contains $K_i = \left\lfloor \frac{M}{D_i} \right\rfloor$ episodes. We collect the cumulative reward R_i from the aforementioned W episodes. The normalized value $R_i/(WH) \in [0, 1]$ hence corresponds to the reward of playing the selected arm in time step i of the bandit problem. Finally, we update the weights of the bandit arms based on the observed reward, using the following update rule specified in the Exp3.P algorithm:

$$s_{i+1}(j) \leftarrow s_i(j) \exp\left(\frac{\gamma}{3(J+1)}\left(\hat{R}_i(j) + \frac{\alpha}{p_i(j)\sqrt{(J+1)\lceil M/W \rceil}}\right)\right),$$

where $\hat{R}_i(j) = R_i \mathbb{I}\{j = A_i\} / \left(WHp_i(j)\right), \forall j = 0, 1, \ldots, J$, and A_i denotes the arm selected at phase i.

The following result states that our Double-Restart Q-UCB algorithm achieves a sublinear dynamic regret in T, without requiring knowledge of the (total) variation budget Δ.

Theorem 12. *(Freedman, no total budgets) For T greater than some polynomial of S, A, Δ and H, and for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the dynamic regret of Double-Restart Q-UCB with Freedman bonuses and no prior knowledge of the total variation budget Δ is bounded by $\widetilde{O}(S^{\frac{1}{3}} A^{\frac{1}{3}} \Delta^{\frac{1}{3}} H T^{\frac{2}{3}} + H^{\frac{3}{4}} T^{\frac{3}{4}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors.*

The regret bound in Theorem 12 consists of two terms: The first term is the dynamic regret of using the optimal candidate value $D^\dagger \in \mathcal{J}$ of the number of epochs. This term is in the same order as the known-variation case (Theorem 11), because we have discretized the candidate value set $\mathcal{J}$ at a proper granularity such that the optimal candidate value $D^\dagger \in \mathcal{J}$ approximates the actual optimal value $D^\star$. The second regret term in Theorem 12 is caused by the regret of learning the optimal candidate value inside $\mathcal{J}$ using the Exp3.P algorithm. Due to the additional step of estimating the unknown variation budget, the overall dynamic regret bound becomes slightly worse in terms of its dependence on T (from $\widetilde{O}(T^{\frac{2}{3}})$ in Theorem 11 to $\widetilde{O}(T^{\frac{3}{4}})$). Such a degradation seems unavoidable under the current framework as it has also appeared in a similar bandit scenario [178].

Remark 5 (Comparison with [149])**.** *We follow the Bandit-over-RL technique to utilize a separate bandit algorithm to select the key parameters for our algorithm. But we have to emphasize that the resulting algorithm*

is simpler and more practical for implementation. This is because our Double-Restart Q-UCB algorithm is essentially running a stationary Q-UCB algorithm in between restarts. In contrast, the algorithm in [149]) relies on a carefully tuned sliding-window update schedule. More importantly, we point out that such a design (together with our new analysis) can lead to an improved dynamic regret bound in terms of S and A (even with the Hoeffding-style bonus terms similar to [149]). This exhibits the advantage of our design compared to that of [149], which combines restart and sliding-window.

3.6 Lower Bounds

In this section, we provide information-theoretical lower bounds of the dynamic regret to characterize the fundamental limits of any algorithm in non-stationary RL.

Theorem 13. *For any algorithm, there exists an episodic non-stationary MDP such that the dynamic regret of the algorithm is at least* $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$.

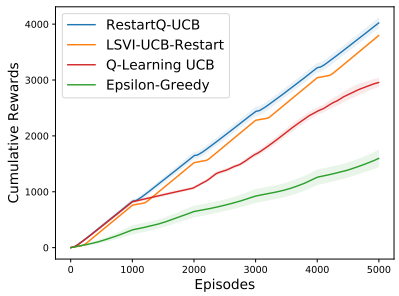
Proof sketch. The proof of our lower bound relies on the construction of a "hard instance" of non-stationary MDPs. The instance we construct is essentially an MDP with piecewise constant dynamics on each *segment* of the horizon, and its dynamics experience an abrupt change at the beginning of each new segment. Specifically, we divide the horizon T into L segments[2], where each segment has $T_0 := \lfloor \frac{T}{L} \rfloor$ steps and contains $M_0 := \lfloor \frac{M}{L} \rfloor$ episodes. Within each segment, the system dynamics of the MDP do not vary, and we construct the dynamics for each segment in a way such that the instance is a hard instance of stationary MDPs on its own. The MDP within each segment is essentially similar to the hard instances constructed in [60], [182]. Between two consecutive segments, the dynamics of the MDP change abruptly, and we let the dynamics vary in a way such that no information learned from previous interactions with the MDP can be used in the new segment. In this sense, the agent needs to learn a new hard MDP in each segment. Finally, optimizing the value of L and the variation magnitude between consecutive segments (subject to the constraints of the total variation budget) leads to our lower bound. □

Remark 6. *We emphasize that in our construction of the worst-case non-stationary MDP, we only let the state transition kernel vary over time but keep the reward functions fixed. By doing so, we are able to provide a lower bound of order* $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$. *Recall that the upper bound stated in Theorem 11 is* $O(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, *and hence our upper and lower bounds match in terms of* Δ $(= \Delta_r + \Delta_p)$.

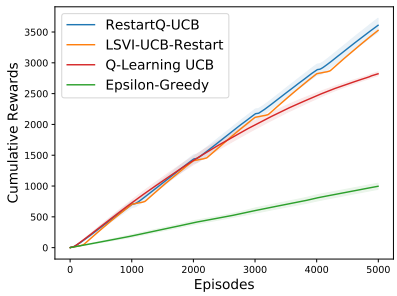
Remark 7 (Tightness of Our Results)**.** *For our setting, we conjecture that the lower bound can be improved. Our current construction of the lower bound relies on a chain of H copies of "JAO MDPs" [96]. The non-stationarity is achieved by changing the transitions abruptly after a fixed time period, and such a change applies* simultaneously *across all H copies of JAO MDPs. One possible direction is to construct the lower bound instances such that the state transition kernel is allowed to vary within the same episode, which we have not taken advantage of. Including this extra ingredient into the construction could potentially lead to a sharper lower bound, and we leave this as future work.*

A useful side result of our proof is the following lower bound for non-stationary RL in the un-discounted setting, which is the same setting as studied in [145], [144] and [22].

[2]The definition of segments is irrelevant to, and should not be confused with, the notion of epochs we previously defined.



(a) Abrupt variations



(b) Gradual variations

Algorithm	Time per episode
RestartQ-UCB	0.102 ms
LSVI-UCB-Restart	57.65 ms
Q-Learning UCB	0.098 ms
Epsilon-Greedy	0.123 ms

(c) Time usage

Figure 3.2: Cumulative rewards of the four algorithms under (a) abrupt variations, and (b) gradual variations, respectively, as well as their (c) time usage. Shaded areas denote the standard deviations of rewards. Note that RestartQ-UCB significantly outperforms Q-Learning UCB and Epsilon-Greedy, and matches LSVI-UCB-Restart while being *much more* time-efficient.

Proposition 1. *Consider a reinforcement learning problem in un-discounted non-stationary MDPs with horizon length T, total variation budget Δ, and maximum MDP diameter D [22]. For any learning algorithm, there exists a non-stationary MDP such that the dynamic regret of the algorithm is at least $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}D^{\frac{2}{3}}T^{\frac{2}{3}})$.*

3.7 Simulations

In this section, we empirically evaluate RestartQ-UCB on reinforcement learning tasks with various types of non-stationarity.

We compare RestartQ-UCB with three baseline algorithms: LSVI-UCB-Restart [147], Q-Learning UCB, and Epsilon-Greedy [183]. LSVI-UCB-Restart is a state-of-the-art non-stationary RL algorithm that combines optimistic least-squares value iteration with periodic restarts. It is originally designed for non-stationary RL in linear MDPs, but in our simulations we reduce it to the tabular case by setting the feature map to be essentially an identity mapping, i.e., the feature dimension is set to be $d = S \times A$. Q-Learning UCB is simply our RestartQ-UCB algorithm with no restart. It is a Q-learning based algorithm that uses upper confidence bounds to guide the exploration. Epsilon-Greedy is also a Q-learning based algorithm with restarts. Compared with RestartQ-UCB, Epsilon-Greedy does not employ a UCB-based bonus term to explicitly force exploration. Instead, it takes the greedy action according to the estimated Q function with a high probability $1 - \varepsilon$, and explores an action from the action set uniformly at random with probability ε.

We evaluate the cumulative rewards of the four algorithms on a variant of a reinforcement learning task named Bidirectional Diabolical Combination Lock [184], [185]. This task is designed to be particularly difficult for *exploration*. At the beginning of each episode, the agent starts at a fixed state. According to its first action, the agent transitions to one of the two paths, or "combination locks", each of length H. Each path is a chain of H states, where the state at the endpoint of each path gives a high reward. At each step on the path, there is only one "correct" action that leads the agent to the next state on the path, while the other $A - 1$ actions lead it to a sinking state that yields a small per-step reward of $\frac{1}{8H}$ ever since. Since we are considering a non-deterministic MDP, each intended transition "succeeds" with probability 0.98; that is, even if the agent takes the correct action at a certain step, there is still a 0.02 probability that it will end in the sinking state. The agent obtains a 0 reward when taking a correct action and gets a $\frac{1}{8H}$ reward at the step when it transitions to the sinking state. Finally, the endpoint state of one path gives a reward of 1, while

the other endpoint only gives a reward of 0.25. As argued in [184], the following properties make this task especially challenging: First, it has sparse high rewards, and uniform exploration only has a A^{-H} probability of reaching a high reward endpoint. Second, it has dense low rewards, and a locally optimal policy will lead to the sinking state quickly. Third, there is no indication which path has the globally optimal reward, and the agent must remember to still visit the other one. Interested readers can refer to Section 5.1 of [184] for detailed descriptions of the task.

We introduce two types of non-stationarity to the Bidirectional Diabolical Combination Lock task, namely *abrupt* variations and *gradual* variations. For abrupt variations, we periodically switch the two high-reward endpoints: One high-reward endpoint gives a reward of 1 at the beginning, and abruptly changes to a reward of 0.25 after a certain number of episodes, and then switches back to the reward of 1 after the same number of episodes. The other high-reward endpoint goes the other way around. For gradual changes, we gradually vary the transition probability at the starting state: At the first episode, one action leads to the first path with 0.98 probability, and to the second path with 0.02 probability. We linearly decrease its probability of leading to the first path and increase its probability to the second path. As a result, at the last episode, this action would lead to the first path with 0.02 probability, and to the second path with 0.98 probability instead. The same is true for the other actions.

For simplicity, we use Hoeffding-based bonus terms in the simulations for RestartQ-UCB. We set $M = 5000, H = 5, S = 10$, and $A = 2$. For abrupt variations, we switch the two high-reward endpoints after every 1000 episodes. The hyper-parameters for each algorithm are optimized individually. For RestartQ-UCB, LSVI-UCB-Restart, and Epsilon-Greedy, we restart the algorithms after every 1000 episodes both for abrupt variations and gradual variations. This is the same frequency as the abrupt variation of the environment (because the restart frequency is optimized as a hyper-parameter), although it turns out that other restart frequencies lead to very similar results. For Epsilon-Greedy, we set the exploration probability to be $\varepsilon = 0.05$. All results are averaged over 30 runs on a laptop with an Intel Core i5-9300H CPU and 16 GB memory.

The cumulative rewards of the four algorithms in the abruptly-changing and gradually-changing environments are shown in Figures 3.2(a) and 3.2(b), respectively. As we can see, RestartQ-UCB outperforms Q-Learning UCB and Epsilon-Greedy under both types of environment variations. For the abruptly-changing environment as an example, RestartQ-UCB achieves 1.36 and 2.52 times of the cumulative rewards of Q-Learning UCB and Epsilon-Greedy, respectively. This demonstrates the importance of both addressing the environment variations (using restarts) and actively exploring the environment (using UCB-based bonus terms) in non-stationary RL. LSVI-UCB-Restart nearly matches the performance of RestartQ-UCB, which is unsurprising because both of them use the restarting strategy and optimistic exploration. Nevertheless, LSVI-UCB-Restart requires a higher time and space complexity. It needs to store all the history information in one epoch and solve a regularized least-squares minimization problem at every time step. This is indeed evidenced by our simulation results (shown in Figure 3.2(c)) that RestartQ-UCB only takes 0.18% of the computation time of LSVI-UCB-Restart.

Remark 8. *The heavy computation in LSVI-UCB-Restart mostly comes from the usage of a high-dimensional feature. In our simulations, we followed Example 2.1 in [164] to convert a linear MDP algorithm to a tabular one, which results in a feature dimension of $d = S \times A$. This is essentially the most efficient feature encoding when no special structure is imposed on the tabular MDP. We believe that designing low-dimensional features for specific MDP instances can possibly reduce the computations for LSVI-UCB-Restart by a large amount, and is an interesting future direction for learning in linear MDPs per se.*

3.8 Application to Multi-Agent RL

In this section, we discuss an application of our non-stationary RL method to multi-agent RL in episodic stochastic games [7], which by nature leads to a non-stationary RL problem from each agent's perspective.

3.8.1 Problem Setup

In general, an N-player episodic stochastic game is defined by a tuple $(\mathcal{N}, H, \mathcal{S}, \{\mathcal{A}^{(i)}\}_{i=1}^N, \{r^{(i)}\}_{i=1}^N, P)$, where (1) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $\mathcal{S}$ is the finite state space; (4) $\mathcal{A}^{(i)}$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r_h^{(i)} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function at step $h \in [H]$ for agent $i \in \mathcal{N}$, where $\mathcal{A} = \times_{i=1}^N \mathcal{A}^{(i)}$; and (6) $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel at step $h \in [H]$, where the next state depends on the current state and the joint actions of all the agents. The game lasts for M episodes, and we let $T = MH$ be the total number of time steps. At each time step (m, h), the agents observe the state $s_h^m \in \mathcal{S}$, and take actions $a_h^{(i),m} \in \mathcal{A}^{(i)}, i \in \mathcal{N}$ simultaneously.[3] We let $a_h^m = (a_h^{(1),m}, \ldots, a_h^{(N),m})$. Agent i receives a reward with an expected value of $r_h^{(i)}(s_h^m, a_h^m)$, and the environment transitions to the next state $s_{h+1}^m \sim P_h(\cdot|s_h^m, a_h^m)$. For each agent i, a policy is a mapping from the time index and state space to (possibly a distribution over) the action space. We denote the set of policies for agent i by $\Pi^{(i)} = \{\pi^{(i)} : [M] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A}^{(i)})\}$. The set of joint policies are denoted by $\Pi = \times_{i=1}^N \Pi^{(i)}$. Each agent seeks to find a policy that maximizes its own reward.

For notational convenience, and without much loss of conceptual generality, we consider two-player games, i.e., $N = 2$. For ease of notations, we consider the problem where we can control the policy of agent 1, while agent 2 is an opponent that is adapting its own policy in an unknown way. Since the two agents play symmetric roles in the problem we study (to be specified later), such a notational simplification is also without loss of generality. Achieving sublinear regret in the face of an arbitrarily changing opponent is known to be computationally hard [152]. Therefore, existing works [152], [153] often focus on a setting where the opponent is only "slowly changing" its policy over time. One such example is when the opponent is using a relatively stable learning algorithm. We also focus on the *decentralized* setting[4], where an agent *cannot* observe the actions and rewards of the other agent. This is generally considered to be a more practical multi-agent RL paradigm, and also more challenging than those that we will compare with in the literature [152], [153].

A joint policy induces a probability measure on the sequence of states and joint actions. For a joint policy $\pi = (\pi^{(1)}, \pi^{(2)}) \in \Pi$, and for each time step $(m, h) \in [M] \times [H]$, state $s \in \mathcal{S}$, we define the state value function *for agent 1* as follows:

$$V_h^{m,\pi}(s) := \mathbb{E}\left[\sum_{h'=h}^{H} r^{(1)}\left(s_{h'}, \pi_{h'}^{(1),m}\left(s_{h'}\right), \pi_{h'}^{(2),m}\left(s_{h'}\right)\right) \mid s_h = s\right].$$

For a joint policy $(\pi^{(1)}, \pi^{(2)})$, we again evaluate the optimality of agent 1's policy $\pi^{(1)}$ in terms of its *dynamic regret*, which compares the agent's policy with the optimal policy of each individual episode in hindsight:

$$\mathcal{R}^{\pi^{(2)}}(\pi^{(1)}, M) := \sum_{m=1}^{M}\left(\sup_{\pi^{(1)\star}} V_1^{m,(\pi^{(1)\star},\pi^{(2)})}(s_1^m) - V_1^{m,(\pi^{(1)},\pi^{(2)})}(s_1^m)\right).$$

[3] Note that we use superscripts in parentheses to index the agents, while a superscript with no parenthesis denotes the index of an episode.

[4] This setting has been studied under various names in the literature, including individual learning [51], decentralized learning [52], [186], online agnostic learning [53], and independent learning [55]. It is also related to the broader category of teams and games with decentralized information structure [56]–[58].

The initial state of each episode s_1^m is again chosen by an oblivious adversary.

3.8.2 Regret Against a Slowly-Changing Opponent

We model the slowly-changing behavior of agent 2 by requiring it to have a low *switching cost* [165], [187]. This is a standard notion in the literature to measure the changing behavior of an RL algorithm. We consider the following definition of the (local) switching cost from [165].

Definition 7. *The switching cost between any pair of policies (π, π') is the number of (h, s) pairs on which π and π' act differently:*

$$n_{switch}(\pi, \pi') := \left|\{(h, s) \in [H] \times \mathcal{S} : \pi_h(s) \neq \pi'_h(s)\}\right|.$$

For a policy trajectory $(\pi^1, \ldots, \pi^M)$ across M episodes, its switching cost is defined as $N_{switch} := \sum_{m=1}^{M} n_{switch}(\pi^m, \pi^{m+1})$.

[165] develops a learning algorithm that achieves a switching cost of $O(SAH^3 \log T)$, while [61] improves the switching cost to $O(SAH^2 \log T)$. For the sake of generality, we characterize the behavior of agent 2 by assuming that the switching cost of its policy trajectory is upper bounded by $O(T^\beta)$ for some $0 < \beta < 1$. Clearly, the two state-of-the-art RL algorithms mentioned above satisfy this upper bound. A direct application of RestartQ-UCB leads to the following result for agent 1:

Theorem 14. *Suppose that the switching cost of agent 2 satisfies $N_{switch} = O(T^\beta)$ for $0 < \beta < 1$. Let agent 1 run the RestartQ-UCB (Hoeffding/Freedman) algorithm. For T large enough, the dynamic regret of agent 1 is upper bounded by $\widetilde{O}(T^{\frac{\beta+2}{3}})$.*

3.8.3 Learning Team-Optimality

Theorem 14 can be readily applied to learning team-optimal policies in "smooth games", which is the setting considered in [152]. This corresponds to the setting where a team of agents learn to collaborate. Before we present our results, a few definitions are in order.

Definition 8. *A two-player stochastic game is called a stochastic team (or simply a* team*) if there exists a reward function $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ such that $r_h^{(i)} = r_h, \forall i \in \{1, 2\}, h \in [H]$.*

Definition 9. *In a two-player team, a joint policy $\pi^\star = (\pi^{(1)\star}, \pi^{(2)\star}) \in \Pi$ is called* team-optimal *if*

$$V_h^{(\pi^{(1)\star}, \pi^{(2)\star})}(s) = \sup_{\pi^{(1)}, \pi^{(2)}} V_h^{(\pi^{(1)}, \pi^{(2)})}(s), \forall s \in \mathcal{S}, h \in [H],$$

where $V_h^{(\pi^{(1)}, \pi^{(2)})}(s) := \mathbb{E}[\sum_{h'=h}^{H} r_{h'}(s_{h'}, \pi_{h'}^{(1)}(s_{h'}), \pi_{h'}^{(2)}(s_{h'})) \mid s_h = s]$ is the value function.

In a stochastic team, the agents share the same objective, and aim to maximize the accumulated reward for the team. Team optimality is achieved when the joint policy of the agents induces the highest possible accumulated reward.

Since we cannot control the behavior of agent 2, its behavior might be sub-optimal and drive us away from team-optimality. To avoid such scenarios, we impose a structural assumption that allows us to quantify the distance from optimality. In particular, we assume that the team is (λ, μ)-smooth, following the definition in [152].

Definition 10. *(Adapted from Definition 1 in [152]) A two-player stochastic team is (λ, μ)-smooth if there exists a pair of policies $(\pi^{(1)\star}, \pi^{(2)\star})$ such that for every policy pair $(\pi^{(1)}, \pi^{(2)})$ and every $h \in [H], s \in \mathcal{S}$:*

$$V_h^{(\pi^{(1)\star}, \pi^{(2)\star})}(s) \geq V_h^{(\pi^{(1)}, \pi^{(2)})}(s),$$
$$V_h^{(\pi^{(1)\star}, \pi^{(2)})}(s) \geq \lambda \cdot V_h^{(\pi^{(1)\star}, \pi^{(2)\star})}(s) - \mu \cdot V^{(\pi^{(1)}, \pi^{(2)})}(s).$$

The (λ, μ)-smoothness ensures that agent 2's sub-optimal behavior only has a bounded negative impact on the joint value. Our definition of smoothness is adapted from [152], where the infinite-horizon average-reward setting is considered. We adapt it to the finite-horizon case. This notion of smoothness is motivated by the definition of smooth games in [115], [188], as stated in [152].

Applying our RestartQ-UCB algorithm for agent 1 would lead to the following theorem, which implies that the time-average return of the agents converges to a $\frac{\lambda}{1+\mu}$ factor of the team-optimal value as T grows. This is the same factor as has been achieved in [152].

Theorem 15. *Let $\pi^{(2)}$ denote the policy of agent 2, and suppose that the switching cost of agent 2 satisfies $N_{switch} = O(T^\beta)$ for $0 < \beta < 1$. Assume that the team problem is (λ, μ)-smooth. Let agent 1 run the RestartQ-UCB algorithm, and let $\pi^{(1)}$ denote its induced policy. For T large enough, the return of the algorithm is lower bounded by:*

$$\sum_{m=1}^{M} V_1^{(\pi^{(1)}, \pi^{(2)})}(s_1^m) \geq \frac{\lambda}{1+\mu} \left[\sum_{m=1}^{M} V_1^{(\pi^{(1)\star}, \pi^{(2)\star})}(s_1^m) - \widetilde{O}(T^{\frac{\beta+2}{3}}) \right].$$

Proof. We first show that when the switching cost of agent 2 satisfies $N_{\text{switch}} = O(T^\beta)$ for $0 < \beta < 1$, the dynamic regret of agent 1 is upper bounded by $\widetilde{O}(T^{\frac{\beta+2}{3}})$. To see this, notice that from the perspective of agent 1, the environment is non-stationary due to the fact that agent 2 is changing its policy over time. Since the switching cost of agent 2 is upper bounded by $O(T^\beta)$, by the definitions of Δ_r and Δ_p in Section 3.2, we know that the variation of the environment from the perspective of agent 1 is upper bounded by $O(T^\beta)$. Substituting the value of Δ with $O(T^\beta)$ in Theorem 10 or Theorem 11 leads to the desired result.

From the (λ, μ)-smoothness of the MDP, it follows that

$$\lambda \cdot V_h^{(\pi^{(1)\star}, \pi^{(2)\star})}(s) - \mu \cdot V_h^{(\pi^{(1)}, \pi^{(2)})}(s) \leq V_h^{(\pi^{(1)\star}, \pi^{(2)})}(s), \forall s \in \mathcal{S}, h \in [H].$$

Therefore, it holds that

$$\begin{aligned}
&\sum_{m=1}^{M} \left(\lambda \cdot V_1^{(\pi^{(1)\star}, \pi^{(2)\star})}(s_1^m) - (1+\mu) \cdot V_1^{(\pi^{(1)}, \pi^{(2)})}(s_1^m) \right) \\
\leq &\sum_{m=1}^{M} \left(V_1^{(\pi^{(1)\star}, \pi^{(2)})}(s_1^m) - V_1^{(\pi^{(1)}, \pi^{(2)})}(s_1^m) \right) \\
\leq &\sum_{m=1}^{M} \left(\sup_{\pi^{(1)\star}} V_1^{(\pi^{(1)\star}, \pi^{(2)})}(s_1^m) - V_1^{(\pi^{(1)}, \pi^{(2)})}(s_1^m) \right) \\
= &\mathcal{R}^{\pi^{(2)}}(\pi^{(1)}, M) = \widetilde{O}(T^{\frac{\beta+2}{3}}),
\end{aligned}$$

where the last step follows from the $\widetilde{O}(T^{\frac{\beta+2}{3}})$ dynamic regret bound of agent 1, as we discussed above. Rearranging the terms leads to the desired result. □

Remark 9. *(Comparison with [152] and [153].) It might first appear to the reader that our regret guarantee is weaker than the bounds of* $O(T^{\max\{1-\frac{3}{7}\alpha,\frac{1}{4}\}})$ *and* $O(T^{\max\{1-\frac{3}{2}\alpha,0\}})$ *given in [152] and [153], respectively, where* α *can be essentially translated*[5] *to* $1-\beta$*. However, we would like to emphasize that our setting significantly generalizes the other two works and is inherently more challenging due to the following facts: First, we are considering a* learning *problem where the transition and reward functions are unknown; the other two works essentially consider* planning *with a known MDP model. Second, we are using the more challenging* dynamic regret *as a measure of optimality, while the other two use the static regret. Third, we study* decentralized *learning, where the agents cannot observe the actions and rewards of each other; the algorithms proposed in the other two works critically rely on the* observation *of one agent on the other agent's* policies.

Remark 10. *(Significance of model-freeness.) Decentralized multi-agent RL is generally only possible with model-free approaches (see, e.g., [52], [53], [55]); model-based methods proceed by explicitly estimating the transition and reward functions, which crucially relies on observing the other agents' actions. This further demonstrates the flexibility and significance of model-free methods, when one addresses the non-stationarity issues in multi-agent RL through the lens of non-stationary RL.*

3.9 Application to Inventory Control Across Related Products

In this section, we discuss the application of our non-stationary RL algorithm to the problem of inventory control across related products. Different from conventional inventory control problems (e.g., [136]) that only consider one product, we investigate the case where a sequence of related products are being sold, and the products share similar but different demand distributions. This is motivated by the sequential launch of related products (e.g., the line of iPhone) that allows us to leverage experience from past products to inform inventory management for future ones. Following [149], [189] (who only consider a single product being sold), we focus on the setting of zero lead time, fixed cost, and lost sales.

3.9.1 Problem Setup

The inventory control problem has M episodes, representing M different but related products. Each episode/product lasts for H time steps.[6] For each time step $h \in [H]$ of an episode $m \in [M]$, the following sequence of events happens in order:

1. The seller observes her stock level $s_h^m \geq 0$ for product m at the beginning of time step h, and decides on the quantity $a_h^m \geq 0$ to order.

2. If $a_h^m > 0$, the order arrives immediately, and the seller's stock level becomes $s_h^m + a_h^m$. The seller pays a fixed cost f and a c per-unit ordering cost.

3. The random demand X_h^m is realized. The seller only observes the actual sales quantity, or *censored demand* $Y_h^m = \min\{X_h^m, s_h^m + a_h^m\}$. She will not know the actual demand if $X_h^m \geq s_h^m + a_h^m$. Following prior works [22], [184] and [189], we assume that the demands X_h^m are independent random variables over m, but they do not necessarily follow identical distributions since we consider different products across the episodes.

[5] The other two works model the slowly-changing behavior of agent 2 using the small "policy change magnitude" criterion. Our setting is in this sense not completely comparable with theirs.

[6] We assume for simplicity that the life cycle of each product is of the same length.

4. All unfulfilled demands are permanently lost and incur a per-unit lost sales cost p. Excess inventory incurs a per-unit holding cost q. The total cost at step h can be expressed as

$$C_h^m(s_h^m, a_h^m) = f \cdot \mathbb{I}[a_h^m > 0] + c \cdot a_h^m + p \cdot [X_h^m - s_h^m - a_h^m]^+ + q \cdot [s_h^m + a_h^m - X_h^m]^+ .$$

5. The inventory carried over to the next step $h+1$ is $s_{h+1}^m = [s_h^m + a_h^m - X_h^m]^+$.

Following [149], [189], we assume that the seller has a finite storage capacity S, in the sense that she can hold at most $S-1$ units of inventory at any time. The seller's objective is to minimize her cumulative cost $\sum_{m=1}^M \sum_{h=1}^H C_h^m(s_h^m, a_h^m)$. At the end of each episode, as a product is reaching the end of its life cycle, we assume for simplicity that the storage is emptied at no cost. Such an inventory control problem can be easily formulated as an instance of the non-stationary RL model that we defined in Section 3.2. Concretely, we treat the stock level s_h^m at the beginning of each time step as the state of the environment, and regard the order quantity a_h^m as the action at the corresponding time step. Consequently, we define the state space of the problem as $\mathcal{S} = \{0, 1, \ldots, S-1\}$, and the state-dependent action space as $\mathcal{A}_s = \{0, 1, \ldots, S-1-s\}$. One can verify that Algorithm 14 and its analysis easily generalize to state-dependent action spaces.

The reward function of the non-stationary MDP is defined as $R_h^m(s_h^m, a_h^m) = -C_h^m(s_h^m, a_h^m)$, and we let $r_h^m(s_h^m, a_h^m) = \mathbb{E}[R_h^m(s_h^m, a_h^m)]$ be the expected value of the reward. For any $s_h^m, s_{h+1}^m \in \mathcal{S}$ and $a_h^m \in \mathcal{A}_s$, we define the state transition function as

$$P_h^m(s_{h+1}^m \mid s_h^m, a_h^m) = \mathbb{P}\left(s_h^m + a_h^m - \min\{s_h^m + a_h^m, X_h^m\} = s_{h+1}^m\right).$$

Our definitions of the policy π, the value function $V_h^{m,\pi}$, the state-action value function $Q_h^{m,\pi}$, as well as the optimal policy $\pi^\star$ and its corresponding value functions $V_h^{m,\star}, Q_h^{m,\star}$ directly carry over from Section 3.2 to this problem instance, and we do not repeat such definitions here for simplicity. The variation budget Δ is also defined in the same way as in Section 3.2, which captures the differences in the products' demand distributions for this problem. The dynamic regret of the agent's policy is defined analogously as

$$\mathcal{R}(\pi, M) = \sum_{m=1}^M \left(V_1^{m,\star}(s_1^m) - V_1^{m,\pi}(s_1^m)\right).$$

3.9.2 Implementation of RestartQ-UCB

Notably, one major difference between the inventory control problem we considered in Section 3.9.1 and our non-stationary MDP formulation in Section 3.2 is that due to demand censoring, the seller cannot calculate the actual cost $C_h^m(s_h^m, a_h^m)$, and hence the immediate reward $R_h^m(s_h^m, a_h^m)$ is also not observable. Nevertheless, we will show that one can bypass such an issue by using a pseudo-reward technique, which was originally introduced for a stationary problem [21]. Specifically, for every time step $h \in [H]$ in episode $m \in [M]$, and for every state $s \in \mathcal{S}$ and action $a \in \mathcal{A}_s$, we define the pseudo-reward as

$$R_h^{m,\text{pseudo}}(s, a) := R_h^m(s, a) + p \cdot X_h^m = -f \cdot \mathbb{I}[a > 0] - c \cdot a - q \cdot [s + a - Y_h^m]^+ + p \cdot Y_h^m,$$

where we recall that the censored demand $Y_h^m = \min\{X_h^m, s + a\}$ is perfectly observable. Similarly, we can also define the mean pseudo-reward as

$$r_h^{m,\text{pseudo}}(s, a) := \mathbb{E}[R_h^{m,\text{pseudo}}(s, a)] = \mathbb{E}[R_h^m(s, a) + p \cdot X_h^m] = r_h^m(s, a) + p \cdot \mathbb{E}[X_h^m].$$

Therefore, the mean pseudo-reward can be considered as shifting the mean reward function uniformly by an amount of $p \cdot \mathbb{E}[X_h^m]$. Without loss of generality, we normalize the pseudo-reward to the range $[0,1]$. We use the tuple $\mathcal{M} = \left\{\mathcal{S}, \mathcal{A}, H, \{P_h^m\}_{m\in[M],h\in[H]}, \{r_h^m\}_{m\in[M],h\in[H]}\right\}$ to denote the non-stationary MDP with respect to the original reward function, and let $\mathcal{M}^{\text{pseudo}} = \left\{\mathcal{S}, \mathcal{A}, H, \{P_h^m\}_{m\in[M],h\in[H]}, \{r_h^{m,\text{pseudo}}\}_{m\in[M],h\in[H]}\right\}$ be the one corresponding to the pseudo-reward. We further define $\pi^{\star,\text{pseudo}}$ to be the (episode-wise) optimal policy for $\mathcal{M}^{\text{pseudo}}$, and let $V_h^{m,\star,\text{pseudo}}$ and $Q_h^{m,\star,\text{pseudo}}$, respectively, be the corresponding value function and state-action value function.

Since only the pseudo-reward is observable, we can only apply our RestartQ-UCB algorithm to $\mathcal{M}^{\text{pseudo}}$ rather than $\mathcal{M}$. A natural question, then, is whether we can generalize the performance guarantee from $\mathcal{M}^{\text{pseudo}}$ to $\mathcal{M}$. Interestingly, the following result (adapted from [21]) shows that, for any (possibly non-Markovian) policy π induced by Algorithm 14, the dynamic regret on $\mathcal{M}^{\text{pseudo}}$ and $\mathcal{M}$ are equal.

Lemma 23. *(Adapted from Lemma 3.1 in [21]). Let $\mathcal{F}_h^m$ be the set of all historical information collected up to the beginning of time step h of episode m. Let π be the (possibly non-Markovian) policy induced by Algorithm 14, such that $\pi_h^m(s_h^m, \mathcal{F}_h^m)$ maps the state and history to a distribution over the action space. Then, π incurs the same dynamic regret on $\mathcal{M}$ and $\mathcal{M}^{pseudo}$:*

$$\sum_{m=1}^{M} \left(V_1^{m,\star}(s_1^m) - V_1^{m,\pi}(s_1^m)\right) = \sum_{m=1}^{M} \left(V_1^{m,\star,pseudo}(s_1^m) - V_1^{m,\pi,pseudo}(s_1^m)\right).$$

Proof. First, we show that

$$\sum_{m=1}^{M} \left(V_1^{m,\star}(s_1^m) - V_1^{m,\star,\text{pseudo}}(s_1^m)\right) = -\sum_{m=1}^{M}\sum_{h=1}^{H} p \cdot \mathbb{E}[X_h^m]. \tag{3.4}$$

Recall that the pseudo-reward is constructed by uniformly shifting the reward function by an amount of $p \cdot \mathbb{E}[X_h^m]$. Since the difference between the reward and the pseudo-reward does not depend on the action taken, for any realization of the demands $\{X_h^m\}_{m\in[M],h\in[H]}$, the optimal policies $\pi^\star$ and $\pi^{\star,\text{pseudo}}$ induce the same distribution over the action space, which in turn leads to the same distribution over state-action trajectories. We can hence conclude that (3.4) holds.

Similarly, one can show by induction that for any realization of the demands $\{X_h^m\}_{m\in[M],h\in[H]}$, Algorithm 14 also induces the same distribution of action sequences on $\mathcal{M}$ and $\mathcal{M}^{\text{pseudo}}$. This leads us to

$$\sum_{m=1}^{M} \left(V_1^{m,\pi}(s_1^m) - V_1^{m,\pi,\text{pseudo}}(s_1^m)\right) = -\sum_{m=1}^{M}\sum_{h=1}^{H} p \cdot \mathbb{E}[X_h^m]. \tag{3.5}$$

Combining (3.4) and (3.5) yields the desired result. □

Together with Theorem 9, we obtain the following dynamic regret bound for running Algorithm 14 on the inventory control problem across related products.

Theorem 16. *For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0,1)$, with probability at least $1-\delta$, the dynamic regret of running Algorithm 14 on the inventory control problem formulated in Section 3.9.1 with pseudo-rewards and Freedman bonuses is bounded by $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors of S, A, T and $1/\delta$.*

Remark 11 (Comparison with [22])**.** *Although both our work and [22] consider applications in inventory control and utilizes techniques from [21], the foci are quite different. In [22], the authors only study single*

product inventory, whereas in contrast, our work studies the setting where there is a sequence of related, but different *products.*

Specifically, in [22], variation budget has been defined with respect to demand changes within a single product selling horizon, and the corresponding regret upper bound scales with this budget, whereas in ours no constraint has been put to limit the demand changes within a single product's selling horizon (an episode), and the variation budget captures the difference across products. This is similar to a meta/transfer learning setting where the goal is to leverage data obtained from inventory learning for similar products to accelerate inventory learning for the new product.

Moreover, as discussed in Section 3.1, a direct application of the results in [22] to this setting may lead to a worse regret upper bound.

Remark 12. *Our results can be extended to a multi-product inventory control problem with a warehouse-capacity constraint, similar to the setting studied in [190]. Specifically, we have an episodic setting with n products and M episodes, where each episode lasts for H time steps. For each time step $h \in [H]$ of an episode $m \in [M]$, a demand is specified for every product $i \in [n]$. In our non-stationary formulation, the demands need not follow identical distributions over time. An overall warehouse capacity constraint is also imposed on the total number of products simultaneously in the inventory. At each time step, the seller observes the stock level and decides on the quantity to order for each product at a certain per-unit ordering cost. Unfulfilled demands are permanently lost and incur a per-unit lost sales cost. Excess inventory also incurs a per-unit holding cost. The seller's objective is to minimize the cumulative cost. Such a multi-product problem can also be cast as an MDP, where we define the stock levels of all products to be the state of the environment, and define the joint ordering quantity across all products as the action of each step. We also let the action space be state-dependent to handle the joint capacity constraint; in particular, an action is considered invalid at a certain state if the corresponding ordering quantity causes the stock levels to exceed the warehouse capacity. Applying Algorithm 14 to such a non-stationary multi-product problem leads to the same dynamic regret bound as in Theorem 16, though the state space should now be interpreted as the possible combinations of stock levels across all products that do not exceed the warehouse capacity, which is significantly larger than the single-product case. The same is true for the action space. A final remark is that the multi-product formulation above does not consider upgrading [191], the situation where a high-quality product is used to serve the demand of a lower-quality one that has been sold out. Upgrading adds an additional element of difficulty to the decision-making process, as the seller now needs to consider the ordering and upgrading decisions jointly. We leave the treatment of such a more intricate scenario to our future work.*

3.10 Proofs of the Technical Lemmas

3.10.1 Proof of Lemma 20

Proof. For each $d \in [D]$, define $\Delta_r^{(d)}$ to be the *local variation* of the mean reward function within epoch d. By definition, we have $\sum_{d=1}^{D} \Delta_r^{(d)} \le \Delta_r$. Further, for each $d \in [D]$ and $h \in [H]$, define $\Delta_{r,h}^{(d)}$ to be the variation of the mean reward at step h in epoch d, i.e.,

$$\Delta_{r,h}^{(d)} := \sum_{m=(d-1)K+1}^{\min\{dK,M\}-1} \sup_{s,a} \left| r_h^m(s,a) - r_h^{m+1}(s,a) \right|.$$

It also holds that $\sum_{h=1}^{H} \Delta_{r,h}^{(d)} = \Delta_r^{(d)}$ by definition. Define $\Delta_p^{(d)}$ and $\Delta_{p,h}^{(d)}$ analogously.

In the following, we will prove a stronger statement: $\left|Q_h^{k_1,\star}(s,a) - Q_h^{k_2,\star}(s,a)\right| \le \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h}^{H} \Delta_{p,h'}^{(1)}$, which implies the statement of the lemma because $\sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} \le \Delta_r^{(1)}$ and $\sum_{h'=h}^{H} \Delta_{p,h'}^{(1)} \le \Delta_p^{(1)}$ by definition. Our proof relies on backward induction on h. First, the statement holds for $h = H$ because for any (s,a), by definition

$$\begin{aligned}\left|Q_H^{k_1,\star}(s,a) - Q_H^{k_2,\star}(s,a)\right| &= \left|r_H^{k_1}(s,a) - r_H^{k_2}(s,a)\right| \le \sum_{k=k_1}^{k_2-1} \left|r_H^{k+1}(s,a) - r_H^{k}(s,a)\right| \\ &\le \sum_{k=1}^{K-1} \left|r_H^{k+1}(s,a) - r_H^{k}(s,a)\right| \le \Delta_{r,H}^{(1)}, \end{aligned} \tag{3.6}$$

where we have used the triangle inequality. Now suppose the statement holds for $h+1$; by the Bellman optimality equation,

$$\begin{aligned} & Q_h^{k_1,\star}(s,a) - Q_h^{k_2,\star}(s,a) \\ =& P_h^{k_1} V_{h+1}^{k_1,\star}(s,a) - P_h^{k_2} V_{h+1}^{k_2,\star}(s,a) + r_h^{k_1}(s,a) - r_h^{k_2}(s,a) \\ \le& P_h^{k_1} V_{h+1}^{k_1,\star}(s,a) - P_h^{k_2} V_{h+1}^{k_2,\star}(s,a) + \Delta_{r,h}^{(1)} \qquad (3.7) \\ =& \sum_{s'\in\mathcal{S}} P_h^{k_1}(s' \mid s,a) V_{h+1}^{k_1,\star}(s') - \sum_{s'\in\mathcal{S}} P_h^{k_2}(s' \mid s,a) V_{h+1}^{k_2,\star}(s') + \Delta_{r,h}^{(1)} \\ =& \sum_{s'\in\mathcal{S}} \left(P_h^{k_1}(s' \mid s,a) Q_{h+1}^{k_1,\star}(s', \pi_{h+1}^{k_1,\star}(s')) - P_h^{k_2}(s' \mid s,a) Q_{h+1}^{k_2,\star}(s', \pi_{h+1}^{k_2,\star}(s')) \right) + \Delta_{r,h}^{(1)}, \qquad (3.8) \end{aligned}$$

where inequality (3.7) holds due to a similar reasoning as in (3.6), and in (3.8) $\pi^{k_1,\star}$ and $\pi^{k_2,\star}$ denote the optimal policy in episodes k_1 and k_2, respectively. Then by our induction hypothesis on $h+1$, for any $s' \in \mathcal{S}$,

$$\begin{aligned} Q_{h+1}^{k_1,\star}(s', \pi_{h+1}^{k_1,\star}(s')) \le& Q_{h+1}^{k_2,\star}(s', \pi_{h+1}^{k_1,\star}(s')) + \sum_{h'=h+1}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h+1}^{H} \Delta_{p,h'}^{(1)} \\ \le& Q_{h+1}^{k_2,\star}(s', \pi_{h+1}^{k_2,\star}(s')) + \sum_{h'=h+1}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h+1}^{H} \Delta_{p,h'}^{(1)}, \end{aligned} \tag{3.9}$$

where inequality (3.9) is due to the optimality of the policy $\pi^{k_2,\star}$ in episode k_2 over $\pi^{k_1,\star}$. Then,

$$\begin{aligned} & Q_h^{k_1,\star}(s,a) - Q_h^{k_2,\star}(s,a) \\ \le& \sum_{s'\in\mathcal{S}} (P_h^{k_1}(s' \mid s,a) - P_h^{k_2}(s' \mid s,a)) Q_{h+1}^{k_2,\star}(s', \pi_{h+1}^{k_2,\star}(s')) + \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h+1}^{H} \Delta_{p,h'}^{(1)} \\ \le& \left\| P_h^{k_1}(\cdot|s,a) - P_h^{k_2}(\cdot|s,a) \right\|_1 \left\| Q_{h+1}^{k_2,\star}(\cdot, \pi_{h+1}^{k_2,\star}(\cdot)) \right\|_\infty + \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h+1}^{H} \Delta_{p,h'}^{(1)} \qquad (3.10) \\ \le& \Delta_{p,h}^{(1)}(H-h) + \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h+1}^{H} \Delta_{p,h'}^{(1)} \qquad (3.11) \\ \le& \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H \sum_{h'=h}^{H} \Delta_{p,h'}^{(1)}, \end{aligned}$$

where (3.10) is by Hölder's inequality, and (3.11) is by the definition of $\Delta_{p,h}^{(1)}$ and by the definition of optimal Q-values that $Q_{h+1}^{k_2,\star}(s,a) \leq H-h, \forall (s,a) \in \mathcal{S}\times\mathcal{A}$. Repeating a similar process gives us $Q_h^{k_2,\star}(s,a) - Q_h^{k_1,\star}(s,a) \leq \sum_{h'=h}^{H} \Delta_{r,h'}^{(1)} + H\sum_{h'=h}^{H}\Delta_{p,h'}^{(1)}$. This completes our proof. □

3.10.2 Proof of Lemma 21

Proof. It should be clear from the way we update $Q_h(s,a)$ that $Q_h^k(s,a)$ is monotonically decreasing in k. We now prove $Q_h^{k,\star}(s,a) \leq Q_h^{k+1}(s,a)$ for all s,a,h,k by induction on k. First, it holds for $k=1$ by our initialization of $Q_h(s,a)$. For $k \geq 2$, now suppose $Q_h^{j,\star}(s,a) \leq Q_h^{j+1}(s,a) \leq Q_h^j(s,a)$ for all s,a,h and $1 \leq j \leq k$. For a fixed triple (s,a,h), we consider the following two cases.

Case 1: $Q_h(s,a)$ is updated in episode k. Then with probability at least $1-2\delta$,

$$
\begin{aligned}
Q_h^{k+1}(s,a) =& \frac{\check{r}_h(s,a)}{\check{N}_h^k(s,a)} + \frac{\check{v}_h(s,a)}{\check{N}_h^k(s,a)} + b_h^k + 2b_\Delta \\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} V_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i}) + \sqrt{\frac{H^2}{\check{n}}\iota} + \sqrt{\frac{\iota}{\check{n}}} + 2b_\Delta && (3.12)\\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} P_h^{\check{l}_i} V_{h+1}^{\check{l}_i,\star}(s,a) + \sqrt{\frac{\iota}{\check{n}}} + 2b_\Delta && (3.13)\\
=& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} \left(Q_h^{\check{l}_i,\star}(s,a) - r_h^{\check{l}_i}(s,a)\right) + \sqrt{\frac{\iota}{\check{n}}} + 2b_\Delta && (3.14)\\
\geq& Q_h^{k,\star}(s,a) + b_\Delta. && (3.15)
\end{aligned}
$$

Inequality (3.12) is by the induction hypothesis that $Q_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i},a) \geq Q_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i},a), \forall a \in \mathcal{A}$, and hence $V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) \geq V_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i})$. Inequality (3.13) follows from the Azuma-Hoeffding inequality. (3.14) uses the Bellman optimality equation. Inequality (3.15) is by the Hoeffding's inequality that $\frac{1}{\check{n}}\left(\sum_{i=1}^{\check{n}} r_h^{\check{l}_i}(s,a) - \check{r}_h(s,a)\right) \leq \sqrt{\frac{\iota}{\check{n}}}$ with high probability, and by Lemma 20 that $Q_h^{\check{l}_i,\star}(s,a) + b_\Delta \geq Q_h^{k,\star}(s,a)$. According to the monotonicity of $Q_h^k(s,a)$, we know that $Q_h^{k,\star}(s,a) \leq Q_h^{k+1}(s,a) \leq Q_h^k(s,a)$. In fact, we have proved the stronger statement $Q_h^{k+1}(s,a) \geq Q_h^{k,\star}(s,a) + b_\Delta$ that will be useful in Case 2 below.

Case 2: $Q_h(s,a)$ is not updated in episode k. Then there are two possibilities:

1. If $Q_h(s,a)$ has never been updated from episode 1 to episode k: It is easy to see that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^1(s,a) = H-h+1 \geq Q_h^{k,\star}(s,a)$ holds.

2. If $Q_h(s,a)$ has been updated at least once from episode 1 to episode k: Let j be the index of the latest episode that $Q_h(s,a)$ was updated. Then, from our induction hypothesis and Case 1, we know that $Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) + b_\Delta$. Since $Q_h(s,a)$ has not been updated from episode $j+1$ to episode k, we know that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) + b_\Delta \geq Q_h^{k,\star}(s,a)$, where the last inequality holds because of Lemma 20.

A union bound over all time steps completes our proof. □

3.10.3 Proof of Lemma 22

Proof. This proof follows a similar structure as the proof of Lemma 21. It should be clear from the way we update $Q_h(s,a)$ that $Q_h^k(s,a)$ is monotonically decreasing in k. We now prove $Q_h^{k,\star}(s,a) - 2(H-h+1)b_\Delta \leq$

$Q_h^{k+1}(s,a)$ for all s,a,h,k by induction on k. First, it holds for $k=1$ by our initialization of $Q_h(s,a)$. For $k \geq 2$, now suppose that $Q_h^{j,\star}(s,a) - 2(H-h+1)b_\Delta \leq Q_h^{j+1}(s,a) \leq Q_h^j(s,a)$ for all s,a,h and $1 \leq j \leq k$. For a fixed triple (s,a,h), we consider the following two cases.

Case 1: $Q_h(s,a)$ is updated in episode k. Then, with probability at least $1-2\delta$,

$$\begin{aligned}
Q_h^{k+1}(s,a) =& \frac{\check{r}_h(s,a)}{\check{N}_h^k(s,a)} + \frac{\check{v}_h(s,a)}{\check{N}_h^k(s,a)} + b_h^k \\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} V_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i}) - 2(H-h)b_\Delta + \sqrt{\frac{H^2}{\check{n}}\iota} + \sqrt{\frac{\iota}{\check{n}}} && (3.16) \\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} P_h^{\check{l}_i} V_{h+1}^{\check{l}_i,\star}(s,a) + \sqrt{\frac{\iota}{\check{n}}} - 2(H-h)b_\Delta && (3.17) \\
=& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(Q_h^{\check{l}_i,\star}(s,a) - r_h^{\check{l}_i}(s,a) \right) + \sqrt{\frac{\iota}{\check{n}}} - 2(H-h)b_\Delta && (3.18) \\
\geq& Q_h^{k,\star}(s,a) - b_\Delta - 2(H-h)b_\Delta. && (3.19)
\end{aligned}$$

Inequality (3.16) is by the induction hypothesis that $Q_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i},a) \geq Q_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i},a) - 2(H-h)b_\Delta, \forall a \in \mathcal{A}$, and hence $V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) \geq V_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i}) - 2(H-h)b_\Delta$. Inequality (3.17) follows from the Azuma-Hoeffding inequality. (3.18) uses the Bellman optimality equation. Inequality (3.19) is by the Hoeffding's inequality that $\frac{1}{\check{n}}\left(\sum_{i=1}^{\check{n}} r_h^{\check{l}_i}(s,a) - \check{r}_h(s,a)\right) \leq \sqrt{\frac{\iota}{\check{n}}}$ with high probability, and by Lemma 20 that $Q_h^{\check{l}_i,\star}(s,a) \geq Q_h^{k,\star}(s,a) - b_\Delta$. According to the monotonicity of $Q_h^k(s,a)$, we know that $Q_h^{k,\star}(s,a) - 2(H-h+1)b_\Delta \leq Q_h^{k+1}(s,a) \leq Q_h^k(s,a)$. In fact, we have proved the stronger statement $Q_h^{k+1}(s,a) \geq Q_h^{k,\star}(s,a) - b_\Delta - 2(H-h)b_\Delta$ that will be useful in Case 2 below.

Case 2: $Q_h(s,a)$ is not updated in episode k. Then there are two possibilities:

1. If $Q_h(s,a)$ has never been updated from episode 1 to episode k: It is easy to see that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^1(s,a) = H-h+1 \geq Q_h^{k,\star}(s,a) - 2(H-h+1)b_\Delta$ holds.
2. If $Q_h(s,a)$ has been updated at least once from episode 1 to episode k: Let j be the index of the latest episode that $Q_h(s,a)$ was updated. Then, from our induction hypothesis and Case 1, we know that $Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) - b_\Delta - 2(H-h)b_\Delta$. Since $Q_h(s,a)$ has not been updated from episode $j+1$ to episode k, we know that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) - b_\Delta - 2(H-h)b_\Delta \geq Q_h^{k,\star}(s,a) - 2(H-h+1)b_\Delta$, where the last inequality holds because of Lemma 20.

A union bound over all time steps completes our proof. □

3.10.4 Proof of Proposition 2

In the following, we will bound each term in Λ_{h+1}^k separately in a series of lemmas.

Lemma 24. *With probability* 1*, we have that*

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}(3b_h^k + 5b_\Delta) \leq O(\sqrt{SAKH^5\iota} + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)}).$$

Proof. First, by the definition of b_Δ, it is easy to see that $\sum_{h=1}^H \sum_{k=1}^K (1+\frac{1}{H})^{h-1} 5b_\Delta \leq \sum_{h=1}^H \sum_{k=1}^K O(\Delta_r^{(1)} + H\Delta_p^{(1)}) \leq O(KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)})$. Recall our definition that $e_1 = H$ and $e_{i+1} = \left\lfloor (1+\frac{1}{H})e_i \right\rfloor, i \geq 1$. For a

fixed $h \in [H]$, since $H^2 \geq 1$,

$$\begin{aligned}
&\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}3b_h^k \leq \sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}12\sqrt{\frac{H^2}{\check{N}_h^k(s_h^k,a_h^k)}\iota} \\
=&12H\sqrt{\iota}\sum_{s,a}\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}\sqrt{\frac{1}{e_j}}\sum_{k=1}^{K}\mathbb{1}\left[(s_h^k,a_h^k)=(s,a),\check{N}_h^k(s_h^k,a_h^k)=e_j\right] \\
=&12H\sqrt{\iota}\sum_{s,a}\sum_{j\geq 1}(1+\frac{1}{H})^{h-1}w(s,a,j)\sqrt{\frac{1}{e_j}},
\end{aligned}$$

where $w(s,a,j) := \sum_{k=1}^{K}\mathbb{1}\left[(s_h^k,a_h^k)=(s,a),\check{N}_h^k(s_h^k,a_h^k)=e_j\right]$, and $w(s,a) := \sum_{j\geq 1}w(s,a,j)$. We then know that $\sum_{s,a}w(s,a) = K$. For a fixed (s,a), let us now find an upper bound of j, denoted as J. Since each stage is $(1+\frac{1}{H})$ times longer than the previous stage, we know for $1 \leq j \leq J$, $w(s,a,j) = \sum_{k=1}^{K}\mathbb{1}\left[(s_h^k,a_h^k)=(s,a),\check{N}_h^k(s_h^k,a_h^k)=e_j\right] = \left\lfloor(1+\frac{1}{H})e_j\right\rfloor$. From $\sum_{j=1}^{J}w(s,a,j) = w(s,a)$, we get $e_J \leq (1+\frac{1}{H})^{J-1} \leq \frac{10}{1+\frac{1}{H}}\frac{w(s,a)}{H}$. Therefore,

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

Finally, by the Cauchy-Schwartz inequality, we have

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}3b_h^k = O\left(H^2\sqrt{\iota}\sum_{s,a}\sum_{j\geq 1}w(s,a,j)\sqrt{\frac{1}{e_j}}\right) \leq \sqrt{SAKH^5\iota}.$$

Combining the bounds for b_h^k and b_Δ completes the proof. □

Lemma 25. *With probability at least $1-\delta$, it holds that*

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\phi_{h+1}^k \leq O(\sqrt{KH^3\iota} + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)}).$$

Proof. We have that

$$\begin{aligned}
&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\phi_{h+1}^k \\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^{\check{l}_i,\star} - V_{h+1}^{k,\pi}\right)(s_h^k,a_h^k) \\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^{\check{l}_i,\star} - V_{h+1}^{k,\star} + V_{h+1}^{k,\star} - V_{h+1}^{k,\pi}\right)(s_h^k,a_h^k) \\
\leq&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}2b_\Delta + \sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^{k,\star} - V_{h+1}^{k,\pi}\right)(s_h^k,a_h^k),
\end{aligned}$$

where the last inequality follows from Lemma 20 and the definition of b_Δ. From the proof of Lemma 24, we

know that the first term can be bounded as

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}2b_{\Delta}\leq O(KH\Delta_r^{(1)}+KH^2\Delta_p^{(1)}).$$

Further, the second term is bounded by the Azuma-Hoeffding inequality as

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\left(P_h^k-\mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^{k,\star}-V_{h+1}^{k,\pi}\right)(s_h^k,a_h^k)\leq O(\sqrt{KH^3\iota}).$$

Combining the two terms completes the proof. □

Lemma 26. *With probability at least* $1-(KH+1)\delta$*, it holds that*

$$\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\xi_{h+1}^k\leq O(\sqrt{SAKH^3\iota}+KH^2\Delta_p^{(1)}).$$

Proof. We have that

$$\begin{aligned}
&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\xi_{h+1}^k\\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^k-\mathbf{e}_{s_{h+1}^{\check{l}_i}}\right)\left(V_{h+1}^{\check{l}_i}-V_{h+1}^{\check{l}_i,\star}\right)(s_h^k,a_h^k)\\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^k-P_h^{\check{l}_i}+P_h^{\check{l}_i}-\mathbf{e}_{s_{h+1}^{\check{l}_i}}\right)\left(V_{h+1}^{\check{l}_i}-V_{h+1}^{\check{l}_i,\star}\right)(s_h^k,a_h^k)\\
\leq&O(KH^2\Delta_p^{(1)})+\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^{\check{l}_i}-\mathbf{e}_{s_{h+1}^{\check{l}_i}}\right)\left(V_{h+1}^{\check{l}_i}-V_{h+1}^{\check{l}_i,\star}\right)(s_h^k,a_h^k),
\end{aligned}\tag{3.20}$$

where the last step is by the fact that $V_{h+1}^{\check{l}_i}(s_h^k,a_h^k)\geq V_{h+1}^{\check{l}_i,\star}(s_h^k,a_h^k)$ from Lemma 21, and then by Hölder's inequality and the triangle inequality. The following proof is analogous to the proof of Lemma 15 in [61]. For completeness we reproduce it here. We have

$$\begin{aligned}
&\sum_{h=1}^{H}\sum_{k=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^{\check{l}_i}-\mathbf{e}_{s_{h+1}^{\check{l}_i}}\right)\left(V_{h+1}^{\check{l}_i}-V_{h+1}^{\check{l}_i,\star}\right)(s_h^k,a_h^k)\\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{j=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}_h^k}\sum_{i=1}^{\check{n}_h^k}\mathbb{1}\left[\check{l}_{h,i}^k=j\right]\left(P_h^j-\mathbf{e}_{s_{h+1}^j}\right)\left(V_{h+1}^j-V_{h+1}^{j,\star}\right)(s_h^k,a_h^k)\\
=&\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{j=1}^{K}(1+\frac{1}{H})^{h-1}\frac{1}{\check{n}_h^k}\sum_{i=1}^{\check{n}_h^k}\mathbb{1}\left[\check{l}_{h,i}^k=j\right]\left(P_h^j-\mathbf{e}_{s_{h+1}^j}\right)\left(V_{h+1}^j-V_{h+1}^{j,\star}\right)(s_h^j,a_h^j),
\end{aligned}\tag{3.21}$$

where (3.21) holds because $\check{l}_{h,i}^k(s_h^k,a_h^k)=j$ if and only if j is in the previous stage of k and $(s_h^k,a_h^k)=(s_h^j,a_h^j)$. For simplicity of notations, we define $\theta_{h+1}^k:=(1+\frac{1}{H})^{h-1}\sum_{j=1}^{K}\frac{1}{\check{n}_h^j}\sum_{i=1}^{\check{n}_h^j}\mathbb{1}\left[\check{l}_{h,i}^j=k\right]$. Then, we further have

(note that we have swapped the notation of j and k)

$$(3.21) = \sum_{h=1}^{H}\sum_{k=1}^{K} \theta_{h+1}^k \left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s_h^k, a_h^k).$$

For $(k,h) \in [K]\times[H]$, let x_h^k denote the number of occurrences of the triple (s_h^k, a_h^k, h) in the current stage. Define also $\tilde{\theta}_{h+1}^k := (1+\frac{1}{H})^{h-1}\frac{\lfloor (1+\frac{1}{H})x_h^k \rfloor}{x_h^k} \le 3$. Define $\mathcal{K} := \{(k,h) : \theta_{h+1}^k = \tilde{\theta}_{h+1}^k\}$, and $\bar{\mathcal{K}} := \{(k,h) \in [K]\times[H] : \theta_{h+1}^k \neq \tilde{\theta}_{h+1}^k\}$. Then, we have that

$$\begin{aligned}(3.21) =& \sum_{h=1}^{H}\sum_{k=1}^{K} \tilde{\theta}_{h+1}^k \left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s_h^k, a_h^k) \\ &+ \sum_{h=1}^{H}\sum_{k=1}^{K} (\theta_{h+1}^k - \tilde{\theta}_{h+1}^k)\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s_h^k, a_h^k).\end{aligned}$$

Since $\tilde{\theta}_{h+1}^k$ is independent of s_{h+1}^k, by the Azuma-Hoeffding inequality, it holds with probability at least $1-\delta$ that

$$\sum_{h=1}^{H}\sum_{k=1}^{K} \tilde{\theta}_{h+1}^k \left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s_h^k, a_h^k) \le O(\sqrt{KH^3\iota}). \tag{3.22}$$

It is easy to see that if k is in a stage that is before the second last stage of the triple (s_h^k, a_h^k, h), then $(k,h) \in \mathcal{K}$. For a triple (s,a,h), define $\mathcal{K}_h^{\perp}(s,a) := \{k \in [K] : k \text{ is in the second last stage of the triple } (s,a,h),\ (s_h^k, a_h^k) = (s,a)\}$. We have that

$$\begin{aligned}&\sum_{h=1}^{H}\sum_{k=1}^{K} (\theta_{h+1}^k - \tilde{\theta}_{h+1}^k)\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s_h^k, a_h^k) \\ =& \sum_{s,a,h}\sum_{k:(k,h)\in\bar{\mathcal{K}}} \mathbb{1}\left[(s_h^k, a_h^k) = (s,a)\right](\theta_{h+1}^k - \tilde{\theta}_{h+1}^k)\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s,a) \\ =& \sum_{s,a,h}(\theta_{h+1}(s,a) - \tilde{\theta}_{h+1}(s,a)) \sum_{k\in\mathcal{K}_h^{\perp}(s,a)}\left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right)\left(V_{h+1}^k - V_{h+1}^{k,\star}\right)(s,a),\end{aligned} \tag{3.23}$$

where for a fixed triple (s,a,h), we have defined $\theta_{h+1}(s,a) := \theta_{h+1}^k$, for any $k \in \mathcal{K}_h^{\perp}(s,a)$. Note that $\theta_{h+1}(s,a)$ is well-defined, because $\theta_{h+1}^{k_1} = \theta_{h+1}^{k_2}, \forall k_1, k_2 \in \mathcal{K}_h^{\perp}(s,a)$. Similarly, let $\tilde{\theta}_{h+1}(s,a) := \tilde{\theta}_{h+1}^k$ for any $k \in \mathcal{K}_h^{\perp}(s,a)$, and $\tilde{\theta}_{h+1}(s,a)$ is also well-defined. By the Azuma-Hoeffding inequality and a union bound, it holds with probability at least $1 - KH\delta$ that

$$\begin{aligned}(3.23) \le& \sum_{s,a,h} O\left(\sqrt{H^2\left|\mathcal{K}_h^{\perp}(s,a)\right|\iota}\right) \\ =& \sum_{s,a,h} O\left(\sqrt{H^2\check{N}_h^{K+1}(s,a)\iota}\right) \\ \le& O\left(\sqrt{SAH^3\iota\sum_{s,a,h}\check{N}_h^{K+1}(s,a)}\right) \qquad (3.24) \\ \le& O\left(\sqrt{SAKH^3\iota}\right) \qquad (3.25)\end{aligned}$$

where $\check{N}_h^{K+1}(s,a)$ is defined to be the total number of visitations to the triple (s,a,h) over the entire K

episodes. (3.24) is by the Cauchy-Schwartz inequality. (3.25) holds because, by the way stages are defined, for each triple (s, a, h), the length of its last two stages is at most an $O(1/H)$ fraction of the total number of visitations.

Combining (3.20), (3.22) and (3.25) completes the proof. □

3.11 Proof of Theorem 9

We introduce a few terms to facilitate the analysis. Denote by s_h^k and a_h^k respectively the state and action taken at step h of episode k. Let $N_h^k(s,a), \check{N}_h^k(s,a), Q_h^k(s,a)$ and $V_h^k(s)$ denote, respectively, the values of $N_h(s,a), \check{N}_h(s,a), Q_h(s,a)$ and $V_h(s)$ at the *beginning* of the k-th episode in Algorithm 14. Further, for the triple (s_h^k, a_h^k, h), let n_h^k be the total number of episodes that this triple has been visited prior to the current stage, and let $l_{h,i}^k$ denote the index of the episode that this triple was visited the i-th time among the total n_h^k times. Similarly, let $\check{n}_h^k$ denote the number of visits to the triple (s_h^k, a_h^k, h) in the stage right before the current stage, and let $\check{l}_{h,i}^k$ be the i-th episode among the $\check{n}_h^k$ episodes right before the current stage. For simplicity, we use l_i and $\check{l}_i$ to denote $l_{h,i}^k$ and $\check{l}_{h,i}^k$, and $\check{n}$ to denote $\check{n}_h^k$, when h and k are clear from the context. We also use $\check{r}_h(s,a)$ and $\check{v}_h(s,a)$ to denote the values of $\check{r}_h(s_h^k, a_h^k)$ and $\check{v}_h(s_h^k, a_h^k)$ when updating the $Q_h(s_h^k, a_h^k)$ value in Line 16 of Algorithm 14.

We now proceed to analyze the dynamic regret in one epoch, and at the very end of Section 3.11, we will see how to combine the dynamic regret over all the epochs to prove Theorem 9. The following analysis will be conditioned on the successful event of Lemma 21.

The dynamic regret of Algorithm 14 in epoch $d = 1$ can hence be expressed as

$$\begin{aligned}\mathcal{R}^{(d)}(\pi, K) &= \sum_{k=1}^{K} \left(V_1^{k,*}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right) \right) \\ &\le \sum_{k=1}^{K} \left(V_1^{k}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right) \right). \end{aligned} \tag{3.26}$$

From the update rules of the value functions in Algorithm 14, we have

$$\begin{aligned} V_h^k(s_h^k) &\le \mathbb{1}\left[n_h^k = 0\right] H + \frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} + \frac{\check{v}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} + b_h^k + 2b_\Delta \\ &= \mathbb{1}\left[n_h^k = 0\right] H + \frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) + b_h^k + 2b_\Delta. \end{aligned}$$

For ease of exposition, we define the following terms:

$$\delta_h^k := V_h^k(s_h^k) - V_h^{k,\star}(s_h^k), \ \ \zeta_h^k := V_h^k(s_h^k) - V_h^{k,\pi}(s_h^k). \tag{3.27}$$

We further define $\tilde{r}_h^k(s_h^k, a_h^k) := \frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} - r_h^k(s_h^k, a_h^k)$. Then by the Hoeffding's inequality, it holds with high probability that

$$\begin{aligned} \tilde{r}_h^k(s_h^k, a_h^k) &\le \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} r_h^{\check{l}_i}(s_h^k, a_h^k) + \sqrt{\frac{\iota}{\check{n}}} - r_h^k(s_h^k, a_h^k) \\ &\le b_h^k + b_\Delta. \end{aligned} \tag{3.28}$$

By the Bellman equation $V_h^{k,\pi}(s_h^k) = Q_h^{k,\pi}(s_h^k, \pi(s_h^k)) = r_h^k(s_h^k, a_h^k) + P_h^k V_{h+1}^{k,\pi}(s_h^k, a_h^k)$, we have

$$\begin{aligned}
\zeta_h^k \leq & \mathbb{1}\left[n_h^k = 0\right] H + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) + b_h^k + 2b_\Delta + \tilde{r}_h^k(s_h^k, a_h^k) - P_h^k V_{h+1}^{k,\pi}(s_h^k, a_h^k) \\
\leq & \mathbb{1}\left[n_h^k = 0\right] H + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} P_h^{\check{l}_i} V_{h+1}^{\check{l}_i}(s_h^k, a_h^k) - P_h^k V_{h+1}^{k,\pi}(s_h^k, a_h^k) + 3b_h^k + 3b_\Delta \qquad (3.29) \\
= & \mathbb{1}\left[n_h^k = 0\right] H + \underbrace{\frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(P_h^{\check{l}_i} - P_h^k\right) V_{h+1}^{\check{l}_i}(s_h^k, a_h^k)}_{①} + \underbrace{\frac{1}{\check{n}} \sum_{i=1}^{\check{n}} P_h^k \left(V_{h+1}^{\check{l}_i} - V_{h+1}^{\check{l}_i,\star}\right)(s_h^k, a_h^k)}_{②} + 3b_h^k + 3b_\Delta \\
& + \underbrace{\frac{1}{\check{n}} \sum_{i=1}^{\check{n}} P_h^k \left(V_{h+1}^{\check{l}_i,\star} - V_{h+1}^{k,\pi}\right)(s_h^k, a_h^k)}_{③}, \qquad (3.30)
\end{aligned}$$

where (3.29) is by the Azuma-Hoeffding inequality and by (3.28). In the following, we bound each term in (3.30) separately. First, by Hölder's inequality, we have

$$① \leq \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \Delta_p^{(1)}(H - h) \leq b_\Delta. \qquad (3.31)$$

Let $\mathbf{e}_j$ denote a standard basis vector of proper dimensions that has a 1 at the j-th entry and 0s at the others, in the form of $(0, \ldots, 0, 1, 0, \ldots, 0)$. Recall the definition of δ_h^k in (3.27), and we have

$$② = \underbrace{\frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(P_h^k - \mathbf{e}_{s_{h+1}^{\check{l}_i}}\right) \left(V_{h+1}^{\check{l}_i} - V_{h+1}^{\check{l}_i,\star}\right)(s_h^k, a_h^k)}_{\xi_{h+1}^k} + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \delta_{h+1}^{\check{l}_i} = \xi_{h+1}^k + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \delta_{h+1}^{\check{l}_i}. \qquad (3.32)$$

Finally, recalling the definition of ζ_h^k in (3.27), we have that

$$\begin{aligned}
③ = & \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(V_{h+1}^{\check{l}_i,\star}(s_{h+1}^k) - V_{h+1}^{k,\pi}(s_{h+1}^k)\right) \\
& + \underbrace{\frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(P_h^k - \mathbf{e}_{s_{h+1}^k}\right) \left(V_{h+1}^{\check{l}_i,\star} - V_{h+1}^{k,\pi}\right)(s_h^k, a_h^k)}_{\phi_{h+1}^k} \\
= & \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \left(V_{h+1}^{\check{l}_i,\star}(s_{h+1}^k) - V_{h+1}^{k,\star}(s_{h+1}^k)\right) + \zeta_{h+1}^k - \delta_{h+1}^k + \phi_{h+1}^k \\
\leq & b_\Delta + \zeta_{h+1}^k - \delta_{h+1}^k + \phi_{h+1}^k \qquad (3.33)
\end{aligned}$$

where inequality (3.33) is by Lemma 20. Combining (3.30), (3.31), (3.32), and (3.33) leads to

$$\zeta_h^k \leq \mathbb{1}\left[n_h^k = 0\right] H + \frac{1}{\check{n}} \sum_{i=1}^{\check{n}} \delta_{h+1}^{\check{l}_i} + \xi_{h+1}^k + \zeta_{h+1}^k - \delta_{h+1}^k + \phi_{h+1}^k + 3b_h^k + 5b_\Delta. \qquad (3.34)$$

To find an upper bound of $\sum_{k=1}^K \zeta_h^k$, we proceed to upper bound each term on the RHS of (3.34) separately.

First, notice that $\sum_{k=1}^{K} \mathbb{1}\left[n_h^k = 0\right] \leq SAH$, because each fixed triple (s,a,h) contributes at most 1 to $\sum_{k=1}^{K} \mathbb{1}\left[n_h^k = 0\right]$. In the following, we upper bound the second term in (3.34). Notice that

$$\sum_{k=1}^{K} \frac{1}{\check{n}_h^k} \sum_{i=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,i}^k} = \sum_{k=1}^{K} \sum_{j=1}^{K} \frac{1}{\check{n}_h^k} \delta_{h+1}^{j} \sum_{i=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,i}^k = j\right] = \sum_{j=1}^{K} \delta_{h+1}^{j} \sum_{k=1}^{K} \frac{1}{\check{n}_h^k} \sum_{i=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,i}^k = j\right]. \tag{3.35}$$

For a fixed episode j, notice that $\sum_{i=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,i}^k = j] \leq 1$, and that $\sum_{i=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,i}^k = j] = 1$ happens if and only if $(s_h^k, a_h^k) = (s_h^j, a_h^j)$ and (j,h) lies in the previous stage of (k,h) with respect to the triple (s_h^k, a_h^k, h). Let $\mathcal{K} := \{k \in [K] : \sum_{i=1}^{\check{n}_h^k} \mathbb{1}[\check{l}_{h,i}^k = j] = 1\}$; then, we know that every element $k \in \mathcal{K}$ has the same value of $\check{n}_h^k$, i.e., there exists an integer $N_j > 0$, such that $\check{n}_h^k = N_j, \forall k \in \mathcal{K}$. Further, by our definition of the stages, we know that $|\mathcal{K}| \leq (1 + \frac{1}{H})N_j$, because the current stage is at most $(1 + \frac{1}{H})$ times longer than the previous stage. Therefore, for every j, we know that

$$\sum_{k=1}^{K} \frac{1}{\check{n}_h^k} \sum_{i=1}^{\check{n}_h^k} \mathbb{1}\left[\check{l}_{h,i}^k = j\right] \leq 1 + \frac{1}{H}. \tag{3.36}$$

Substituting it back into (3.35) leads to

$$\sum_{k=1}^{K} \frac{1}{\check{n}_h^k} \sum_{i=1}^{\check{n}_h^k} \delta_{h+1}^{\check{l}_{h,i}^k} \leq (1 + \frac{1}{H}) \sum_{k=1}^{K} \delta_{h+1}^{k}. \tag{3.37}$$

Combining (3.34) and (3.37), we now have that

$$\begin{aligned}
\sum_{k=1}^{K} \zeta_h^k \leq & SAH^2 + \frac{1}{H} \sum_{k=1}^{K} \delta_{h+1}^k + \sum_{k=1}^{K} \left(\xi_{h+1}^k + \zeta_{h+1}^k + \phi_{h+1}^k + 3b_h^k + 5b_\Delta\right) \\
\leq & SAH^2 + (1 + \frac{1}{H}) \sum_{k=1}^{K} \zeta_{h+1}^k + \sum_{k=1}^{K} \underbrace{\left(\xi_{h+1}^k + \phi_{h+1}^k + 3b_h^k + 5b_\Delta\right)}_{\Lambda_{h+1}^k},
\end{aligned} \tag{3.38}$$

where in (3.38) we have used the fact that $\delta_{h+1}^k \leq \zeta_{h+1}^k$, which in turn is due to the optimality that $V_h^{k,\star}(s_h^k) \geq V_h^{k,\pi}(s_h^k)$. Notice that we have ζ_h^k on the LHS of (3.38) and ζ_{h+1}^k on the RHS. By iterating (3.38) over $h = H, H-1, \ldots, 1$, we conclude that

$$\sum_{k=1}^{K} \zeta_1^k \leq O\left(SAH^3 + \sum_{h=1}^{H} \sum_{k=1}^{K} (1 + \frac{1}{H})^{h-1} \Lambda_{h+1}^k\right). \tag{3.39}$$

We bound $\sum_{h=1}^{H} \sum_{k=1}^{K} (1 + \frac{1}{H})^{h-1} \Lambda_{h+1}^k$ in the proposition below. Its proof relies on a series of lemmas in Section 3.10 that upper bound each term in Λ_{h+1}^k separately.

Proposition 2. *With probability at least* $1 - (KH+2)\delta$*, it holds that*

$$\sum_{h=1}^{H} \sum_{k=1}^{K} (1 + \frac{1}{H})^{h-1} \Lambda_{h+1}^k \leq \widetilde{O}(\sqrt{SAKH^5} + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)}).$$

Now we are ready to prove Theorem 9.

Proof. (of Theorem 9) By (3.26) and (3.39), and by replacing δ with $\frac{\delta}{KH+2}$ in Proposition 2, we know that the dynamic regret in epoch $d=1$ can be upper bounded with probability at least $1-\delta$ by:

$$\mathcal{R}^{(d)}(\pi, K) \leq \widetilde{O}(SAH^3 + \sqrt{SAKH^5} + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)}), \tag{3.40}$$

and this holds for every epoch $d \in [D]$. Suppose $T = \Omega(SA\Delta H^2)$; summing up the dynamic regret over all the D epochs gives us an upper bound of $\widetilde{O}(D\sqrt{SAKH^5} + \sum_{d=1}^{D} KH\Delta_r^{(d)} + \sum_{d=1}^{D} KH^2\Delta_p^{(d)})$. Recall the definition that $\sum_{d=1}^{D}\Delta_r^{(d)} \leq \Delta_r$, $\sum_{d=1}^{D}\Delta_p^{(d)} \leq \Delta_p$, $\Delta = \Delta_r + \Delta_p$, and that $K = \Theta(\frac{T}{DH})$. By setting $D = S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}H^{-\frac{2}{3}}T^{\frac{1}{3}}$, the dynamic regret over the entire T steps is bounded by $\mathcal{R}(\pi, M) \leq \widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{5}{3}}T^{\frac{2}{3}})$, which completes the proof. □

3.12 Proof Sketch of Theorem 10

Proof sketch. We only outline the difference with respect to the proof of Theorem 9. The reader should have no difficulty recovering the complete proof by following the same routine as in the proof of Theorem 9. Specifically, it suffices to investigate the steps that are involved with Lemma 21.

The dynamic regret of the new algorithm in epoch $d=1$ can now be expressed as

$$\mathcal{R}^{(d)}(\pi, K) = \sum_{k=1}^{K}\left(V_1^{k,*}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right)\right) \leq \sum_{k=1}^{K}\left(V_1^{k}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right)\right) + 2KHb_\Delta, \tag{3.41}$$

where we applied the results of Lemma 22 instead of Lemma 21. The reader should bear in mind that from the new update rules of the value functions, we now have

$$V_h^k(s_h^k) \leq \mathbb{1}\big[n_h^k = 0\big]H + \frac{\check{r}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} + \frac{\check{v}_h(s_h^k, a_h^k)}{\check{N}_h^k(s_h^k, a_h^k)} + b_h^k, \tag{3.42}$$

where the RHS no longer has the additional bonus term b_Δ. If we define ζ_h^k, ξ_{h+1}^k, and ϕ_{h+1}^k in the same way as before, the reader can easily verify that all the derivations until Equation (3.39) still holds, although the value of Λ_{h+1}^k should be re-defined as $\Lambda_{h+1}^k := \xi_{h+1}^k + \phi_{h+1}^k + 3b_h^k + 3b_\Delta$ due to the new upper bound in (3.42) that is independent of b_Δ. Proposition 2 also follows analogously, though some additional attention should be paid to the proof of Lemma 26 where the results of Lemma 21 have been utilized. Finally, we obtain the dynamic regret upper bound in epoch $d=1$ as follows:

$$\mathcal{R}^{(d)}(\pi, K) \leq \widetilde{O}\left(SAH^3 + \sqrt{SAKH^5} + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)}\right) + 2KHb_\Delta, \tag{3.43}$$

where the additional term $2KHb_\Delta$ comes from (3.41). From our definition of b_Δ, we can easily see that $2KHb_\Delta \leq O(KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)})$. Therefore, we can conclude that the dynamic regret upper bound in one epoch remains the same order, which leaves the dynamic regret over the entire horizon also unchanged. □

3.13 Proof of Theorem 11

Similar to the proofs of Theorems 9 and 10, we start with the dynamic regret in one epoch, and then extend to all epochs in the end. The proof follows the same routine as in the proofs of Theorems 9 and 10. Given that a rigorous analysis on the Freedman-based bonus with variance reduction is present in [61], one should

not find it difficult to extend our Hoeffding-based algorithm to a Freedman-based one. Therefore, rather than providing a complete proof of Theorem 11, in the following, we sketch the differences and highlight the additional analysis needed that is not covered by the proof of Theorem 10 and [61].

To facilitate the analysis, first recall a few notations $N_h^k, \check{N}_h^k, Q_h^k(s,a), V_h^k(s), n_h^k, l_{h,i}^k, \check{n}_h^k, \check{l}_{h,i}^k, l_i$ and $\check{l}_i$ that we have defined in Section 3.11. In addition, when (h,k) is clear from the context, we drop the time indices and simply use $\check{\mu}, \check{\sigma}, \mu^{\text{ref}}, \sigma^{\text{ref}}$ to denote their corresponding values in the computation of the $Q_h(s_h^k, a_h^k)$ value in Line 16 of Algorithm 14.

We start with the following lemma, which is an analogue of Lemma 22 but requires a more careful treatment of variations accumulated in μ^{ref} and $\check{\mu}_h$. It states that the optimistic $Q_h^k(s,a)$ is an "upper bound" of the optimal $Q_h^{k,\star}(s,a)$ subject to an error term of the order $2(H-h+1)b_\Delta$ with high probability.

Lemma 27. *(Freedman, no local budgets) For $\delta \in (0,1)$, with probability at least $1-2KH\delta$, it holds that $Q_h^{k,\star}(s,a) - 4(H-h+1)b_\Delta \le Q_h^{k+1}(s,a) \le Q_h^k(s,a), \forall (s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$.*

Proof. It should be clear from the way we update $Q_h(s,a)$ that $Q_h^k(s,a)$ is monotonically decreasing in k. We now prove $Q_h^{k,\star}(s,a) - 4(H-h+1)b_\Delta \le Q_h^{k+1}(s,a)$ for all s,a,h,k by induction on k. First, it holds for $k=1$ by our initialization of $Q_h(s,a)$. For $k \ge 2$, now suppose $Q_h^{j,\star}(s,a) - 4(H-h+1)b_\Delta \le Q_h^j(s,a)$ for all s,a,h and $1 \le j \le k$. For a fixed triple (s,a,h), we consider the following two cases.

Case 1: $Q_h(s,a)$ is updated in episode k. Notice that it suffices to analyze the case where $Q_h(s,a)$ is updated using $\underline{b}_h^k$, because the other case of b_h^k would be exactly the same as in Lemma 22. With probability at least $1-\delta$,

$$
\begin{aligned}
Q_h^{k+1}(s,a) =& \frac{\check{r}_h(s,a)}{\check{N}_h^k(s,a)} + \frac{\mu^{\text{ref}}(s,a)}{N_h^k(s,a)} + \frac{\check{\mu}_h(s,a)}{\check{N}_h^k(s,a)} + 2\underline{b}_h^k \\
=& \frac{\check{r}_h(s,a)}{\check{n}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(V_{h+1}^{\text{ref},l_i}(s_{h+1}^{l_i}) - P_h^{l_i}V_{h+1}^{\text{ref},l_i}(s,a)\right)}_{\chi_1} \\
&+ \underbrace{\frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left[\left(V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) - V_{h+1}^{\text{ref},\check{l}_i}(s_{h+1}^{\check{l}_i})\right) - \left(P_h^{\check{l}_i}V_{h+1}^{\check{l}_i} - P_h^{\check{l}_i}V_{h+1}^{\text{ref},\check{l}_i}\right)(s,a)\right]}_{\chi_2} \\
&+ \underbrace{\frac{1}{n}\sum_{i=1}^{n}P_h^{l_i}V_{h+1}^{\text{ref},l_i} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^{\check{l}_i}V_{h+1}^{\check{l}_i} - P_h^{\check{l}_i}V_{h+1}^{\text{ref},\check{l}_i}\right)(s,a)}_{\chi_3} + 2\underline{b}_h^k. \qquad (3.44)
\end{aligned}
$$

In the following, we will bound each term in (3.44) separately. First, we have that

$$
\begin{aligned}
\chi_3 =& \frac{1}{n}\sum_{i=1}^{n}\left(P_h^{l_i}V_{h+1}^{\text{ref},l_i} - P_h^{k}V_{h+1}^{\text{ref},l_i}\right)(s,a) & (3.45)\\
&- \frac{1}{\check{n}}\sum_{i=1}^{\check{n}}\left(P_h^{\check{l}_i}V_{h+1}^{\text{ref},\check{l}_i} - P_h^{k}V_{h+1}^{\text{ref},\check{l}_i}\right)(s,a) & (3.46)\\
&+ \frac{1}{n}\sum_{i=1}^{n}P_h^{k}V_{h+1}^{\text{ref},l_i}(s,a) - \frac{1}{\check{n}}\sum_{i=1}^{\check{n}}P_h^{k}V_{h+1}^{\text{ref},\check{l}_i}(s,a) + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}}P_h^{\check{l}_i}V_{h+1}^{\check{l}_i}(s,a) & (3.47)\\
\ge& \frac{1}{\check{n}}\sum_{i=1}^{\check{n}}P_h^{\check{l}_i}V_{h+1}^{\check{l}_i}(s,a) - 2b_\Delta, & (3.48)
\end{aligned}
$$

where (3.45)$\geq -b_\Delta$ and (3.46)$\geq -b_\Delta$ by Hölder's inequality and the definition of b_Δ. In (3.47), we have that $\frac{1}{n}\sum_{i=1}^{n} P_h^k V_{h+1}^{\text{ref},l_i}(s,a) - \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} P_h^k V_{h+1}^{\text{ref},\check{l}_i}(s,a) \geq 0$, because $V_{h+1}^{\text{ref},k}(s)$ is non-increasing in k.

Following a similar procedure as in Lemma 10, Lemma 12, and Lemma 13 in [61], we can further bound $|\chi_1|$ and $|\chi_2|$ as follows:

$$|\chi_1| \leq 2\sqrt{\frac{\nu^{\text{ref}}\iota}{n}} + \frac{5H\iota^{\frac{3}{4}}}{n^{\frac{3}{4}}} + \frac{2\sqrt{\iota}}{Tn} + \frac{2H\iota}{n}, \tag{3.49}$$

$$|\chi_2| \leq 2\sqrt{\frac{\check{\nu}\iota}{\check{n}}} + \frac{5H\iota^{\frac{3}{4}}}{\check{n}^{\frac{3}{4}}} + \frac{2\sqrt{\iota}}{T\check{n}} + \frac{2H\iota}{\check{n}}, \tag{3.50}$$

where $\nu^{\text{ref}} := \frac{\sigma^{\text{ref}}}{n} - \left(\frac{\mu^{\text{ref}}}{n}\right)^2$ and $\check{\nu} := \frac{\check{\sigma}}{\check{n}} - \left(\frac{\check{\mu}}{\check{n}}\right)^2$. These are the steps where Freedman's inequality [192] come into use, and we omit these steps since they are essentially the same as the derivations in [61]. We can see from (3.49), (3.50), and the definition of $\underline{b}_h^k$ that $|\chi_1| + |\chi_2| \leq \underline{b}_h^k$.

Substituting the results on χ_1, χ_2 and χ_3 back to (3.44), it holds that with probability at least $1-\delta$,

$$\begin{aligned}
Q_h^{k+1}(s,a) =& \frac{\check{r}_h(s,a)}{\check{n}} + \chi_1 + \chi_2 + \chi_3 + 2\underline{b}_h^k \\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} P_h^{\check{l}_i} V_{h+1}^{\check{l}_i}(s,a) + \underline{b}_h^k - 2b_\Delta && (3.51)\\
\geq& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} P_h^{\check{l}_i} V_{h+1}^{\check{l}_i,\star}(s,a) - 4(H-h)b_\Delta + \underline{b}_h^k - 2b_\Delta && (3.52)\\
=& \frac{\check{r}_h(s,a)}{\check{n}} + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} \left(Q_h^{\check{l}_i,\star}(s,a) - r_h^{\check{l}_i}(s,a)\right) + \underline{b}_h^k - 4(H-h)b_\Delta - 2b_\Delta \\
\geq& \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} Q_h^{\check{l}_i,\star}(s,a) - 4(H-h)b_\Delta - 2b_\Delta \geq Q_h^{k,\star}(s,a) - 4(H-h)b_\Delta - 3b_\Delta, && (3.53)
\end{aligned}$$

where in (3.51) we used (3.48), (3.49), (3.50), and the definition of $\underline{b}_h^k$ in Algorithm 14. (3.52) is by the induction hypothesis that $Q_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i},a) \geq Q_{h+1}^{\check{l}_i,\star}(s_{h+1}^{\check{l}_i},a) - 2(H-h)b_\Delta, \forall a \in \mathcal{A}, 1 \leq \check{l}_i \leq k$. The second to last inequality holds due to the Hoeffding's inequality that $\frac{1}{\check{n}}\left(\sum_{i=1}^{\check{n}} r_h^{\check{l}_i}(s,a) - \check{r}_h(s,a)\right) \leq \sqrt{\frac{\iota}{\check{n}}} \leq \underline{b}_h^k$ with high probability. Finally, the last inequality follows from Lemma 20.

According to the monotonicity of $Q_h^k(s,a)$, we can conclude from (3.53) that $Q_h^{k,\star}(s,a) - 4(H-h+1)b_\Delta \leq Q_h^{k+1}(s,a) \leq Q_h^k(s,a)$. In fact, we have proved the stronger statement $Q_h^{k+1}(s,a) \geq Q_h^{k,\star}(s,a) - 4(H-h+1)b_\Delta + b_\Delta$ that will be useful in Case 2 below.

Case 2: $Q_h(s,a)$ is not updated in episode k. Then, there are two possibilities:

1. If $Q_h(s,a)$ has never been updated from episode 1 to episode k: It is easy to see that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^1(s,a) = H-h+1 \geq Q_h^{k,\star}(s,a)$ holds.

2. If $Q_h(s,a)$ has been updated at least once from episode 1 to episode k: Let j be the index of the latest episode that $Q_h(s,a)$ was updated. Then, from our induction hypothesis and Case 1, we know that $Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) - 4(H-h+1)b_\Delta + b_\Delta$. Since $Q_h(s,a)$ has not been updated from episode $j+1$ to episode k, we know that $Q_h^{k+1}(s,a) = Q_h^k(s,a) = \cdots = Q_h^{j+1}(s,a) \geq Q_h^{j,\star}(s,a) - 4(H-h+1)b_\Delta + b_\Delta \geq Q_h^{k,\star}(s,a) - 4(H-h+1)b_\Delta$, where the last inequality holds because of Lemma 20.

A union bound over all time steps completes our proof. □

Conditional on the successful event of Lemma 27, the dynamic regret of RestartQ-UCB Freedman in epoch $d = 1$ can hence be expressed as

$$\mathcal{R}^{(d)}(\pi, K) = \sum_{k=1}^{K} \left(V_1^{k,*}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right) \right) \leq \sum_{k=1}^{K} \left(V_1^{k}\left(s_1^k\right) - V_1^{k,\pi}\left(s_1^k\right) \right) + 4KHb_\Delta. \tag{3.54}$$

From the update rules of the value functions in Algorithm 14, we have

$$\begin{aligned} V_h^k(s_h^k) &\leq \mathbb{1}\left[n_h^k = 0\right] H + \frac{\check{r}_h(s_h^k, a_h^k)}{\check{n}} + \frac{\mu_h^{\mathrm{ref},k}}{n} + \frac{\check{\mu}_h^k}{\check{n}} + 2\underline{b}_h^k \\ &= \mathbb{1}\left[n_h^k = 0\right] H + \frac{\check{r}_h(s_h^k, a_h^k)}{\check{n}} + \frac{1}{n}\sum_{i=1}^{n} V_{h+1}^{\mathrm{ref},l_i}(s_{h+1}^{l_i}) + \frac{1}{\check{n}}\sum_{i=1}^{\check{n}} (V_{h+1}^{\check{l}_i}(s_{h+1}^{\check{l}_i}) - V_{h+1}^{\mathrm{ref},\check{l}_i}(s_{h+1}^{\check{l}_i})) + 2\underline{b}_h^k. \end{aligned}$$

If we again define $\zeta_h^k := V_h^k(s_h^k) - V_h^{k,\pi}(s_h^k)$, we can follow a similar routine as in the proof of Theorem 9 (details can be found in [61]) and obtain

$$\sum_{k=1}^{K} \zeta_1^k \leq O\left(SAH^3 + \sum_{h=1}^{H}\sum_{k=1}^{K} (1 + \frac{1}{H})^{h-1} \Lambda_{h+1}^k \right),$$

where $\Lambda_{h+1}^k := \psi_{h+1}^k + \xi_{h+1}^k + \phi_{h+1}^k + 4\underline{b}_h^k + 4b_\Delta$ with the following definitions:

$$\begin{aligned} \psi_{h+1}^k &:= \frac{1}{n_h^k} \sum_{i=1}^{n_h^k} \left(P_h^k V_{h+1}^{\mathrm{ref},l_i} - P_h^k V_{h+1}^{\mathrm{ref},K+1} \right)(s_h^k, a_h^k), \\ \xi_{h+1}^k &:= \frac{1}{\check{n}_h^k} \sum_{i=1}^{\check{n}_h^k} \left(P_h^k - \mathbf{e}_{s_{h+1}^{\check{l}_i}} \right) \left(V_{h+1}^{\check{l}_i} - V_{h+1}^{\check{l}_i,\star} \right)(s_h^k, a_h^k), \\ \phi_{h+1}^k &:= \left(P_h^k - \mathbf{e}_{s_{h+1}^k} \right) \left(V_{h+1}^{\check{l}_i,\star} - V_{h+1}^{k,\pi} \right)(s_h^k, a_h^k). \end{aligned}$$

An upper bound on the first four terms in Λ_{h+1}^k is derived in the proof of Lemma 7 in [61] (There is an extra term of $\sqrt{\frac{1}{\check{n}}\iota}$ in our defnition of $\underline{b}_h^k$ compared to theirs, but it does not affect the leading term in the upper bound). By further recalling the definition of b_Δ, we can obtain the following lemma.

Lemma 28. *(Lemma 7 in [61]) With probability at least* $(1 - O(H^2T^4\delta))$*, it holds that*

$$\sum_{h=1}^{H}\sum_{k=1}^{K} (1 + \frac{1}{H})^{h-1} \Lambda_{h+1}^k = O\left(\sqrt{SAH^3K\iota} + \sqrt{KH^3\iota}\log(KH) + S^2A^{\frac{3}{2}}H^{\frac{33}{4}}K^{\frac{1}{4}}\iota + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)} \right).$$

Combined with (3.54) and the definition of ζ_h^k, we obtain the dynamic regret bound in a single epoch:

$$\mathcal{R}^{(d)}(\pi, K) = O\left(\sqrt{SAH^3K\iota} + \sqrt{KH^3\iota}\log(KH) + S^2A^{\frac{3}{2}}H^{\frac{33}{4}}K^{\frac{1}{4}}\iota + KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)} + KHb_\Delta \right), \forall d \in [D].$$

From our definition of b_Δ, we can easily see that $KHb_\Delta \leq O(KH\Delta_r^{(1)} + KH^2\Delta_p^{(1)})$. Finally, suppose T is greater than a polynomial of S, A, Δ and H, $\sqrt{SAH^3K\iota}$ would be the leading term of the dynamic regret in

a single epoch. In this case, summing up the dynamic regret over all the D epochs gives us an upper bound of

$$\widetilde{O}\left(D\sqrt{SAH^3K}+\sum_{d=1}^{D}KH\Delta_r^{(d)}+\sum_{d=1}^{D}KH^2\Delta_p^{(d)}\right). \tag{3.55}$$

Recall that $\sum_{d=1}^{D}\Delta_r^{(d)}\leq\Delta_r$, $\sum_{d=1}^{D}\Delta_p^{(d)}\leq\Delta_p$, $\Delta=\Delta_r+\Delta_p$, and that $K=\Theta(\frac{T}{DH})$. By setting $D=S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}}$, the dynamic regret over the entire T steps is bounded by

$$\mathcal{R}(\pi,M)\leq\widetilde{O}\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}}\right).$$

This completes the proof of Theorem 11.

3.14 Proof of Theorem 12

First, we define $D^\dagger$ to be the optimal candidate value in $\mathcal{J}$ that leads to the lowest dynamic regret. Recall that since $\mathcal{J}$ is a discretized set and only covers values in the range of $\left[\left\lfloor\frac{T}{SAH^2W}\right\rfloor,\left\lfloor\frac{T}{SAH^2}\right\rfloor\right]$, it might not contain the actual optimal value $D^\star=S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}}$ for the number of epochs D. Further, let $R_i(D)$ be the cumulative reward collected in phase i due to choosing the value D for the number of total epochs. Then, the dynamic regret of Algorithm 15 can be decomposed into two parts:

$$\begin{aligned}\mathcal{R}(\pi,M)&=\sum_{m=1}^{M}\left(V_1^{m,\star}\left(s_1^m\right)-V_1^{m,\pi}\left(s_1^m\right)\right)\\&=\left[\sum_{m=1}^{M}V_1^{m,\star}\left(s_1^m\right)-\sum_{i=1}^{\lceil M/W\rceil}R_i(D^\dagger)\right]+\left[\sum_{i=1}^{\lceil M/W\rceil}R_i(D^\dagger)-\sum_{i=1}^{\lceil M/W\rceil}R_i(D_i)\right],\end{aligned} \tag{3.56}$$

where the first term is the dynamic regret of using the optimal candidate value $D^\dagger$ of the number of epochs, and the second term is caused by the regret of learning the optimal candidate value using the Exp3.P algorithm. Applying the regret bound of the Exp3.P algorithm ([107]), for any choice of $D^\dagger$, the second term in (3.56) is upper bounded by

$$\sum_{i=1}^{\lceil M/W\rceil}R_i(D^\dagger)-\sum_{i=1}^{\lceil M/W\rceil}R_i(D_i)\leq\widetilde{O}(WH\sqrt{\lceil M/W\rceil\left(J+1\right)})=\widetilde{O}(\sqrt{HTW})=\widetilde{O}(H^{\frac{3}{4}}T^{\frac{3}{4}}), \tag{3.57}$$

where in the last step we used that $W=\sqrt{HT}$.

From the proof of Theorem 11 (e.g., Equation (3.55) with the fact that $K=\Theta(\frac{T}{DH})$), and applying the Azuma-Hoeffding inequality and a union bound, we can upper bound the first term in (3.56) by

$$\sum_{m=1}^{M}V_1^{m,\star}\left(s_1^m\right)-\sum_{i=1}^{\lceil M/W\rceil}R_i(D^\dagger)\leq\widetilde{O}\left(\sqrt{SATD^\dagger H^2}+\frac{TH\Delta}{D^\dagger}+\sqrt{TH}\right). \tag{3.58}$$

To derive a further upper bound of (3.58), we need to distinguish between two cases: Whether $D^\star$ is covered in the range of $\mathcal{J}$ or not. Since we have assumed that the horizon is sufficiently long, i.e., T is greater than some polynomial of S,A,Δ and H, it holds that $D^\star=S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}}\leq\left\lfloor\frac{T}{SAH^2}\right\rfloor$. Therefore, to determine whether $D^\star$ is covered in the range of $\mathcal{J}$, we only need to compare $D^\star$ with the lower bound $\left\lfloor\frac{T}{SAH^2W}\right\rfloor$ in $\mathcal{J}$.

- If $D^\star$ is covered in the range of $\mathcal{J}$, i.e., $D^\star \geq \left\lfloor \frac{T}{SAH^2W} \right\rfloor$: Since $\mathcal{J}$ is discretized in a way that two consecutive values differ from each other by a factor of at most $W^{1/J}$, we know that there exists a value $D^\dagger \in \mathcal{J}$, such that $D^\star \leq D^\dagger \leq W^{1/J} D^\star$. In this case, we can upper bound the RHS of (3.58) by

$$\widetilde{O}\left(\sqrt{SATD^\dagger H^2} + \frac{TH\Delta}{D^\dagger} + \sqrt{TH}\right) \leq \widetilde{O}\left(\sqrt{SATW^{1/J}D^\star H^2} + \frac{TH\Delta}{D^\star}\right) \leq \widetilde{O}\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}}\right),$$

where in the last step we used the facts that $D^\star = S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}}$ and that $W^{1/J} = W^{1/\lceil \ln W \rceil} \leq \exp(1)$.

- If $D^\star$ is not covered in the range of $\mathcal{J}$, i.e., $D^\star < \left\lfloor \frac{T}{SAH^2W} \right\rfloor$: Since $D^\star = S^{-\frac{1}{3}}A^{-\frac{1}{3}}\Delta^{\frac{2}{3}}T^{\frac{1}{3}} < \left\lfloor \frac{T}{SAH^2W} \right\rfloor$, it implies that $\Delta < S^{-1}A^{-1}H^{-\frac{15}{4}}T^{\frac{1}{4}}$. The optimal candidate value in $\mathcal{J}$ would be the smallest one, and hence $D^\dagger = \left\lfloor \frac{T}{SAH^2W} \right\rfloor$. In this case, we can upper bound the RHS of (3.58) by

$$\widetilde{O}\left(\sqrt{SATD^\dagger H^2} + \frac{TH\Delta}{D^\dagger} + \sqrt{TH}\right) \leq \widetilde{O}\left(\sqrt{SATH^2\left\lfloor \frac{T}{SAH^2W} \right\rfloor} + \frac{TH\Delta}{\left\lfloor \frac{T}{SAH^2W} \right\rfloor}\right) \leq \widetilde{O}\left(H^{\frac{3}{4}}T^{\frac{3}{4}}\right),$$

where in the last step we used that $\Delta < S^{-1}A^{-1}H^{-\frac{15}{4}}T^{\frac{1}{4}}$ and $W = \sqrt{HT}$.

Combining the above two cases with (3.56), (3.57), and (3.58), we can conclude that the dynamic regret of Algorithm 15 is upper bounded by

$$\mathcal{R}(\pi, M) \leq \widetilde{O}\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}} + H^{\frac{3}{4}}T^{\frac{3}{4}}\right).$$

This completes the proof of Theorem 12.

3.15 Proof of Theorem 13

The proof of our lower bound relies on the construction of a "hard instance" of non-stationary MDPs. The instance we construct is essentially a switching-MDP: an MDP with piecewise constant dynamics on each *segment* of the horizon, and its dynamics experience an abrupt change at the beginning of each new segment. More specifically, we divide the horizon T into L segments[7], where each segment has $T_0 := \left\lfloor \frac{T}{L} \right\rfloor$ steps and contains $M_0 := \left\lfloor \frac{M}{L} \right\rfloor$ episodes, each episode having a length of H. Within each such segment, the system dynamics of the MDP do not vary, and we construct the dynamics for each segment in a way such that the instance is a hard instance of stationary MDPs on its own. The MDP within each segment is essentially similar to the hard instances constructed in stationary RL problems [60], [182]. Between two consecutive segments, the dynamics of the MDP change abruptly, and we let the dynamics vary in a way such that no information learned from previous interactions with the MDP can be used in the new segment. In this sense, the agent needs to learn a new hard stationary MDP in each segment. Finally, optimizing the value of L and the variation magnitude between consecutive segments (subject to the constraints of the total variation budget) leads to our lower bound.

We start with a simplified episodic setting where the transition kernels and reward functions are held constant within each episode, i.e., $P_1^m = \cdots = P_h^m = \ldots P_H^m$ and $r_1^m = \cdots = r_h^m = \ldots r_H^m, \forall m \in [M]$. This is a popular but less challenging episodic setting, and its stationary counterpart has been studied in [97]. We further require that when the environment varies due to the non-stationarity, all steps in one episode should

[7]The definition of segments is irrelevant to, and should not be confused with, the notion of epochs we previously defined.

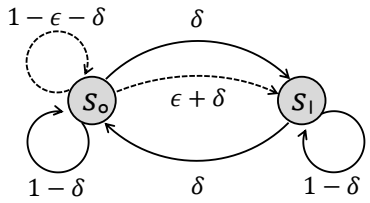


Figure 3.3: The "JAO MDP" constructed in [96]. Dashed lines denote transitions related to the good action $a^\star$.

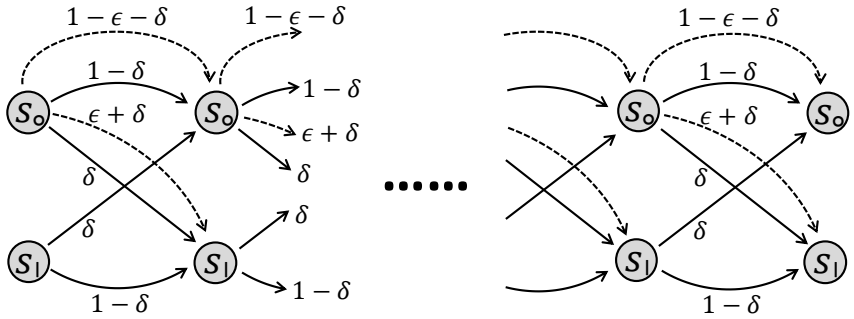


Figure 3.4: A chain with H copies of JAO MDPs correlated in time. At the end of an episode, the state should deterministically transition from any state in the last copy to the $s_\circ$ state in the first copy of the chain, the arrows of which are not shown in the figure. Also, the $s_|$ state in the first copy is actually never reached and is redundant.

vary simultaneously in the same way. This simplified setting is easier to analyze, and its analysis conveniently leads to a lower bound for the un-discounted setting as a side result along the way. Later we will show how the analysis can be naturally extended to the more general setting we introduced in Section 3.2, using techniques that have also been utilized in [60]. For simplicity of notations, we temporarily drop the h indices and use P^m and r^m to denote the transition kernel and reward function whenever there is no ambiguity.

Consider a two-state MDP as depicted in Figure 3.3. This MDP was initially proposed in [96] as a hard instance of stationary MDPs, and following [60] we will refer to this construction as the "JAO MDP". This MDP has 2 states $\mathcal{S} = \{s_\circ, s_|\}$ and SA actions $\mathcal{A} = \{1, 2, \ldots, SA\}$. The reward does not depend on actions: state $s_|$ always gives reward 1 whatever action is taken, and state $s_\circ$ always gives reward 0. Any action taken at state $s_|$ takes the agent to state $s_\circ$ with probability δ, and to state $s_|$ with probability $1-\delta$. At state $s_\circ$, for all but a single "good" action $a^\star$, the agent is taken to state $s_|$ with probability δ, and for the good action $a^\star$, the agent is taken to state $s_|$ with probability $\delta + \varepsilon$ for some $0 < \varepsilon < \delta$. The exact values of δ and ε will be chosen later. Note that this is not an MDP with S states and A actions as we desire, but the extension to an MDP with S states and A actions is routine [96], and is hence omitted here.

To apply the JAO MDP to the simplified episodic setting, we "concatenate" H copies of exactly the same JAO MDP into a chain as depicted in Figure 3.4, denoting the H steps in an episode. The initial state of this MDP is the $s_\circ$ state in the first copy of the chain, and after each episode the state is "reset" to the initial state. In the following, we first show that the constructed MDP is a hard instance of stationary MDPs, without worrying about the evolution of the system dynamics. The techniques that we will be using are essentially the same as in the proofs of the lower bound in the multi-armed bandit problem [107] or the reinforcement learning problem in the un-discounted setting [96].

The good action $a^\star$ is chosen uniformly at random from the action space $\mathcal{A}$, and we use $\mathbb{E}_\star[\cdot]$ to denote the expectation with respect to the random choice of $a^\star$. We write $\mathbb{E}_a[\cdot]$ for the expectation conditioned on action a being the good action $a^\star$. Finally, we use $\mathbb{E}_{\text{unif}}[\cdot]$ to denote the expectation when there is no good action in the MDP, i.e., every action in $\mathcal{A}$ takes the agent from state $s_\circ$ to $s_\shortmid$ with probability δ. Define the probability notations $\mathbb{P}_\star(\cdot), \mathbb{P}_a(\cdot)$, and $\mathbb{P}_{\text{unif}}(\cdot)$ analogously.

Consider running a reinforcement learning algorithm on the constructed MDP for T_0 steps, where $T_0 = M_0 H$. It has been shown in [107] and [96] that it is sufficient to consider deterministic policies. Therefore, we assume that the algorithm maps deterministically from a sequence of observations to an action a_t at time t. Define the random variables $N_\shortmid, N_\circ$ and $N_\circ^\star$ to be the total number of visits to state $s_\shortmid$, the total number of visits to $s_\circ$, and the total number of times that $a^\star$ is taken at state $s_\circ$, respectively. Let s_t denote the state observed at time t, and a_t the action taken at time t. When there is no chance of ambiguity, we sometimes also use s_h^m to denote the state at step h of episode m, which should be interpreted as the state s_t observed at time $t = (m-1) \times H + h$. The notation a_h^m is used analogously. Since $s_\circ$ is assumed to be the initial state, we have that

$$
\begin{aligned}
&\mathbb{E}_a[N_\shortmid] = \sum_{t=1}^{T_0} \mathbb{P}_a(s_t = s_\shortmid) = \sum_{m=1}^{M_0} \sum_{h=2}^{H} \mathbb{P}_a(s_h^m = s_\shortmid) \\
=&\sum_{m=1}^{M_0} \sum_{h=2}^{H} \left(\mathbb{P}_a(s_{h-1}^m = s_\circ) \cdot \mathbb{P}_a(s_h^m = s_\shortmid \mid s_{h-1}^m = s_\circ) + \mathbb{P}_a(s_{h-1}^m = s_\shortmid) \cdot \mathbb{P}_a(s_h^m = s_\shortmid \mid s_{h-1}^m = s_\shortmid) \right) \\
=&\sum_{m=1}^{M_0} \sum_{h=2}^{H} \left(\delta \mathbb{P}_a(s_{h-1}^m = s_\circ, a_h^m \neq a^\star) + (\delta + \varepsilon) \mathbb{P}_a(s_{h-1}^m = s_\circ, a_h^m = a^\star) + (1-\delta) \mathbb{P}_a(s_{h-1}^m = s_\shortmid) \right) \\
\leq&\delta \mathbb{E}_a[N_\circ - N_\circ^\star] + (\delta + \varepsilon) \mathbb{E}_a[N_\circ^\star] + (1-\delta) \mathbb{E}_a[N_\shortmid],
\end{aligned}
$$

and rearranging the last inequality gives us $\mathbb{E}_a[N_\shortmid] \leq \mathbb{E}_a[N_\circ - N_\circ^\star] + (1 + \frac{\varepsilon}{\delta}) \mathbb{E}_a[N_\circ^\star]$.

For this proof only, define the random variable $W(T_0)$ to be the total reward of the algorithm over the horizon T_0, and define $G(T_0)$ to be the (static) regret with respect to the optimal policy. Since for any algorithm, the probability of staying in state $s_\circ$ under $\mathbb{P}_a(\cdot)$ is no larger than under $\mathbb{P}_{\text{unif}}(\cdot)$, it follows that

$$
\begin{aligned}
&\mathbb{E}_a[W(T_0)] \leq \mathbb{E}_a[N_\shortmid] \leq \mathbb{E}_a[N_\circ - N_\circ^\star] + (1 + \frac{\varepsilon}{\delta}) \mathbb{E}_a[N_\circ^\star] \\
=&\mathbb{E}_a[N_\circ] + \frac{\varepsilon}{\delta} \mathbb{E}_a[N_\circ^\star] \leq \mathbb{E}_{\text{unif}}[N_\circ] + \frac{\varepsilon}{\delta} \mathbb{E}_a[N_\circ^\star] \\
=&T_0 - \mathbb{E}_{\text{unif}}[N_\shortmid] + \frac{\varepsilon}{\delta} \mathbb{E}_a[N_\circ^\star].
\end{aligned} \tag{3.59}
$$

Let $\tau_{\circ\shortmid}^m$ denote the first step that the state transits from state $s_\circ$ to $s_\shortmid$ in the m-th episode; then

$$
\begin{aligned}
\mathbb{E}_{\text{unif}}[N_\shortmid] &= \sum_{m=1}^{M_0} \sum_{h=1}^{H} \mathbb{P}_{\text{unif}}(\tau_{\circ\shortmid}^m = h) \mathbb{E}_{\text{unif}}[N_\shortmid \mid \tau_{\circ\shortmid}^m = h] = \sum_{m=1}^{M_0} \sum_{h=1}^{H} (1-\delta)^{h-1} \delta \mathbb{E}_{\text{unif}}[N_\shortmid \mid \tau_{\circ\shortmid}^m = h] \\
&\geq \sum_{m=1}^{M_0} \sum_{h=1}^{H} (1-\delta)^{h-1} \delta \frac{H-h}{2} = \sum_{m=1}^{M_0} \left(\frac{H}{2} - \frac{1}{2\delta} + \frac{(1-\delta)^H}{2\delta} \right) \\
&\geq \frac{T_0}{2} - \frac{M_0}{2\delta}.
\end{aligned} \tag{3.60}
$$

Since the algorithm is a deterministic mapping from the observation sequence to an action, the random variable $N_\circ^\star$ is also a function of the observations up to time T. In addition, since the immediate reward only

depends on the current state, $N_\circ^\star$ can further be considered as a function of just the state sequence up to T. Therefore, the following lemma from [96], which in turn was adapted from Lemma A.1 in [107], also applies in our setting.

Lemma 29. *(Lemma 13 in [96]) For any finite constant B, let $f : \{s_\circ, s_\shortmid\}^{T_0+1} \to [0, B]$ be any function defined on the state sequence $\mathbf{s} \in \{s_\circ, s_\shortmid\}^{T_0+1}$. Then, for any $0 < \delta \leq \frac{1}{2}$, any $0 < \varepsilon \leq 1 - 2\delta$, and any $a \in \mathcal{A}$, it holds that*

$$\mathbb{E}_a[f(\mathbf{s})] \leq \mathbb{E}_{unif}[f(\mathbf{s})] + \frac{B}{2} \cdot \frac{\varepsilon}{\sqrt{\delta}} \sqrt{2\mathbb{E}_{unif}[N_\circ^\star]}.$$

Since $N_\circ^\star$ itself is a function from the state sequence to $[0, T_0]$, we can apply Lemma 29 and arrive at

$$\mathbb{E}_a[N_\circ^\star] \leq \mathbb{E}_{\text{unif}}[N_\circ^\star] + \frac{T_0}{2} \cdot \frac{\varepsilon}{\sqrt{\delta}} \sqrt{2\mathbb{E}_{\text{unif}}[N_\circ^\star]}. \tag{3.61}$$

From (3.60), we have that $\sum_{a=1}^{SA} \mathbb{E}_{\text{unif}}[N_\circ^\star] = T_0 - \mathbb{E}_{\text{unif}}[N_\shortmid] \leq \frac{T_0}{2} + \frac{M_0}{2\delta}$. By the Cauchy-Schwarz inequality, we further have that $\sum_{a=1}^{SA} \sqrt{2\mathbb{E}_{\text{unif}}[N_\circ^\star]} \leq \sqrt{SA(T_0 + \frac{M_0}{\delta})}$. Therefore, from (3.61), we obtain

$$\sum_{a=1}^{SA} \mathbb{E}_a[N_\circ^\star] \leq \frac{T_0}{2} + \frac{M_0}{2\delta} + \frac{T_0}{2} \cdot \frac{\varepsilon}{\sqrt{\delta}} \sqrt{SA(T_0 + \frac{M_0}{\delta})}.$$

Together with (3.59) and (3.60), it holds that

$$\begin{aligned}
\mathbb{E}_\star[W(T_0)] \leq & \frac{1}{SA} \sum_{a=1}^{SA} \mathbb{E}_a[W(T_0)] \\
\leq & \frac{T_0}{2} + \frac{M_0}{2\delta} + \frac{\varepsilon}{\delta} \frac{1}{SA} \left(\frac{T_0}{2} + \frac{M_0}{2\delta} + \frac{T_0}{2} \cdot \frac{\varepsilon}{\sqrt{\delta}} \sqrt{SA(T_0 + \frac{M_0}{\delta})} \right).
\end{aligned} \tag{3.62}$$

3.15.1 The Un-discounted Setting

Let us now momentarily deviate from the episodic setting and consider the un-discounted setting (with $M_0 = 1$). This is the case of the JAO MDP in Figure 3.3 where there is not reset. We could calculate the stationary distribution and find that the optimal average reward for the JAO MDP is $\frac{\delta+\varepsilon}{2\delta+\varepsilon}$. It is also easy to calculate that the diameter of the JAO MDP is $D = \frac{1}{\delta}$. Therefore, the expected (static) regret with respect to the randomness of a^* can be lower bounded by

$$\begin{aligned}
\mathbb{E}_\star[G(T_0)] = & \frac{\delta + \varepsilon}{2\delta + \varepsilon} T_0 - \mathbb{E}_\star[W(T_0)] \\
\geq & \frac{\varepsilon T_0}{4\delta + 2\varepsilon} - \frac{D}{2} - \frac{\varepsilon D(T_0 + D)}{2SA} - \frac{\varepsilon^2 T_0 D\sqrt{D}}{2\sqrt{SA}} (\sqrt{T_0} + \sqrt{D}).
\end{aligned}$$

By assuming $T_0 \geq DSA$ (which in turn suggests $D \leq \sqrt{\frac{T_0 D}{SA}}$) and setting $\varepsilon = c\sqrt{\frac{SA}{T_0 D}}$ for $c = \frac{3}{40}$, we further have that

$$\begin{aligned}
\mathbb{E}_\star[G(T_0)] \geq & \left(\frac{c}{6} - \frac{c}{2SA} - \frac{cD}{2SAT_0} - \frac{c^2}{2} - \frac{c^2}{2} \sqrt{\frac{D}{T_0}} \right) \sqrt{SAT_0 D} - \frac{D}{2} \\
\geq & \left(\frac{3}{20} c - c^2 - \frac{1}{200} \right) \sqrt{SAT_0 D} = \frac{1}{1600} \sqrt{SAT_0 D}.
\end{aligned}$$

It is easy to verify that our choice of δ and ε satisfies our assumption that $0 < \varepsilon < \delta$. So far, we have recovered the (static) regret lower bound of $\Omega(\sqrt{SAT_0D})$ in the un-discounted setting, which was originally proved in [96].

Based on this result, let us now incorporate the non-stationarity of the MDP and derive a lower bound for the dynamic regret $\mathcal{R}(T)$. Recall that we are constructing the non-stationary environment as a switching-MDP. For each segment of length T_0, the environment is held constant, and the regret lower bound for each segment is $\Omega(\sqrt{SAT_0D})$. At the beginning of each new segment, we uniformly sample a new action a^* at random from the action space $\mathcal{A}$ to be the good action for the new segment. In this case, the learning algorithm cannot use the information it learned during its previous interactions with the environment, even if it knows the switching structure of the environment. Therefore, the algorithm needs to learn a new (static) MDP in each segment, which leads to a dynamic regret lower bound of $\Omega(L\sqrt{SAT_0D}) = \Omega(\sqrt{SATLD})$, where let us recall that L is the number of segments. Every time the good action a^* varies, it will cause a variation of magnitude 2ε in the transition kernel. The constraint of the overall variation budget requires that $2\varepsilon L = \frac{3}{20}\sqrt{\frac{SA}{T_0D}}L \leq \Delta$, which in turn requires $L \leq 4\Delta^{\frac{2}{3}}T^{\frac{1}{3}}D^{\frac{1}{3}}S^{-\frac{1}{3}}A^{-\frac{1}{3}}$. Finally, by assigning the largest possible value to L subject to the variation budget, we obtain a dynamic regret lower bound of $\Omega\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}D^{\frac{2}{3}}T^{\frac{2}{3}}\right)$. This completes the proof of Proposition 1.

3.15.2 The Episodic Settings

Now let us go back to our simplified episodic setting, as depicted in Figure 3.4. One major difference with the previous un-discounted setting is that we might not have time to mix between $s_\circ$ and $s_\imath$ in H steps. (Note that we only need to reach the stationary distribution over the $(s_\circ, s_\imath)$ pair in each step h, rather than the stationary distribution over the entire MDP. In fact, the latter case is never possible because the entire MDP is not aperiodic.) It can be shown that the optimal policy on this MDP has a mixing time of $\Theta\left(\frac{1}{\delta}\right)$ [60], and, hence, we can choose δ to be slightly larger than $\Theta(\frac{1}{H})$ to guarantee sufficient time to mix. All the analysis up to inequality (3.62) carries over to the episodic setting, and essentially we can set δ to be $\Theta\left(\frac{1}{H}\right)$ to get a (static) regret lower bound of $\Omega(\sqrt{SAT_0H})$ in each segment. Another difference with the previous setting lies in the usage of the variation budget. Since we require that all the steps in the same episode should vary simultaneously, it now takes a variation budget of $2\varepsilon H$ each time we switch to a new action a^* at the beginning of a new segment. Therefore, the overall variation budget now puts a constraint of $2\varepsilon HL \leq O(\Delta)$ on the magnitude of each switch. Again, by choosing $\varepsilon = \Theta\left(\sqrt{\frac{SA}{T_0H}}\right)$ and optimizing over possible values of L subject to the budget constraint, we obtain a dynamic regret lower bound of $\Omega\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{1}{3}}T^{\frac{2}{3}}\right)$ in the simplified episodic setting.

Finally, we consider the standard episodic setting as introduced in Section 3.2. In this setting, we essentially will be concatenating H distinct JAO MDPs, each with an independent good action a^*, into a chain like Figure 3.4. The transition kernels in these JAO MDPs are also allowed to vary asynchronously in each step h, although our construction of the lower bound does not make use of this property. As argued similarly in [60], the number of observations for each specific JAO MDP is only T_0/H, instead of T_0. Therefore, we can assign a slightly larger value to ε and the learning algorithm would still not be able to identify the good action given the fewer observations. Setting $\delta = \Theta\left(\frac{1}{H}\right)$ and $\varepsilon = \Theta\left(\sqrt{\frac{SA}{T_0}}\right)$ leads to a (static) regret lower bound of $\Omega(H\sqrt{SAT_0})$ in the stationary RL problem. Again, the transition kernels in all the H JAO MDPs vary simultaneously at the beginning of each new segment. By optimizing L subject to the overall budget constraint $2\varepsilon HL \leq O(\Delta)$, we obtain a dynamic regret lower bound of $\Omega\left(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}}\right)$ in the episodic

setting. This completes our proof of Theorem 13.

3.16 Concluding Remarks

In this chapter, we have considered model-free reinforcement learning in non-stationary episodic MDPs. We have proposed an algorithm named RestartQ-UCB that adopts a simple restarting strategy. RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret of $\widetilde{O}(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}HT^{\frac{2}{3}})$, which nearly matches the information-theoretical lower bound $\Omega(S^{\frac{1}{3}}A^{\frac{1}{3}}\Delta^{\frac{1}{3}}H^{\frac{2}{3}}T^{\frac{2}{3}})$. We have further presented a parameter-free algorithm named Double-Restart Q-UCB that removes the assumption on knowing the variation budget. Numerical experiments have validated the advantages of RestartQ-UCB in terms of both cumulative rewards and computational efficiency. Examples in multi-agent RL and inventory control have been discussed as applications to illustrate the power of our method. An interesting future direction would be to close the $\widetilde{O}(H^{\frac{1}{3}})$ factor gap between the upper and lower bounds that we have established for the non-stationary RL problem. It would also be interesting to explore if non-stationary RL can be helpful in other multi-agent RL or inventory control scenarios.

Chapter 4

Meta-Learning in Markov Games

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 chapter, 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.

4.1 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 [193], 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 [194], [195], 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 [196], [197], 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.

Meta-learning, or learning-to-learn [24]–[27], 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 [194], [195], [198], 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 chapter, 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) [28] 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 4.3.1). 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 4.3.2). 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 4.4.1). In addition, with a properly chosen policy update rule, we prove the non-asymptotic convergence of the exact MAML algorithm in MPGs (Section 4.4.2), despite the convoluted learning dynamics of multiple loosely-coupled agents. 4) For learning coarse correlated equilibria (CCE) in general-sum Markov games (Section 4.5), 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 4.6).

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 [28] 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 [199]–[203]. In particular, [200], [203] have established the convergence of MAML to first-order stationarity for non-convex objectives. [202] has designed an unbiased gradient estimator for MAML in reinforcement learning tasks. Various first-order approximations [28], [200], [204] of MAML have been proposed to avoid the heavy computation of the Hessian. Meta-learning has also been studied in online convex optimization [205]–[208], where regret bounds have been established under different

metrics of task similarity. Another line of research [209]–[211] 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) [7]. Due to the fundamental difficulty of computing NE in generic games [15], most MARL research has focused on learning NE in games with special structures (such as zero-sum Markov games [13], [14], [34], [54], [55], [70], [122], [212], [213] and Markov potential games [89], [91], [214]–[218]) or learning weaker solution concepts such as (coarse) correlated equilibria [16], [17], [72], [123], [219]–[221]. The most relevant works are [121], [122], 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, [91], [214], [215] 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, [222] has studied meta-learning in normal-form games. Under different notions of game similarities, [222] has shown faster convergences of meta-learning in zero-sum, general-sum, and Stackelberg games. [223] has investigated no-regret learning in time-varying zero-sum normal-form games. Compared to [222], [223], 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 [224] and meta-safe RL for quick adaptation in constrained Markov decision processes (CMDPs) under task similarity [225]. Finally, meta-learning has also been empirically applied to many important MARL scenarios, including multi-intersection traffic signal control [198], multi-agent communication with natural language [195], and multi-agent collaboration with first-person pixel observations in open-ended tasks [226].

4.2 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) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $\mathcal{S}$ is the finite state space; (4) $\mathcal{A}_i$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r_i : [H] \times \mathcal{S} \times \mathcal{A}_{\text{all}} \to [0, 1]$ 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 [219], [220] 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 $\boldsymbol{a}_h = (a_{h,1}, \ldots, a_{h,N})$, 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 $\pi = (\pi_1, \ldots, \pi_N) \in \Pi$ 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 [30], 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:

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

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

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

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 [89], [215], [227]. MPGs cover Markov teams [76], 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 chapter, we consider the classic full-information feedback setting [36], [54], [85], [115],

[212], 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 [16], [34], [91], [220].

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 [28] 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 $\mathbb{G}^k$:

$$\pi^{k,t} \leftarrow \psi(\pi^{k,t-1}; \mathbb{G}^k), \forall t \in [T]. \tag{4.1}$$

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:

$$\pi^{k+1,0} \leftarrow \Psi(\{\pi^{k',t}\}_{k' \in [k], t \in [T]}; \mathbb{G}^1, \ldots, \mathbb{G}^k). \tag{4.2}$$

In this chapter, 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.

4.3 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 4.3.1, we start by proposing a new base algorithm (4.1) for zero-sum Markov games that has a refined initialization-dependent convergence guarantee. Based on that, we present our meta-algorithm (4.2) in Section 4.3.2 and establish its sharper convergence rates.

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

Algorithm 16 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 16 performs optimistic online mirror descent (OMD) [114], [115] for policy updates (Lines 5 and 6), in contrast to the popular optimistic follow the regularized leader (FTRL) method in recent MARL policy optimization [121], [122]. 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 16: Optimistic Online Mirror Descent for Zero-Sum Markov Games

Input: Initial policies $\tilde{\mu} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{A})$ and $\tilde{\nu} : [\bar{\tau}] \times [H] \times \mathcal{S} \to \Delta(\mathcal{B})$;
Set stage index $\tau \leftarrow 1$, $t_\tau^{\text{start}} \leftarrow 1$, and $L_\tau \leftarrow H$;
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 $Q_h^\tau \leftarrow \mathbf{0}, \forall h \in [H]$;
for *iteration* $t \leftarrow 1$ *to* T **do**
    Auxiliary policy update: for each step $h \in [H]$ and state $s \in \mathcal{S}$:

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

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

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

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

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

        $\tau \leftarrow \tau + 1$; $\mu_h^t = \hat{\mu}_h^t \leftarrow \tilde{\mu}_h^\tau$, $\nu_h^t = \hat{\nu}_h^t \leftarrow \tilde{\nu}_h^\tau, \forall h \in [H]$;
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 (4.3). If we were to use the celebrated $\alpha_t = \frac{H+1}{H+t}$ learning rate [60] 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 [104]. While a stabilization technique [228] 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 [61]. 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) [23], [61]. 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 38). 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 [17] that can be easily addressed by (optimistic) OMD.

We introduce a few notations before presenting the convergence analysis of Algorithm 16. 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

$$\mathrm{reg}^{\tau}_{h,1}(s) := \max_{\mu_h^{\tau,\dagger}(\cdot|s) \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^\tau \nu_h^j \right\rangle (s). \tag{4.3}$$

The per-state regret $\mathrm{reg}^{\tau}_{h,2}(s)$ for the min-player can be defined symmetrically (see (4.14) in Section 4.8). We define the maximal regret (over the states and the two players) as $\mathrm{reg}^{\tau}_h := \max_{s\in\mathcal{S}} \max_{i=1,2}\{\mathrm{reg}^{\tau}_{h,i}(s)\}$. An upper bound for the per-state regrets is provided in Lemma 37 of Section 4.8, which is useful in the analysis of Algorithm 16. 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 16 is as follows.

Theorem 17. *If Algorithm 16 is run on a two-player zero-sum Markov game for T iterations with a learning rate $\eta \le 1/(8H^2)$, the output policy pair $(\bar{\mu}, \bar{\nu})$ satisfies:*

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

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

$$\text{NE-gap}(\bar{\mu}, \bar{\nu}) \le \frac{768H^5 \log T \log(AB)}{T}. \tag{4.4}$$

Compared to existing results [121], [122], Theorem 17 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 (4.4) still matches the best-known result in the literature [122] except for an extra factor of $O(\log T)$. When suppressing the logarithmic terms, Theorem 17 immediately implies that for any $\varepsilon > 0$, Algorithm 16 takes no more than $T = \widetilde{O}(H^5/\varepsilon)$ steps to learn an ε-approximate NE in an individual zero-sum Markov game.

4.3.2 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 16 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 16 actually delivers.

Our meta-learning procedure proceeds as follows: Within each game $\mathbb{G}^k$, we run Algorithm 16 as our base algorithm (4.1) to find a NE of $\mathbb{G}^k$. In a new game $\mathbb{G}^{k+1}$, the initial policy of Algorithm 16 is given by the following meta-updates in the outer loop (4.2), which essentially averages the best response policies of the previous tasks under α-greedy parameterization:

$$\tilde{\mu}^{k+1} = \frac{1}{k}\sum_{k'=1}^{k} [\mu^{k',\dagger}]_\alpha, \quad \text{and} \quad \tilde{\nu}^{k+1} = \frac{1}{k}\sum_{k'=1}^{k} [\nu^{k',\dagger}]_\alpha. \tag{4.5}$$

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 18. *In a sequence of K two-player zero-sum Markov games, if Algorithm 16 is run for T iterations as the base algorithm and (4.5) with $\alpha = 1/\sqrt{K}$ as the meta-updates, we have*

$$\frac{1}{K}\sum_{k=1}^{K} \text{NE-gap}(\bar{\mu}^k, \bar{\nu}^k) \le \frac{192H^5}{T}\left(\frac{\Delta_{\mu,\nu}}{KH^2} + \frac{10(A+B)\log K}{\sqrt{K}H^2} + \frac{16\log T \log(ABK)}{\sqrt{K}}\right). \tag{4.6}$$

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 (4.6) become negligible. Hence, compared to the best-known results $\widetilde{O}(H^5/T)$ of learning each game individually, Theorem 18 implies a significantly sharper convergence rate when the games are similar, i.e., when $\Delta_{\mu,\nu} \ll KH^2$.

4.4 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 [91] provides initialization-dependent bounds. Building on it, in Section 4.4.1, 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 [28]. In Section 4.4.2, 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 [200], [202], [203].

4.4.1 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 [89], [91], [215], [227]. A detailed description of the setup is provided in Section 4.9 for completeness. Equivalent results for the finite-horizon episodic setting (as we defined in Section 4.2) can be derived in a straightforward way. We choose an existing state-of-the-art algorithm, namely independent projected Q-descent [91], as our base algorithm (4.1). Specifically, in an MPG $\mathbb{G}^k$, each agent independently runs policy gradient ascents to update its own policy for T iterations:

$$\pi_i^{k,t}(\cdot|s) \leftarrow \text{Proj}_{\Delta(\mathcal{A}_i)}\left(\pi_i^{k,t-1}(\cdot|s) + \alpha \bar{Q}_i^{\pi^{k,t-1}}(s,\cdot)\right), \forall t \in [T], \tag{4.7}$$

where $\bar{Q}_i^{\pi}$ is the "averaged" Q-function formally defined in Section 4.9. Let $\Phi(\cdot\,;\mathbb{G}^k)$ denote the potential function of $\mathbb{G}^k$. Through a simple refinement of the analysis in [91], we can establish the following initialization-dependence bound for our base algorithm (4.7).

Proposition 3. *(Theorem 1 in [91]) Suppose that all players in a Markov potential game $\mathbb{G}^k$ run independent projected Q-descent (4.7) for T iterations with $\alpha \leq \frac{(1-\gamma)^4}{8\kappa^3 N A_{\max}}$. 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 Section 4.9.

Proposition 3 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:

$$\Delta_\Phi := \sum_{k=1}^{K-1} \max_{\pi} \left(\Phi(\pi;\mathbb{G}^k) - \Phi(\pi;\mathbb{G}^{k+1})\right). \tag{4.8}$$

As for the meta-updates, we simply instantiate (4.2) 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 (4.8) 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 19. *In a sequence of K Markov potential games, if (4.7) 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.*

4.4.2 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 [28]. 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 [28], [202], [203] can be formulated as

$$\max_{\theta\in\Theta} F_T(\theta) := \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\zeta(\ldots(\zeta(\theta;\mathbb{G}))\ldots);\mathbb{G}\right)\right], \tag{4.9}$$

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 4.2, the MAML procedure consists of two nested stages. For the inner stage (4.1), 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 [218]) that $T = O(1/\varepsilon^2)$ steps will find an ε-approximate NE for each individual MPG. For the outer stage (4.2), MAML directly performs gradient ascents with respect to the meta-objective (4.9). The gradient of F_T can be written in closed-form as

$$\nabla F_T(\theta) = \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\left(\prod_{t=0}^{T-1}\left(I + \alpha\nabla^2\Phi(\theta^{(t)};\mathbb{G})\right)\right)\nabla\Phi(\theta^{(T)};\mathbb{G})\right]. \tag{4.10}$$

A detailed discussion of MAML and its instantiation in our problem are provided in Section 4.9. Most importantly, Section 4.9 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 (4.10) 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 (4.9) in a non-asymptotic manner. A key step of the proof is to prove (rather than assume, as in existing works [200], [203]) that the meta-objective is Lipschitz smooth in the policy parameter θ. The smoothness constant can also be written in a closed form (Lemma 43).

Theorem 20. *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 43 of Section 4.9.*

4.5 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 4.3, we start by developing an initialization-dependent algorithm, followed by investigating the sharper convergence of meta-learning.

Algorithm 17: 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 $\tau \leftarrow 1$, $t_\tau^{\text{start}} \leftarrow 1$, and $L_\tau \leftarrow H$;
Initialize: $\pi_h^0 = \hat{\pi}_h^0 \leftarrow \tilde{\pi}_h^1$, and $Q_h^\tau \leftarrow \mathbf{0}, \forall h \in [H]$;
for *iteration* $t \leftarrow 1$ *to* T **do**
  Auxiliary policy update: for each player $i \in \mathcal{N}$, step $h \in [H]$ and state $s \in \mathcal{S}$:

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

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

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

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

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

    $\tau \leftarrow \tau + 1$; $\pi_h^t = \hat{\pi}_h^t \leftarrow \tilde{\pi}_h^\tau, \forall h \in [H]$;
Output policy: Sample $t \sim \text{Unif}([T])$. Output $\bar{\pi} := \bar{\pi}_1^t$ as defined in Algorithm 18.

Our base algorithm for learning CC in a general-sum Markov game is presented in Algorithm 17. Similar to Algorithm 16 for zero-sum Markov games, Algorithm 17 performs optimistic online mirror descent [114], [115] for policy updates in order to establish initialization-dependent convergence. Algorithm 17 also utilizes stage-based value updates to avoid the need for a complicated no-weighted-regret analysis. Different from Algorithm 16, the output policy $\bar{\pi}$ of Algorithm 17 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 18.

We introduce a few more notations to facilitate the analysis. 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 17. Similar to zero-sum games (4.3), 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 the states and all the players) as

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

Finally, we 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 $\delta_h^t := \max_{i \in \mathcal{N}} \delta_{h,i}^t$. Lemma 46 provides an upper

Algorithm 18: Construction of $\bar{\pi}_h^t$

Input: Policy trajectory $\{\pi_h^t\}_{h\in[H],t\in[T]}$ of Algorithm 17;
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_h^j for step h;
    Set $t \leftarrow j$;

bound of the per-state regret, which further leads us to the following initialization-dependent convergence guarantee of Algorithm 17.

Theorem 21. *If Algorithm 17 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

$$\text{CCE-gap}(\bar{\pi}) \le \frac{12N^{\frac{2}{3}}H^{\frac{8}{3}}\log T \log A_{\max}}{T^{\frac{2}{3}}}. \tag{4.11}$$

Compared to existing results, Theorem 21 directly associates the convergence rate with the quality of the initial policy $\tilde{\pi}$. With uniform initialization, the convergence rate in (4.11) has a slightly worse dependence on T than the best known result $\widetilde{O}(\sqrt{N}H^{11/4}/T^{3/4})$ [121]. 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 47), unlike in optimistic FTRL. Although we believe that our rate in (4.11) 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 chapter.

Let $\tilde{\pi}^k$ and $\bar{\pi}^k$, respectively, denote the initial policy and output policy of Algorithm 17 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 22. *In a sequence of K general-sum Markov games, if Algorithm 17 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.*

4.6 Simulations

In this section, we present our simulation results. We first evaluate our algorithms on a sequence of handcrafted two-player zero-sum Markov games (Section 4.6.1) and Markov potential games (Section 4.6.2). Then, in Section 4.6.3, we further demonstrate the scalability of our methods by considering larger-scale tasks, including a simplified version of the Poker endgame considered in [222] and a 1D linear-quadratic tracking task [229].

4.6.1 Zero-Sum Markov Games

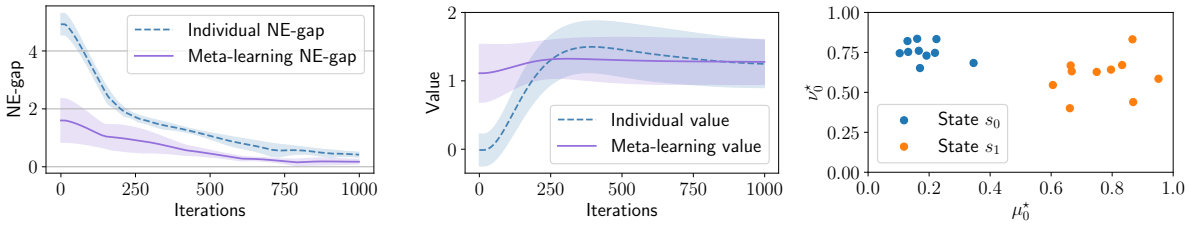
We first evaluate our meta-learning procedure presented in Section 4.3 on a sequence of $K = 10$ two-player zero-sum Markov games. We generate a sequence of K similar games by first specifying a "base game" and then adding random perturbations to its reward function to get K slightly different games. 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 4.1. 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 4.1: 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 4.1 (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 4.1 (c) only plots the NE pairs with respect to the perturbed matrix games as defined in Table 4.1. 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 4.1: 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 16. We evaluate the convergences of the algorithms in terms of 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 4.1 (a) compares the average NE-gap over the K games between individual learning and meta-learning. Figure 4.1 (b) further compares the average values achieved by the two methods. 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.

4.6.2 Markov Potential Games

We now evaluate our meta-learning algorithm from Section 4.4 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 4.2. We add independent $\mathcal{N}(0, 0.1)$ Gaussian perturbation to each entry of the reward matrix to generate $K = 10$ slightly different games.

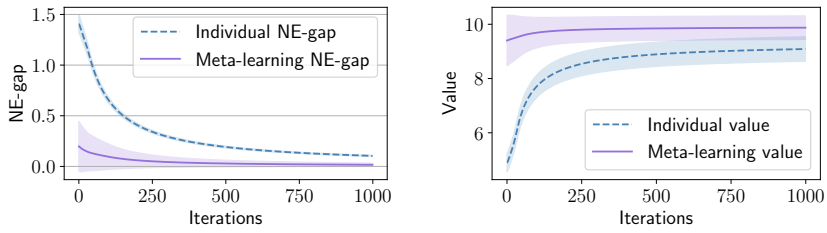
s_0	b_0	b_1
a_0	0.1	0.5
a_1	0.5	1

s_1	b_0	b_1
a_0	0.8	0.2
a_1	0.2	0.8

Table 4.2: Reward matrices for both players in the base game.

The state transition function is defined in the same way as in Section 4.6.1: 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.

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 (4.7). We evaluate the convergences of the algorithms in terms of 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 4.2 (a) compares the average NE-gap over the K games between individual learning and meta-learning. Figure 4.2 (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.



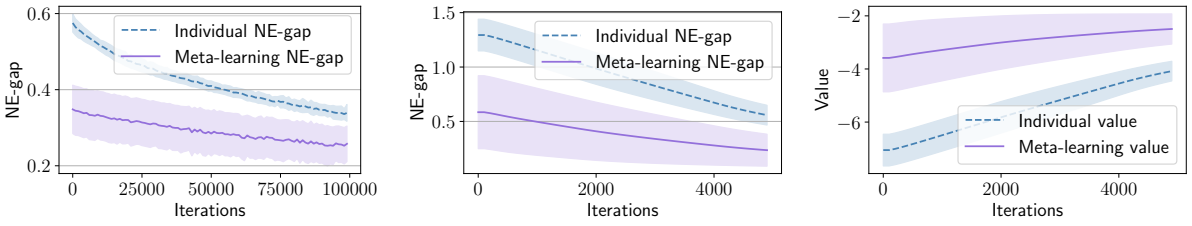
(a) Potential game NE-gap (b) Potential game value

Figure 4.2: 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.

4.6.3 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 [222]. We use a public River endgame ("Endgame A" of [222]) that was released in the Brains vs AI competition [3]. 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 NE-gap over the K games in Figure 4.3(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 NE-gap (b) LQ tracking NE-gap (c) LQ tracking value

Figure 4.3: Average NE-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 [229]–[231], 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 4.3(b) and 4.3(c) demonstrate that our meta-learning method achieves faster NE-gap and value convergences than individual learning in the linear-quadratic tracking task.

4.7 Technical Lemmas

Lemma 30. *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 [232]),

$$\mathrm{KL}(x\|y) - \mathrm{KL}(\tilde{x}\|y) = \mathrm{KL}(x\|\tilde{x}) + \langle \ln\tilde{x} - \ln y, x - \tilde{x}\rangle \tag{4.12}$$

The first term in (4.12) 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 (4.12) is bounded as

$$\langle \ln\tilde{x} - \ln y, x - \tilde{x}\rangle \leq \|\ln\tilde{x} - \ln y\|_\infty \|x - \tilde{x}\|_1. \tag{4.13}$$

We handle the two terms in (4.13) 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 (4.13), 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, (4.12) 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. □

Lemma 31. *(Proposition B.1 of [208]) 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 32. *(Lemma 2 of [202]) 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 33. *(Lemma 3 of [202]) 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 $\|\nabla_\theta \log p(i; \theta)\| \leq B_p$ 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 34. *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 $\|C\| \leq \max_{1\leq i\leq N} \|A_i\|$.*

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

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

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,

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

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

$C_{k-1} = \begin{bmatrix} A_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & A_{k-1} \end{bmatrix}$ is a block diagonal matrix consisting of $k-1$ matrices. Invoking the induction hypothesis for $N = k-1$, we know that $\|C_{k-1}\| \leq \max_{1 \leq i \leq k-1} \|A_i\|$. Finally, using the induction hypothesis for $N = 2$, we conclude that $\|C\| \leq \max\{\|C_{k-1}\|, \|A_k\|\} \leq \max_{1 \leq i \leq k} \|A_i\|$. This completes the induction proof. □

Lemma 35. *Consider a block matrix $A(\theta)$ with $N \times N$ blocks parameterized by $\theta \in \mathbb{R}^d$:*

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

where $A_{i,j}(\theta) \in \mathbb{R}^{d_i \times d_j}, \forall 1 \leq i,j \leq N$ 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., $\|A_{i,j}(\theta) - A_{i,j}(\theta')\| \leq L_{i,j} \|\theta - \theta'\|, \forall \theta, \theta' \in \mathbb{R}^d, 1 \leq i,j \leq N$. Let $L = \max\{L_{i,j} : 1 \leq i,j \leq N\}$. Then, the norm of $A(\theta)$ is also Lipschitz, i.e.,

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

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 $x_i \in \mathbb{R}^{d_i}, \forall 1 \leq i \leq N$. We have

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

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 $\|A_{i,j}(\theta) - A_{i,j}(\theta')\| \leq L_{i,j} \|\theta - \theta'\|$ yields

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

where the last step uses the facts that $L_{i,j} \leq L, \forall 1 \leq i,j \leq N$ 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

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

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

4.8 Proofs for Section 4.3

4.8.1 Proof of Theorem 17

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 16 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 chapter, 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

$$\begin{aligned}
\text{reg}_{h,1}^\tau(s) &:= \max_{\mu_h^{\tau,\dagger} \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^\tau \nu_h^j \right\rangle (s), \\
\text{reg}_{h,2}^\tau(s) &:= \max_{\nu_h^{\tau,\dagger} \in \Delta(\mathcal{B})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left\langle \nu_h^j - \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j \right\rangle (s).
\end{aligned} \tag{4.14}$$

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 16, which we re-state and prove as follows.

Theorem 17. If we run Algorithm 16 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 36 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 [122]. For this purpose, Lemma 37 provides an upper bound on the per-state regrets, while Lemma 38 and Lemma 39 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 $\eta \le 1/(8H^2)$, by plugging in the results of Lemma 37 and Lemma 38 to Lemma 36, we obtain that

$$\begin{aligned}
&\text{NE-gap}(\bar{\mu}, \bar{\nu}) \le \frac{2}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}} L_\tau \max_{s\in\mathcal{S}} \left(\text{reg}_{h,1}^{\tau}(s) + \text{reg}_{h,2}^{\tau}(s)\right) + \frac{2}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}} L_\tau \delta_h^\tau \\
\le &\frac{16H^2}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}} \max_{s\in\mathcal{S}} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s))\right) \\
&+ \frac{192H^2}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}}\sum_{h'=h+1}^{H} \max_{s\in\mathcal{S}} \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s))\right) \\
\le &\frac{192H^2}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}}\sum_{h'=h}^{H} \max_{s\in\mathcal{S}} \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s))\right) \\
\le &\frac{192H^3}{T}\sum_{h=1}^{H}\sum_{\tau=1}^{\bar{\tau}} \max_{s} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s))\right), \qquad (4.15)
\end{aligned}$$

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 $D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) \le \log A, \forall s \in \mathcal{S}, h \in [H]$. A similar result holds for $D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s))$. We can hence obtain that

$$\max_{s} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^\tau(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^\tau(\cdot|s))\right) \le \log(AB). \qquad (4.16)$$

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 $(1+1/H)^{\bar{\tau}} \ge T/H$. Using the Taylor series expansion, one can show that $(1+\frac{1}{H})^H \ge e - \frac{e}{2H}$. Hence, it reduces to finding a minimum $\bar{\tau}$ such that

$$\left(e - \frac{e}{2H}\right)^{\bar{\tau}/H} \ge \frac{T}{H}. \qquad (4.17)$$

One can easily see that any $\bar{\tau} \ge \frac{H\log T}{\log(e/2)}$ satisfies the condition. Together with (4.15) and (4.16), we obtain that

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

This completes the proof of the theorem. □

4.8.2 Supporting Lemmas for Section 4.3

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 16). 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 36. *Let $(\bar{\mu}, \bar{\nu})$ be the output policies of Algorithm 16. 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 [121], we know that

$$\begin{aligned}
&\text{NE-gap}(\bar{\mu}, \bar{\nu}) \\
=&V_1^{\dagger,\bar{\nu}}(s_1) - V_1^\star(s_1) + V_1^\star(s_1) - V_1^{\bar{\mu},\dagger}(s_1) \\
\leq& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \left[\langle \mu_h^\dagger, Q_h^\star \bar{\nu}_h \rangle - \langle \nu_h^\dagger, (Q_h^\star)^\top \bar{\mu}_h \rangle \right](s) \right\} \\
=& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^\star \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^\star)^\top \mu_h^t \rangle \right](s) \right\} \\
\leq& 2 \sum_{h=1}^{H} \max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^{\tau(t)} \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^{\tau(t)})^\top \mu_h^t \rangle \right](s) \right\} + \frac{2}{T} \sum_{h=1}^{H} \sum_{t=1}^{T} \delta_h^t,
\end{aligned} \tag{4.18}$$

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 (4.18), notice that

$$\begin{aligned}
&\max_s \left\{ \max_{\mu_h^\dagger, \nu_h^\dagger} \frac{1}{T} \sum_{t=1}^{T} \left[\langle \mu_h^\dagger, Q_h^{\tau(t)} \nu_h^t \rangle - \langle \nu_h^\dagger, (Q_h^{\tau(t)})^\top \mu_h^t \rangle \right](s) \right\} \\
\leq& \frac{1}{T} \sum_{\tau=1}^{\bar{\tau}} \max_s \left\{ \max_{\mu_h^{\tau,\dagger}, \nu_h^{\tau,\dagger}} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \left[\langle \mu_h^{\tau,\dagger}, Q_h^\tau \nu_h^j \rangle - \langle \nu_h^{\tau,\dagger}, (Q_h^\tau)^\top \mu_h^j \rangle \right](s) \right\} \\
\leq& \frac{1}{T} \sum_{\tau=1}^{\bar{\tau}} L_\tau \max_s \left(\text{reg}_{h,1}^\tau(s) + \text{reg}_{h,2}^\tau(s) \right).
\end{aligned} \tag{4.19}$$

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 (4.19) 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 (4.18), we use the fact that with stage-based value updates, the value estimation error δ_h^t does not change within a stage. Therefore,

$$\frac{2}{T} \sum_{h=1}^{H} \sum_{t=1}^{T} \delta_h^t = \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} \sum_{j=t_\tau^{\text{start}}}^{t_\tau^{\text{end}}} \delta_h^j = \frac{2}{T} \sum_{h=1}^{H} \sum_{\tau=1}^{\bar{\tau}} L_\tau \delta_h^\tau. \tag{4.20}$$

Finally, substituting (4.19) and (4.20) back to (4.18) completes the proof. □

Lemma 37. *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:*

$$\begin{aligned}
\mathrm{reg}_{h,1}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2 \\
& - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2, \qquad (4.21) \\
\mathrm{reg}_{h,2}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2 \\
& - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2. \qquad (4.22)
\end{aligned}$$

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

$$\begin{aligned}
\mathrm{reg}_{h,1}^{\tau}(s) + \mathrm{reg}_{h,2}^{\tau}(s) \leq & \frac{1}{\eta L_\tau} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) \right) \\
& - \sum_{j=t_\tau^{start}}^{t_\tau^{end}} \frac{4\eta H^3}{L_\tau} \left(\|\nu_h^j(\cdot|s) - \nu_h^{j-1}(\cdot|s)\|_1^2 + \|\mu_h^j(\cdot|s) - \mu_h^{j-1}(\cdot|s)\|_1^2 \right). \qquad (4.23)
\end{aligned}$$

Proof. We prove the regret bound for the max-player, i.e., $\mathrm{reg}_{h,1}^{\tau}(s)$. The bound for the min-player holds analogously. Notice that the policy update steps in Algorithm 16 are exactly the same as the optimistic online mirror descent algorithm [114], [115], 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 [121], [122]. This allows us to directly apply the standard optimistic OMD results (e.g., Lemma 1 in [114] and Proposition 5 in [115]) to obtain

$$\begin{aligned}
\mathrm{reg}_{h,1}^{\tau}(s) = & \max_{\mu_h^{\tau,\dagger} \in \Delta(\mathcal{A})} \frac{1}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\langle \mu_h^{\tau,\dagger} - \mu_h^j, Q_h^\tau \nu_h^j \right\rangle (s) \\
\leq & \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{\eta}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\| [Q_h^\tau \nu_h^j - Q_h^\tau \nu_h^{j-1}](s,\cdot) \right\|_\infty^2 \qquad (4.24) \\
& - \frac{1}{8\eta L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2. \qquad (4.25)
\end{aligned}$$

To further upper bound the term in (4.24), 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 (4.25) yields

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

This completes the proof of (4.21). The regret bound in (4.22) can be shown via symmetry.

Combining (4.21) and (4.22) leads to

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

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

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

This completes the proof of the lemma. □

Lemma 38. *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 39 (similar to Lemma C.2 of [121]) to establish the following recursion for the value estimation error:

$$\delta_h^t \leq \delta_{h+1}^{\tau-1} + \mathrm{reg}_{h+1}^{\tau-1}, \tag{4.26}$$

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 37, we can upper bound the individual regrets $\mathrm{reg}_{h,1}^{\tau}(s)$ and $\mathrm{reg}_{h,2}^{\tau}(s)$ by

$$\mathrm{reg}_{h,1}^{\tau}(s) \leq \frac{1}{\eta L_\tau} D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\| \nu_h^j(\cdot \mid s) - \nu_h^{j-1}(\cdot \mid s) \right\|_1^2, \tag{4.27}$$

$$\mathrm{reg}_{h,2}^{\tau}(s) \leq \frac{1}{\eta L_\tau} D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) + \frac{2\eta H^2}{L_\tau} \sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}} \left\| \mu_h^j(\cdot \mid s) - \mu_h^{j-1}(\cdot \mid s) \right\|_1^2. \tag{4.28}$$

where we have dropped the negative terms in (4.21) and (4.22). Following a similar approximate non-negativity argument as in Lemma 5 of [122] (reproduced in Lemma 40 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 (4.23) in Lemma 37, we obtain that

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

Since the above inequality holds for any state $s \in \mathcal{S}$, substituting it back to (4.27) and (4.28) yields

$$\mathrm{reg}_h^{\tau} \leq \max_s \frac{3}{2\eta L_\tau} \left(D_R(\mu_h^{\tau,\dagger}, \tilde{\mu}_h^{\tau}(\cdot|s)) + D_R(\nu_h^{\tau,\dagger}, \tilde{\nu}_h^{\tau}(\cdot|s)) \right) + \frac{\delta_h^{\tau}}{H}. \tag{4.29}$$

We can further substitute the regret bound above back to the recursion 4.26 to get that

$$\delta_h^{\tau} \leq \frac{3}{2\eta L_{\tau-1}} \max_s \left(D_R(\mu_{h+1}^{\tau-1,\dagger}, \tilde{\mu}_{h+1}^{\tau-1}(\cdot|s)) + D_R(\nu_{h+1}^{\tau-1,\dagger}, \tilde{\nu}_{h+1}^{\tau-1}(\cdot|s)) \right) + (1 + \frac{1}{H})\delta_{h+1}^{\tau-1}, \tag{4.30}$$

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 (4.30) leads us to the following result:

$$\begin{aligned}
\delta_h^{\tau} \leq &\sum_{h'=h+1}^{H} \frac{3}{2\eta L_{\tau-h'+h}} \left(1 + \frac{1}{H}\right)^{h'-h-1} \max_s \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right) \\
\leq &\frac{3}{2\eta L_\tau} \sum_{h'=h+1}^{H} \left(1 + \frac{1}{H}\right)^{2(h'-h)-1} \max_s \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right) \\
\leq &\frac{3}{2\eta L_\tau} \sum_{h'=h+1}^{H} \left(1 + \frac{1}{H}\right)^{2H} \max_s \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right) \\
\leq &\frac{12}{\eta L_\tau} \sum_{h'=h+1}^{H} \max_s \left(D_R(\mu_{h'}^{\tau-h'+h,\dagger}, \tilde{\mu}_{h'}^{\tau-h'+h}(\cdot|s)) + D_R(\nu_{h'}^{\tau-h'+h,\dagger}, \tilde{\nu}_{h'}^{\tau-h'+h}(\cdot|s)) \right),
\end{aligned} \tag{4.31}$$

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 (4.31) is due to the fact that $(1 + 1/H)^H \leq e \approx 2.71828$. This completes the proof of the lemma. □

Lemma 39. *(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 [121]. 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 $\text{reg}_{h+1}^{\tau-1}$ and $\delta_{h+1}^{\tau-1}$, the above inequality further leads to

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

where the last step is due to the value update rule in Algorithm 16. This implies that

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

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

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

Combining both directions yields the desired result. □

Lemma 40. *(Approximate non-negativity) For any $\tau \in [\bar{\tau}]$ and $h \in [H]$, we have that*

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

Proof. This lemma can be considered as a stage-based variant of Lemma 5 in [122]. From the definitions of

$\mathrm{reg}_{h,1}^{\tau}(s)$ and $\mathrm{reg}_{h,2}^{\tau}(s)$, we have that

$$
\begin{aligned}
&\mathrm{reg}_{h,1}^{\tau}(s)+\mathrm{reg}_{h,2}^{\tau}(s)\\
=&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}}\frac{1}{L_\tau}\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle\mu_h^{\tau,\dagger},Q_h^{\tau}\nu_h^{j}\right\rangle-\left\langle\nu_h^{\tau,\dagger},(Q_h^{\tau})^{\top}\mu_h^{j}\right\rangle\right)(s)\\
=&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}}\frac{1}{L_\tau}\Bigg[\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle\mu_h^{\tau,\dagger},Q_h^{\star}\nu_h^{j}\right\rangle-\left\langle\nu_h^{\tau,\dagger},(Q_h^{\star})^{\top}\mu_h^{j}\right\rangle\right)(s)\\
&+\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle\mu_h^{\tau,\dagger},(Q_h^{\tau}-Q_h^{\star})\nu_h^{j}\right\rangle-\left\langle\nu_h^{\tau,\dagger},(Q_h^{\tau}-Q_h^{\star})^{\top}\mu_h^{j}\right\rangle\right)(s)\Bigg]\\
\geq&\max_{\mu_h^{\tau,\dagger},\nu_h^{\tau,\dagger}}\frac{1}{L_\tau}\left[\sum_{j=t_\tau^{\mathrm{start}}}^{t_\tau^{\mathrm{end}}}\left(\left\langle\mu_h^{\tau,\dagger},Q_h^{\star}\nu_h^{j}\right\rangle-\left\langle\nu_h^{\tau,\dagger},(Q_h^{\star})^{\top}\mu_h^{j}\right\rangle\right)(s)\right]-2\delta_h^{\tau},
\end{aligned}
\tag{4.32}
$$

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 (4.32) is always non-negative:

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

Plugging the above inequality back into (4.32) completes the proof. □

4.8.3 Proof of Theorem 18

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 17) can be written more succinctly as

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

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 16 on a sequence of K games, we have that

$$\frac{1}{K}\sum_{k=1}^{K}\text{NE-gap}(\bar{\mu}^k, \bar{\nu}^k) \le \frac{192H^3}{KT}\sum_{k=1}^{K}\left(\text{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right) + \text{KL}\left(\nu^{k,\dagger}\|\tilde{\nu}^k\right)\right). \tag{4.33}$$

In the following, we will focus on the term for the maximizing player in (4.33). 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 30, we obtain that

$$\frac{1}{K}\sum_{k=1}^{K}\text{KL}\left(\mu^{k,\dagger}\|\tilde{\mu}^k\right) \le \frac{1}{K}\sum_{k=1}^{K}\text{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) + 4H\bar{\tau}\alpha\ln\frac{A}{\alpha}. \tag{4.34}$$

Notice that the conditions of Lemma 30 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

$$\begin{aligned}\sum_{k=1}^{K}\text{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) &= \min_{\mu}\sum_{k=1}^{K}\text{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right) + \min_{\mu}\sum_{k=1}^{K}\left(\text{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) - \text{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right)\right)\\ &\le \min_{\mu}\sum_{k=1}^{K}\text{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right) + \frac{8A(1+\ln K)}{\alpha},\end{aligned} \tag{4.35}$$

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 (4.35), which reduces to bounding the following regret where the loss functions are given by the Bregman divergences:

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

It is known that the unique minimum of $\sum_{k'=1}^{k}\text{KL}([\mu^{k',\dagger}]_\alpha\|\cdot)$ is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\mu^{k',\dagger}]_\alpha$ (see Proposition 1 of [233] 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 $\sum_{k=1}^{K}\text{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$. We can then invoke the logarithmic regret guarantee of FTL with respect to Bregman divergences, which was established in [208] and was reproduced as Lemma 31 in Section 4.7 for completeness. To show that Lemma 31 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 31. 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 $\text{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$ is $\frac{2A}{\alpha}$ within the $\frac{\alpha}{A}$-restricted domain. This allows us to apply Lemma 31 to obtain the result in (4.35).

Moving forward from (4.35), we again apply the property that the unique minimum of $\sum_{k=1}^{K}\text{KL}([\mu^{k,\dagger}]_\alpha\|\cdot)$

is attained at $\mu = \frac{1}{K}\sum_{k=1}^{K}[\mu^{k,\dagger}]_\alpha$, which leads to

$$\begin{aligned}\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\tilde{\mu}^k\right) &\le \min_{\mu}\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|\mu\right)+\frac{8A(1+\ln K)}{\alpha}\\ &=\sum_{k=1}^{K}\mathrm{KL}\left([\mu^{k,\dagger}]_\alpha\|[\mu^\star]_\alpha\right)+\frac{8A(1+\ln K)}{\alpha}\\ &\le(1-\alpha)\sum_{k=1}^{K}\mathrm{KL}\left(\mu^{k,\dagger}\|\mu^\star\right)+\frac{8A(1+\ln K)}{\alpha},\end{aligned}\tag{4.36}$$

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 (4.36) to (4.34) 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 (4.33) 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 (4.17) 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. □

4.9 Proofs for Section 4.4

4.9.1 Definitions

To be consistent with existing results in the literature, we consider an infinite-horizon γ-discounted reward setting for MPGs [89], [91], [215], [227]. 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:

$$V_i^s(\theta; \mathbb{G}) := \mathbb{E}_{\theta, \mathbb{G}}\Big[\sum_{t=0}^{\infty} \gamma^t r_i(s^t, a^t) \mid s^0 = s\Big], \tag{4.37}$$
$$Q_i^{s,a}(\theta; \mathbb{G}) := \mathbb{E}_{\theta, \mathbb{G}}\Big[\sum_{t=0}^{\infty} \gamma^t r_i(s^t, a^t) \mid s^0 = s, a^0 = a\Big].$$

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 11. *(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 $\theta_i, \theta_i' \in \Theta_i$, $\theta_{-i} \in \Theta_{-i}$:

$$\Phi_s(\theta_i, \theta_{-i}; \mathbb{G}) - \Phi_s(\theta_i', \theta_{-i}; \mathbb{G}) = V_i^s(\theta_i, \theta_{-i}; \mathbb{G}) - V_i^s(\theta_i', \theta_{-i}; \mathbb{G}). \tag{4.38}$$

Intuitively, MPGs capture the variations of the agents' individual values by a single global potential function.

[1] For example, direct parameterization requires that $\theta_{s,a_i} \geq 0$ 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 $\Theta_i = \mathbb{R}^{|\mathcal{S}||\mathcal{A}_i|}$.

MPGs cover Markov teams [76] 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 $\kappa(\mathbb{G}) = \sup_{\theta\in\Theta} \left\| d_\theta^\rho(\cdot\,; \mathbb{G})/\rho \right\|_\infty$. For a set $\mathcal{G}$ of games, we let $\kappa = \sup_{\mathbb{G}\in\mathcal{G}} \kappa(\mathbb{G})$.

4.9.2 Proof of Theorem 19

Proof. Proposition 3 implies that if the agents run projected Q-descent on the Markov potential game $\mathbb{G}^k$ for T iterations, we have

$$\sum_{t=0}^{T-1} \max_{i\in\mathcal{N}} \left(\max_{\theta_i'\in\Theta_i} V_i^\rho(\theta_i', \theta_{-i}^{k,t}; \mathbb{G}^k) - V_i^\rho(\theta_i^{k,t}, \theta_{-i}^{k,t}; \mathbb{G}^k) \right) \leq \sqrt{\frac{\kappa(\mathbb{G}^k) T (\Phi(\theta^{k,T}; \mathbb{G}^k) - \Phi(\theta^{k,0}, \mathbb{G}^k))}{\alpha(1-\gamma)^2}}. \tag{4.39}$$

From the Cauchy-Schwarz inequality, we have that

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

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 (4.39), we have that

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

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

4.9.3 Model-Agnostic Meta-Learning in Markov Potential Games

In what follows, we study meta-learning in MPG under the same formulation as MAML [28], [200], [203]. 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

$$\max_{\theta\in\Theta} F_1(\theta) := \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\theta + \alpha\nabla\Phi(\theta;\mathbb{G});\mathbb{G}\right)\right], \tag{4.40}$$

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 (4.40) can be written as

$$\max_{\theta\in\Theta} F_T(\theta) := \mathbb{E}_{\mathbb{G}\sim p(\mathcal{G})}\left[\Phi\left(\zeta(\ldots(\zeta(\theta;\mathbb{G}))\ldots);\mathbb{G}\right)\right], \tag{4.41}$$

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:

$$\theta^{k,t+1}(\mathbb{G}) \leftarrow \psi(\theta^{k,t}(\mathbb{G});\mathbb{G}), \text{ for } 0 \le t \le T-1, \tag{4.42}$$

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

$$\theta^{k+1} \leftarrow \Psi(\theta^k, \mathcal{G}), \tag{4.43}$$

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 [201] 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 [200], [202], [203].

In the following, we develop a meta-learning procedure (ψ, Ψ) that finds a stationary point of the meta-objective (4.41) 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

$$\theta_i^{k,t+1}(\mathbb{G}) \leftarrow \theta_i^{k,t}(\mathbb{G}) + \alpha \nabla_{\theta_i} V_i^{\rho}(\theta^{k,t}(\mathbb{G}); \mathbb{G}), \forall 0 \leq t \leq T-1. \tag{4.44}$$

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 [215], [234], the gradient $\nabla_{\theta_i} V_i^{\rho}(\theta; \mathbb{G})$ can be calculated as

$$\frac{\partial V_i^{\rho}(\theta; \mathbb{G})}{\partial \theta_i(a_i|s)} = \frac{1}{1-\gamma} d_{\rho}^{\theta}(s; \mathbb{G}) \pi_{\theta_i}(a_i|s) \bar{A}_i^{s,a_i}(\theta; \mathbb{G}), \tag{4.45}$$

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 [235]. For simplicity, we assume that the exact policy gradients are given. It follows from the definition of the potential function (4.38) 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 (4.44) is equivalent to running centralized gradient ascents with respect to the potential function (4.42). Hence, the base algorithm for each individual game can be executed in a decentralized way. Finally, we invoke Theorem 5 of [218] to show that under mild assumptions, our policy gradient updates with softmax parameterization (4.44) 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 (4.40). The gradient of F_T can be written as

$$\nabla F_T(\theta) = \mathbb{E}_{\mathbb{G} \sim p(\mathcal{G})} \left[\Big(\prod_{t=0}^{T-1} \big(I + \alpha \nabla^2 \Phi(\theta^{(t)}(\mathbb{G}); \mathbb{G}) \big) \Big) \nabla \Phi(\theta^{(T)}(\mathbb{G}); \mathbb{G}) \right], \tag{4.46}$$

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 (4.43) as

$$\theta^{k+1} \leftarrow \theta^k + \frac{\eta}{|\mathcal{G}|} \sum_{\mathbb{G} \in \mathcal{G}} \Big(\prod_{t=0}^{T-1} \big(I + \alpha \nabla^2 \Phi(\theta^{k,t}(\mathbb{G}); \mathbb{G}) \big) \Big) \nabla \Phi(\theta^{k,T}(\mathbb{G}); \mathbb{G}), \tag{4.47}$$

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 44). 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 [43], a popular strategy used for training MARL algorithms.

In order to establish the convergence of (4.47) to the stationary point of the meta-objective (4.40), we first show the smoothness of the meta-objective through the following sequence of lemmas.

Lemma 41. *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 42. *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* $L_H := \frac{56N^{3/2}}{(1-\gamma)^4}$.

Lemma 43. *(Meta-objective smoothness). Consider running (4.44) with softmax parameterization and $\alpha = \frac{(1-\gamma)^3}{2N\gamma A_{\max}}$ as the inner stage and running (4.47) as the outer stage. Then, the meta-objective (4.41) 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 20).

4.9.4 Proof of Lemma 41

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., [235])

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

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

$$\|\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|s)\| \leq \sqrt{2}. \tag{4.48}$$

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., [202] 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 [202], reproduced as Lemmas 32 and 33 in Section 4.7. Since $\|\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|s)\| \leq 2$, from Lemma 33, 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

$\|A\|_2^2 \le \|A\|_1 \|A\|_\infty$, we can conclude that

$$\left\| \left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}] \right) \left(\mathbf{1}_{s,a_i'} - \mathbb{E}_{a_i'' \sim \pi_{\theta_i}(a_i''|s)}[\mathbf{1}_{s,a_i''}] \right)^\top \right\| \le 2. \tag{4.49}$$

Further, by Lemma 32, the term in (4.49) is Lipschitz continuous with constant 8. By applying Lemma 33 one more time, we know that

$$\left\| \nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s) \right\| \le 2, \text{ and } \left\| \nabla^2_{\theta_i} \log \pi_{\theta_i}(a_i|s) - \nabla^2_{\theta_i'} \log \pi_{\theta_i'}(a_i|s) \right\| \le 12 \left\| \theta_i - \theta_i' \right\|. \tag{4.50}$$

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 34 to show that

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

This completes the proof of result (ii). To show result (iii), we again apply Lemma 34 to conclude that

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

where the last step is by (4.50). This completes the proof of the lemma. □

4.9.5 Proof of Lemma 42

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 42, 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:

$$\nabla^2\Phi(\theta) = \begin{bmatrix} \nabla^2_{1,1}\Phi(\theta) & \dots & \nabla^2_{1,N}\Phi(\theta) \\ \vdots & \ddots & \vdots \\ \nabla^2_{N,1}\Phi(\theta) & \dots & \nabla^2_{N,N}\Phi(\theta) \end{bmatrix}, \tag{4.51}$$

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 [236]. For clarity of presentation, we defer its proof to Section 4.9.6.

Lemma 44. *Each matrix block $\nabla^2_{i,j}\Phi(\theta)$ in the policy Hessian matrix (4.51) 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*

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

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 Section 4.9.7.

Lemma 45. *Each matrix block* $\nabla^2_{i,j}\Phi(\theta)$ *in the policy Hessian matrix* (4.51) *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 44 and Lemma 45, we are now ready to prove Lemma 42.

Proof (of Lemma 42).

Proof of (i): From the definition of the potential function (4.38), we know that $\nabla_{\theta_i}\Phi(\theta) = \nabla_{\theta_i}V_i^\rho(\theta)$, and hence $\nabla\Phi(\theta) = (\nabla_{\theta_1}V_1^\rho(\theta),\ldots,\nabla_{\theta_N}V_N^\rho(\theta))$. 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 (4.48) from Lemma 42 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 [218].

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 45 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 $L_{ij} \le \frac{56\sqrt{N}}{(1-\gamma)^4}$. We can then use Lemma 35 to conclude that $\nabla^2\Phi(\theta)$ is also Lipschitz

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

This completes the proof of Lemma 42. □

4.9.6 Proof of Lemma 44

Proof. The proof follows steps similar to those used in the proof of Theorem 3 in [236]. 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

$$p_\theta(s^{0:t}, a^{0:t} \mid \rho) := \mathbb{P}_\theta(s^{0:t}, a^{0:t} | s^0 \sim \rho) = \rho(s^0) \prod_{\tau=0}^{t-1} \left(\pi_\theta(a^\tau|s^\tau) P(s^{\tau+1}|s^\tau, a^\tau)\right) \pi_\theta(a^t|s^t). \tag{4.52}$$

From the definition in (4.37), 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 (4.38), 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

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

From (4.52), we can see that $\nabla_{\theta_i \theta_j}^2 \log p_\theta(s^{0:t}, a^{0:t} \mid \rho) = \sum_{\tau=0}^t \nabla_{\theta_i \theta_j}^2 \log \pi_\theta(a^\tau|s^\tau)$. Hence, the first term in the above equation can be written as

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

The second term can be written as

$$
\begin{aligned}
② = &\sum_{t=0}^{\infty}\sum_{\tau=0}^{t}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^\tau|s^\tau)\nabla_{\theta_j}^\top\log\pi_\theta(a^\tau|s^\tau)r_i(s^t,a^t)\\
&+\sum_{t=0}^{\infty}\sum_{\tau_2=0}^{t}\sum_{\tau_1=0}^{\tau_2-1}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^{\tau_1}|s^{\tau_1})\nabla_{\theta_j}^\top\log\pi_\theta(a^{\tau_2}|s^{\tau_2})r_i(s^t,a^t) \qquad (4.53)\\
&+\sum_{t=0}^{\infty}\sum_{\tau_1=0}^{t}\sum_{\tau_2=0}^{\tau_1-1}\sum_{a^{0:t}}\sum_{s^{0:t}}\gamma^t p_\theta(s^{0:t},a^{0:t}|\rho)\nabla_{\theta_i}\log\pi_\theta(a^{\tau_1}|s^{\tau_1})\nabla_{\theta_j}^\top\log\pi_\theta(a^{\tau_2}|s^{\tau_2})r_i(s^t,a^t).
\end{aligned}
$$

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 (4.53) is equal to $\mathcal{H}_1^{i,j}(\theta)$. The second and third terms on the RHS of (4.53) 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 [236]. □

4.9.7 Proof of Lemma 45

Proof. Recall from Lemma 44 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 (4.48) from Lemma 41 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 41 implies that $\nabla_{\theta_i}\log\pi_\theta(a|s)$ is bounded by $\sqrt{2}$ and is 2-Lipschitz continuous. Further using Lemma 32, we can conclude that

$$\left\|h_1^{i,j}(\theta)\right\|\le\frac{2}{1-\gamma}\quad\text{and}\quad\left\|h_1^{i,j}(\theta)-h_1^{i,j}(\theta')\right\|\le\frac{2\sqrt{2}(2-\gamma)}{(1-\gamma)^2}\left\|\theta-\theta'\right\|. \qquad (4.54)$$

(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 $i \neq j$, we simply have $\nabla^2_{\theta_i\theta_j} \log \pi_\theta(a|s) = 0$. For $i = j$, we know from (4.50) that $\|\nabla^2_{\theta_i\theta_j} \log \pi_\theta(a|s)\| \leq 2$, and $\|\nabla^2_{\theta_i\theta_j} \log \pi_\theta(a|s) - \nabla^2_{\theta_i\theta_j} \log \pi_{\theta'}(a|s)\| \leq 12 \|\theta_i - \theta_i'\|$. Therefore, we obtain from Lemma 32 that

$$h_2^{i,j}(\theta) = 0, \text{ if } i \neq j; \text{ and } \left\|h_2^{i,j}(\theta)\right\| \leq \frac{2}{1-\gamma}, \left\|h_2^{i,j}(\theta) - h_2^{i,j}(\theta')\right\| \leq \frac{8(2-\gamma)}{(1-\gamma)^2} \|\theta - \theta'\|, \text{ if } i = j. \tag{4.55}$$

(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 [237] 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 (4.37),

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

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

$$\begin{aligned}
&\left|\nabla_{\theta_j} Q_i^{s,a}(\theta) - \nabla_{\theta_j} Q_i^{s,a}(\theta')\right| \\
\leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t \left| p_\theta(s^{0:t}, a^{0:t}|s, a) \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau) \right| \\
\leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t \left| p_\theta(s^{0:t}, a^{0:t}|s, a) - p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \right| \left\| \sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) \right\| \qquad (4.56) \\
&+ \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s, a) \left\| \sum_{\tau=1}^{t} \left(\nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau) \right) \right\|. \qquad (4.57)
\end{aligned}$$

In the following, we upper bound each of the two terms above separately. To analyze (4.56), we first apply

the mean-value theorem to the function $\prod_{\tau=1}^{t} \pi_\theta(a^\tau|s^\tau)$ of θ and obtain

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

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 41 (i). Using the above property, we obtain

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

Substituting the above equation back into (4.56) yields

$$\begin{aligned}(4.56) \leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \sqrt{2Nt}\gamma^t p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s,a) \cdot \left\|\sum_{\tau=1}^{t} \nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau)\right\| \cdot \|\theta-\theta'\| \\ \leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} 2\sqrt{N}t^2\gamma^t p_{\tilde{\theta}}(s^{0:t}, a^{0:t}|s,a)\,\|\theta-\theta'\|\,,\end{aligned}$$

where the second step uses (4.48) from Lemma 41 and the fact that π_θ is a product policy.

To upper bound (4.57), we apply Lemma 41 (ii) and obtain

$$\begin{aligned}(4.57) \leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} \gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s,a) \sum_{\tau=1}^{t} \left\|\nabla_{\theta_j} \log \pi_\theta(a^\tau|s^\tau) - \nabla_{\theta_j} \log \pi_{\theta'}(a^\tau|s^\tau)\right\| \\ \leq& \sum_{t=0}^{\infty} \sum_{a^{0:t}} \sum_{s^{0:t}} 2t\gamma^t p_{\theta'}(s^{0:t}, a^{0:t}|s,a)\,\|\theta-\theta'\|\,.\end{aligned}$$

Substituting the above upper bounds back into (4.56) and (4.57), 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 32, we can conclude that

$$\left\|h_{12}^{i,j}(\theta)\right\| \leq \frac{2\gamma}{(1-\gamma)^2} \text{ and } \left\|h_{12}^{i,j}(\theta) - h_{12}^{i,j}(\theta')\right\| \leq \frac{6\sqrt{2N}\gamma(1+\gamma)}{(1-\gamma)^3}. \tag{4.58}$$

(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 (4.54), (4.55), and (4.58) that

$$\|h(\theta)\| \leq \frac{4}{(1-\gamma)^2}, \text{ and } \|h(\theta) - h(\theta')\| \leq \frac{50\sqrt{N}}{(1-\gamma)^3} \|\theta - \theta'\|. \tag{4.59}$$

Recall from Lemma 44 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,

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

The third step uses the upper bounds from (4.59). The fourth step can be derived by using the following result from Equation (A.67) of [237]:

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

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

4.9.8 Proof of Lemma 43

Proof. Recall from (4.46) 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

$$\Big(\prod_{t=0}^{T-1}\big(I+\alpha\nabla^2\Phi(\theta^{(t)}(\mathbb{G});\mathbb{G})\big)\Big)\nabla\Phi(\theta^{(T)}(\mathbb{G});\mathbb{G}) \tag{4.60}$$

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 (4.60). First, from Lemma 42(ii), we know that $\left\|I+\alpha\nabla^2\Phi(\theta^{(t)})\right\| \le 1+\alpha L_G, \forall 0\le t\le T-1$. 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 $\left\|I+\alpha\nabla^2\Phi(\theta^{(t)})\right\| \le 1+\alpha L_G, \forall 0\le t\le T-1$, we know that $\theta^{(t)}$ is Lipschitz continuous with constant $(1+\alpha L_G)^t$. Further, combining Lemma 42 (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 $T\ge 2$, Lemma 32 thus implies that the $\prod_{t=0}^{T-1}(I+\alpha\nabla^2\Phi(\theta^{(t)}))$ factor from (4.60) 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 (4.60), we know from Lemma 42(i) that it is bounded by B_G. Using Lemma 42(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 32 to obtain that (4.60) 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 $L_F = (\alpha T B_G L_H + L_G)2^{2T}$. □

4.9.9 Proof of Theorem 20

Proof. Based on the aforementioned series of lemmas, we are now ready to establish Theorem 20. The proof follows from standard analysis in non-convex optimization. Since the meta-objective function is L_F-smooth (Lemma 43), the smoothness property implies that

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

Using the outer stage update rule (4.47) 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. □

4.10 Proofs for Section 4.5

4.10.1 Proof of Theorem 21

Proof. From the construction of $\bar{\pi}$ (Algorithm 18) and the definition of CCE-gap, we have

$$\begin{aligned}
\text{CCE-gap}(\bar{\pi}) =& \max_{i \in \mathcal{N}} V_{1,i}^{\dagger, \bar{\pi}_{-i}}(s_1) - V_{1,i}^{\bar{\pi}}(s_1) \\
\leq& \frac{1}{T} \sum_{t=1}^{T} \max_{i \in \mathcal{N}} \max_{s \in \mathcal{S}} \left(V_{1,i}^{\dagger, \bar{\pi}_{1,-i}^t}(s) - V_{1,i}^{\bar{\pi}_1^t}(s) \right) \\
\leq& \frac{1}{T} \sum_{t=1}^{T} \delta_1^t.
\end{aligned}$$

Using Lemma 48, the above term can be further bounded by

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

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 17 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 $\pi^{\tau,\dagger}_{h,i}$ lies in the simplex, when we initialize $\tilde{\pi}^{\tau}_{h,i}(\cdot|s) = \mathbf{1}/A_i$ to be a uniform distribution, we naturally have $D_R(\pi^{\tau,\dagger}_{h,i}, \tilde{\pi}^{\tau}_{h,i}(\cdot|s)) \le \log A_i, \forall i \in \mathcal{N}, s \in \mathcal{S}$, 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 $(1+1/H)^{\bar{\tau}} \ge T/H$. Using the Taylor series expansion, one can show that $(1+\frac{1}{H})^H \ge e - \frac{e}{2H}$. Hence, it reduces to finding a minimum $\bar{\tau}$ such that

$$\left(e - \frac{e}{2H}\right)^{\bar{\tau}/H} \ge \frac{T}{H}. \tag{4.61}$$

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

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

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

4.10.2 Supporting Lemmas for Section 4.5

Lemma 46. *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:*

$$\text{reg}^{\tau}_{h,i}(s) \le \frac{1}{\eta L_\tau} D_R(\pi^{\tau,\dagger}_{h,i}, \tilde{\pi}^{\tau}_{h,i}(\cdot|s)) + 36(N-1)^2\eta^2 H^3. \tag{4.62}$$

Proof. Notice that the policy update steps in Algorithm 17 are exactly the same as the optimistic online mirror descent algorithm [114], [115], with the loss vector $g^t = [Q^{\tau}_{h,i}\pi^t_{h,-i}](s,\cdot)$ and the recency bias $M^t = [Q^{\tau}_{h,i}\pi^{t-1}_{h,-i}](s,\cdot)$. 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 [121], [122]. This allows us to directly apply the standard optimistic OMD results (e.g., Lemma 1 in [114] and Proposition 5 in [115]) to obtain

$$\begin{aligned}
\text{reg}^{\tau}_{h,i}(s) =& \max_{\pi^{\tau,\dagger}_{h,i} \in \Delta(\mathcal{A}_i)} \frac{1}{L_\tau} \sum_{j=t^{\text{start}}_\tau}^{t^{\text{end}}_\tau} \left\langle \pi^{\tau,\dagger}_{h,i} - \pi^j_{h,i}, Q^{\tau}_{h,i}\pi^j_{h,-i} \right\rangle(s) \\
\le& \frac{1}{\eta L_\tau} D_R(\pi^{\tau,\dagger}_{h,i}, \tilde{\pi}^{\tau}_{h,i}(\cdot|s)) + \frac{\eta}{L_\tau} \sum_{j=t^{\text{start}}_\tau}^{t^{\text{end}}_\tau} \left\| [Q^{\tau}_{h,i}\pi^j_{h,-i} - Q^{\tau}_{h,i}\pi^{j-1}_{h,-i}](s,\cdot) \right\|^2_\infty \\
& - \frac{1}{8\eta L_\tau} \sum_{j=t^{\text{start}}_\tau}^{t^{\text{end}}_\tau} \left\| \pi^j_{h,i}(\cdot \mid s) - \pi^{j-1}_{h,i}(\cdot \mid s) \right\|^2_1 \\
\le& \frac{1}{\eta L_\tau} D_R(\pi^{\tau,\dagger}_{h,i}, \tilde{\pi}^{\tau}_{h,i}(\cdot|s)) + \frac{\eta}{L_\tau} \sum_{j=t^{\text{start}}_\tau}^{t^{\text{end}}_\tau} 2H^2 \left\| \pi^j_{h,-i}(\cdot \mid s) - \pi^{j-1}_{h,-i}(\cdot \mid s) \right\|^2_1,
\end{aligned} \tag{4.63}$$

where in the last step we used the Hölder's inequality and the fact that $\|Q^{\tau}_h(s,\cdot)\|_\infty \le H$. To further upper

bound (4.63), we apply Lemma 47 to obtain that for any $t \in [t_\tau^{\text{start}}, t_\tau^{\text{end}}]$,

$$\left\|\pi_{h,-i}^t(\cdot \mid s) - \pi_{h,-i}^{t-1}(\cdot \mid s)\right\|_1^2 \leq 18(N-1)^2\eta H. \tag{4.64}$$

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 [121] has shown a $\left\|\pi_{h,-i}^t(\cdot \mid s) - \pi_{h,-i}^{t-1}(\cdot \mid s)\right\|_1^2 \leq 16(N-1)^2\eta^2H^2$ 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 [121]). Plugging (4.64) back into (4.63) leads to the desired result. □

Lemma 47. *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 17 satisfy:*

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

Consequently,

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

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

$$\left\|\pi_i^t - \pi_i^{t-1}\right\|_1 \leq \left\|\pi_i^t - \hat{\pi}_i^t\right\|_1 + \left\|\hat{\pi}_i^t - \hat{\pi}_i^{t-1}\right\|_1 + \left\|\hat{\pi}_i^{t-1} - \pi_i^{t-1}\right\|_1. \tag{4.65}$$

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

$$\left\|\pi_i^t - \hat{\pi}_i^t\right\|_1 \leq \sqrt{2\,\mathrm{KL}\left(\pi_i^t \| \hat{\pi}_i^t\right)}. \tag{4.66}$$

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 17 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 [238]) 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 (4.66) 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 (4.65). 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 48. *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 39 for the zero-sum game setting, one can establish the following recursion for the value estimation error:

$$\delta_h^t \le \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\mathrm{start}}}^{t_{\tau-1}^{\mathrm{end}}} \delta_{h+1}^j + \mathrm{reg}_h^{\tau-1}, \tag{4.67}$$

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 46, 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 4.67 to get that

$$\delta_h^t \le \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \frac{1}{\eta L_{\tau-1}} D_R(\pi_{h,i}^{\tau-1,\dagger}, \tilde{\pi}_{h,i}^{\tau-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^3 + \frac{1}{L_{\tau-1}} \sum_{j=t_{\tau-1}^{\text{start}}}^{t_{\tau-1}^{\text{end}}} \delta_{h+1}^j. \tag{4.68}$$

Notice that according to the definition in Algorithm 18, 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 (4.68) can hence be rewritten more succinctly as

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

Applying the above inequality recursively over h leads to

$$\begin{aligned}
\delta_h^t &\le \sum_{h'=h}^{H} \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \frac{1}{\eta L_{\tau-h'+h-1}} D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger}, \tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^3(H-h+1) \\
&\le \sum_{h'=h}^{H} \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} \frac{1}{\eta L_\tau} \left(1+\frac{1}{H}\right)^{h'-h+1} D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger}, \tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^4 \\
&\le \frac{3}{\eta L_\tau} \sum_{h'=h}^{H} \max_{i\in\mathcal{N}} \max_{s\in\mathcal{S}} D_R(\pi_{h',i}^{\tau-h'+h-1,\dagger}, \tilde{\pi}_{h',i}^{\tau-h'+h-1}(\cdot|s)) + 36(N-1)^2\eta^2 H^4,
\end{aligned} \tag{4.69}$$

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 (4.69) is due to the fact that $(1+1/H)^H \le e \approx 2.71828$. □

4.10.3 Proof of Theorem 22

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 21) can be written more succinctly as

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

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 17 on a sequence of K games, we have that

$$\frac{1}{K}\sum_{k=1}^{K}\text{CCE-gap}(\bar{\pi}^k) \le \frac{3}{\eta K T}\sum_{k=1}^{K}\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right) + 36N^2\eta^2H^4. \tag{4.70}$$

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 30, we obtain that

$$\frac{1}{K}\sum_{k=1}^{K}\text{KL}\left(\pi^{k,\dagger}\|\tilde{\pi}^k\right) \le \frac{1}{K}\sum_{k=1}^{K}\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right) + 4H\bar{\tau}\alpha\ln\frac{A_{\max}}{\alpha}. \tag{4.71}$$

Notice that the conditions of Lemma 30 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

$$\begin{aligned}\sum_{k=1}^{K}\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right) &= \min_\pi\sum_{k=1}^{K}\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right) + \min_\pi\sum_{k=1}^{K}\left(\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right) - \text{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)\right)\\ &\le \min_\pi\sum_{k=1}^{K}\text{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right) + \frac{8A_{\max}(1+\ln K)}{\alpha},\end{aligned} \tag{4.72}$$

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 (4.72), which reduces to bounding the following regret where the loss functions are given by the Bregman divergences:

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

It is known that the unique minimum of $\sum_{k'=1}^{k}\text{KL}([\pi^{k',\dagger}]_\alpha\|\cdot)$ is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\pi^{k',\dagger}]_\alpha$ (see Proposition 1 of [233] 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 $\sum_{k=1}^{K}\text{KL}([\pi^{k,\dagger}]_\alpha\|\cdot)$. We can then invoke the logarithmic regret guarantee of FTL with respect to Bregman divergences, which was established in [208] and is reproduced as Lemma 31 in Section 4.7 for completeness.

To show that Lemma 31 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 31. 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 $\text{KL}([\pi_i^{k,\dagger}]_\alpha\|\cdot)$ is $\frac{2A_{\max}}{\alpha}$ within the $\frac{\alpha}{A_{\max}}$-restricted domain. This allows us to apply Lemma 31 to obtain the result in (4.72).

Moving forward from (4.72), we again apply the property that the unique minimum of $\sum_{k'=1}^{k}\text{KL}([\pi^{k',\dagger}]_\alpha\|\cdot)$

is attained at $\frac{1}{k}\sum_{k'=1}^{k}[\pi^{k',\dagger}]_\alpha$, which leads to

$$\begin{aligned}\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\tilde{\pi}^k\right) &\le \min_{\pi}\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|\pi\right)+\frac{8A_{\max}(1+\ln K)}{\alpha}\\ &=\sum_{k=1}^{K}\mathrm{KL}\left([\pi^{k,\dagger}]_\alpha\|[\pi^\star]_\alpha\right)+\frac{8A_{\max}(1+\ln K)}{\alpha}\\ &\le(1-\alpha)\sum_{k=1}^{K}\mathrm{KL}\left(\pi^{k,\dagger}\|\pi^\star\right)+\frac{8A_{\max}(1+\ln K)}{\alpha},\end{aligned}\tag{4.73}$$

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 (4.73) to (4.71) yields

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

Further substituting the above result back into (4.70) 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

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

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 (4.61) 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. □

4.11 Concluding Remarks

In this chapter, 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 future work, our convergence rate for learning CCE (Theorem 21) 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).

Chapter 5

Concluding Remarks

In this dissertation, we have discussed a series of results toward theoretical understandings of multi-agent reinforcement learning. First, we have presented decentralized MARL algorithms for learning (coarse) correlated equilibria in general-sum Markov games. We have started by introducing the V-learning OMD algorithm and established the first line of sample complexity guarantees in this setting. We have strengthened these results by proposing stage-based V-learning algorithms with simplified analysis and improved sample complexity bounds. We have then extended the V-learning framework to the full-information setting and derived their near-optimal convergence rates accordingly. Second, we have proposed a series of restart-based RL algorithms for learning in non-stationary environments, a common challenge arising in many MARL scenarios. We have proved near-optimal dynamic regret bounds of our algorithms and illustrated how our non-stationary RL method can be readily applied to learning the team-optimal policies in cooperative smooth games. Third, we have studied the use of meta-learning to transfer useful information across multiple Markov games. We have developed multiple MARL algorithms with initialization-dependent convergence guarantees as the basis, and derived the faster convergence rates of meta-learning to different equilibria in a sequence of similar games. Our efforts have mostly been devoted to developing MARL algorithms with convergence or sample complexity guarantees for the class of nonzero-sum Markov games, where very few results were previously known.

The research conducted in the dissertation opens up several potential avenues for future research. An important future direction would be to further tighten the bounds established in this dissertation, including the sample complexity upper and lower bounds for learning CCE/CE in general-sum Markov games (Chapter 2) and closing the $\widetilde{O}(H^{\frac{1}{3}})$ gap for the dynamic regret bounds in non-stationary RL (Chapter 3). Our initialization-dependent convergence rate for learning CCE is slightly less competitive than the best-known results when our policies are initialized conservatively, which might also be improved via a refined policy stability analysis (Chapter 4). In addition, in this dissertation, we have primarily considered the fully observable MARL setup where the agents have full access to the state information. This is in contrast to the more general partially observable Markov games [239] or decentralized partially observable Markov decision processes [58], [129], where each agent has only a private partial view of the environment state. Learning or even computing a NE under partial observability is much more challenging and would be an interesting future direction. Finally, another promising direction is to see how the theoretical results established in this dissertation can be applied to real-world application scenarios [240], [241]. In our own efforts, we have explored the opportunities of applying our methodology to the resource management problem in cloud computing [242], [243]. In particular,

we have investigated the use of the mean-field approximation in MARL to deal with the scalability issues in multi-tenant serverless computing platforms [244], [245] and applied the idea of meta-learning to address the heterogeneity of the workloads in resource autoscaling [196], [246]. It would be interesting to identify other real-world implications of our methods or results, as well.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2018.

[4] J. Schrittwieser, I. Antonoglou, T. Hubert, *et al.*, "Mastering Atari, Go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[6] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics, 1998.

[7] L. S. Shapley, "Stochastic games," *Proceedings of the National Academy of Sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[8] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *International Conference on Machine Learning*, 1994, pp. 157–163.

[9] O. Vinyals, I. Babuschkin, W. M. Czarnecki, *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

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

[11] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[12] 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.

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

[14] 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*, 2020, pp. 3674–3682.

[15] X. Chen, X. Deng, and S.-H. Teng, "Settling the complexity of computing two-player Nash equilibria," *Journal of the ACM*, vol. 56, no. 3, pp. 1–57, 2009.

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

[17] 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*, 2022, pp. 15 007–15 049.

[18] H. Cai, K. Ren, W. Zhang, *et al.*, "Real-time bidding by reinforcement learning in display advertising," in *International Conference on Web Search and Data Mining*, 2017, pp. 661–670.

[19] J. Lu, C. Yang, X. Gao, L. Wang, C. Li, and G. Chen, "Reinforcement learning with sequential information clustering in real-time bidding," in *International Conference on Information and Knowledge Management*, 2019, pp. 1633–1641.

[20] S. Chawla, N. R. Devanur, A. R. Karlin, and B. Sivan, "Simple pricing schemes for consumers with evolving values," in *ACM-SIAM Symposium on Discrete Algorithms*, 2016, pp. 1476–1490.

[21] S. Agrawal and R. Jia, "Learning in structured MDPs with convex cost functions: Improved regret bounds for inventory management," in *ACM Conference on Economics and Computation*, 2019, pp. 743–744.

[22] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Reinforcement learning for non-stationary Markov decision processes: The blessing of (more) optimism," *arXiv preprint arXiv:2006.14389*, 2020.

[23] 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*, PMLR, 2021, pp. 7447–7458.

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

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

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

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

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

[29] W. Mao, H. Qiu, C. Wang, *et al.*, "Multi-agent meta-reinforcement learning: Sharper convergence rates with task similarity," in *Conference on Neural Information Processing Systems*, 2023.

[30] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[31] H. Moulin and J.-P. Vial, "Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon," *International Journal of Game Theory*, vol. 7, no. 3-4, pp. 201–221, 1978.

[32] R. J. Aumann, "Correlated equilibrium as an expression of Bayesian rationality," *Econometrica: Journal of the Econometric Society*, pp. 1–18, 1987.

[33] C. H. Papadimitriou and T. Roughgarden, "Computing correlated equilibria in multi-player games," *Journal of the ACM*, vol. 55, no. 3, pp. 1–29, 2008.

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

[35] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

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

[37] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science*, IEEE, 1995, pp. 322–331.

[38] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on Machine Learning*, 2003, pp. 928–936.

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

[40] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," *AAAI Conference on Artificial Intelligence*, vol. 1998, no. 746-752, p. 2, 1998.

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

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

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

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

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

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

[47] S. Kar, J. M. F. Moura, and H. V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

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

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

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

[51] D. S. Leslie and E. J. Collins, "Individual Q-learning in normal form games," *SIAM Journal on Control and Optimization*, vol. 44, no. 2, pp. 495–514, 2005.

[52] G. Arslan and S. Yüksel, "Decentralized Q-learning for stochastic teams and games," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1545–1558, 2016.

[53] Y. Tian, Y. Wang, T. Yu, and S. Sra, "Provably efficient online agnostic learning in Markov games," *arXiv preprint arXiv:2010.15020*, 2020.

[54] 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," *Annual Conference on Learning Theory*, 2021.

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

[56] Y.-C. Ho, "Team decision theory and information structures," *Proceedings of the IEEE*, vol. 68, no. 6, pp. 644–654, 1980.

[57] A. Nayyar, A. Gupta, C. Langbort, and T. Başar, "Common information based Markov perfect equilibria for stochastic games with asymmetric information: Finite games," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 555–570, 2013.

[58] A. Nayyar, A. Mahajan, and D. Teneketzis, "Decentralized stochastic control with partial history sharing: A common information approach," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1644–1658, 2013.

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

[60] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" In *Advances in Neural Information Processing Systems*, 2018, pp. 4863–4873.

[61] Z. Zhang, Y. Zhou, and X. Ji, "Almost optimal model-free reinforcement learning via reference-advantage decomposition," *arXiv preprint arXiv:2004.10019*, 2020.

[62] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer Science & Business Media, 2012.

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

[64] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, and T. Başar, "$\widetilde{O}(T^{-1})$ convergence to (coarse) correlated equilibria in full-information general-sum Markov games," *arXiv preprint arXiv:2403.07890, also to appear in Annual Learning for Dynamics and Control Conference*, 2024.

[65] A. Blum and Y. Mansour, "From external to internal regret," *Journal of Machine Learning Research*, vol. 8, no. 6, 2007.

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

[67] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[68] T. D. Hansen, P. B. Miltersen, and U. Zwick, "Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor," *Journal of the ACM*, vol. 60, no. 1, pp. 1–16, 2013.

[69] M. L. Littman and C. Szepesvári, "A generalized reinforcement-learning model: Convergence and applications," in *International Conference on Machine Learning*, 1996, pp. 310–318.

[70] Y. Bai and C. Jin, "Provable self-play algorithms for competitive reinforcement learning," in *International Conference on Machine Learning*, 2020, pp. 551–560.

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

[72] 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.

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

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

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

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

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

[78] A. Zehfroosh and H. G. Tanner, "PAC reinforcement learning algorithm for general-sum Markov games," *arXiv preprint arXiv:2009.02605*, 2020.

[79] A. Greenwald and K. Hall, "Correlated-Q learning," in *International Conference on Machine Learning*, 2003, pp. 242–249.

[80] J. Pérolat, F. Strub, B. Piot, and O. Pietquin, "Learning Nash equilibrium for general-sum Markov games from batch data," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 232–241.

[81] H. Prasad, P. LA, and S. Bhatnagar, "Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games," in *International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1371–1379.

[82] 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*, PMLR, 2021, pp. 7001–7010.

[83] Y. Viossat and A. Zapechelnyuk, "No-regret dynamics and fictitious play," *Journal of Economic Theory*, vol. 148, no. 2, pp. 825–842, 2013.

[84] S. Hart and A. Mas-Colell, "Uncoupled dynamics do not lead to Nash equilibrium," *American Economic Review*, vol. 93, no. 5, pp. 1830–1836, 2003.

[85] Y. Freund and R. E. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 79–103, 1999.

[86] R. Kleinberg, G. Piliouras, and É. Tardos, "Multiplicative updates outperform generic no-regret learning in congestion games," in *ACM Symposium on Theory of Computing*, 2009, pp. 533–542.

[87] J. Cohen, A. Héliou, and P. Mertikopoulos, "Learning with bandit feedback in potential games," in *International Conference on Neural Information Processing Systems*, 2017, pp. 6372–6381.

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

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

[90] D. Mguni, Y. Wu, Y. Du, *et al.*, "Learning in nonzero-sum stochastic games with potentials," *arXiv preprint arXiv:2103.09284*, 2021.

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

[92] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 373–396, 2009.

[93] J. R. Marden, G. Arslan, and J. S. Shamma, "Cooperative control and potential games," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1393–1407, 2009.

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

[95] R. I. Brafman and M. Tennenholtz, "R-max-a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, vol. 3, no. Oct, pp. 213–231, 2002.

[96] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, pp. 1563–1600, 2010.

[97] M. G. Azar, I. Osband, and R. Munos, "Minimax regret bounds for reinforcement learning," in *International Conference on Machine Learning*, 2017, pp. 263–272.

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

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

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

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

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

[103] A. S. Nemirovskij and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

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

[105] H. Fang, N. Harvey, V. Portella, and M. Friedlander, "Online mirror descent and dual averaging: Keeping pace in the dynamic case," in *International Conference on Machine Learning*, 2020, pp. 3008–3017.

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

[107] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

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

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

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

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

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

[113] C. Daskalakis, A. Deckelbaum, and A. Kim, "Near-optimal no-regret algorithms for zero-sum games," in *ACM-SIAM Symposium on Discrete Algorithms*, 2011, pp. 235–254.

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

[115] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire, "Fast convergence of regularized learning in games," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2989–2997, 2015.

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

[117] X. Chen and B. Peng, "Hedging in games: Faster convergence of external and swap regrets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 990–18 999, 2020.

[118] C. Daskalakis, M. Fishelson, and N. Golowich, "Near-optimal no-regret learning in general games," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 604–27 616, 2021.

[119] I. Anagnostides, C. Daskalakis, G. Farina, M. Fishelson, N. Golowich, and T. Sandholm, "Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 736–749.

[120] I. Anagnostides, G. Farina, C. Kroer, C.-W. Lee, H. Luo, and T. Sandholm, "Uncoupled learning dynamics with $O(\log T)$ swap regret in multiplayer games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3292–3304, 2022.

[121] 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.

[122] 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.

[123] 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.

[124] Y.-G. Hsieh, K. Antonakopoulos, and P. Mertikopoulos, "Adaptive learning in continuous games: Optimal regret bounds and convergence to Nash equilibrium," in *Conference on Learning Theory*, 2021, pp. 2388–2422.

[125] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[126] W. Hoeffding and J. Wolfowitz, "Distinguishability of sets of distributions," *The Annals of Mathematical Statistics*, vol. 29, no. 3, pp. 700–718, 1958.

[127] S. Yüksel and T. Başar, *Stochastic Networked Control Systems: Stabilization and Optimization under Information Constraints*. Springer Science & Business Media, 2013.

[128] S. Bhatt, W. Mao, A. Koppel, and T. Başar, "Semiparametric information state embedding for policy search under imperfect information," in *IEEE Conference on Decision and Control*, 2021, pp. 4501–4506.

[129] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[130] W. Mao, K. Zhang, Z. Yang, and T. Başar, "Decentralized learning of finite-memory policies in Dec-POMDPs," *IFAC World Congress*, vol. 56, no. 2, pp. 2601–2607, 2023.

[131] W. Mao, Z. Zheng, and F. Wu, "Pricing for revenue maximization in IoT data markets: An information design perspective," in *IEEE Conference on Computer Communications*, 2019, pp. 1837–1845.

[132] C. Chen, H. Wei, N. Xu, *et al.*, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 3414–3421.

[133] S. M. Shortreed, E. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy, "Informing sequential clinical decision-making through reinforcement learning: An empirical study," *Machine Learning*, vol. 84, no. 1-2, pp. 109–136, 2011.

[134] S. R. Balseiro and Y. Gur, "Learning in repeated auctions with budgets: Regret minimization and equilibrium," *Management Science*, vol. 65, no. 9, pp. 3952–3968, 2019.

[135] J. R. Birge, H. Chen, N. B. Keskin, and A. Ward, "To interfere or not to interfere: Information revelation and price-setting incentives in a multiagent learning environment," *SSRN 3864227*, 2021.

[136] W. T. Huh and P. Rusmevichientong, "A nonparametric asymptotic analysis of inventory planning with censored demand," *Mathematics of Operations Research*, vol. 34, no. 1, pp. 103–123, 2009.

[137] H. Zhang, X. Chao, and C. Shi, "Closing the gap: A learning algorithm for the lost-sales inventory system with lead times," *Management Science*, vol. 66, no. 5, pp. 1962–1980, 2019.

[138] H. Bastani, D. Simchi-Levi, and R. Zhu, "Meta dynamic pricing: Transfer learning across experiments," *Management Science (Forthcoming)*, 2021.

[139] A. Tirinzoni, R. Poiani, and M. Restelli, "Sequential transfer in reinforcement learning with a generative model," *arXiv preprint arXiv:2007.00722*, 2020.

[140] E. Brunskill and L. Li, "Sample complexity of multi-task reinforcement learning," in *Uncertainty in Artificial Intelligence*, 2013, p. 122.

[141] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in *International Conference on Machine Learning*, 2018, pp. 2497–2506.

[142] D. Abel, Y. Jinnai, S. Y. Guo, G. Konidaris, and M. Littman, "Policy and value transfer in lifelong reinforcement learning," in *International Conference on Machine Learning*, 2018, pp. 20–29.

[143] Y. Sun, X. Yin, and F. Huang, "Temple: Learning template of transitions for sample efficient multi-task RL," *arXiv preprint arXiv:2002.06659*, 2020.

[144] R. Ortner, P. Gajane, and P. Auer, "Variational regret bounds for reinforcement learning," in *Uncertainty in Artificial Intelligence*, 2019, pp. 81–90.

[145] P. Gajane, R. Ortner, and P. Auer, "A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions," *arXiv preprint arXiv:1805.10066*, 2018.

[146] O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko, "A kernel-based approach to non-stationary reinforcement learning in metric spaces," *arXiv preprint arXiv:2007.05078*, 2020.

[147] H. Zhou, J. Chen, L. R. Varshney, and A. Jagmohan, "Nonstationary reinforcement learning with linear function approximation," *arXiv preprint arXiv:2010.04244*, 2020.

[148] A. Touati and P. Vincent, "Efficient learning in non-stationary linear Markov decision processes," *arXiv preprint arXiv:2010.12870*, 2020.

[149] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Non-stationary reinforcement learning: The blessing of (more) optimism," *SSRN Preprint 3397818*, 2020.

[150] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Advances in Neural Information Processing Systems*, 2014, pp. 199–207.

[151] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Başar, "Model-free non-stationary RL: Near-optimal regret and applications in multi-agent RL and inventory control," *arXiv preprint arXiv:2010.03161, also to appear in Management Science*, 2024.

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

[153] C.-W. Lee, H. Luo, C.-Y. Wei, and M. Zhang, "Linear last-iterate convergence for matrix games and stochastic games," *arXiv preprint arXiv:2006.09517v1*, 2020.

[154] Y. Fei, Z. Yang, Z. Wang, and Q. Xie, "Dynamic regret of policy optimization in non-stationary environments," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[155] C.-Y. Wei and H. Luo, "Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach," *arXiv preprint arXiv:2102.05406*, 2021.

[156] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *arXiv preprint arXiv:2005.10619*, 2020.

[157] J. Y. Yu and S. Mannor, "Online learning in Markov decision processes with arbitrarily changing rewards and transitions," in *International Conference on Game Theory for Networks*, IEEE, 2009, pp. 314–322.

[158] G. Neu, A. Antos, A. György, and C. Szepesvári, "Online Markov decision processes under bandit feedback," in *Advances in Neural Information Processing Systems*, 2010, pp. 1804–1812.

[159] R. Arora, O. Dekel, and A. Tewari, "Deterministic MDPs with adversarial rewards and bandit feedback," in *Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 93–101.

[160] Y. A. Yadkori, P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári, "Online learning in Markov decision processes with adversarially chosen transition probability distributions," in *Advances in Neural Information Processing Systems*, 2013, pp. 2508–2516.

[161] T. Dick, A. Gyorgy, and C. Szepesvari, "Online learning in Markov decision processes with changing cost sequences," in *International Conference on Machine Learning*, 2014, pp. 512–520.

[162] J. Wang, Y. Liu, and B. Li, "Reinforcement learning with perturbed rewards," *arXiv preprint arXiv:1810.01032*, 2018.

[163] T. Lykouris, M. Simchowitz, A. Slivkins, and W. Sun, "Corruption robust exploration in episodic reinforcement learning," *arXiv preprint arXiv:1911.08689*, 2019.

[164] C. Jin, T. Jin, H. Luo, S. Sra, and T. Yu, "Learning adversarial MDPs with bandit feedback and unknown transition," *arXiv preprint arXiv:1912.01192*, 2019.

[165] Y. Bai, T. Xie, N. Jiang, and Y.-X. Wang, "Provably efficient Q-learning with low switching cost," in *Advances in Neural Information Processing Systems*, 2019, pp. 8004–8013.

[166] O. Besbes, Y. Gur, and A. Zeevi, "Optimal exploration–exploitation in a multi-armed bandit problem with non-stationary rewards," *Stochastic Systems*, vol. 9, no. 4, pp. 319–337, 2019.

[167] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *International Conference on Algorithmic Learning Theory*, 2011, pp. 174–188.

[168] N. B. Keskin and A. Zeevi, "Chasing demand: Learning and earning in a changing environment," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 277–307, 2017.

[169] R. Allesiardo, R. Féraud, and O.-A. Maillard, "The non-stationary stochastic multi-armed bandit problem," *International Journal of Data Science and Analytics*, vol. 3, no. 4, pp. 267–283, 2017.

[170] Z. S. Karnin and O. Anava, "Multi-armed bandits: Competing with optimal sequences," in *Advances in Neural Information Processing Systems*, 2016, pp. 199–207.

[171] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Hedging the drift: Learning to optimize under non-stationarity," *arXiv preprint arXiv:1903.01461*, 2019.

[172] P. Auer, P. Gajane, and R. Ortner, "Adaptively tracking the best bandit arm with an unknown number of distribution changes," in *Conference on Learning Theory*, 2019, pp. 138–158.

[173] C. Tekin and M. Liu, "Online algorithms for the multi-armed bandit problem with Markovian rewards," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 1675–1682.

[174] W. Ma, "Improvements and generalizations of stochastic knapsack and Markovian bandits approximation algorithms," *Mathematics of Operations Research*, vol. 43, no. 3, pp. 789–812, 2018.

[175] X. Zhou, Y. Xiong, N. Chen, and X. Gao, "Regime switching bandits," *arXiv preprint arXiv:2001.09390*, 2020.

[176] H. Luo, C.-Y. Wei, A. Agarwal, and J. Langford, "Efficient contextual bandits in non-stationary worlds," in *Conference On Learning Theory*, 2018, pp. 1739–1776.

[177] Y. Chen, C.-W. Lee, H. Luo, and C.-Y. Wei, "A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free," *arXiv preprint arXiv:1902.00980*, 2019.

[178] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Learning to optimize under non-stationarity," in *International Conference on Artificial Intelligence and Statistics*, 2019, pp. 1079–1087.

[179] P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou, "A simple approach for non-stationary linear bandits," in *International Conference on Artificial Intelligence and Statistics*, vol. 2020, 2020.

[180] W. Mao, K. Zhang, Q. Xie, and T. Başar, "POLY-HOOT: Monte-Carlo planning in continuous space MDPs with non-asymptotic analysis," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[181] N. Chen, C. Wang, and L. Wang, "Learning and optimization with seasonal patterns," *arXiv preprint arXiv:2005.08088*, 2020.

[182] I. Osband and B. Van Roy, "On lower bounds for regret in reinforcement learning," *arXiv preprint arXiv:1608.02732*, 2016.

[183] C. J. C. H. Watkins, "Learning from delayed rewards," *PhD thesis, King's College, University of Cambridge*, 1989.

[184] A. Agarwal, M. Henaff, S. Kakade, and W. Sun, "PC-PG: Policy cover directed exploration for provable policy gradient learning," *arXiv preprint arXiv:2007.08459*, 2020.

[185] D. Misra, M. Henaff, A. Krishnamurthy, and J. Langford, "Kinematic state abstraction and provably efficient rich-observation reinforcement learning," in *International Conference on Machine Learning*, 2020, pp. 6961–6971.

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

[187] M. Gao, T. Xie, S. S. Du, and L. F. Yang, "A provably efficient algorithm for linear Markov decision process with low switching cost," *arXiv preprint arXiv:2101.00494*, 2021.

[188] T. Roughgarden, "Intrinsic robustness of the price of anarchy," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, 2009, pp. 513–522.

[189] H. Yuan, Q. Luo, and C. Shi, "Marrying stochastic gradient descent with bandits: Learning algorithms for inventory systems with fixed costs," *Management Science*, 2021.

[190] C. Shi, W. Chen, and I. Duenyas, "Nonparametric data-driven algorithms for multiproduct inventory systems with censored demand," *Operations Research*, vol. 64, no. 2, pp. 362–370, 2016.

[191] Y. Yu, X. Chen, and F. Zhang, "Dynamic capacity management with general upgrading," *Operations Research*, vol. 63, no. 6, pp. 1372–1389, 2015.

[192] D. A. Freedman, "On tail probabilities for martingales," *The Annals of Probability*, pp. 100–118, 1975.

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

[194] 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*, 2019, pp. 707–714.

[195] 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*, vol. 34, pp. 16 899–16 912, 2021.

[196] H. Qiu, W. Mao, C. Wang, *et al.*, "AWARE: Automate workload autoscaling with reinforcement learning in production cloud systems," in *USENIX Annual Technical Conference*, 2023.

[197] S. Xue, C. Qu, X. Shi, *et al.*, "A meta reinforcement learning approach for predictive autoscaling in the cloud," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4290–4299.

[198] 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*, 2021, pp. 18–25.

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

[200] 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*, 2020, pp. 1082–1092.

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

[202] 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*, vol. 34, pp. 3096–3107, 2021.

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

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

[205] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *International Conference on Machine Learning*, PMLR, 2019, pp. 1920–1930.

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

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

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

[209] 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.

[210] P. A. Ortega, J. X. Wang, M. Rowland, *et al.*, "Meta-learning of sequential strategies," *arXiv preprint arXiv:1905.03030*, 2019.

[211] 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*, PMLR, 2022, pp. 13 793–13 806.

[212] S. Cen, Y. Wei, and Y. Chi, "Fast policy extragradient methods for competitive games with entropy regularization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 952–27 964, 2021.

[213] S. Zeng, T. Doan, and J. Romberg, "Regularized gradient descent ascent for two-player zero-sum Markov games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 34 546–34 558, 2022.

[214] 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.

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

[216] 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.

[217] 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*, 2022, pp. 4414–4425.

[218] 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.

[219] 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.

[220] 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.

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

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

[223] 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*, 2022, pp. 26 772–26 808.

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

[225] 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.

[226] A. A. Team, J. Bauer, K. Baumli, *et al.*, "Human-timescale adaptation in an open-ended task space," *arXiv preprint arXiv:2301.07608*, 2023.

[227] 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.

[228] 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*, vol. 23, no. 1, pp. 5271–5308, 2022.

[229] M. Lauriere, S. Perrin, S. Girgin, *et al.*, "Scalable deep reinforcement learning algorithms for mean field games," in *International Conference on Machine Learning*, PMLR, 2022, pp. 12 078–12 095.

[230] 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*, vol. 33, pp. 13 199–13 213, 2020.

[231] W. Mao, H. Qiu, C. Wang, *et al.*, "A mean-field game approach to cloud resource management with function approximation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 243–36 258, 2022.

[232] G. Chen and M. Teboulle, "Convergence analysis of a proximal-like minimization algorithm using Bregman functions," *SIAM Journal on Optimization*, vol. 3, no. 3, pp. 538–543, 1993.

[233] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. 10, 2005.

[234] 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*, vol. 12, 1999.

[235] 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*, vol. 22, no. 98, pp. 1–76, 2021.

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

[237] K. Zhang, A. Koppel, H. Zhu, and T. Başar, "Global convergence of policy gradient methods to (almost) locally optimal policies," *SIAM Journal on Control and Optimization*, vol. 58, no. 6, pp. 3586–3612, 2020.

[238] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[239] Q. Liu, C. Szepesvári, and C. Jin, "Sample-efficient reinforcement learning of partially observable Markov games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 296–18 308, 2022.

[240] X. Zhang, W. Mao, S. Mowlavi, M. Benosman, and T. Başar, "Controlgym: Large-scale control environments for benchmarking reinforcement learning algorithms," *arXiv preprint arXiv:2311.18736, also to appear in Annual Learning for Dynamics and Control Conference*, 2024.

[241] X. Zhang, W. Mao, H. Qiu, and T. Başar, "Decision transformer as a foundation model for partially observable continuous control," *arXiv preprint arXiv:2404.02407*, 2024.

[242] H. Qiu, W. Mao, C. Wang, *et al.*, "When green computing meets performance and resilience SLOs," in *Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2024.

[243] H. Qiu, W. Mao, A. Patke, *et al.*, "Efficient interactive LLM serving with proxy model-based sequence length prediction," in *Proceedings of the 5th International Workshop on Cloud Intelligence / AIOps*, 2024.

[244] H. Qiu, W. Mao, A. Patke, *et al.*, "Reinforcement learning for resource management in multi-tenant serverless platforms," in *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, 2022, pp. 20–28.

[245] H. Qiu, W. Mao, A. Patke, *et al.*, "SIMPPO: A scalable and incremental online learning framework for serverless resource management," in *Proceedings of the 13th Symposium on Cloud Computing*, 2022, pp. 306–322.

[246] H. Qiu, W. Mao, C. Wang, *et al.*, "On the promise and challenges of foundation models for learning-based cloud systems management," in *Machine Learning for Systems Workshop at 37th NeurIPS Conference*, 2023.

Appendix A

Publications of Weichao Mao Related to the Thesis

Publications Directly Related to the Thesis

[1] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, and T. Başar, "$\widetilde{O}(T^{-1})$ Convergence to (Coarse) Correlated Equilibria in Full-Information General-Sum Markov Games," to appear in Annual Conference on Learning for Dynamics and Control, 2024.

[2] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Başar, "Model-free non-stationary RL: Near-optimal regret and applications in multi-agent RL and inventory control," arXiv preprint arXiv:2010.03161, also to appear in Management Science, 2024.

[3] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, R. Iyer, and T. Başar, "Multi-agent meta-reinforcement learning: Sharper convergence rates with task similarity," in Conference on Neural Information Processing Systems, 2023.

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

[5] 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, 2022.

[6] 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, 2021.

Publications Independent of the Thesis

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

[2] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, R. K. Iyer, and T. Başar, "A mean-field game approach to cloud resource management with function approximation", in Conference on Neural Information Processing Systems, 2022.

[3] H. Qiu, W. Mao, A. Patke, C. Wang, H. Franke, Z. Kalbarczyk, T. Başar, and R. K. Iyer, "SIMPPO: A scalable and incremental online learning framework for serverless resource management", In ACM Symposium on Cloud Computing, 2022.

[4] S. Bhatt, W. Mao, A. Koppel, and T. Başar, "Semiparametric information state embedding for policy search under imperfect information", in Conference on Decision and Control, 2021.

[5] W. Mao, K. Zhang, Q. Xie, and T. Başar, "POLY-HOOT: Monte-Carlo planning in continuous space MDPs with non-asymptotic analysis", in Conference on Neural Information Processing Systems, 2020.
[6] W. Mao, K. Zhang, E. Miehling, and T. Başar, "Information state embedding in partially observable cooperative multi-agent reinforcement learning," in IEEE Conference on Decision and Control, 2020.