Breaking the Curse of Multiagents in a Large State Space: RL in Markov Games with Independent Linear Function Approximation

Qiwen Cui [1] **Kaiqing Zhang** [2] **Simon S. Du** [1]

Abstract

We propose a new model, *independent linear Markov game*, for multi-agent reinforcement learning with a large state space and a large number of agents. This is a class of Markov games with *independent* linear function approximation, where each agent has its own function approximation for the state-action value functions that are *marginalized* by other players' policies. We design new algorithms for learning the Markov coarse correlated equilibria (CCE) and Markov correlated equilibria (CE) with sample complexity bounds that only scale polynomially with *each agent's own function class complexity*, thus breaking the curse of multiagents. In contrast, existing works for Markov games with function approximation have sample complexity bounds scale with the size of the *joint action space* when specialized to the canonical tabular Markov game setting, which is exponentially large in the number of agents. Our algorithms rely on two key technical innovations: (1) utilizing policy replay to tackle *non-stationarity* incurred by multiple agents and the use of function approximation; (2) separating learning Markov equilibria and exploration in the Markov games, which allows us to use the full-information no-regret learning oracle instead of the stronger bandit-feedback no-regret learning oracle used in the tabular setting. Furthermore, we propose an iterative-best-response type algorithm that can learn pure Markov Nash equilibria in independent linear Markov potential games, with applications in learning in congestion games. In the tabular case, by adapting the policy replay mechanism for independent linear Markov games, we propose an algorithm with $\widetilde{O}(\epsilon^{-2})$ sample complexity to learn Markov CCE, which improves the state-of-the-art result $\widetilde{O}(\epsilon^{-3})$ in (Daskalakis et al., 2022), where ϵ is the desired accuracy, and also significantly improves other problem parameters. Furthermore, we design the first provably efficient algorithm for learning Markov CE that breaks the curse of multiagents.

1. Introduction

Decision-making under uncertainty in a multi-agent system has shown its potential to approach artificial intelligence, with superhuman performance in Go games (Silver et al., 2017), Poker (Brown and Sandholm, 2019), and real-time strategy games (Vinyals et al., 2019), etc. All these successes can be generally viewed as examples of multi-agent reinforcement learning (MARL), a generalization of single-agent reinforcement learning (RL) (Sutton and Barto, 2018) where multiple RL agents interact and make sequential decisions in a common environment (Zhang et al., 2021a). Despite the impressive empirical achievements of MARL, the theoretical understanding of MARL is still far from complete due to the complex interactions among agents.

One of the most prominent challenges in RL is the curse of *large state-action* spaces. In real-world applications, the number of states and actions is exponentially large so that the *tabular* RL algorithms are not applicable. For example, there are 3^{361} potential states in Go games, and it is impossible to enumerate all of them. In single-agent RL, plenty of works attempt to tackle this issue via function approximation so that the sample complexity only depends on the complexity of the function class, thus successfully breaking the curse of large state-action spaces (Wen and Van Roy, 2017; Jiang et al., 2017; Yang and Wang, 2020; Du et al., 2019; Jin et al., 2020; Weisz et al., 2021; Wang et al., 2020; Zanette et al., 2020; Jin et al., 2021a; Du et al., 2021; Foster et al., 2021).

However, it is still unclear what is the proper function approximation model for multi-agent RL. The existing theoretical analyses in MARL exclusively focus on a *global* function approximation paradigm, i.e., a function class capturing the state-joint-action value $Q_i(s, a_1, \cdots, a_m)$ where

[1] University of Washington, Seattle, USA [2] University of Maryland, College Park, USA. Correspondence to: Qiwen Cui <qwcui@uw.edu>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023. Copyright 2023 by the author(s).

s is the state and a_i is the action of player $i \in [m]$ (Xie et al., 2020; Huang et al., 2021; Chen et al., 2021; Jin et al., 2022; Chen et al., 2022; Ni et al., 2022). Unfortunately, these algorithms would *suffer from the curse of multiagents* when specialized to tabular Markov games, one of the most canonical models in MARL. Specifically, the sample complexity depends on the number of joint actions $\prod_{i\in[m]} A_i$, where A_i is the number of actions for player i, which is exponentially worse than the best algorithms specified to the tabular Markov game whose sample complexity only depends on $\max_{i\in[m]} A_i$ (Jin et al., 2021b; Song et al., 2021; Mao et al., 2022; Daskalakis et al., 2022).

On the other hand, empirical algorithms with *independent* function approximation such as Independent PPO have surprisingly good performance, where only the independent state-individual-action value function $Q_i(s, a_i)$ is modeled (de Witt et al., 2020; Yu et al., 2021). This can be surprising because the independent state-action value function $Q_i(s, a_i)$ does not reflect the change of other players' policies, a.k.a. the *non-stationarity* from multiple agents, which should fail to allow learning at first glance. In addition, single-agent RL with function approximation already suffers from the nonstationarity of applying function approximation (Baird, 1995), making it even harder for MARL. This gap between theoretical and empirical research leads to the following question:

Can we design provably efficient MARL algorithms for Markov games with independent function approximation that can break the curse of multiagents?

In this paper, we provide an affirmative answer to this question. We highlight our contributions and technical novelties below. Due to space limitation, tables and related works are deferred to Appendix A and Appendix B.

1.1. Main Contributions and Technical Novelties

1. Multi-player general-sum Markov games with independent linear function approximation. We propose independent linear Markov games, which is the first provably efficient model in MARL that allows each agent to have its own independent function approximation. We show that independent linear Markov games capture several important instances, namely tabular Markov games (Shapley, 1953), linear Markov decision processes (MDP) (Jin et al., 2020), and congestion games (Rosenthal, 1973). Then we provide the first provably efficient algorithm in MARL that breaks the curse of multiagents and the curse of large state and action spaces at the same time, i.e., the sample complexity only has polynomial dependence on the complexity of the independent function class complexity. See Table 1 for comparisons between our work and prior works.

Our algorithm design relies on two high-level technical ideas which we detail here:

- **Policy replay to tackle non-stationarity.** Different from experience replay that incrementally adds new on-policy data to a dataset, *policy replay* maintains a policy set and completely renews the dataset at each episode by collecting fresh data using the policy set. We propose a new policy replay mechanism for learning equilibria in independent linear Markov games, which allows efficient exploration while adapting to the non-stationarity induced by both multiple agents and function approximation at the same time.
- **Separating exploration and learning Markov equilibria.** States and actions in independent linear Markov games are correlated through the feature map, so we can no longer resort to adversarial bandit oracles as in algorithms for tabular Markov games (Jin et al., 2021b; Song et al., 2021; Mao et al., 2022; Daskalakis et al., 2022). In particular, the adversarial contextual linear bandit oracles would be a potential substitute, while the existence of such oracles remains largely an open problem (see Section 29.4 in (Lattimore and Szepesvári, 2020)). To tackle this issue, we exploit the fact that under the self-play setting, other players are not adversarial but *under control*, so we can sample multiple i.i.d. feedback to derive an accurate estimate instead of just a single bandit feedback. We separate the exploration in Markov games from learning equilibria so that any no-regret algorithms with *full-information feedback* are sufficient for our MARL algorithm, which is significantly weaker than the adversarial bandit oracle used in all the previous works that break the curse of multiagents in the tabular setting.

2. Learning Nash equilibria in Linear Markov potential games. We provide an algorithm to learn Markov Nash equilibria (NE) when the underlying independent linear Markov game is also a Markov potential game. The algorithm is based on the reduction from learning NE in independent linear Markov potential games to learning the optimal policy in linear MDPs. In addition, the result directly implies a provable efficient decentralized algorithm for learning NE in congestion games, which improves the previous state-of-the-art sample complexity result in (Cui et al., 2022).

3. Improved sample complexity for tabular multi-player general-sum Markov games. Aside from our contributions to Markov games with function approximation, we design an algorithm for tabular Markov games with improved sample complexity for learning Markov CCE by adapting the policy replay mechanism we proposed for the independent linear Markov games. Our sample complexity for learning

Markov CCE is $\widetilde{O}(H^6 S^2 A_{\max}\epsilon^{-2})$, which significantly improves the prior state-of-the-art result $\widetilde{O}(H^{11} S^3 A_{\max}\epsilon^{-3})$ in (Daskalakis et al., 2022), where H is the time horizon, S is the number of the states, $A_{\max} = \max_{i\in[m]} A_i$ is the maximum action space and ϵ is the desired accuracy.[*] Furthermore, our analysis is simpler. In addition, we provide the first provably efficient algorithm for learning Markov CE with sample complexity $\widetilde{O}(H^6 S^2 A_{\max}^2 \epsilon^{-2})$.

Notation. For a finite set X, we use $\Delta(X)$ to denote the space of distributions over X. For $n \in \mathbb{N}^+$, we use $[n]$ to denote $\{1, 2, \cdots, n\}$. We use $\|\cdot\|$ to denote the Euclidean norm $\|\cdot\|_2$ and $\langle\cdot,\cdot\rangle$ to denote the Euclidean inner product. We define $\mathrm{proj}_{[a,b]}(x) := \min\{\max\{x, a\}, b\}$ and $x \vee y := \max\{x, y\}$. An arbitrary tie-breaking rule can be used for determining $\mathrm{argmax}_x f(x)$.

2. Preliminaries

Multi-player general-sum Markov games are defined by the tuple $(\mathcal{S}, \{A_i\}_{i=1}^m, H, \mathbb{P}, \{r_i\}_{i=1}^m)$, where $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$, m is the number of the players, $\mathcal{A}_i$ is the action space for player i with $|\mathcal{A}_i| = A_i$, H is the length of the horizon, $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is the collection of the transition kernels such that $\mathbb{P}_h(\cdot \mid s, \mathbf{a})$ gives the distribution of the next state given the current state s and joint action $\mathbf{a} = (a_1, a_2, \cdots, a_m)$ at step h, and $r_i = \{r_{h,i}\}_{h\in[H]}$ is the collection of random reward functions for each player such that $r_{h,i}(s, \mathbf{a}) \in [0, 1]$ is the random reward with mean $R_{h,i}(s, \mathbf{a})$ for player i given the current state s and the joint action $\mathbf{a}$ at step h. We use $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_m$ to denote the joint action space, $\mathbf{r}_h = (r_{h,1}, r_{h,2}, \cdots, r_{h,m})$ to denote the joint reward profile at step h, and $A_{\max} = \max_{i\in[m]} A_i$. In the rest of the paper, we will simplify "multi-player general-sum Markov games" to "Markov games" when it is clear from the context.

Markov games will start at a fixed initial state s_1 for each episode.[†] At each step $h \in [H]$, each player i will observe the current state s_h and choose some action $a_{h,i}$ simultaneously, and receive their own reward realization $\widetilde{r}_{h,i} \sim r_{h,i}(s_h, \mathbf{a}_h)$ where $\mathbf{a}_h = (a_{h,1}, a_{h,2}, \cdots, a_{h,m})$. Then the state will transition according to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, \mathbf{a}_h)$. The game will terminate when state s_{H+1} is reached and the goal of each player is to maximize their own expected total reward $\mathbb{E}\left[\sum_{h=1}^H \widetilde{r}_{h,i}\right]$. We consider the bandit-feedback setting where only the reward for the chosen action is revealed, and there is no simulator and thus exploration is necessary.

Policy. A Markov joint policy is denoted by $\pi = \{\pi_h\}_{h=1}^H$ where each $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$ is the joint policy at step h. We say that a Markov joint policy is a Markov product policy if there are policies $\{\pi_i\}_{i=1}^m$ such that $\pi_h(\mathbf{a} \mid s) = \prod_{i=1}^m \pi_{h,i}(a_i \mid s)$ for each $h \in [H]$, where $\pi_i = \{\pi_{h,i}\}_{h=1}^H$ is the collection of Markov policies $\pi_{h,i} : \mathcal{S} \to \Delta(\mathcal{A}_i)$ for player i. In other words, a Markov product policy means that the policies of each player are not correlated. For a Markov joint policy π, we use π_{-i} to denote the Markov joint policy for all the players except player i. We will simplify the terminology by using "policy" instead of "Markov joint policy" when it is clear from the context as we will only focus on Markov policies.

Value function. For a policy π, it can induce a random trajectory $(s_1, \mathbf{a}_1, \mathbf{r}_1, s_2, \cdots, s_H, \mathbf{a}_H, \mathbf{r}_H, s_{H+1})$ such that $\mathbf{a}_h \sim \pi_h(\cdot \mid s_h)$, $\mathbf{r}_h \sim \mathbf{r}_h(s_h, \mathbf{a}_h)$, and $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, \mathbf{a}_h)$ for all $h \in [H]$. For simplicity, we will denote $\mathbb{E}_\pi[\cdot] = \mathbb{E}_{(s_1,\mathbf{a}_1,\mathbf{r}_1,s_2,\cdots,s_H,\mathbf{a}_H,\mathbf{r}_H,s_{H+1})\sim\pi}[\cdot]$. We define the state value function under policy π for each player $i \in [m]$ to be $V_{h,i}^\pi(s_h) := \mathbb{E}_\pi\left[\sum_{t=h}^H r_{t,i}(s_t, \mathbf{a}_t) \,\middle|\, s_h\right], \forall s_h \in \mathcal{S}$, which is the expected total reward for player i if all the players are following policy π starting from state s_h at step h.

Best response and strategy modification. Suppose all the players except player i are playing according to a fixed policy π_{-i}, then the best response of player i is the policy that can achieve the highest total reward for player i. Concretely, π_i is the best response to π_{-i} if $\pi_i = \mathrm{argmax}_{\pi_i'\in\Pi_i} V_{1,i}^{\pi_i',\pi_{-i}}(s_1)$, where Π_i consists of all the possible policies for player i. We will use $V_{h,i}^{\dagger,\pi_{-i}}(s)$ to denote the best-response value $\max_{\pi_i'\in\Pi_i} V_{h,i}^{\pi_i',\pi_{-i}}(s)$ for all $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ and $\mathbb{E}_{\dagger,\pi_{-i}}[\cdot]$ to be the expectation over the corresponding best-response policy. Note that if all the other players are playing a fixed policy, then player i is in an MDP and the best response is the corresponding optimal policy, which can always be deterministic and achieve the optimal value $\max_{\pi_i'\in\Pi_i} V_{h,i}^{\pi_i',\pi_{-i}}(s)$ for all $h \in [H]$ and $s \in \mathcal{S}$ simultaneously.

A strategy modification $\psi_i = \{\psi_{h,i}\}_{h=1}^H$ for player i is a collection of maps $\psi_{h,i} : \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i$, which will map the action chosen at any state to another action.[‡] For a Markov joint policy π, we use $\psi_i \diamond \pi$ to denote the modified Markov joint policy such that $(\psi_i \diamond \pi)_h(\mathbf{a} \mid s) = \sum_{\mathbf{a}':\psi_{h,i}(a_i'|s)=a_i,\mathbf{a}_{-i}'=\mathbf{a}_{-i}} \pi_h(\mathbf{a}' \mid s)$. In words, if the policy

[*] We use $\widetilde{O}(\cdot)$ to omit logarithmic dependence on all the parameters.

[†] It is straightforward to generalize to stochastic initial state $s_1 \sim p_1(\cdot)$ by adding a dummy state s_0 instead, which will transition to $s_1 \sim p_1(\cdot)$ no matter what action is chosen.

[‡] We only consider deterministic strategy modification as it is known that the optimal strategy modification can always be deterministic (Jin et al., 2021b).

π_h assigns action a_i to player i at state s, it will be modified to action $\psi_{h,i}(a_i \mid s)$. We use Ψ_i to denote all the possible strategy modifications for player i. As Ψ_i contains all the constant modifications, we have $\max_{\psi_i\in\Psi_i} V_{1,i}^{\psi_i\diamond\pi}(s_1) \ge \max_{\pi_i'} V_{1,i}^{\pi_i',\pi_{-i}}(s_1) = V_{1,i}^{\dagger,\pi_{-i}}(s_1)$, which means strategy modification is stronger than best response.

Notions of equilibria. A Markov Nash equilibrium is a Markov product policy where no player can increase their total reward by changing their own policy.

Definition 2.1. (Markov Nash equilibrium) A Markov product policy π is an ϵ-approximate Nash equilibrium if $\mathrm{NashGap}(\pi) := \max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1)\right) \le \epsilon$.

In general, it is intractable to compute Nash equilibrium even in normal-form general-sum games, which are Markov games with $H = 1$ and $S = 1$ (Daskalakis et al., 2009; Chen et al., 2009). In this paper, we will focus on the following two relaxed equilibrium notions, which allow computationally efficient learning.

Definition 2.2. (Markov Coarse Correlated Equilibrium) A Markov joint policy π is a Markov coarse correlated equilibrium if $\mathrm{CCEGap}(\pi) := \max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1)\right) \le \epsilon$.

Definition 2.3. (Markov Correlated Equilibrium) A Markov joint policy π is a Markov correlated equilibrium if $\mathrm{CEGap}(\pi) := \max_{i\in[m]}\left(\max_{\psi_i\in\Psi_i} V_{1,i}^{\psi_i\diamond\pi}(s_1) - V_{1,i}^{\pi}(s_1)\right) \le \epsilon$.

It is known that every Markov NE is a Markov CE and every Markov CE is a Markov CCE, and in two-player zero-sum Markov games, these three notions are equivalent. In this work, we will focus on Markov equilibria, which are more refined compared with non-Markov equilibria considered in (Jin et al., 2021b; Song et al., 2021; Mao et al., 2022). For a detailed discussion regarding the difference, we refer the readers to (Daskalakis et al., 2022).

Two important special cases of Markov games are two-player zero-sum Markov games and Markov potential games, which have computationally efficient algorithms for learning Markov NE. Two-player zero-sum Markov games are Markov games with the number of players $m = 2$ and reward function satisfying $r_{h,1}(s,\mathbf{a}) + r_{h,2}(s,\mathbf{a}) = 0$ for all $(s,\mathbf{a}) \in \mathcal{S}\times\mathcal{A}$ and $h \in [H]$. Markov potential games are Markov games with a potential function $\Phi : \Pi \to [0,\Phi_{\max}]$, where Π is the set of all possible Markov product policies $\pi_1 \times \pi_2 \cdots \times \pi_m$, such that for any player $i \in [m]$, two policies π_i, π_i' of player i and policy π_{-i} for the other players, we have $V_{1,i}^{\pi_i,\pi_{-i}}(s_1) - V_{1,i}^{\pi_i',\pi_{-i}}(s_1) = \Phi(\pi_i,\pi_{-i}) - \Phi(\pi_i',\pi_{-i})$. Immediately, we have $\Phi_{\max} \le mH$ by varying π_i for each player i for one time. One special case of Markov potential games is Markov cooperative games, where all the players share the same reward function.

3. MARL with Independent Linear Function Approximation

In this section, we will introduce the independent linear Markov game model and demonstrate the advantage of this model over existing Markov games with function approximation. Intuitively, independent linear Markov games assume that if other players are following some fixed Markov product policies, then player i is approximately in a linear MDP (Jin et al., 2020). This is fundamentally different from previous global function approximation formulations, which basically assume that the Markov game is a big linear MDP where the action is the joint action $\mathbf{a} = (a_1, a_2, \cdots, a_m)$.

Feature and independent linear function class. For each player i, they have access to their own feature map $\phi_i : \mathcal{S}\times\mathcal{A}_i \to \mathbb{R}^{d_i}$ and we assume that $\sup_{(s,a_i)\in\mathcal{S}\times\mathcal{A}_i} \|\phi_i(s,a_i)\|_2 \le 1$. For player i, given parameters $\theta_i = (\theta_{1,i},\cdots,\theta_{H,i})$, the corresponding linear state-action value function for player i would be $f_i^{\theta} = (f_{1,i}^{\theta_{1,i}}, f_{2,i}^{\theta_{2,i}}, \cdots, f_{H,i}^{\theta_{H,i}})$ where $f_{h,i}^{\theta_{h,i}}(s,a_i) = \langle \phi_i(s,a_i), \theta_{h,i}\rangle$ for all $(s,a_i) \in \mathcal{S}\times\mathcal{A}_i$. We consider the following linear state-action value function class for player i: $\mathcal{Q}_i^{\mathrm{lin}} = \left\{ f_i^{\theta_i} \mid \|\theta_{h,i}\|_2 \le H\sqrt{d}, \forall h \in [H]\right\}$. We also define the state value function class $\mathcal{V} = \{(V_1,\cdots,V_{H+1}) \mid V_h(s) \in [0, H+1-h], \forall h, s\}$.

Given the state value function $V \in \mathcal{V}$ and other players' policies π_{-i}, we can define the independent state-action value function for all $h \in [H]$ and $(s_h, a_{h,i}) \in \mathcal{S}\times\mathcal{A}_i$ as: $Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) = \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[r_{h,i}(s_h,a_{h,i},a_{h,-i}) + V_{h+1}(s_{h+1})\right]$. Now we define Markov games with independent linear function approximation, which generalizes the misspecified MDPs with linear function approximation model proposed in (Zanette and Wainwright, 2022) to the Markov games setting.

Definition 3.1. For any player i, feature map ϕ_i is ν-misspecified with policy set Π^{estimate} if for any rollout policy $\overline{\pi}$, target policy $\widetilde{\pi}$, we have for any $V \in \mathcal{V}$,

$$\max_{\pi\in\Pi^{\mathrm{estimate}}} \left| \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}}\left[\mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s_h,a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V}\right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i})\right]\right| \le \nu,$$

where Π^{estimate} is the collection of Markov product policies that need to be evaluated and

$$\theta_h^{\overline{\pi},\pi_{-i},V} = \tag{1}$$

$$\underset{\|\theta\|\le H\sqrt{d}}{\operatorname{argmin}} \mathbb{E}_{\overline{\pi}} \left(\langle \phi_i(s_h, a_{h,i}), \theta\rangle - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i})\right)^2$$

is the parameter for the best linear function fit to $Q_{h,i}^{\pi_{-i},V}$ under rollout policy $\overline{\pi}$. We say a multi-player general-sum Markov game with features $\{\phi_i\}_{i\in[m]}$ is a ν-misspecified linear Markov game with Π^{estimate} if for any player i, the feature map ϕ_i is ν-misspecified with Π^{estimate}. In addition, we define $d_{\max} := \max_{i\in[m]} d_i$ as the complexity measure of the linear Markov game.

The policy estimation set Π^{estimate} consists of policies that need to be estimated in the algorithm, which reflects the inductive bias of the algorithm. We emphasize that all of our algorithms do not require any knowledge of the policy estimation set Π^{estimate} or the misspecification error ν, which is known as the *agnostic setting* (Agarwal et al., 2020c;a). We give some concrete examples to serve as the special cases of the independent linear Markov game in Appendix C. Note that the complexity of tabular Markov games would be $d = S\prod_{i\in[m]} A_i$ if we apply the global function approximation models in (Chen et al., 2022; Ni et al., 2022), which is exponentially larger than $d_{\max} = S\max_{i\in[m]} A_i$, as in the tabular setting when model-based approaches are used (Bai and Jin, 2020; Zhang et al., 2020; Liu et al., 2021). See Table 1 for a detailed comparison.

4. Algorithms and Analyses for Linear Markov Games

4.1. Experience Replay and Policy Replay

Before getting into the details of our algorithm, we will first review two popular exploration paradigms in single-agent RL, namely *experience replay* and *policy replay*. Experience replay is utilized in most empirical and theoretical algorithms, which adds new on-policy data to a dataset and then uses the dataset to retrain a new policy (Mnih et al., 2013; Azar et al., 2017; Jin et al., 2020). By carefully designing how to train the new policy to explore the underlying MDP, the dataset will contain more and more information about the MDP and thus we can learn the optimal policy without any simulator.

Another popular approach is called policy replay, which is also known as policy cover. Instead of incrementally maintaining a dataset, the algorithm will maintain a policy set, and at each episode renew the dataset by drawing fresh samples using the policies in this policy set. As the dataset is completely refreshed at each episode, policy replay is able to tackle non-stationarity and enjoy better robustness in many different settings. In (Agarwal et al., 2020a), it is used to address the "catastrophic forgetting" problem in policy gradient methods while being robust to the so-called transfer error. In (Zanette and Wainwright, 2022; Daskalakis et al., 2022), it is used to tackle the non-stationarity in Q-learning with function approximation and non-stationarity of multiple agents in tabular Markov games, respectively.

In independent linear Markov games, non-stationarity comes from both multiple agents and function approximation. In particular, the change in other players' policies will lead to a different independent state-action value function to estimate, and the change in the next-step value function estimate will lead to changing targets for regression. In our algorithm, we will show that policy replay can tackle both types of non-stationarity at the same time as we use it to create a stationary environment with fixed regression targets, which leads to provably efficient algorithms for independent linear Markov games. Policy replay also guarantees that if each player has a misspecified feature, the final guarantee will only have a linear dependence on the misspecification error. In addition, we will provide a carefully designed policy-replay-type algorithm for tabular Markov games which has significant improvement over (Daskalakis et al., 2022) in Section 6.

4.2. Algorithm

One technical difficulty in designing algorithms for linear Markov games is that we can no longer resort to adversarial bandits oracles, which is utilized in all algorithms that can break the curse of multiagents (Jin et al., 2021b; Song et al., 2021; Mao et al., 2022; Daskalakis et al., 2020). This is because adversarial contextual linear bandits oracle is necessary to avoid dependence on S and A_i. However, to the best of our knowledge, the only relevant result considers i.i.d. context with known covariance (Neu and Olkhovskaya, 2020), which can not fit into Markov games. Indeed, adversarial linear bandits with changing action set is still an open problem (See Section 29.4 in (Lattimore and Szepesvári, 2020)).

Perhaps surprisingly, our algorithms only require no-regret learning with full-information feedback oracle (Protocol 1 in Appendix D). This oracle is considerably easier than the previous (weighted) high-probability adversarial bandit with noisy bandit feedback oracles (Jin et al., 2021b; Daskalakis et al., 2022). The intuition is that as all the players are using the same algorithm, the environment is not completely adversarial and we can take multiple i.i.d. samples so that the full-information feedback can be constructed with the batched data.

For learning CCE and CE, the no-regret learning oracle needs to satisfy the following no-external-regret and no-swap-regret properties, respectively. We will use the minimax optimal no-external-regret and no-swap-regret algorithms while any other no-regret algorithms are eligible. Assumption 4.1 and Assumption 4.2 can be achieved by EXP3 (Freund and Schapire, 1997) and BM-EXP3 (Blum and Mansour, 2007), respectively.

Assumption 4.1. (No-external-regret with full-information feedback) For any loss sequence $l_1, \ldots, l_T \in \mathbb{R}^B$ bounded between $[0,1]$, the no-regret learning oracle (Protocol 1) enjoys external-regret (Freund and Schapire, 1997): $\max_{b \in \mathcal{B}} \sum_{t=1}^T (\langle p_t, l_t \rangle - l_t(b)) \le \mathrm{Reg}(T) := O(\sqrt{\log(B)T})$.

Assumption 4.2. (No-swap-regret with full-information feedback) For any loss sequence $l_1, \ldots, l_T \in \mathbb{R}^B$ bounded between $[0,1]$, the no-regret learning oracle (Protocol 1) enjoys swap-regret (Blum and Mansour, 2007; Ito, 2020): $\max_{\psi \in \Psi} \sum_{t=1}^T (\langle p_t, l_t \rangle - \langle \psi \diamond p_t, l_t \rangle) \le \mathrm{SwapReg}(T) := O(\sqrt{B\log(B)T})$, where Ψ denote the set $\{\psi : \mathcal{B} \to \mathcal{B}\}$ which consists of all possible strategy modifications.

Due to space limitation, our algorithm **PReFI** (Algorithm 1) is deferred to Appendix D and the decentralized implementation details to Appendix D.2. We will explain the algorithm for learning Markov CCE and the only difference in learning Markov CE is to use the no-swap-regret oracle to replace the no-external-regret one. For simplicity, here we assume the model misspecification error $\nu = 0$. The algorithm has two main components: learning Markov CCE with policy cover and policy cover update. For the first part, given a policy cover Π, we will compute an approximate optimistic CCE under the distribution induced by the policy cover. Specifically, we use a value-iteration-type algorithm that computes the CCE and the corresponding value function from step H to 1 (Line 5). At each step h, each player will run a no-regret algorithm for T steps (Line 8). In this inner loop, we will generate a dataset $\mathcal{D}_{h,i}^{k,t}$ by using policies in the policy cover concatenated with the current policies from the no-regret oracle (Line 11). Then we compute an optimistic independent Q function $\overline{Q}_{h,i}^{k,t}$ via constrained least squares, which has the following guarantee: $0 \le \overline{Q}_{h,i}^{k,t}(s,a_i) - Q_{h,i}^{\pi_{h,-i}^{k,t}, \overline{V}_{h+1}^k}(s,a_i) \le O(\beta \|\phi_i(s,a_i)\|_{[\Sigma_{h,i}^k]^{-1}}), \forall\ (s,a_i) \in \mathcal{S} \times \mathcal{A}_i$, where $\Sigma_{h,i}^k$ is the population covariance matrix of the underlying random design least squares. Then we feed $\overline{Q}_{h,i}^{k,t}$ to the no-regret oracle as the full-information feedback (Line 19 and Line 23). At the end of the no-regret loop, we will compute the optimistic value function, which will be an upper bound of the best response value (Line 26): $\overline{V}_{h,i}^k(s) \ge \max_{a_i \in \mathcal{A}_i} \frac{1}{T} \sum_{t=1}^T \overline{Q}_{h,i}^{k,t}(s,a_i) \ge V_{h,i}^{\dagger, \pi_{-i}^k}(s), \forall\ s \in \mathcal{S}$.

For the policy cover update part, we utilize a lazy update to ensure that the algorithm will end within $K \le K_{\max} := \widetilde{O}(mHd_{\max})$ episodes with high probability, which can significantly improve the final sample complexity bound, similar to the single-agent MDP case studied in (Zanette and Wainwright, 2022). We will show that for any player $i \in [m]$ and step $h \in [H]$, the number of the triggering events (Line 42) is bounded by $\widetilde{O}(d_{\max})$. We maintain a counter $T_{h,i}$ for each player i at each step h, which estimates the information gained by adding the current policy π^k and its weight n^k to the existing policy cover (Line 38). Note that $T_{h,i}$ is the sum of $\|\phi_i(s,a_i)\|^2_{[\Sigma_{h,i}^{k,1}]^{-1}}$, where (s,a_i) is drawn from the trajectory induced by policy π^k and $\Sigma_{h,i}^{k,1}$ is the empirical covariance matrix of the policy cover. Whenever an update is triggered, $T_{h,i} \ge T_{\mathrm{Trig}}$, it means that we will collect a good amount of new information by playing policy π^k for n^k times, which can be measured by the growth of the determinant of the covariance matrix: $\det(\Sigma_{h,i}^{k+1}) \ge \left(1 + \frac{T_{\mathrm{Trig}}}{4}\right) \det(\Sigma_{h,i}^k)$. We will show that such update can only happen $\widetilde{O}(d_{\max})$ times. In addition, the algorithm will terminate when the dataset size reaches N (Line 42 and Line 29) so that the sample complexity is always upper bounded by $O(mHTK_{\max}N)$.

We also have a policy certification part, where similar ideas have been utilized in (Dann et al., 2019; Liu et al., 2021; Ni et al., 2022) to convert regret-based analysis to sample complexity. Specifically, we maintain a pessimistic value estimate $\underline{V}_{1,i}^k(s_1)$, which satisfies $\underline{V}_{1,i}^k(s_1) \le V_{1,i}^{\pi^k}(s_1)$ with high probability (Line 22). Thus the output policy is the best approximation of Markov CCE in the policy cover. This technique can be applied to most no-regret algorithms in RL to transform regret bounds to sample complexity bounds with a better dependence on the failure probability δ.[§]

4.3. Guarantees

Our algorithm, **PReFI**, has the following guarantees for learning Markov CCE and Markov CE in linear Markov games. The sample complexity only has polynomial dependence on $d_{\max}$, which exponentially improves all the previous results for Markov games with function approximation. Note that the $\widetilde{O}(\cdot)$ notation here only hide polylog dependence on $m, H, d_{\max}, \epsilon, \delta$, and the $\log(A_{\max})$ factor in the bound can be replaced by $d_{\max}$ as in adversarial linear bandits (Bubeck et al., 2012).

Theorem 4.3. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.1. Then for ν-misspecified independent linear Markov games with $\Pi^{\mathrm{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 1 will output an $(\epsilon + 4\nu)$-approximate Markov CCE with sample complexity $O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4 \log(A_{\max})\epsilon^{-4})$.*

Theorem 4.4. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption*

[§] In (Jin et al., 2018), they show how to transform regret bounds to sample complexity bounds while the dependence on failure probability becomes $1/\delta$. This technique can improve it to $\log(1/\delta)$.

4.2. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 1 will output an $(\epsilon + 4\nu)$-approximate Markov CE with sample complexity $\widetilde{O}(m^4 H^{10} d_{\max}^4 A_{\max} \log(A_{\max})\epsilon^{-4})$.

The choice of input parameters and the proofs are deferred to Appendix D. As Markov CCE is equivalent to Markov NE in two-player zero-sum Markov games, we have the following Corollary.

Corollary 4.5. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.1. Then for ν-misspecified independent linear two-player zero-sum Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 1 will output an $(\epsilon + 4\nu)$-approximate Markov NE with sample complexity $\widetilde{O}(m^4 H^{10} d_{\max}^4 \log(A_{\max})\epsilon^{-4})$.*

5. Learning Markov NE in Independent Linear Markov Potential Games

In this section, we will focus on a special class of independent linear Markov games, namely *independent linear Markov potential games*. The existence of the potential function guarantees that the stationary points of the potential function are NE (Leonardos et al., 2021), which means the iterative best-response dynamic can converge to NE as it is similar to coordinate descent (Durand, 2018). Specifically, our algorithm **Lin-Nash-CA** (Algorithm 4) can learn pure Markov NE in independent linear Markov potential games, which generalizes the algorithm for tabular Markov potential games in (Song et al., 2021). See Table 2 for a detailed comparison with previous results.

As when the other players are fixed, player $i \in [m]$ will be in a misspecified linear MDP, existing algorithms for misspecified linear MDP can all serve as the best-response oracle. The algorithm will use the following oracle LINEARMDP_SOLVER that can solve misspecified linear MDPs. Here misspecified linear MDPs are the degenerated cases of misspecified independent linear Markov games with only one player and thus no Π^{estimate} is included, which is similar to the model in (Zanette and Wainwright, 2022).

Definition 5.1. A Markov decision process with feature ϕ is a ν-misspecified linear MDP if $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ satisfies for any rollout policy $\overline{\pi}$, target policy $\widetilde{\pi}$, value function $V \in \mathcal{V}$, we have,

$$\left|\sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}}\left[\operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi(s_h,a_h), \theta_h^{\overline{\pi},V}\right\rangle\right) - Q_h^V(s_h,a_h)\right]\right| \le \nu$$

where

$$\theta_h^{\overline{\pi},V} = \operatorname*{argmin}_{\|\theta\| \le H\sqrt{d}} \mathbb{E}_{\overline{\pi}}\left(\langle \phi(s_h,a_h), \theta\rangle - Q_h^V(s_h,a_h)\right)^2,$$

$$Q_h^V(s_h,a_h) = \mathbb{E}\left[r_h(s_h,a_h) + V_{h+1}(s_{h+1})\right].$$

Assumption 5.2. For any ν-misspecified linear MDP with feature $\phi(s,a) \in \mathbb{R}^d$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, LINEARMDP_SOLVER takes features $\phi(\cdot,\cdot)$ as input and can interact with the underlying linear MDP. Then it can output an $(\epsilon + O(\nu))$-approximate optimal policy with sample complexity $\text{LinearMDP\_SC}(\epsilon,\delta,d)$ with probability at least $1-\delta$. Without loss of generality, we assume that $\text{LinearMDP\_SC}(\epsilon,\delta,d)$ is non-decreasing w.r.t. d.

In Appendix F, we will adapt Algorithm 1 to the single-agent case to serve as LINEARMDP_SOLVER with $\text{LinearMDP\_SC}(\epsilon,\delta,d) = \widetilde{O}(H^6 d^4 \epsilon^{-2})$ and output an $(\epsilon + 4\nu)$-optimal policy (See Algorithm 3). With the best-response oracle, we provide our MARL algorithm **Lin-Nash-CA** for linear Markov potential games (Algorithm 4 in Appendix G). It is easy to see that Algorithm 4 can be implemented in the same decentralized way as Algorithm 1. Below we provide the sample complexity guarantees.

Theorem 5.3. *For ν-misspecified independent linear Markov potential games with $\Pi^{\text{estimate}} = \{\pi^k\}_{k=1}^K$, with probability at least $1-\delta$, Algorithm 4 will output an $(\epsilon + O(\nu))$-approximate pure Markov NE. The sample complexity is $O(m^2 H \epsilon^{-1} \cdot \text{LinearMDP\_SC}(\epsilon/8, \delta/(10m^2 H\epsilon^{-1}), d_{\max}))$.*

As the congestion game is a special case of linear Markov potential game (Proposition C.2), we have the following corollary if we replace the linear MDP solver with a linear bandit solver with $\text{LinearBandit\_SC}(\epsilon,\delta,d))$ sample complexity.

Corollary 5.4. *For congestion games, with probability at least $1-\delta$, Algorithm 4 will output an ϵ-approximate pure NE. The sample complexity is $O(m^2\epsilon^{-1} \cdot \text{LinearBandit\_SC}(\epsilon/8, \delta/(10m^2 H\epsilon^{-1}), F))$.*

If we use Algorithm 3 as the oracle, the sample complexity for linear Markov potential games would be $\widetilde{O}(m^2 H^7 d_{\max}^4 \epsilon^{-3})$. For linear bandits, it is easy to adapt the $\widetilde{O}(d\sqrt{K})$ regret algorithm in (Abbasi-Yadkori et al., 2011) to sample complexity $\widetilde{O}(d^2\epsilon^{-2})$, which leads to $\widetilde{O}(m^2 F^2 \epsilon^{-3})$ sample complexity for congestion games.[¶] Our algorithm significantly improves the previous result for the decentralized algorithm, which has sample complexity $\widetilde{O}(m^{12} F^6 \epsilon^{-6})$ (Cui et al., 2022).

[¶] E.g., we can use policy certification as in Algorithm 1 to find the best policy among all the policies played with no additional sample complexity.

6. Improved Sample Complexity in Tabular Case

In this section, we will present an algorithm specialized to tabular Markov games based on the policy cover technique in Algorithm 1. The sample complexity for learning an ϵ-approximate Markov CCE is $\widetilde{O}(H^6 S^2 A_{\max}\epsilon^{-2})$, which significantly improves the previous state-of-the-art result $\widetilde{O}(H^{11} S^3 A_{\max}\epsilon^{-3})$ (Daskalakis et al., 2022), and is only worse than learning an ϵ-approximate *non-Markov* CCE by a factor of HS (Jin et al., 2021b). In addition, our algorithm can learn an ϵ-approximate Markov CE with $\widetilde{O}(H^6 S^2 A_{\max}^2\epsilon^{-2})$ sample complexity, which is the first provably efficient result for learning Markov CE in tabular Markov games.

For learning CCE and CE, the adversarial bandit algorithm (Protocol 2 in Appendix H) needs to satisfy the following no-external-regret and no-swap-regret properties, respectively. The following two assumptions can be achieved by leveraging the results in (Neu, 2015) and (Blum and Mansour, 2007), which is shown in (Jin et al., 2021b).[‖]

Assumption 6.1. (No-external-regret with bandit-feedback) For any loss sequence $l^1, \ldots, l^T \in \mathbb{R}^B$ bounded between $[0, 1]$, the adversarial bandit oracle satisfies that with probability at least $1-\delta$, for all $t \leq T$, $\max_{b\in\mathcal{B}} \sum_{i=1}^t (\langle p_i, l_i\rangle - l_i(b)) \leq \mathrm{BReg}(t) := O\left(\sqrt{Bt}\log(Bt/\delta)\right)$.

Assumption 6.2. (No-swap-regret with bandit-feedback) For any loss sequence $l^1, \ldots, l^T \in \mathbb{R}^B$ bounded between $[0, 1]$, the adversarial bandit oracle satisfies that with probability at least $1-\delta$, for all $t \leq T$, $\max_{\psi\in\Psi} \sum_{i=1}^t (\langle p_i, l_i\rangle - \langle \psi \diamond p_i, l_i\rangle) \leq \mathrm{BSwapReg}(t) := O\left(B\sqrt{t}\log(Bt/\delta)\right)$. where Ψ denotes the set $\{\psi : \mathcal{B} \to \mathcal{B}\}$ which consist of all possible strategy modifications.

Due to space limitation, our algorithm **PReBO** (Algorithm 5) and the proofs are deferred to Appendix H. Here we emphasize several major differences between Algorithm 5 and Algorithm 1. First, the states and actions are no longer entangled through the feature map as in independent linear Markov games. As a result, we can use the adversarial bandit oracle to explore *individual action space* while using policy cover to explore the *shared state space*. Then there will be no inner loop for estimating the full-information feedback and saving $\widetilde{O}(\epsilon^{-2})$ factors. Second, for independent linear Markov games, each player has its own feature space so that the exploration progress is different and communication is required to synchronize. However, in tabular Markov games, all the players explore in the shared state space, so the exploration progress is inherently synchronous and no communication is required. The triggering event is that whenever a state visitation is approximately doubled, the policy cover will update, which guarantees that with high probability, the number of episodes is bounded by $\widetilde{O}(HS)$.

Theorem 6.3. *Suppose Algorithm 5 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 6.1. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 5 will output an ϵ-approximate Markov CCE with sample complexity $\widetilde{O}(H^6 S^2 A_{\max}\epsilon^{-2})$.*

Theorem 6.4. *Suppose Algorithm 5 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 6.2. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 5 will output an ϵ-approximate Markov CE with sample complexity is $\widetilde{O}(H^6 S^2 A_{\max}^2\epsilon^{-2})$.*

7. Conclusion

In this paper, we propose the independent function approximation model for Markov games and provide algorithms for different types of Markov games that can break the curse of multiagents in a large state space. We hope this work can serve as the first step towards understanding the empirical success of MARL with independent function approximation.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in neural information processing systems*, 33:13399–13412, 2020a.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20095–20107. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper/2020/file/e894d787e2fd6c133af47140aa156f00-Paper.pdf.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020c.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In

[‖] They proved a stronger version for weighted regret while we only require the unweighted version.

International Conference on Machine Learning, pages 263–272. PMLR, 2017.

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*, pages 551–560. PMLR, 2020.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

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

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012.

Shicong Cen, Yuejie Chi, Simon S Du, and Lin Xiao. Faster last-iterate convergence of policy optimization in zero-sum markov games. *arXiv preprint arXiv:2210.01050*, 2022.

Fan Chen, Song Mei, and Yu Bai. Unified algorithms for rl with decision-estimation coefficients: No-regret, pac, and reward-free learning. *arXiv preprint arXiv:2209.11745*, 2022.

Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.

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

Qiwen Cui and Simon S Du. When is offline two-player zero-sum markov game solvable? *arXiv preprint arXiv:2201.03522*, 2022a.

Qiwen Cui and Simon S Du. Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. *arXiv preprint arXiv:2206.00159*, 2022b.

Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S Du. Learning in congestion games with bandit feedback. *arXiv preprint arXiv:2206.01880*, 2022.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019.

Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

Constantinos Daskalakis, Dylan J Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33:5527–5540, 2020.

Constantinos Daskalakis, Noah Golowich, and Kaiqing Zhang. The complexity of markov equilibrium in stochastic games. *arXiv preprint arXiv:2204.03991*, 2022.

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *International Conference on Machine Learning*, pages 5166–5220. PMLR, 2022.

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019.

Stéphane Durand. *Analysis of Best Response Dynamics in Potential Games*. PhD thesis, Université Grenoble Alpes, 2018.

Liad Erez, Tal Lancewicki, Uri Sherman, Tomer Koren, and Yishay Mansour. Regret minimization and convergence to equilibria in general-sum markov games. *arXiv preprint arXiv:2207.14211*, 2022.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55 (1):119–139, 1997.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In *Conference on Learning Theory*, pages 1877–1893. PMLR, 2020.

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

Shinji Ito. A tight lower bound and efficient reduction for swap regret. *Advances in Neural Information Processing Systems*, 33:18550–18559, 2020.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021a.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021b.

Chi Jin, Qinghua Liu, and Tiancheng Yu. The power of exploiter: Provable multi-agent rl in large state spaces. In *International Conference on Machine Learning*, pages 10251–10279. PMLR, 2022.

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

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

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

Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049. PMLR, 2022.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

Gergely Neu and Julia Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Conference on Learning Theory*, pages 3049–3068. PMLR, 2020.

Chengzhuo Ni, Yuda Song, Xuezhou Zhang, Chi Jin, and Mengdi Wang. Representation learning for general-sum low-rank markov games. *arXiv preprint arXiv:2210.16976*, 2022.

Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized Q-learning in zero-sum markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

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

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Ziang Song, Song Mei, and Yu Bai. When can we learn general-sum markov games with a large number of players sample-efficiently? *arXiv preprint arXiv:2110.04184*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

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

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

Yuanhao Wang, Qinghua Liu, Yu Bai, and Chi Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent rl with function approximation. *arXiv preprint arXiv:2302.06606*, 2023.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021.

Zheng Wen and Benjamin Van Roy. Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42 (3):762–782, 2017.

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

Wei Xiong, Han Zhong, Chengshuai Shi, Cong Shen, Liwei Wang, and Tong Zhang. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent mdp and markov game. *arXiv preprint arXiv:2205.15512*, 2022.

Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. Model-based reinforcement learning is minimax-optimal for offline zero-sum markov games. *arXiv preprint arXiv:2206.04044*, 2022.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

Yuepeng Yang and Cong Ma. $O(T^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum markov games. *arXiv preprint arXiv:2209.12430*, 2022.

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

Andrea Zanette and Martin J Wainwright. Stabilizing q-learning with linear architectures for provably efficient learning. *arXiv preprint arXiv:2206.00796*, 2022.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Kaiqing Zhang, Sham Kakade, Tamer Basar, and Lin Yang. Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity. *Advances in Neural Information Processing Systems*, 33:1166–1178, 2020.

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021a.

Runyu Zhang, Zhaolin Ren, and Na Li. Gradient play in stochastic games: stationary points, convergence, and sample complexity. *arXiv preprint arXiv:2106.00198*, 2021b.

Runyu Zhang, Qinghua Liu, Huan Wang, Caiming Xiong, Na Li, and Yu Bai. Policy optimization for markov games: Unified framework and faster convergence. *arXiv preprint arXiv:2206.02640*, 2022.

Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. *arXiv preprint arXiv:2202.07511*, 2022.

A. Tables

Algorithms	Game	Equilibrium	Sample complexity	Sample complexity (tabular)	BCM
(Liu et al., 2021)	MG	NE/CE/CCE	$H^4S^2\prod_{i=1}^m A_i\epsilon^{-2}$	-	×
(Jin et al., 2021b)	ZSMG	NE	$H^5SA_{\max}\epsilon^{-2}$	-	-
(Jin et al., 2021b)	MG	NM-CCE	$H^5SA_{\max}\epsilon^{-2}$	-	✓
(Jin et al., 2021b)	MG	NM-CE	$H^5SA_{\max}^2\epsilon^{-2}$	-	✓
(Daskalakis et al., 2022)	MG	CCE	$H^{11}S^3A_{\max}\epsilon^{-3}$	-	✓
(Xie et al., 2020)	ZSMG	NE	$H^4d^3\epsilon^{-2}$	$d = SA_1A_2$	-
(Chen et al., 2021)	ZSMG	NE	$H^3d^2\epsilon^{-2}$	$d = SA_1A_2$	-
(Huang et al., 2021)	ZSMG	NE	$H^3W^2A_{\max}\epsilon^{-2}$	$W = SA_1A_2$	-
(Jin et al., 2022)	ZSMG	NE	$H^2d^2\epsilon^{-2}$	$d = SA_1A_2$	-
(Chen et al., 2022)	MG	NE/CE/CCE	$S^3(\prod_{i\in[m]} A_i)^2H^3\epsilon^{-2}$	-	×
(Ni et al., 2022)	MG	NE/CE/CCE	$H^6d^4(\prod_{i=1}^m A_i)^2\log(\lvert\Phi\rvert\lvert\Psi\rvert)\epsilon^{-2}$	$d = S\prod_{i\in[m]} A_i$	×
(Ni et al., 2022)	MG	NE/CE/CCE	$m^4H^6d^{2(L+1)^2}A_{\max}^{2(L+1)}\epsilon^{-2}$	$d = S\prod_{i\in[m]} A_i$	×
Algorithm 1 (**PReFI**)	MG	CCE	$m^4H^{10}d_{\max}^4\epsilon^{-4}$	$d_{\max} = SA_{\max}$	✓
Algorithm 1 (**PReFI**)	MG	CE	$m^4H^{10}d_{\max}^4A_{\max}\epsilon^{-4}$	$d_{\max} = SA_{\max}$	✓
Algorithm 5 (**PReBO**)	MG	CCE	$H^6S^2A_{\max}\epsilon^{-2}$	-	✓
Algorithm 5 (**PReBO**)	MG	CE	$H^6S^2A_{\max}^2\epsilon^{-2}$	-	✓

Table 1: Comparison of the models and the most related sample complexity results for MARL in Markov games. S is the number of states, m is the number of players, A_i is the number of actions for player i with $A_{\max} = \max_{i\in[m]} A_i$, ϵ is the target accuracy, and d or W is the complexity of the corresponding function class. We use **MG** to denote multi-player general-sum Markov games, **ZSMG** to denote two-player zero-sum Markov games, **NE/CE/CCE** to denote Markov Nash equilibria, Markov correlated equilibria, and Markov coarse correlated equilibria, respectively. We use the prefix (NM-) to denote non-Markov equilibria. For algorithms with function approximation, we show the parameters when applied to the tabular setting and whether breaking the curse of multiagents (**BCM**) or not in the last two columns. Polylog dependence on relevant parameters is omitted in the sample complexity results.

Algorithms	Game type	Sample complexity
(Leonardos et al., 2021)	Markov potential game	$\mathrm{poly}(\kappa, m, A_{\max}, S, H, \epsilon)$
(Ding et al., 2022)	Markov potential game	$\mathrm{poly}(\kappa, m, A_{\max}, d, H, \epsilon)$
(Song et al., 2021)	Markov potential game	$m^2H^4SA_{\max}\epsilon^{-3}$
(Cui et al., 2022) (Centralized)	Congestion game	$m^2F\epsilon^{-2}$
(Cui et al., 2022) (Decentralized)	Congestion game	$m^{12}F^6\epsilon^{-6}$
Algorithm 4 (**Lin-Nash-CA**)	Linear Markov potential game	$m^2H^7d_{\max}^4\epsilon^{-3}$
Algorithm 4 (**Lin-Nash-CA**)	Congestion game	$m^2F^2\epsilon^{-3}$

Table 2: Comparison of algorithms for learning NE in Markov potential games. κ is the distribution mismatch coefficient, S is the number of states, m is the number of players, A_i is the number of actions for player i, $A_{\max} = \max_{i\in[m]} A_i$, F is the number of facilities in congestion games, ϵ is accuracy, and $d_{\max}$ is the complexity of the function class. For (Leonardos et al., 2021; Ding et al., 2022), κ can be arbitrarily large as no exploration is considered.

B. Related Work

Tabular Markov games. Markov games, also known as stochastic games, are introduced in the seminal work (Shapley, 1953). We first discuss works that consider bandit feedback as in our paper. (Bai and Jin, 2020) provide the first provably sample-efficient MARL algorithm for two-player zero-sum Markov games, which is later improved in (Bai et al., 2020). For multi-player general-sum Markov games, (Liu et al., 2021) provide the first provably efficient algorithm with sample complexity depending on the size of joint action space $\prod_{i\in[m]} A_i$. Jin et al. (2021b); Song et al. (2021); Mao et al. (2022) utilize a decentralized algorithm to break the curse of multiagents. However, the output policy therein is non-Markov. Recently, Daskalakis et al. (2022) provide the first algorithm that can learn Markov CCE and break the curse of multiagents at the same time. Several other lines of research consider full-information feedback setting in Markov games and have attempted to prove convergence to NE/CE/CCE and/or sublinear individual regret (Sayin et al., 2021; Zhang et al., 2022; Cen et al., 2022; Yang and Ma, 2022; Erez et al., 2022; Ding et al., 2022), and offline learning setting where a dataset is given and no further interaction with the environment is permitted (Cui and Du, 2022a; Zhong et al., 2022; Yan et al., 2022; Xiong et al., 2022; Cui and Du, 2022b).

Markov games with function approximation. To tackle the curse of large state and action spaces, it is natural to incorporate existing function approximation frameworks for single-agent RL into MARL algorithms. Xie et al. (2020); Chen et al. (2021) consider linear function approximation in two-player zero-sum Markov games, which originate from linear MDP and linear mixture MDP in single-agent RL, respectively (Jin et al., 2020; Yang and Wang, 2020). Huang et al. (2021); Jin et al. (2022); Chen et al. (2022); Ni et al. (2022) consider different kinds of general function approximation, which also originate from single-agent RL literature (Jiang et al., 2017; Du et al., 2019; Agarwal et al., 2020b; Wang et al., 2020; Zanette et al., 2020; Jin et al., 2021a; Foster et al., 2021; Du et al., 2021). It is notable that all of these frameworks are based on *global* function approximation, which is centralized and suffers from the curse of multiagents when applied to tabular Markov games.

Markov potential games. Markov potential games incorporate Markovian state transition to potential games (Monderer and Shapley, 1996). Most existing results consider full-information feedback or well-explored setting and prove fast convergence of policy gradient methods to NE (Leonardos et al., 2021; Zhang et al., 2021b; Ding et al., 2022). Song et al. (2021) provide a best-response type algorithm that can *explore* in tabular Markov potential games. One important class of potential games is congestion games (Rosenthal, 1973). Cui et al. (2022) give the first non-asymptotic analysis for general congestion games with bandit feedback. We refer the readers to (Cui et al., 2022) for a more detailed background about learning in potential/congestion games. It is worth noting that for congestion games, each player is in a combinatorial bandit if other players' policies are fixed, which can be directly handled by our independent linear Markov games model, while applying potential game results lead to polynomial dependence on $A_{\max}$, which could be exponentially large in the number of the facilities in the congestion game.

Comparison with (Wang et al., 2023). There is a concurrent and independent work (Wang et al., 2023). The two works share quite a bit of results, e.g., the use of a similar function approximation model, similar algorithm design and sample complexity results for learning Markov CCE in tabular Markov games, similar discussions on the improved result by using additional communication among agents, etc. Here we highlight several differences in learning Markov CCE with linear function approximation. First, they utilize a novel second-order regret oracle and Bernstein-type concentration bounds, so that they can leverage the *single-sample* estimate instead of the *batched* estimate in our algorithm, which results in better dependence on $d_{\max}$, ϵ and H compared with our sample complexity. On the other hand, our result has no dependence on the number of actions, which is aligned with the single-agent linear MDP sample complexity, while theirs has a polynomial dependence on $A_{\max}$.[**] This difference is because they use a uniform policy to sample at the last step while we always use the on-policy samples. In fact, neither of the sample complexity bounds is strictly better than the other one and is not directly comparable as the assumptions are not the same. Second, our algorithm can use arbitrary full-information no-regret learning oracles while their results are specialized to the Expected Follow-the-Perturbed-Leader (E-FTPL) oracle (Hazan and Minasyan, 2020), which makes the policy class Π^{estimate} therein the linear argmax policy class. Our Π^{estimate} is induced by the full-information oracle being used, and the result is in this sense more agnostic. On the other hand, if we use E-FTPL, the induced Π^{estimate} has a more complicated form than the linear argmax policy class. This is because we use the optimistic estimation of the Q function in our algorithm. Third, our algorithm can work with agnostic model misspecification which is not considered in (Wang et al., 2023). Besides the differences in linear function approximation results mentioned above and the similar algorithms and sample complexity for the tabular case, we also have results for learning NE in Markov potential games, as well as learning Markov CE in general-sum Markov games, while they provide a policy mirror-descent-type algorithm for other function approximation settings, such as linear quadratic games and the settings with low Eluder dimension, with a weaker version of CCE called policy-class-restricted CCE.

C. Examples of Independent Linear Markov Games

Example 1. *(Tabular Markov games) Let $d_i = SA_i$ and set $\phi_i(s, a_i) = e_{(s,a_i)}$ be the canonical basis in $\mathbb{R}^{d_i}$ for all $i \in [m]$. Then we recover tabular Markov game with misspecification error $\nu = 0$.*

[**] In Theorem 4.3, there is a $\log(A_{\max})$ factor, which can be replaced by $d_{\max}$ by using a covering argument as in adversarial linear bandits (Bubeck et al., 2012).

Example 2. *(State abstraction Markov games) Suppose we have an abstraction function $\psi : \mathcal{S} \to \mathcal{Z}$ for all $h \in [H]$, where $\mathcal{Z}$ is a finite set as the "state abstractions" such that states with the same images have similar properties. For any $z \in \mathcal{Z}$, the model misspecification is defined as*

$$\epsilon_h(z) := \max_{s,s':\psi(s)=\psi(s')=z; i\in[m], h\in[H], \mathbf{a}\in\mathcal{A}} \left\{ |r_{h,i}(s,\mathbf{a}) - r_{h,i}(s',\mathbf{a})|, \|\mathbb{P}_h(\cdot \mid s,\mathbf{a}) - \mathbb{P}_h(\cdot \mid s',\mathbf{a})\|_1 \right\}.$$

We define ν-misspecified state abstraction Markov games to satisfy that for any policy π, we have

$$\left| \sum_{h=1}^{H} \mathbb{E}_\pi \left[\epsilon_h(\psi(s_h)) \right] \right| \le \nu,$$

which means the misspecification error is bounded under any policy π.

Proposition C.1. *ν-misspecified state abstraction Markov games (Example 2) are $H\nu$-misspecified independent linear Markov games with $\Pi^{\text{abstraction}} = \{\pi \mid \pi_h(\cdot \mid s) = \pi_h(\cdot \mid s'), \psi(s) = \psi(s')\}$, $d_i = |\mathcal{Z}|A_i$ for all $i \in [m]$ and feature $\phi_i(s, a_i) = e_{\psi(s),a_i}$ to be the canonical basis in $\mathbb{R}^{d_i}$.*

Proof. For all player i, we will let $d_i = |\mathcal{Z}|A_i$ and $\phi_i(s,a_i) = e_{(\psi(s),a_i)}$ be the canonical basis in $\mathbb{R}^{d_i}$. For any policy $\pi \in \Pi^{\text{estimate}}$, by the definition of $\theta_h^{\overline{\pi},\pi_{-i},V}$ (See Equation (1)), we have

$$\theta_h^{\overline{\pi},\pi_{-i},V}(z,a_i) = \frac{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s) Q_{h,i}^{\pi_{-i},V}(s,a_i)}{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s)} \in [0, H+1-h],$$

where $d_h^{\overline{\pi}}(\cdot)$ is the distribution over $\mathcal{S}$ induced by following policy $\overline{\pi}$ till step h. Thus we have

$$\begin{aligned}
&\operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s_h,a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V} \right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) \\
=& \operatorname{proj}_{[0,H+1-h]}\left(\theta_h^{\overline{\pi},\pi_{-i},V}(\psi(s_h),a_{h,i})\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) \\
=& \theta_h^{\overline{\pi},\pi_{-i},V}(\psi(s_h),a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) \\
=& \frac{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s) \left(Q_{h,i}^{\pi_{-i},V}(s,a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) \right)}{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s)}.
\end{aligned}$$

On the other hand, for any $z = \psi(s_h) = \psi(s'_h)$, $i \in [m]$, $h \in [H]$, $V \in \mathcal{V}$ and $\pi \in \Pi^{\text{estimate}}$, we have

$$\begin{aligned}
&\left| Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s'_h,a_{h,i}) \right| \\
=& \Big| \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)} \left[r_{h,i}(s_h,a_{h,i},a_{h,-i}) + V_{h+1}(s_{h+1}) \right] \\
&- \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s'_h)} \left[r_{h,i}(s'_h,a_{h,i},a_{h,-i}) + V_{h+1}(s_{h+1}) \right] \Big| \\
\le& \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)} \Big[|r_{h,i}(s_h,\mathbf{a}_{h,i}) - r_{h,i}(s'_h,\mathbf{a}_{h,i})| \\
&+ \left| \mathbb{E}_{s_{h+1}\sim\mathbb{P}_h(\cdot|s_h,\mathbf{a}_{h,i})} \left[V_{h+1}(s_{h+1}) \right] - \mathbb{E}_{s_{h+1}\sim\mathbb{P}_h(\cdot|s'_h,\mathbf{a}_{h,i})} \left[V_{h+1}(s_{h+1}) \right] \right| \Big] \\
&\hfill (\text{For } \pi \in \Pi^{\text{estimate}}, \text{ we have } \pi_{h,-i}(\cdot \mid s_h) = \pi_{h,-i}(\cdot \mid s'_h)) \\
\le& \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)} \left[\epsilon_h(z) + \left| \sum_{s_{h+1}\in\mathcal{S}} \left(\mathbb{P}_h(s_{h+1} \mid s_h,\mathbf{a}_{h,i}) - \mathbb{P}_h(s_{h+1} \mid s'_h,\mathbf{a}_{h,i}) \right) V_{h+1}(s_{h+1}) \right| \right] \\
\le& \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)} \left[\epsilon_h(z) + (H-h)\epsilon_h(z) \right] \\
=& (H-h+1)\epsilon_h(z).
\end{aligned}$$

Thus we have

$$\left| \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[\operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s_h,a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V} \right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i}) \right] \right|$$

$$
\begin{aligned}
&\le \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[\left| \operatorname{proj}_{[0,H+1-h]} \left(\left\langle \phi_i(s_h, a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V} \right\rangle \right) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \right| \right] \\
&= \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[\left| \frac{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s) \left(Q_{h,i}^{\pi_{-i},V}(s, a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \right)}{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s)} \right| \right] \\
&\le \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[(H-h+1)\epsilon_h(\psi(s_h)) \right] \\
&\le H\nu,
\end{aligned}
$$

where the last inequality is by the definition of ν-misspecified state abstraction Markov games. □

Example 3. *(Congestion games) Congestion games are normal-form general-sum games defined by the tuple $(\mathcal{F}, \{A_i\}_{i=1}^m, \{r^f\}_{f\in\mathcal{F}})$, where $\mathcal{F}$ is the facility set with $F = |\mathcal{F}|$, $\mathcal{A}_i \subseteq 2^{\mathcal{F}}$ is the action set for player $i \in [m]$, and $r^f(n) \in [0, 1/F]$ is a random reward function with mean $R^f(n)$ for all $n \in [m]$. For a joint action $\mathbf{a} = (a_1, \cdots, a_m)$, $n^f(\mathbf{a}) = \sum_{i=1}^m \mathbf{1}\{f \in a_i\}$ is the number of players choosing facility f and the reward collected for player i is $r_i(\mathbf{a}) = \sum_{f\in a_i} r^f(n^f(\mathbf{a}))$, which is sum of the reward from the facilities they choose.*

Proposition C.2. *Congestion games (Example 3) are independent linear Markov games with $S = 1$, $H = 1$ and $d_i = F$ for all $i \in [m]$ and misspecification error $\nu = 0$.*

Proof. As $S = 1$ and $H = 1$, we will ignore s and h in the notation. For all player i and action $a_i \in \mathcal{A}_i$, we set $\phi_i(a_i) \in \{0,1\}^F$ such that

$$
[\phi_i(a_i)]_f = \begin{cases} 1, & \forall f \in a_i \\ 0, & \forall f \notin a_i. \end{cases}
$$

We only need to construct $\theta_i^{\pi_{-i}}$ such that $\left\|\theta_i^{\pi_{-i}}\right\| \le \sqrt{F}$ and $\left\langle \phi_i(a_i), \theta_i^{\pi_{-i}} \right\rangle = \mathbb{E}_{a_{-i}\sim\pi_{-i}}[R_i(\mathbf{a})] \in [0,1]$ for all policy π and then we will have

$$
\mathbb{E}_{a_i\sim\widetilde{\pi}_i} \left[\operatorname{proj}_{[0,1]} \left\langle \phi_i(a_i), \theta_i^{\pi_{-i}} \right\rangle - \mathbb{E}_{a_{-i}\sim\pi_{-i}}[R_i(\mathbf{a})] \right] = 0
$$

for all $\widetilde{\pi}$.

For any player i and product policy π_{-i}, we can set

$$
\left[\theta_i^{\pi_{-i}}\right]_f = \mathbb{E}_{a_{-i}\sim\pi_{-i}} \left[R^f(n^f(a_{-i}) + 1) \right], \forall f \in \mathcal{F},
$$

where we use $n^f(a_{-i})$ to denote the number of players except i using facility f. As each element in $\theta_i^{\pi_{-i}}$ is bounded between $[0,1]$, we have $\left\|\theta_i^{\pi_{-i}}\right\| \le \sqrt{F}$. In addition, we have

$$
\begin{aligned}
\left\langle \phi_i(a_i), \theta_i^{\pi_{-i}} \right\rangle &= \mathbb{E}_{a_{-i}\sim\pi_{-i}} \left[\sum_{f\in a_i} (R^f(n^f(a_{-i}) + 1)) \right] = \mathbb{E}_{a_{-i}\sim\pi_{-i}} \left[\sum_{f\in a_i} R^f(n^f(\mathbf{a})) \right] \\
&= \mathbb{E}_{a_{-i}\sim\pi_{-i}} \left[R_i(\mathbf{a}) \right],
\end{aligned}
$$

which concludes the proof. □

These examples demonstrate the generality of the linear Markov games we defined.

D. Missing Parts in Section 4

D.1. No-regret Learning with Full-information Feedback Oracle

NO_REGRET_UPDATE subroutine. Consider the expert problem with B experts (Freund and Schapire, 1997). We use $\mathcal{B}$ to denote the action set with $|\mathcal{B}| = B$, and the policy $p \in \Delta(\mathcal{B})$. At round t, the adversary chooses some loss l_t (also known as the "expert advice"). Then the learner observes the loss l_t and updates the policy to p_{t+1}, which is denoted as $p_{t+1} \leftarrow$ NO_REGRET_UPDATE(l_t).

Protocol 1 No-regret Learning Algorithm

```
Initialize: Action set $\mathcal{B}$, and $p_1$ to be the uniform distribution over $\mathcal{B}$.
for $t = 1, 2, \ldots, T$ do
    Adversary chooses loss $l_t$.
    Observe loss $l_t$.
    Update $p_{t+1} \leftarrow \text{NO\_REGRET\_UPDATE}(l_t)$.
end for
```

D.2. Decentralized Implementation

Now we discuss the implementation details of the algorithm. Our algorithm can be implemented in a decentralized manner as specified below:

1. All players know the input parameters of the algorithm.
2. Each player only knows their own features $\phi_i(\cdot, \cdot)$ and observes the states, individual actions, and individual rewards in each sample trajectory.
3. All players have shared random seeds to sample from the output Markov joint policy π^{output}.
4. All players have shared random seeds to sample from the Markov joint policy π^k, which is the policy learned at episode k.
5. All players can communicate $O(1)$ bit at each episode $k \in [K]$.

V-learning (Jin et al., 2021b; Song et al., 2021; Mao et al., 2022) can be implemented with (1), (2) and (3), and SPoCMAR (Daskalakis et al., 2022) can be implemented with (1), (2), (3) and (4). Similar to the algorithm proposed in (Daskalakis et al., 2022), our algorithm can be implemented in a decentralized way with shared random seeds to enable sampling from the Markov joint policy π^k. In details, when the players want to sample $\mathbf{a} \sim \pi_h^k(\mathbf{a} \mid s) = \frac{1}{T}\sum_{t=1}^{T}\prod_{i\in[m]} \pi_{h,i}^{k,t}(a_i \mid s)$, each player samples $t \sim \mathrm{Unif}(T)$ with the shared random seed and then independently samples $a_i \sim \pi_{h,i}^{k,t}(a_i \mid s)$. Our algorithm also requires $O(1)$ communication for broadcasting the policy cover update (Line 42) and the output policy (Line 30) at each episode.[††] The total communication complexity is bounded by $O(K_{\max}) = \widetilde{O}(mHd_{\max})$ with only polylog dependence on the accuracy ϵ.

In Appendix E, we present another algorithm for MARL in independent linear Markov games without communication, which can be implemented with (1), (2), (3) and (4). To remove communication, we utilize agile policy cover update and the number of episodes becomes $K = \widetilde{O}(m^2H^4d_{\max}^2\epsilon^{-2})$. As a result, the final sample complexity will be worse than Algorithm 1. It would be an interesting future direction to study this tradeoff between communication and sample complexity.

D.3. Proofs for Algorithm 1

We will set the parameters for Algorithm 1 to be

- $\lambda = \frac{2\log(16d_{\max}mNHT/\delta)}{\log(36/35)}$
- $W = H\sqrt{d_{\max}}$
- $\beta = 16(W+H)\sqrt{\lambda + d_{\max}\log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)}$
- $T_{\mathrm{Trig}} = 64\log(8mHN^2/\delta)$
- $K_{\max} = \min\{\frac{2Hmd_{\max}\log(N+\lambda)}{\log(1+T_{\mathrm{Trig}}/4)}, N\}$

[††] Line 30 can be implemented with $O(1)$ communication at each episode by maintaining the best index and corresponding value up to the current episode k.

Algorithm 1 Policy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (**PReFI**)

```
Input: $\epsilon$, $\delta$, $d_{\max}$, $\lambda$, $\beta$, $T_{\text{Trig}}$, $K_{\max}$, $T$, $N$
Initialization: Policy Cover $\Pi = \emptyset$. $n^{\text{tot}} = 0$.
for episode $k = 1, 2, \ldots, K_{\max}$ do
    Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$, $n^k = 0$.
    for $h = H, H-1, \ldots, 1$ do                              ▷ Retrain policy with the current policy cover
        Initialize $\pi^{k,1}_{h,i}$ to be uniform policy for all player $i$. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.
        Each player $i$ initializes a no-regret learning instance (Protocol 1) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for which we will use $\text{NO\_REGRET\_UPDATE}_{h,i,s}(\cdot)$ to denote the update.
        for $t = 1, 2, \ldots, T$ do
            for $i \in [m]$ do
                Set Dataset $\mathcal{D}^{k,t}_{h,i} = \emptyset$.
                for $l = 1, 2, \ldots, \sum_{j=1}^{k-1} n^j$ do
                    Sample $\pi^l \in \Pi = \{\pi^j\}_{j=1}^{k-1}$ with probability $n^l / \sum_{j=1}^{k-1} n^j$.
                    Draw a joint trajectory $(s^l_1, \mathbf{a}^l_1, r^l_{1,i}, \ldots, s^l_h, \mathbf{a}^l_h, r^l_{h,i}, s^l_{h+1})$ from $\pi^l_{1:h-1} \circ \left(\pi^l_{h,i}, \pi^{k,t}_{h,-i}\right)$, which is the policy that follows $\pi^l$ for the first $h-1$ steps and follows $\pi^l_{h,i}, \pi^{k,t}_{h,-i}$ for step $h$.
                    Add $(s^l_h, a^l_{h,i}, r^l_{h,i}, s^l_{h+1})$ to $\mathcal{D}^{k,t}_{h,i}$.
                end for
                Set $\Sigma^{k,t}_{h,i} = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \phi_i(s,a)\phi_i(s,a)^\top$.
                Set $\overline{\theta}^{k,t}_{h,i} = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d_{\max}}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \overline{V}^k_{h+1,i}(s')\right)^2$.
                Set $\underline{\theta}^{k,t}_{h,i} = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d_{\max}}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \underline{V}^k_{h+1,i}(s')\right)^2$.
                Set $\overline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \overline{\theta}^{k,t}_{h,i} \right\rangle + \beta \|\phi_i(\cdot,\cdot)\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.
                Set $\underline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \underline{\theta}^{k,t}_{h,i} \right\rangle - \beta \|\phi_i(\cdot,\cdot)\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.
                Update $\overline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\overline{V}^k_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\overline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.
                Update $\underline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\underline{V}^k_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\underline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.
                Update the no-regret learning instance for all state $s$ at step $h$: $\pi^{k,t+1}_{h,i}(\cdot \mid s) \leftarrow \text{NO\_REGRET\_UPDATE}_{h,i,s}(1 - \overline{Q}^{k,t}_{h,i}(s,\cdot)/H)$.
            end for
        end for
        Set $\overline{V}^k_{h,i}(s) \leftarrow \operatorname{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot \text{(Swap)Reg}(T)\right)$ for all $(i,s) \in [m] \times \mathcal{S}$.
    end for
    Set $\pi^k$ to be the Markov joint policy such that $\pi^k_h(\mathbf{a}|s) = \frac{1}{T}\sum_{t=1}^T \prod_{i \in [m]} \pi^{k,t}_{h,i}(a_i|s)$.
    if $n^{\text{tot}} = N$ then
        Output $\pi^{\text{output}} = \pi^{k^{\text{output}}}$, where $k^{\text{output}} = \operatorname{argmin}_{k' \in [k]} \max_{i \in [m]} \overline{V}^{k'}_{1,i}(s_1) - \underline{V}^{k'}_{1,i}(s_1)$.
    end if
    Set $T_{h,i} = 0$, for all $h \in [H], i \in [m]$.
    repeat                                                      ▷ Update policy cover
        Reset to $s = s_1$, $n^k = n^k + 1$, $n^{\text{tot}} = n^{\text{tot}} + 1$.
        for $h = 1, 2, \ldots, H$ do
            Play $\mathbf{a} = \pi^k_h(\cdot|s)$.
            for $i \in [m]$ do
                $T_{h,i} \to T_{h,i} + \|\phi_i(s,a_i)\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}}$.
            end for
            Get next state $s'$, $s \to s'$.
        end for
    until $\exists h \in [H], i \in [m]$ such that $T_{h,i} \ge T_{\text{Trig}}$ or $n^{\text{tot}} = N$.
    Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.
end for
```

- $T = \widetilde{O}(H^4 \log(A_{\max})\epsilon^{-2})$ for Markov CCE and $T = \widetilde{O}(H^4 A_{\max} \log(A_{\max})\epsilon^{-2})$ for Markov CE
- $N = \widetilde{O}(m^2 H^4 d_{\max}^2 \epsilon^{-2})$.

We will use subscript k, t to denote the variables in episode k and inner loop t, and subscript h, i to denote the variables at step h and for player i. We will use K to denote the episode that the Algorithm 1 ends ($n^{\mathrm{tot}} = N$ or $K = K_{\max}$) . Immediately we have $K \le K_{\max} \le N$.

By the definition of the no-regret learning oracle (Assumption 4.1 and Assumption 4.2), we have the following two lemmas.

Lemma D.1. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.1. For all* $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$ *and* $s \in \mathcal{S}$ *we have*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s, a_i) \ge \max_{a_i \in \mathcal{A}_i} \frac{1}{T}\sum_{t=1}^{T} \overline{Q}_{h,i}^{k,t}(s, a_i) - \frac{H}{T}\cdot \mathrm{Reg}(T).$$

Lemma D.2. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.2. For all* $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$ *and* $s \in \mathcal{S}$ *we have*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s, a_i) \ge \max_{\psi_i \in \Psi_i} \frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s, \psi_h(a_i \mid s))\\ - \frac{H}{T}\cdot \mathrm{SwapReg}(T).\end{aligned}$$

D.3.1. CONCENTRATION

The population covariance matrix for episode k, inner loop t, step h and player i is defined as

$$\Sigma_{h,i}^{k} := \mathbb{E}\left[\Sigma_{h,i}^{k,t}\right] = \lambda I + \sum_{l=1}^{k-1} n^l \Sigma_{h,i}^{\pi^l},$$

where $\Sigma_{h,i}^{\pi^k} = \mathbb{E}_{\pi^k}\left[\phi_i(s_h, a_{h,i})\phi_i(s_h, a_{h,i})^\top\right]$. Note that $s_h^l, a_{h,i}^l$ is sampled following the same policy for each inner loop t, so the expected covariance is the same for different t.

We define $\pi^{k,\mathrm{cov}}$ to be the mixture policy of the policy cover Π^k, where policy π^l is given weight/probability $\frac{n^l}{\sum_{j=1}^{k-1} n^j}$. Then we define the on-policy population fit to be

$$\widetilde{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\| \le W} \mathbb{E}_{(s_h, a_{h,i}) \sim \pi^{k,\mathrm{cov}}}\left\{\langle \phi_i(s_h, a_{h,i}), \theta\rangle - \mathbb{E}_{a_{h,-i} \sim \pi_{h,-i}^{k,t}(\cdot \mid s)}\left[r_{h,i}(s_h, \mathbf{a}_h) + \overline{V}_{h+1,i}^{k}(s')\right]\right\}^2,$$

$$\widehat{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\| \le W} \mathbb{E}_{(s_h, a_{h,i}) \sim \pi^{k,\mathrm{cov}}}\left\{\langle \phi_i(s_h, a_{h,i}), \theta\rangle - \mathbb{E}_{a_{h,-i} \sim \pi_{h,-i}^{k,t}(\cdot \mid s)}\left[r_{h,i}(s_h, \mathbf{a}_h) + \underline{V}_{h+1,i}^{k}(s')\right]\right\}^2.$$

Lemma D.3. *(Concentration) With probability at least* $1 - \delta/2$, *for all* $k \in [K]$, $h \in [H]$, $t \in [T]$, $i \in [m]$, *we have*

$$\left\|\overline{\theta}_{h,i}^{k,t} - \widetilde{\theta}_{h,i}^{k,t}\right\|_{\Sigma_{h,i}^{k}} \le 8(W + H)\sqrt{\lambda + d_i \log(32WN(W + H)) + 4\log(8mK_{\max}HT/\delta)} \le \beta/2, \tag{2}$$

$$\left\|\underline{\theta}_{h,i}^{k,t} - \widehat{\theta}_{h,i}^{k,t}\right\|_{\Sigma_{h,i}^{k}} \le 8(W + H)\sqrt{\lambda + d_i \log(32WN(W + H)) + 4\log(8mK_{\max}HT/\delta)} \le \beta/2, \tag{3}$$

$$\frac{1}{2}\Sigma_{h,i}^{k,t} \preceq \Sigma_{h,i}^{k} \preceq \frac{3}{2}\Sigma_{h,i}^{k,t}. \tag{4}$$

Proof. By applying Lemma I.8 with $Y_{\max} = H$ and union bound, (2) and (3) holds with probability at least $1 - \delta/4$. For (4), we can prove it holds with probability at least $1 - \delta/4$ by applying Lemma I.9 with $\lambda > \frac{2\log(16 d_i m K_{\max} HT/\delta)}{\log(36/35)}$ and union bound. □

Lemma D.4. *With probability at least $1-\delta/2$, the following two events hold:*

- *Suppose at episode k, Line 42: $T_{h,i} \geq T_{\mathrm{Trig}}$ is triggered, then we have*

$$\mathbb{E}_{\pi^k} \left\| \phi_i(s_h, a_{h,i}) \right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} \geq \frac{1}{2n^k} \sum_{j=1}^{n^k} \left\| \phi_i(s_h^{k,j}, a_{h,i}^{k,j}) \right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} \geq \frac{T_{\mathrm{Trig}}}{2n^k},$$

where j denotes the j-th trajectory collected in the policy cover update (Line 33).

- *For any $k \in [K_{\max}]$, $h \in [H]$, $i \in [m]$, we have*

$$\mathbb{E}_{\pi^k} \left\| \phi_i(s_h, a_{h,i}) \right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} \leq \frac{2T_{\mathrm{Trig}}}{n^k}.$$

Proof. Note that if at episode k, $T_{h,i} \geq T_{\mathrm{Trig}}$ is triggered, we will have $n^k \leq N$ as otherwise $n^{\mathrm{tot}} = N$ will be triggered. By Lemma I.2 with $X_j = \left\| \phi_i(s_h^{k,j}, a_{h,i}^{k,j}) \right\|_{[\Sigma^{1,k}_{h,i}]^{-1}}$, $n_{\max} = N$ and $T_{\mathrm{Trig}} \geq 64 \log(8mHK_{\max}N/\delta)$, we have that the argument holds with probability at least $1 - \delta/(2mK_{\max}H)$ for any fixed $k \in [K_{\max}]$, $h \in [H]$ and $i \in [m]$. Then we can prove the lemma by applying union bound. □

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma D.3 and Lemma D.4 hold, which is with probability at least $1-\delta$ by Lemma D.3 and Lemma D.4.

We define the misspecification error to be

$$\overline{\Delta}^{k,t}_{h,i}(s, a_i) := \mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \overline{V}^k_{h+1,i}(s') \right] - \mathrm{proj}_{[0,H+1-h]} \left(\left\langle \phi_i(s, a_i), \widetilde{\theta}^{k,t}_{h,i} \right\rangle \right),$$

$$\underline{\Delta}^{k,t}_{h,i}(s, a_i) := \mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \underline{V}^k_{h+1,i}(s') \right] - \mathrm{proj}_{[0,H+1-h]} \left(\left\langle \phi_i(s, a_i), \widehat{\theta}^{k,t}_{h,i} \right\rangle \right).$$

Then by the definition of ν-misspecified linear Markov games, we have the following lemma.

Lemma D.5. *For any policy π, we have*

$$\left| \sum_{h=1}^H \mathbb{E}_\pi \left[\overline{\Delta}^{k,t}_{h,i}(s, a_i) \right] \right| \leq \nu, \qquad \left| \sum_{h=1}^H \mathbb{E}_\pi \left[\underline{\Delta}^{k,t}_{h,i}(s, a_i) \right] \right| \leq \nu.$$

D.3.2. Proofs for Markov CCE

Lemma D.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$ we have*

$$\begin{aligned} -\overline{\Delta}^{k,t}_{h,i}(s, a_i) &\leq \overline{Q}^{k,t}_{h,i}(s, a_i) - \left[\mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \overline{V}^k_{h+1,i}(s') \right] \right] \\ &\leq 3\beta \left\| \phi_i(s, a_i) \right\|_{[\Sigma^k_{h,i}]^{-1}} - \overline{\Delta}^{k,t}_{h,i}(s, a_i), \end{aligned}$$

$$\begin{aligned} -3\beta \left\| \phi_i(s, a_i) \right\|_{[\Sigma^k_{h,i}]^{-1}} - \underline{\Delta}^{k,t}_{h,i}(s, a_i) &\leq \underline{Q}^{k,t}_{h,i}(s, a_i) - \left[\mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \underline{V}^k_{h+1,i}(s') \right] \right] \\ &\leq -\underline{\Delta}^{k,t}_{h,i}(s, a_i). \end{aligned}$$

Proof. We only prove the first argument and the second one holds similarly.

By Lemma D.3, for any $s \in \mathcal{S}$, $a_i \in \mathcal{A}_i$, $h \in [H]$, $i \in [m]$, $k \in [K]$, we have

$$\left| \left\langle \phi_i(s, a_i), \overline{\theta}^{k,t}_{h,i} - \widetilde{\theta}^{k,t}_{h,i} \right\rangle \right| \leq \left\| \phi_i(s, a_i) \right\|_{[\Sigma^k_{h,i}]^{-1}} \left\| \overline{\theta}^{k,t}_{h,i} - \widetilde{\theta}^{k,t}_{h,i} \right\|_{\Sigma^k_{h,i}} \leq \beta/2 \left\| \phi_i(s, a_i) \right\|_{[\Sigma^k_{h,i}]^{-1}},$$

where the first inequality is from Cauchy-Schwarz inequality. As a result, we have

$$\begin{aligned}
\overline{Q}_{h,i}^{k,t}(s,a_i) =& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}_{h,i}^{k,t}\right\rangle + \beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k,t}]^{-1}}\right) \\
\geq& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}_{h,i}^{k,t}\right\rangle + \frac{1}{2}\beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k}]^{-1}}\right) && \text{(Lemma D.3)} \\
\geq& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}_{h,i}^{k,t}\right\rangle\right) \\
=& \mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^{k}(s')\right] - \overline{\Delta}_{h,i}^{k,t}(s,a_i)
\end{aligned}$$

and

$$\begin{aligned}
\overline{Q}_{h,i}^{k,t}(s,a_i) =& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}_{h,i}^{k,t}\right\rangle + \beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k,t}]^{-1}}\right) \\
\leq& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}_{h,i}^{k,t}\right\rangle + 2\beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k}]^{-1}}\right) && \text{(Lemma D.3)} \\
\leq& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}_{h,i}^{k,t}\right\rangle + 3\beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k}]^{-1}}\right) \\
\leq& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}_{h,i}^{k,t}\right\rangle\right) + 3\beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k}]^{-1}} \\
=& \mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^{k}(s')\right] - \overline{\Delta}_{h,i}^{k,t}(s,a_i) + 3\beta \left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^{k}]^{-1}},
\end{aligned}$$

which concludes the proof. □

Lemma D.7. *(Optimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$\overline{V}_{1,i}^{k}(s_1) \geq V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[\frac{1}{T}\sum_{t=1}^{T} \overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right] \geq V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) - \nu.$$

Proof. For any $k \in [K]$, $i \in [m]$, under the good event $\mathcal{G}$, we have

$$\begin{aligned}
&\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
=& \operatorname{proj}_{[0,H]}\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i\in\mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) + \frac{H}{T}\cdot \operatorname{Reg}(T)\right) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
\geq& \operatorname{proj}_{[0,H]}\left(\max_{a_{1,i}\in\mathcal{A}_i} \frac{1}{T}\sum_{t=1}^{T} \overline{Q}_{1,i}^{k,t}(s_1,a_{1,i})\right) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) && \text{(Lemma D.1)} \\
\geq& \max_{a_{1,i}\in\mathcal{A}_i} \frac{1}{T}\sum_{t=1}^{T}\left\{\mathbb{E}_{a_{-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s,\mathbf{a}) + \overline{V}_{2,i}^{k}(s')\right] - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right\} - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) && \text{(Lemma D.6)} \\
\geq& \mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[r_{1,i}(s_1,\mathbf{a}_1) + \overline{V}_{2,i}^{k}(s') - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
=& \mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[\overline{V}_{2,i}^{k}(s_2) - V_{2,i}^{\dagger,\pi_{-i}^{k}}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] \\
\geq& -\mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right] \\
\geq& -\nu, && \text{(Lemma D.5)}
\end{aligned}$$

where we use $\mathbb{E}_{\dagger,\pi_{-i}^{k}}$ to denote $\mathbb{E}_{\pi_i',\pi_{-i}^{k}}$ such that π_i' is a best response of π_{-i}^{k}. □

Lemma D.8. *(Pessimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$\underline{V}_{1,i}^{k}(s_1) \leq V_{1,i}^{\pi^k}(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\pi^k} \frac{1}{T} \left[\sum_{t=1}^{T} \underline{\Delta}_{h,i}^{k,t}(s_h, a_{h,i}) \right] \leq V_{1,i}^{\pi^k}(s_1) + \nu.$$

Proof. For any $k \in [K]$, $i \in [m]$, under the good event $\mathcal{G}$, we have

$$\begin{aligned}
&\underline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1) \\
=&\frac{1}{T}\sum_{t=1}^{T}\sum_{a \in \mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\underline{Q}_{1,i}^{k,t}(s_1, a_{1,i}) - V_{1,i}^{\pi^k}(s_1) \\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i} \in \mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\left[\mathbb{E}_{a_{1,-i} \sim \pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}_{2,i}^{k}(s_2)\right] - \underline{\Delta}_{1,i}^{k,t}(s_1, a_i)\right] - V_{1,i}^{\pi^k}(s_1) \quad \text{(Lemma D.6)} \\
=&\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^k(\cdot|s_1)}\left[r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}_{2,i}^{k}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1, a_i)\right] - V_{1,i}^{\pi^k}(s_1) \\
=&\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^k(\cdot|s_1)}\left[\underline{V}_{2,i}^{k}(s_2) - V_{2,i}^{\pi^k}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1, a_i)\right] \\
\leq&-\sum_{h=1}^{H}\mathbb{E}_{\pi^k}\left[\frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{h,i}^{k,t}(s_h, a_{h,i})\right] \\
\leq&\nu, \quad \text{(Lemma D.5)}
\end{aligned}$$

which concludes the proof. □

Lemma D.9. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) - V_{1,i}^{\pi^k}(s_1) - 2\nu \leq \overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1) \leq 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^{k}]^{-1}} + \frac{H^2}{T}\cdot\mathrm{Reg}(T) + 2\nu.$$

Proof. The first inequality is from Lemma D.7 and Lemma D.8. Now we prove the second argument. Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have

$$\begin{aligned}
&\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1) \\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\overline{Q}_{1,i}^{k,t}(s_1, a_{1,i}) + \frac{H}{T}\cdot\mathrm{Reg}(T) - \frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i} \in \mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\underline{Q}_{1,i}^{k,t}(s_1, a_{1,i}) \\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i} \in \mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\Big(\Big[\mathbb{E}_{a_{1,-i} \sim \pi_{1,-i}^{k,t}(\cdot|s)}\Big[r_{h,i}(s_1, \mathbf{a}_1) + \overline{V}_{2,i}^{k}(s_2)\Big]\Big] + 3\beta\|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^{k}]^{-1}} \\
&-\overline{\Delta}_{1,i}^{k,t}(s, a_{1,i})\Big) - \frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i} \in \mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i} \mid s_1)\Big(\Big[\mathbb{E}_{a_{1,-i} \sim \pi_{1,-i}^{k,t}(\cdot|s_1)}\Big[r_{h,i}(s_1, \mathbf{a}_1) + \underline{V}_{2,i}^{k}(s_2)\Big]\Big] \\
&-3\beta\|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^{k}]^{-1}} - \underline{\Delta}_{1,i}^{k,t}(s, a_{1,i})\Big) + \frac{H}{T}\cdot\mathrm{Reg}(T) \quad \text{(Lemma D.6)} \\
\leq&\frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^{k,t}(\cdot|s_1)}\left[\overline{V}_{2,i}^{k}(s_2) - \underline{V}_{2,i}^{k}(s_2)\right]\right] + \frac{H}{T}\cdot\mathrm{Reg}(T) \\
&+\mathbb{E}_{a_{1,i} \sim \pi_{1,i}^{k,t}(\cdot|s_1)}\left[6\beta\|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^{k}]^{-1}} - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) - \frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i})\right]
\end{aligned}$$

$$\begin{aligned}
=&\mathbb{E}_{\pi_1^k}\left[\overline{V}^k_{2,i}(s_2) - \underline{V}^k_{2,i}(s_2)\right] + \frac{H}{T}\cdot \mathrm{Reg}(T) \\
&+ \mathbb{E}_{a_{1,i}\sim\pi^{k,t}_{1,i}(\cdot|s_1)}\left[6\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma^k_{1,i}]^{-1}} - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}^{k,t}_{1,i}(s_1,a_{1,i}) - \frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}^{k,t}_{1,i}(s_1,a_{1,i})\right] \\
\le& 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma^k_{h,i}]^{-1}} - \mathbb{E}_{\pi^k}\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\left(\overline{\Delta}^{k,t}_{h,i}(s_h,a_{h,i}) + \underline{\Delta}^{k,t}_{h,i}(s_h,a_{h,i})\right) + \frac{H^2}{T}\cdot\mathrm{Reg}(T) \\
\le& 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma^k_{h,i}]^{-1}} + 2\nu + \frac{H^2}{T}\cdot\mathrm{Reg}(T),
\end{aligned}$$

which completes the proof. □

Lemma D.10. *Under the good event $\mathcal{G}$, for all $i \in [m]$, we have*

$$\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}} \le 4T_{\mathrm{Trig}} d_i \log\left(1 + \frac{N}{d_i\lambda}\right).$$

Proof. First, by the triggering condition, we have

$$\sum_{j=1}^{n^k}\left\|\phi_i(s_h^j,a_{h,i}^j)\right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} = \sum_{j=1}^{n^k-1}\left\|\phi_i(s_h^j,a_{h,i}^j)\right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} + \left\|\phi_i(s_h^{n^k},a_{h,i}^{n^k})\right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} \le T_{\mathrm{Trig}} + 1,$$

where j denotes the j-th trajectory collected in the policy cover update (Line 33). By Lemma D.4, we have

$$n^k\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}} \le 2n^k\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}} \le 4T_{\mathrm{Trig}}.$$

Then by Lemma I.6, we have

$$n^k\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}} \le 4T_{\mathrm{Trig}}\log\frac{\det(\Sigma^{k+1}_{h,i})}{\det(\Sigma^k_{h,i})}.$$

Thus we have

$$\begin{aligned}
\sum_{k=1}^{K} n^k\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}} \le& \sum_{k=1}^{K} 4T_{\mathrm{Trig}}\log\frac{\det(\Sigma^{k+1}_{h,i})}{\det(\Sigma^k_{h,i})} \\
=& 4T_{\mathrm{Trig}}\log\frac{\det(\Sigma^{K+1}_{h,i})}{\det(\Sigma^1_{h,i})} \\
\le& 4T_{\mathrm{Trig}}\left[d_i\log\left(\frac{d_i\lambda+N}{d_i}\right) - d_i\log(\lambda)\right] \\
=& 4T_{\mathrm{Trig}} d_i\log\left(1+\frac{N}{d_i\lambda}\right),
\end{aligned}$$

where we utilized the fact that

$$\log\det(\Sigma^{K+1}_{h,i}) \le d_i\log\left(\frac{\mathrm{trace}(\Sigma^{K+1}_{h,i})}{d_i}\right) \le d_i\log\left(\frac{d_i\lambda+N}{d_i}\right),$$

and complete the proof. □

Lemma D.11. *Under the good event $\mathcal{G}$, we have*

$$\begin{aligned}
&\sum_{k=1}^{K} n^k\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \\
\le& 6mH\beta\sqrt{4N(T_{\mathrm{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)} + \frac{H^2N}{T}\cdot\mathrm{Reg}(T) + 2\nu N.
\end{aligned}$$

Proof. By Lemma D.12, under the good event $\mathcal{G}$, we have $\sum_{k=1}^{K} n^k = n^{\text{tot}} = N$. Thus we have

$$\begin{aligned}
&\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \\
\leq &\sum_{k=1}^{K} n^k \max_{i\in[m]} \left[6\beta\mathbb{E}_{\pi^k} \sum_{h=1}^{H} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}\right] + \frac{H^2}{T}\sum_{k=1}^{K} n^k \text{Reg}(T) + 2\nu N && \text{(Lemma D.9)} \\
\leq &6\beta \sum_{i\in[m]} \sum_{h=1}^{H} \sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \sqrt{\|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}^2} + \frac{H^2 N}{T} \cdot \text{Reg}(T) + 2\nu N \\
\leq &6\beta \sum_{i\in[m]} \sum_{h=1}^{H} \sum_{k=1}^{K} n^k \sqrt{\mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}^2} + \frac{H^2 N}{T} \cdot \text{Reg}(T) + 2\nu N && \text{(Concavity of } f(x) = \sqrt{x}\text{)} \\
\leq &6\beta \sum_{i\in[m]} \sum_{h=1}^{H} \sqrt{\sum_{k=1}^{K} n^k} \sqrt{\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}^2} + \frac{H^2 N}{T} \cdot \text{Reg}(T) + 2\nu N && \text{(Cauchy–Schwarz inequality)} \\
\leq &6\beta \sum_{i\in[m]} \sum_{h=1}^{H} \sqrt{N4(T_{\text{Trig}} + 1) d_i \log\left(1 + \frac{N}{d_i\lambda}\right)} + \frac{H^2 N}{T} \cdot \text{Reg}(T) + 2\nu N && \text{(Lemma D.10)} \\
\leq &6\beta m H \sqrt{N4(T_{\text{Trig}} + 1) d_{\max} \log\left(1 + \frac{N}{\lambda}\right)} + \frac{H^2 N}{T} \cdot \text{Reg}(T) + 2\nu N.
\end{aligned}$$

□

Lemma D.12. *Under the good event $\mathcal{G}$, we have*

$$K \leq \frac{2Hmd_{\max}\log(N+\lambda)}{\log(1 + T_{\text{Trig}}/4)},$$

which means $K < K_{\max}$ and Algorithm 1 ends due to Line 42 ($n^{\text{tot}} = N_{\max}$).

Proof. By Lemma D.4, for any player i and $h \in [H]$, whenever $T_{h,i}^k \geq T_{\text{Trig}}$ is triggered, with probability at least $1-\delta$ we have

$$\begin{aligned}
n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}^2 \geq &\frac{1}{2} n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^{k,1}]^{-1}}^2 && \text{(Lemma D.3)} \\
\geq &\frac{1}{4} \sum_{j=1}^{n^k} \left\|\phi_i(s_h^j, a_{h,i}^j)\right\|_{[\Sigma_{h,i}^{k,1}]^{-1}}^2 && \text{(Lemma D.4)} \\
\geq &\frac{T_{\text{Trig}}}{4}.
\end{aligned}$$

Then by Lemma I.6, we have

$$\frac{\det(\Sigma_{h,i}^{k+1})}{\det(\Sigma_{h,i}^k)} \geq 1 + n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}^2 \geq 1 + \frac{T_{\text{Trig}}}{4}.$$

Suppose $s_{h,i}$ is the number of triggering $T_{h,i}^k \geq T_{\text{Trig}}$ at level h and player i, then we have

$$\frac{\det(\Sigma_{h,i}^{K+1})}{\det(\Sigma_{h,i}^1)} \geq \left(1 + \frac{T_{\text{Trig}}}{4}\right)^{s_{h,i}}.$$

In addition, we have

$$\log(\det(\Sigma_{h,i}^1)) = d_i \log(\lambda), \log\det((\Sigma_{h,i}^{K+1})) \leq d_i \log\left(\frac{\text{trace}(\Sigma_{h,i}^{K+1})}{d_i}\right) \leq d_i \log\left(\frac{d_i\lambda + N}{d_i}\right),$$

which gives

$$s_{h,i} \leq \frac{d_i \log(N/d_i + \lambda)}{\log(1 + T_{\text{Trig}}/4)}.$$

Thus, the total number of triggering is bounded by

$$\sum_{i\in[m]} \sum_{h\in[H]} s_{h,i} + 1 \leq \frac{2mHd_{\max} \log(N + \lambda)}{\log(1 + T_{\text{Trig}}/4)},$$

where the additional 1 is from the event $n^{\text{tot}} = N$. □

Theorem 4.3. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.1. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1 - \delta$, Algorithm 1 will output an $(\epsilon + 4\nu)$-approximate Markov CCE with sample complexity $O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4 \log(A_{\max})\epsilon^{-4})$.*

Proof. Under the good event $\mathcal{G}$, by Lemma $D.12$, the algorithm ends by $n^{\text{tot}} = N$. By Lemma D.11, under the good event $\mathcal{G}$, which happens with probability at least $1 - \delta$ (Lemma D.3 and Lemma D.4), we have

$$\begin{aligned}
&\min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \\
\leq &\frac{1}{N} \sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \\
\leq &6mH\beta \sqrt{4(T_{\text{Trig}} + 1)d_{\max} \log\left(1 + \frac{N}{\lambda}\right)/N} + \frac{H^2}{T} \cdot \text{Reg}(T) + 2\nu.
\end{aligned}$$

By setting $N = \widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T = \widetilde{O}(H^4 \log(A_{\max})\epsilon^{-2})$, we can have

$$\min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \leq \epsilon + 2\nu.$$

Then by Lemma D.9 we have

$$\begin{aligned}
\max_{i\in[m]} \left(V_{1,i}^{\dagger,\pi_{-i}^{\text{output}}}(s_1) - V_{1,i}^{\pi^{\text{output}}}(s_1)\right) &\leq \max_{i\in[m]} \left(\overline{V}_{1,i}^{k^{\text{output}}}(s_1) - \underline{V}_{1,i}^{k^{\text{output}}}(s_1)\right) + 2\nu \\
&= \min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) + 2\nu \\
&\leq \epsilon + 4\nu,
\end{aligned}$$

which completes the proof. □

By Proposition C.1, we have the following corollary for state abstraction Markov games, which is defined in Appendix C. Note that the feature is the same $\phi_i(s, a_i) = \phi_i(s', a_i)$ if $\psi(s) = \psi(s')$, so $\pi^{k,t} \in \Pi^{\text{abstraction}}$ for all $(k, t) \in [K] \times [T]$ as the full-information feedback would be the same for s and s' mapped to the same abstraction and then the policy would be same as well.

Corollary D.13. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.1. Then for ν-misspecified state abstraction Markov games, with probability at least $1 - \delta$, Algorithm 1 will output an $(\epsilon + 4H\nu)$-approximate Markov NE. The sample complexity is $O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}|\mathcal{Z}|^4A_{\max}^4 \log(A_{\max})\epsilon^{-4})$.*

D.3.3. Proofs for Markov CE

Lemma D.14. *(Optimism) Let $\psi_i^k = \text{argmax}_{\psi_i} V_{1,i}^{\psi\diamond\pi^k}(s_1)$ for all $k \in [K]$ and $i \in [m]$. Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$\overline{V}_{1,i}^k(s_1) \geq \max_{\psi_i} V_{1,i}^{\psi_i\diamond\pi^k}(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\psi_i^k\diamond\pi^k} \left[\frac{1}{T} \sum_{t=1}^{T} \overline{\Delta}_{h,i}^{k,t}(s_h, a_{h,i})\right] \geq \max_{\psi_i} V_{1,i}^{\psi_i\diamond\pi^k}(s_1) - \nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s_1 \in \mathcal{S}$, we have

$$
\begin{aligned}
&\overline{V}^k_{1,i}(s_1) - \max_{\psi_i} V^{\psi_i \diamond \pi^k}_{1,i}(s_1) \\
=& \operatorname{proj}_{[0,H]} \left(\frac{1}{T} \sum_{t=1}^{T} \sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s_1) \overline{Q}^{k,t}_{1,i}(s_1, a_{1,i}) + \frac{H}{T} \cdot \mathrm{SwapReg}(T) \right) - V^{\dagger, \pi^k_{-i}}_{1,i}(s_1) \\
\geq& \operatorname{proj}_{[0,H]} \left(\max_{\psi_{1,i}} \frac{1}{T} \sum_{t=1}^{T} \sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s_1) \overline{Q}^{k,t}_{1,i}(s_1, \psi_1(a_{1,i} \mid s_1)) \right) - V^{\dagger, \pi^k_{-i}}_{1,i}(s_1) && \text{(Lemma D.1)} \\
\geq& \max_{\psi_{1,i}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\mathbf{a}_1 \sim \psi_{1,i} \diamond \pi^{k,t}_1(\cdot \mid s_1)} \left[r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}^k_{2,i}(s_2) - \overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) \right] - V^{\dagger, \pi^k_{-i}}_{1,i}(s_1) && \text{(Lemma D.6)} \\
\geq& \mathbb{E}_{\psi^k_{1,i} \diamond \pi^k_1} \left[r_{1,i}(s_1, \mathbf{a}_1) + \overline{V}^k_{2,i}(s') - \frac{1}{T} \sum_{t=1}^{T} \overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) \right] - V^{\dagger, \pi^k_{-i}}_{1,i}(s_1) \\
=& \mathbb{E}_{\psi^k_{1,i} \diamond \pi^k_1} \left[\overline{V}^k_{2,i}(s_2) - V^{\dagger, \pi^k_{-i}}_{2,i}(s_2) - \frac{1}{T} \sum_{t=1}^{T} \overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) \right] \\
\geq& - \mathbb{E}_{\psi^k_i \diamond \pi^k} \left[\sum_{h=1}^{H} \frac{1}{T} \sum_{t=1}^{T} \overline{\Delta}^{k,t}_{h,i}(s_h, a_{h,i}) \right] \\
\geq& - \nu, && \text{(Lemma D.5)}
\end{aligned}
$$

which concludes the proof. □

Lemma D.15. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$
\begin{aligned}
\max_{\psi_i} V^{\psi_i \diamond \pi^k}_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1) - 2\nu &\leq \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \\
&\leq 6\beta \mathbb{E}_{\pi^k} \sum_{h=1}^{H} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma^k_{h,i}]^{-1}} + \frac{H^2}{T} \cdot \mathrm{SwapReg}(T) + 2\nu.
\end{aligned}
$$

Proof. The first inequality is from Lemma D.14 and Lemma D.8. Now we prove the second inequality. Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have

$$
\begin{aligned}
&\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \\
\leq& \frac{1}{T} \sum_{t=1}^{T} \sum_{a_{1,i} \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s) \overline{Q}^{k,t}_{1,i}(s_1, a_{1,i}) + \frac{H}{T} \cdot \mathrm{SwapReg}(T) \\
&- \frac{1}{T} \sum_{t=1}^{T} \sum_{a_{1,i} \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s_1) \underline{Q}^{k,t}_{1,i}(s_1, a_{1,i}) \\
\leq& \frac{1}{T} \sum_{t=1}^{T} \sum_{a_{1,i} \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s_1) \Big(\Big[\mathbb{E}_{a_{1,-i} \sim \pi^{k,t}_{1,-i}(\cdot \mid s)} \left[r_{1,i}(s_1, \mathbf{a}_1) + \overline{V}^k_{2,i}(s_2) \right] \Big] + 3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma^k_{1,i}]^{-1}} \\
&- \overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) \Big) - \frac{1}{T} \sum_{t=1}^{T} \sum_{a_{1,i} \in \mathcal{A}_i} \pi^{k,t}_{1,i}(a_{1,i} \mid s_1) \Big(\Big[\mathbb{E}_{a_{1,-i} \sim \pi^{k,t}_{1,-i}(\cdot \mid s_1)} \left[r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}^k_{2,i}(s_2) \right] \Big] \\
&- 3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma^k_{1,i}]^{-1}} - \underline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) \Big) + \frac{H}{T} \cdot \mathrm{SwapReg}(T) && \text{(Lemma D.6)} \\
=& \frac{1}{T} \sum_{t=1}^{T} \Big(\mathbb{E}_{\mathbf{a}_1 \sim \pi^{k,t}_1(\cdot \mid s_1)} \left[\overline{V}^k_{2,i}(s_2) - \underline{V}^k_{2,i}(s_2) \right] + \mathbb{E}_{a_{1,i} \sim \pi^{k,t}_{1,i}(\cdot \mid s_1)} \Big[6\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma^k_{1,i}]^{-1}}
\end{aligned}
$$

$$
\begin{aligned}
&-\overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) - \underline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i})\Big]\Big) + \frac{H}{T} \cdot \mathrm{SwapReg}(T) \\
=&\mathbb{E}_{\pi_1^k}\left[\overline{V}^k_{2,i}(s_2) - \underline{V}^k_{2,i}(s_2)\right] + \mathbb{E}_{a_{1,i}\sim\pi^k_{1,i}}\left[6\beta\,\|\phi_i(s_1, a_{1,i})\|_{[\Sigma^k_{1,i}]^{-1}} - \overline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i}) - \underline{\Delta}^{k,t}_{1,i}(s_1, a_{1,i})\right] \\
&+ \frac{H}{T} \cdot \mathrm{SwapReg}(T) \\
\le& 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h, a_{h,i})\|_{[\Sigma^k_{h,i}]^{-1}} - \mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left(\overline{\Delta}^{k,t}_{h,i}(s_h, a_{h,i}) + \underline{\Delta}^{k,t}_{h,i}(s_h, a_{h,i})\right) + \frac{H^2}{T} \cdot \mathrm{SwapReg}(T) \\
\le& 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h, a_{h,i})\|_{[\Sigma^k_{h,i}]^{-1}} + \frac{H^2}{T} \cdot \mathrm{SwapReg}(T) + 2\nu.
\end{aligned}
$$

□

Lemma D.16. *Under the good event $\mathcal{G}$, we have*

$$
\begin{aligned}
&\sum_{k=1}^{K} n^k \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \\
\le& 6mH\beta\sqrt{4N(T_{\mathrm{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)} + \frac{H^2N}{T} \cdot \mathrm{SwapReg}(T) + 2\nu.
\end{aligned}
$$

Proof. The proof is similar to the proof for Lemma D.11, where the only difference is that we replace Lemma D.11 with Lemma D.15 in the proof. □

Theorem 4.4. *Suppose Algorithm 1 is instantiated with no-regret learning oracles satisfying Assumption 4.2. Then for ν-misspecified independent linear Markov games with $\Pi^{\mathrm{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 1 will output an $(\epsilon + 4\nu)$-approximate Markov CE with sample complexity $\widetilde{O}(m^4H^{10}d^4_{\max}A_{\max}\log(A_{\max})\epsilon^{-4})$.*

Proof. Under the good event $\mathcal{G}$, by Lemma $D.12$, the algorithm ends by $n^{\mathrm{tot}} = N$. By Lemma D.16, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma D.3 and Lemma D.4), we have

$$
\begin{aligned}
&\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \\
\le&\frac{1}{N}\sum_{k=1}^{K} n^k \sum_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \\
\le& 6mH\beta\sqrt{4(T_{\mathrm{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)/N} + \frac{H^2}{T} \cdot \mathrm{SwapReg}(T) + 2\nu.
\end{aligned}
$$

By setting $N = \widetilde{O}(m^2H^4d^3_{\max}\epsilon^{-2})$ and $T = \widetilde{O}(H^4A_{\max}\log(A_{\max})\epsilon^{-2})$, we can have

$$
\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon + 2\nu.
$$

Then by Lemma D.15, we have

$$
\begin{aligned}
\max_{i\in[m]}\left(\max_{\psi_i} V^{\psi_i\diamond\pi^{\mathrm{output}}}_{1,i}(s_1) - V^{\pi^{\mathrm{output}}}_{1,i}(s_1)\right) \le& \max_{i\in[m]}\left(\overline{V}^{k^{\mathrm{output}}}_{1,i}(s_1) - \underline{V}^{k^{\mathrm{output}}}_{1,i}(s_1)\right) + 2\nu \\
=& \min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) + 2\nu \\
\le& \epsilon + 4\nu,
\end{aligned}
$$

which thus completes the proof. □

E. Algorithms for Learning Markov CCE/CE without Communication

In this section, we present a communication-free algorithm for independent linear Markov games. The key difference is that we leverage an agile policy cover update scheme, i.e., the policy cover is updated whenever a new π^k is learned (Line 25), and the policy certification is replaced by a uniform sampling procedure (Line 27).

Algorithm 2 Communication-free **P**olicy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (Communication-free **PReFI**)

```
 1: **Input:** $\lambda, \beta, K, T$
 2: **Initialization:** Policy Cover $\Pi = \emptyset$.
 3: **for** episode $k = 1, 2, \ldots, K$ **do**
 4:     Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$.
 5:     **for** $h = H, H-1, \ldots, 1$ **do**                    ▷ Retrain policy with the current policy cover
 6:         Initialize $\pi^{1,k}_{h,i}$ to be uniform policy for all player $i$. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.
 7:         Each player $i$ initializes a no-regret learning instance (Protocol 1) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for
    which we will use NO_REGRET_UPDATE$_{h,i,s}(\cdot)$ to denote the update.
 8:         **for** $t = 1, 2, \ldots, T$ **do**
 9:             **for** $i \in [m]$ **do**
10:                 Set Dataset $\mathcal{D}^{k,t}_{h,i} = \emptyset$
11:                 **for** $l = 1, 2, \ldots, k-1$ **do**
12:                     Draw a joint trajectory $(s^l_1, \mathbf{a}^l_1, r^l_{1,i}, \ldots, s^l_h, \mathbf{a}^l_h, r^l_{h,i}, s^l_{h+1})$ from $\pi^l_{1:h-1} \circ \left(\pi^l_{h,i}, \pi^{k,t}_{h,-i}\right)$, where $\pi^l$
    is the policy learned at episode $l$ stored in policy cover $\Pi$.
13:                     Add $(s^l_h, a^l_{h,i}, r^l_{h,i}, s^l_{h+1})$ to $\mathcal{D}^{k,t}_{h,i}$.
14:                 **end for**
15:                 Set $\Sigma^{k,t}_{h,i} = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \phi_i(s,a)\phi_i(s,a)^\top$.
16:                 Set $\overline{\theta}^{k,t}_{h,i} = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \overline{V}^k_{h+1,i}(s')\right)^2$.
17:                 Set $\overline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \overline{\theta}^{k,t}_{h,i} \right\rangle + \beta \left\|\phi_i(\cdot,\cdot)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.
18:                 Update $\overline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\overline{V}_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\overline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.
19:                 Update the no-regret learning instance at step $h$ and state $s$: $\pi^{k,t+1}_{h,i}(\cdot \mid s) \leftarrow$
    NO_REGRET_UPDATE$_{h,i,s}(1 - \overline{Q}^{k,t}_{h,i}(s,\cdot)/H)$ for all $s \in \mathcal{S}$.
20:             **end for**
21:         **end for**
22:         Set $\overline{V}^k_{h,i}(s) \leftarrow \mathrm{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot \text{(Swap)Reg}(T)\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.
23:     **end for**
24:     Set $\pi^k$ to be the Markov joint policy such that $\pi^k_h(\mathbf{a}|s) = \frac{1}{T}\sum_{t=1}^T \prod_{i \in [m]} \pi^{k,t}_{h,i}(a_i|s)$.
25:     Update $\Pi \leftarrow \Pi \bigcup \{\pi^k\}$.                    ▷ Policy cover update
26: **end for**
27: Sample $k \sim \mathrm{Unif}(K)$ and output $\pi^{\text{output}} = \pi^k$.
```

We will set the parameters for Algorithm 2 to be

- $\lambda = \frac{2\log(16 d_{\max} mKHT/\delta)}{\log(36/35)}$
- $W = H\sqrt{d_{\max}}$
- $\beta = 16(W+H)\sqrt{\lambda + d_{\max}\log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)}$
- $T = \widetilde{O}(H^4\log(A_{\max})\epsilon^{-2})$ for Markov CCE and $T = \widetilde{O}(H^4 A_{\max}\log(A_{\max})\epsilon^{-2})$ for Markov CE
- $K = \widetilde{O}(m^2 H^4 d^2_{\max}\epsilon^{-2})$.

E.1. Concentration

The population covariance matrix for episode k, inner loop t, step h and player i is defined as

$$\Sigma^k_{h,i} := \mathbb{E}\left[\widehat{\Sigma}^{k,t}_{h,i}\right] = \lambda I + \sum_{l=1}^{k-1} \Sigma^{\pi^l}_{h,i},$$

where $\Sigma^{\pi^k}_{h,i} = \mathbb{E}_{\pi^k}\left[\phi_i(s_h, a_{h,i})\phi_i(s_h, a_{h,i})^\top\right]$. Note that $s^l_h, a^l_{h,i}$ is sampled following the same policy for each inner loop t, so the expected covariance is the same for different t.

We define $\pi^{k,\mathrm{cov}}$ to be the mixture policy in $\Pi^k = \{\pi^l\}_{l=1}^{k-1}$, where policy π^l is given weight/probability $\frac{1}{k-1}$, and also define

$$\widetilde{\theta}^{k,t}_{h,i} := \operatorname*{argmin}_{\|\theta\|\le W} \mathbb{E}_{(s_h,a_{h,i})\sim\pi^{k,\mathrm{cov}}} \left\{\langle\phi_i(s_h,a_{h,i}),\theta\rangle - \mathbb{E}_{a_{h,-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s_h,\mathbf{a}_h) + \overline{V}^k_{h+1,i}(s')\right]\right\}^2,$$

$$\widehat{\theta}^{k,t}_{h,i} := \operatorname*{argmin}_{\|\theta\|\le W} \mathbb{E}_{(s_h,a_{h,i})\sim\pi^{k,\mathrm{cov}}} \left\{\langle\phi_i(s_h,a_{h,i}),\theta\rangle - \mathbb{E}_{a_{h,-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s_h,\mathbf{a}_h) + \underline{V}^k_{h+1,i}(s')\right]\right\}^2.$$

Lemma E.1. *(Concentration) With probability at least* $1-\delta/2$, *for all* $k\in[K]$, $h\in[H]$, $t\in[T]$, $i\in[m]$, *we have*

$$\left\|\overline{\theta}^{k,t}_{h,i} - \widetilde{\theta}^{k,t}_{h,i}\right\|_{\Sigma^k_{h,i}} \le 8(W+H)\sqrt{\lambda + d_i\log(32WK(W+H)) + 4\log(8mKHT/\delta)} \le \beta/2, \tag{5}$$

$$\left\|\underline{\theta}^{k,t}_{h,i} - \widehat{\theta}^{k,t}_{h,i}\right\|_{\Sigma^k_{h,i}} \le 8(W+H)\sqrt{\lambda + d_i\log(32WK(W+H)) + 4\log(8mKHT/\delta)} \le \beta/2, \tag{6}$$

$$\frac{1}{2}\Sigma^{k,t}_{h,i} \preceq \Sigma^k_{h,i} \preceq \frac{3}{2}\Sigma^{k,t}_{h,i}. \tag{7}$$

Proof. The proof is the same as the proof for Lemma D.3. □

With a slight abuse of the notation, we will still denote the high probability event in Lemma E.1 as $\mathcal{G}$ Now we define

$$\overline{\Delta}^{k,t}_{h,i}(s,a_i) = \mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s,a_i),\widetilde{\theta}^{k,t}_{h,i}\right\rangle\right),$$

$$\underline{\Delta}^{k,t}_{h,i}(s,a_i) = \mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \underline{V}^k_{h+1,i}(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s,a_i),\widehat{\theta}^{k,t}_{h,i}\right\rangle\right).$$

Lemma E.2. *Under good event* $\mathcal{G}$, *for all* $k\in[K]$, $t\in[T]$, $h\in[H]$, $i\in[m]$, $s\in\mathcal{S}$ *and* $a_i\in\mathcal{A}_i$ *we have*

$$\begin{aligned}
-\overline{\Delta}^{k,t}_{h,i}(s,a_i) &\le \overline{Q}^{k,t}_{h,i}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s')\right]\right] \\
&\le 3\beta\,\|\phi_i(s,a_i)\|_{[\Sigma^k_{h,i}]^{-1}} - \overline{\Delta}^{k,t}_{h,i}(s,a_i),
\end{aligned}$$

$$\begin{aligned}
-3\beta\,\|\phi_i(s,a_i)\|_{[\Sigma^k_{h,i}]^{-1}} - \underline{\Delta}^{k,t}_{h,i}(s,a_i) &\le \underline{Q}^{k,t}_{h,i}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \underline{V}^k_{h+1,i}(s')\right]\right] \\
&\le -\underline{\Delta}^{k,t}_{h,i}(s,a_i).
\end{aligned}$$

Proof. The proof is the same as the proof for Lemma D.6. □

E.2. Proofs for Learning Markov CCE with Algorithm 2

Lemma E.3. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1) - \nu \le \overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \le 3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H}{T}\cdot \mathrm{Reg}(T) + \nu.$$

Proof. The first inequality is from Lemma D.7. Now we prove the second argument:

$$\begin{aligned}
&\overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1)\\
\le&\frac{1}{T}\sum_{t=1}^T\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) + \frac{H}{T}\cdot\mathrm{Reg}(T) - V_{1,i}^{\pi^k}(s_1)\\
\le&\frac{1}{T}\sum_{t=1}^T\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\Big(\Big[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\Big[r_{h,i}(s_1,\mathbf{a}_1) + \overline{V}_{2,i}^k(s_2)\Big]\Big]\\
&+3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\Big) + \frac{H}{T}\cdot\mathrm{Reg}(T) - V_{1,i}^{\pi^k}(s_1) && \text{(Lemma E.2)}\\
\le&\frac{1}{T}\sum_{t=1}^T\Big(\Big[\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t}(\cdot|s_1)}\Big[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2)\Big]\Big]\\
&+\mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)}\Big[3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\Big]\Big) + \frac{H}{T}\cdot\mathrm{Reg}(T)\\
\le&\mathbb{E}_{\pi_1^k}\Big[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2)\Big] + \mathbb{E}_{a_{1,i}\sim\pi_{1,i}^k(\cdot|s_1)}\left[3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \frac{1}{T}\sum_{t=1}^T\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]\\
&+\frac{H}{T}\cdot\mathrm{Reg}(T)\\
\le&3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^H\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}} - \mathbb{E}_{\pi^k}\sum_{h=1}^H\frac{1}{T}\sum_{t=1}^T\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i}) + \frac{H^2}{T}\cdot\mathrm{Reg}(T)\\
\le&3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^H\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T}\cdot\mathrm{Reg}(T) + \nu. && \text{(Lemma D.5)}
\end{aligned}$$

□

Lemma E.4. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^K\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma_{h,i}^k]^{-1}} \le d_i\log(1+\frac{K}{d_i\lambda}).$$

Proof. As $\|\phi_i(s_h,a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}} \le 1$, by Lemma I.6 we have

$$\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma_{h,i}^k]^{-2}} \le \log\frac{\det(\Sigma_{h,i}^{k+1})}{\det(\Sigma_{h,i}^k)}.$$

Thus we have

$$\begin{aligned}
\sum_{k=1}^K\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma_{h,i}^k]^{-1}} \le&\sum_{k=1}^K\log\frac{\det(\Sigma_{h,i}^{k+1})}{\det(\Sigma_{h,i}^k)}\\
=&\log\frac{\det(\Sigma_{h,i}^{K+1})}{\det(\Sigma_{h,i}^1)}
\end{aligned}$$

$$\leq d_i \log(1+\frac{K}{d_i\lambda}),$$

where we utilized the fact that

$$\log\det(\Sigma_{h,i}^{K+1}) \leq d_i \log\left(\frac{\mathrm{trace}(\Sigma_{h,i}^{K+1})}{d_i}\right) \leq d_i \log\left(\frac{d_i\lambda + K}{d_i}\right).$$

□

Lemma E.5. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} \max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) \leq 3mH\beta\sqrt{Kd_{\max}\log\left(1+\frac{K}{\lambda}\right)} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K.$$

Proof.

$$\begin{aligned}
&\sum_{k=1}^{K} \max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right)\\
\leq& 3\beta\sum_{k=1}^{K}\max_{i\in[m]}\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T}\sum_{k=1}^{K}\mathrm{Reg}(T) + \nu K && \text{(Lemma E.3)}\\
=& 3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K}\mathbb{E}_{\pi^k}\sqrt{\|\phi_i(s_h,a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K\\
\leq& 3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K}\sqrt{\mathbb{E}_{\pi^k}\|\phi_i(s_h,a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K && \text{(Concavity of } f(x)=\sqrt{x}\text{)}\\
\leq& 3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sqrt{K\sum_{k=1}^{K}\mathbb{E}_{\pi^k}\|\phi_i(s_h,a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K && \text{(Cauchy–Schwarz inequality)}\\
\leq& 3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sqrt{Kd_i\log\left(1+\frac{K}{d_i\lambda}\right)} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K && \text{(Lemma D.10)}\\
\leq& 3mH\beta\sqrt{Kd_{\max}\log\left(1+\frac{K}{\lambda}\right)} + \frac{H^2K}{T}\cdot\mathrm{Reg}(T) + \nu K.
\end{aligned}$$

□

Theorem E.6. *Suppose Algorithm 2 is instantiated with no-regret learning oracles satisfying Assumption 4.1. Then for ν-misspecified linear Markov games, with probability 0.9, Algorithm 2 will output an $(\epsilon + 2\nu)$-approximate Markov CCE. The sample complexity is $O(mHTK^2) = \widetilde{O}(m^5H^{13}d_{\max}^6\log(A_{\max})\epsilon^{-6})$, where $d_{\max} = \max_{i\in[m]} d_i$ and $A_{\max} = \max_{i\in[m]} A_i$.*

Proof. By Lemma E.5, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma D.3), we have

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) \leq& \frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) + \nu && \text{(Lemma D.7)}\\
\leq& 3mH\beta\sqrt{d_{\max}\log\left(1+\frac{K}{\lambda}\right)/K} + \frac{H^2}{T}\cdot\mathrm{Reg}(T) + 2\nu. && \text{(Lemma E.5)}
\end{aligned}$$

By Markov's inequality, we set $K = \widetilde{O}(m^2 H^4 d_{\max}^3 \epsilon^{-2})$ and $T = \widetilde{O}(H^4 \log(A_{\max})\epsilon^{-2})$, with probability 0.9 we have

$$\max_{i\in[m]} \left(V_{1,i}^{\dagger,\pi_{-i}^{\text{output}}}(s_1) - V_{1,i}^{\pi^{\text{output}}}(s_1) \right) \le \epsilon + 2\nu.$$

□

E.3. Proofs for Learning Markov CE with Algorithm 2

Lemma E.7. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$\begin{aligned} \max_{\psi_i} V_{1,i}^{\psi_i \diamond \pi^k}(s_1) - V_{1,i}^{\pi^k}(s_1) - \nu &\le \overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \\ &\le 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H}{T} \cdot \text{SwapReg}(T) + \nu. \end{aligned}$$

Proof. The first inequality is from Lemma D.14. Now we prove the second argument.

$$\begin{aligned}
&\overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \\
\le& \frac{1}{T}\sum_{t=1}^T \sum_{a_{1,i}\in\mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s)\overline{Q}_{1,i}^{k,t}(s, a_{1,i}) + \frac{H}{T}\cdot \text{SwapReg}(T) - V_{1,i}^{\pi^k}(s_1) \\
\le& \frac{1}{T}\sum_{t=1}^T \sum_{a_{1,i}\in\mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1) \Big(\Big[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)} \Big[r_{1,i}(s_1, \mathbf{a}_1) + \overline{V}_{2,i}^k(s_2) \Big] \Big] \\
&+ 3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \Big) + \frac{H}{T}\cdot\text{SwapReg}(T) - V_{1,i}^{\pi^k}(s_1) \qquad \text{(Lemma E.2)} \\
\le& \frac{1}{T}\sum_{t=1}^T \Big(\Big[\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t}(\cdot|s_1)} \Big[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2) \Big] \Big] + 3\beta \mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)} \|\phi_i(s, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} \\
&- \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \Big) + \frac{H}{T}\cdot\text{SwapReg}(T) \\
\le& \mathbb{E}_{\pi_1^k} \Big[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2) \Big] + \mathbb{E}_{a_{1,i}\sim\pi_{1,i}^k(\cdot|s_1)} \Big[3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} - \frac{1}{T}\sum_{t=1}^T \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \Big] \\
&+ \frac{H}{T}\cdot\text{SwapReg}(T) \\
\le& 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} - \mathbb{E}_{\pi^k} \sum_{h=1}^H \frac{1}{T}\sum_{t=1}^T \overline{\Delta}_{h,i}^{k,t}(s_h, a_{h,i}) + \frac{H^2}{T}\cdot\text{SwapReg}(T) \\
\le& 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T}\cdot\text{SwapReg}(T) + \nu, \qquad \text{(Lemma D.5)}
\end{aligned}$$

which completes the proof. □

Lemma E.8. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^K \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \right) \le 3mH\beta \sqrt{K d_{\max} \log\left(1 + \frac{K}{\lambda}\right)} + \frac{H^2 K}{T}\cdot \text{SwapReg}(T) + \nu.$$

Proof. The proof is the same as the proof for Lemma E.5 where we replace Lemma E.3 with Lemma E.7 in the proof. □

Theorem E.9. *Suppose Algorithm 2 is instantiated with no-regret learning oracles satisfying Assumption 4.2. Then for ν-misspecified linear Markov games, with probability* 0.9*, Algorithm 2 will output an $(\epsilon + 2\nu)$-approximate Markov CCE. The sample complexity is $O(mHTK^2) = \widetilde{O}(m^5 H^{13} d_{\max}^6 A_{\max} \log(A_{\max})\epsilon^{-6})$, where $d_{\max} = \max_{i\in[m]} d_i$ and $A_{\max} = \max_{i\in[m]} A_i$.*

Proof. By Lemma E.8, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma E.1), we have

$$
\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\max_{\psi_i} V_{1,i}^{\psi_i\diamond\pi^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right)\\
\leq&\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) + \nu && \text{(Lemma D.14)}\\
\leq&3mH\beta\sqrt{d_{\max}\log\left(1+\frac{K}{\lambda}\right)/K} + \frac{mH^2}{T}\cdot\mathrm{SwapReg}(T) + 2\nu. && \text{(Lemma E.8)}
\end{aligned}
$$

By Markov's inequality, we set $K=\widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T=\widetilde{O}(H^4A_{\max}\log(A_{\max})\epsilon^{-2})$, with probability 0.9, we have

$$
\max_{i\in[m]}\max_{\psi_i}\left(V_{1,i}^{\psi_i\diamond\pi^{\mathrm{output}}}(s_1) - V_{1,i}^{\pi^{\mathrm{output}}}(s_1)\right) \leq \epsilon,
$$

which completes the proof. □

F. Algorithms for Learning Optimal Policies in Misspecified Linear MDP

In this section, we adapt Algorithm 1 to the linear MDP setting. As the single-agent degeneration of independent linear Markov games, we can remove the no-regret learning loop in Algorithm 1 and achieve better sample complexity. The analysis is almost the same as the analysis for Algorithm 1 in Appendix D with $T=1$ and $m=1$.

We will set the parameters for Algorithm 3 to be

- $\lambda=\frac{2\log(16dNH/\delta)}{\log(36/35)}$
- $W=H\sqrt{d}$
- $\beta=16(W+H)\sqrt{\lambda+d\log(32W(W+H))+4\log(8K_{\max}H/\delta)}$
- $T_{\mathrm{Trig}}=64\log(8HN^2/\delta)$
- $K_{\max}=\min\{\frac{2Hd\log(N+\lambda)}{\log(1+T_{\mathrm{Trig}}/4)},N\}$
- $N=\widetilde{O}(H^4d^2\epsilon^{-2})$.

We will use K to denote the episode that Algorithm 3 ends ($n^{\mathrm{tot}}=N$ or $K=K_{\max}$). Immediately we have $K\leq K_{\max}\leq N$.

The population covariance matrix for episode k, step h is defined as

$$
\Sigma_h^k := \mathbb{E}\left[\widehat{\Sigma}_h^k\right] = \lambda I + \sum_{l=1}^{k-1} n^l \Sigma_h^{\pi^l},
$$

where $\Sigma_h^{\pi^k} = \mathbb{E}_{\pi^k}\left[\phi(s_h,a_h)\phi(s_h,a_h)^\top\right]$.

We define $\pi^{k,\mathrm{cov}}$ to be the mixture policy in $\Pi^k=\{(\pi^l,n^l)\}_{l=1}^{k-1}$, where policy π^l is given weight/probability $\frac{n^l}{\sum_{j=1}^{k-1}n^j}$. Then we define the on-policy population fit to be

$$
\widetilde{\theta}_h^k := \operatorname*{argmin}_{\|\theta\|\leq W}\mathbb{E}_{(s_h,a_h)\sim\pi^{k,\mathrm{cov}}}\left\{\langle\phi(s_h,a_h),\theta\rangle - \mathbb{E}\left[r_h(s_h,a_h)+\overline{V}_{h+1}^k(s')\right]\right\}^2,
$$

$$
\widehat{\theta}_h^k := \operatorname*{argmin}_{\|\theta\|\leq W}\mathbb{E}_{(s_h,a_h)\sim\pi^{k,\mathrm{cov}}}\left\{\langle\phi(s_h,a_h),\theta\rangle - \mathbb{E}\left[r_h(s_h,a_h)+\underline{V}_{h+1}^k(s')\right]\right\}^2.
$$

Algorithm 3 Policy Replay for Misspecified MDP with linear function approximation

```
Input: $\epsilon$, $\delta$, $\lambda$, $\beta$, $T_{\mathrm{Trig}}$, $K_{\max}$, $N$
Initialization: Policy Cover $\Pi = \emptyset$. $n^{\mathrm{tot}} = 0$.
for episode $k = 1, 2, \dots, K_{\max}$ do
    Set $\overline{V}_{H+1}^k(\cdot) = \underline{V}_{H+1}^k(\cdot) = 0$, $n^k = 0$.
    for $h = H, H-1, \dots, 1$ do                         ▷ Retrain policy with the current policy cover
        Initialize $\overline{V}_h^k(\cdot) = \underline{V}_h^k(\cdot) = 0$.
        Set Dataset $\mathcal{D}_h^k = \emptyset$.
        for $l = 1, 2, \dots, \sum_{j=1}^{k-1} n^j$ do
            Sample $\pi^l$ with probability $n^l / \sum_{j=1}^{k-1} n^j$.
            Draw a joint trajectory $(s_1^l, a_1^l, r_1^l, \dots, s_H^l, a_H^l, r_H^l, s_{H+1}^l)$ from $\pi^l$.
            Add $(s_h^l, a_{h,i}^l, r_{h,i}^l, s_{h+1}^l)$ to $\mathcal{D}_h^k$.
        end for
        Set $\widehat{\Sigma}_h^k = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}_h^k} \phi(s,a)\phi(s,a)^\top$.
        Set $\overline{\theta}_h^k = \mathrm{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}_h^k} \left( \langle \phi(s,a), \theta \rangle - r - \overline{V}_{h+1}^k(s') \right)^2$.
        Set $\underline{\theta}_h^k = \mathrm{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}_h^k} \left( \langle \phi(s,a), \theta \rangle - r - \underline{V}_{h+1}^k(s') \right)^2$.
        Set $\overline{Q}_h^k(\cdot,\cdot) = \mathrm{proj}_{[0,H+1-h]} \left( \left\langle \phi(\cdot,\cdot), \overline{\theta}_h^k \right\rangle + \beta \|\phi(\cdot,\cdot)\|_{[\widehat{\Sigma}_h^k]^{-1}} \right)$.
        Set $\underline{Q}_h^k(\cdot,\cdot) = \mathrm{proj}_{[0,H+1-h]} \left( \left\langle \phi(\cdot,\cdot), \underline{\theta}_h^k \right\rangle - \beta \|\phi(\cdot,\cdot)\|_{[\widehat{\Sigma}_h^k]^{-1}} \right)$.
        Set $\overline{V}_h^k(\cdot) = \max_{a \in \mathcal{A}} \overline{Q}_h^k(\cdot, a)$.
        Set $\overline{V}_h^k(\cdot) = \underline{Q}_h^k(\cdot, \mathrm{argmax}_{a \in \mathcal{A}} \overline{Q}_h^k(\cdot, a))$
    end for
    Set $\pi^k$ to be the policy such that $\pi_h^k(s) = \mathrm{argmax}_{a \in \mathcal{A}} \overline{Q}_h^k(s,a)$ for all $(h,s) \in [H] \times \mathcal{S}$.
    if $n^{\mathrm{tot}} = N$ then
        Set $k^{\mathrm{output}} = \mathrm{argmin}_k \left( \overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \right)$.
        Output $\pi^{\mathrm{output}} = \pi^{k^{\mathrm{output}}}$.
    end if
    Set $T_{h,i} = 0$, for all $h \in [H], i \in [m]$.
    repeat                                                ▷ Update policy cover
        Reset to $s = s_1$, $n^k = n^k + 1$, $n^{\mathrm{tot}} = n^{\mathrm{tot}} + 1$.
        for $h = 1, 2, \dots, H$ do
            Play $a = \pi_h^k(\cdot|s)$.
            $T_h \to T_h + \|\phi(s,a)\|^2_{[\widehat{\Sigma}_h^k]^{-1}}$.
            Get next state $s'$, $s \to s'$.
        end for
    until $\exists h \in [H]$ such that $T_h \ge T_{\mathrm{Trig}}$ or $n^{\mathrm{tot}} = N$.
    Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.
end for
```

We define the misspecification error to be

$$\overline{\Delta}_h^k(s,a) := \mathbb{E}\left[r_h(s,a) + \overline{V}_{h+1}^k(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi(s,a), \widetilde{\theta}_h^k \right\rangle\right),$$

$$\underline{\Delta}_h^k(s,a) := \mathbb{E}\left[r_h(s,a) + \underline{V}_{h+1}^k(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi(s,a), \widehat{\theta}_h^k \right\rangle\right).$$

Lemma F.1. *(Concentration) With probability at least $1-\delta/2$, for all $k \in [K]$, $h \in [H]$, we have*

$$\left\|\overline{\theta}_h^k - \widetilde{\theta}_h^k\right\|_{\Sigma_h^k} \le 8(W+H)\sqrt{\lambda + d\log(32WN(W+H)) + 4\log(8K_{\max}H/\delta)} \le \beta/2, \tag{8}$$

$$\left\|\underline{\theta}_h^{k,t} - \widehat{\theta}_h^k\right\|_{\Sigma_h^k} \le 8(W+H)\sqrt{\lambda + d\log(32WN(W+H)) + 4\log(8K_{\max}H/\delta)} \le \beta/2, \tag{9}$$

$$\frac{1}{2}\widehat{\Sigma}_h^k \preceq \Sigma_h^k \preceq \frac{3}{2}\widehat{\Sigma}_h^k. \tag{10}$$

Proof. The proof is the same as the proof for Lemma D.3. □

Lemma F.2. *With probability at least $1-\delta/2$, the following two events hold:*

- *Suppose at episode k, Line 34: $T_h \ge T_{\mathrm{Trig}}$ is triggered, then we have*

$$\mathbb{E}_{\pi^k}\left\|\phi(s_h,a_h)\right\|^2_{[\widehat{\Sigma}_h^k]^{-1}} \ge \frac{1}{2n^k}\sum_{j=1}^{n^k}\left\|\phi(s_h^{k,j}, a_h^{k,j})\right\|^2_{[\widehat{\Sigma}_h^k]^{-1}} \ge \frac{T_{\mathrm{Trig}}}{2n^k},$$

  where j denotes the j-th trajectory collected in the policy cover update (Line 27).

- *For any $k \in [K_{\max}]$, $h \in [H]$, we have*

$$\mathbb{E}_{\pi^k}\left\|\phi(s_h,a_h)\right\|^2_{[\widehat{\Sigma}_h^k]^{-1}} \le \frac{2T_{\mathrm{Trig}}}{n^k}.$$

Proof. The proof is the same as the proof for Lemma D.4. □

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma F.1 and Lemma F.2 hold, which holds with probability at least $1-\delta$ by Lemma F.1 and Lemma F.2.

Lemma F.3. *Under good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we have*

$$-\overline{\Delta}_h^k(s,a) \le \overline{Q}_h^k(s,a) - \left[\mathbb{E}\left[r_h(s,a) + \overline{V}_{h+1}^k(s')\right]\right] \le 3\beta\left\|\phi(s,a)\right\|_{[\Sigma_h^k]^{-1}} - \overline{\Delta}_h^k(s,a),$$

$$-3\beta\left\|\phi(s,a)\right\|_{[\Sigma_h^k]^{-1}} - \underline{\Delta}_h^k(s,a) \le \underline{Q}_h^k(s,a) - \left[\mathbb{E}\left[r_h(s,a) + \underline{V}_{h+1}^k(s')\right]\right] \le -\underline{\Delta}_h^k(s,a).$$

Proof. The proof is the same as the proof for Lemma D.6. □

Lemma F.4. *(Optimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$\overline{V}_1^k(s_1) \ge V_1^*(s_1) - \sum_{h=1}^H \mathbb{E}_{\pi^*}\left[\overline{\Delta}_h^k(s_h,a_h)\right] \ge V_1^*(s_1) - \nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, we have

$$\begin{aligned}&\overline{V}_1^k(s_1) - V_1^*(s_1)\\ =&\max_{a_1\in\mathcal{A}}\overline{Q}_1^k(s_1,a_1) - V_1^*(s_1)\end{aligned}$$

$$\begin{aligned}
&\geq \overline{Q}_1^k(s_1, \pi_1^*(s_1)) - Q_1^*(s_1, \pi_1^*(s_1)) \\
&\geq \mathbb{E}\left[r_1(s_1, \pi_1^*(s_1)) + \overline{V}_2^k(s_2)\right] - \overline{\Delta}_1^k(s_1, \pi_1^*(s_1)) - Q_1^*(s_1, \pi_1^*(s_1)) && \text{(Lemma F.3)} \\
&= \mathbb{E}_{\pi^*}\left[\overline{V}_2^k(s_2) - V_2^*(s_2)\right] - \overline{\Delta}_1^k(s_1, \pi_1^*(s_1)) \\
&\geq - \mathbb{E}_{\pi^*}\left[\sum_{h=1}^{H} \overline{\Delta}_h^k(s_h, a_h)\right] \\
&\geq - \nu. && \text{(Lemma D.5)}
\end{aligned}$$

□

Lemma F.5. *(Pessimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$\underline{V}_1^k(s_1) \leq V_1^{\pi^k}(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\pi^k}\left[\underline{\Delta}_h^k(s_h, a_h)\right] \leq V_1^{\pi^k}(s_1) + \nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, we have

$$\begin{aligned}
& \underline{V}_1^k(s_1) - V_1^{\pi^k}(s_1) \\
&= \underline{Q}_1^k(s_1, \pi_1^k(s_1)) - V_1^{\pi^k}(s_1) \\
&\leq \mathbb{E}_{a_1 = \pi_1^k(s_1)}\left[r_1(s_1, a_1) + \underline{V}_2^k(s_2) - \underline{\Delta}_1^k(s_1, a_1)\right] - V_1^{\pi^k}(s_1) && \text{(Lemma F.3)} \\
&= \mathbb{E}_{a_1 = \pi_1^k(s_1)}\left[\underline{V}_2^k(s_2) - V_2^{\pi^k}(s_2) - \underline{\Delta}_1^k(s_1, a_1)\right] \\
&\leq - \mathbb{E}_{\pi^k}\left[\sum_{h=1}^{H} \underline{\Delta}_h^k(s_h, a_h)\right] \\
&\leq \nu. && \text{(Lemma D.5)}
\end{aligned}$$

□

Lemma F.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$V_1^*(s_1) - V_1^{\pi^k}(s_1) - 2\nu \leq \overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \leq 6\beta \mathbb{E}_{\pi^k} \sum_{h=1}^{H} \|\phi(s_h, a_h)\|_{[\Sigma_h^k]^{-1}} + 2\nu.$$

Proof. The proof is the same as the proof for Lemma D.9. □

Lemma F.7. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \|\phi(s_h, a_h)\|_{[\Sigma_h^k]^{-1}}^2 \leq 4T_{\mathrm{Trig}} d \log\left(1 + \frac{N}{d\lambda}\right).$$

Proof. The proof is the same as the proof for Lemma D.10. □

Lemma F.8. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) \leq 6H\beta \sqrt{4N(T_{\mathrm{Trig}} + 1) d \log\left(1 + \frac{N}{\lambda}\right)} + 2\nu N.$$

Proof. The proof is the same as the proof for Lemma D.11. □

Lemma F.9. *Under the good event $\mathcal{G}$, we have*

$$K \leq \frac{2Hd\log(N+\lambda)}{\log(1+T_{\text{Trig}}/4)},$$

which means $K < K_{\max}$ and Algorithm 3 ends due to $n^{\text{tot}} = N_{\max}$.

Proof. The proof is the same as the proof for Lemma D.12. □

Theorem F.10. *For ν-misspecified linear MDP, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4\nu)$-approximate optimal policy. The sample complexity is $O(HK_{\max}N) = \widetilde{O}(H^6 d^4 \epsilon^{-2})$.*

Proof. Under the good event $\mathcal{G}$, by Lemma F.9, the algorithm ends by $n^{\text{tot}} = N$. By Lemma F.8, we have

$$\begin{aligned}\min_{k\in[K]}\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) \leq& \frac{1}{N}\sum_{k=1}^{K} n^k\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right)\\ \leq& 6H\beta\sqrt{4(T_{\text{Trig}}+1)d\log\left(1+\frac{N}{\lambda}\right)/N} + 2\nu.\end{aligned}$$

By setting $N = \widetilde{O}(H^4 d^3 \epsilon^{-2})$, we have

$$\min_{k\in[K]} \overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \leq \epsilon + 2\nu.$$

Then by Lemma F.6, we have

$$\begin{aligned}V_1^*(s_1) - V_1^{\pi^{\text{output}}}(s_1) \leq& \overline{V}_1^{k^{\text{output}}}(s_1) - \underline{V}_1^{k^{\text{output}}}(s_1) + 2\nu = \min_{k\in[K]}\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) + 2\nu\\ \leq& \epsilon + 4\nu.\end{aligned}$$

□

G. Proofs for Learning in Markov Potential Games

G.1. Proofs for Learning Markov NE with Algorithm 4

We will set the parameter for Algorithm 4 to be

- $K = 5mH\epsilon^{-1}$

Lemma G.1. *With probability at least $1-\delta/2$, for all $k \in [K]$ and $i \in [m]$, $\widehat{\pi}_i^{k+1}$ is an $(\epsilon/8 + O(\nu))$-approximate optimal policy in the ν-misspecified linear MDP induced by all the players except player i following policy π_{-i}^k.*

Proof. The argument follows from the property of LINEARMDP_SOLVER (Assumption 5.2) and a union bound. □

Lemma G.2. *Suppose for all $k \in [K]$ and $i \in [m]$, we execute policy π^k and $(\widehat{\pi}_i^{k+1}, \pi_{-i}^k)$ for $\widetilde{O}(H^2\epsilon^{-2})$ episodes, With probability at least $1-\delta/2$, for all $k \in [K]$ and $i \in [m]$, we have*

$$\left|\widehat{V}_{1,i}^{\pi^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right| \leq \frac{\epsilon}{8}, \qquad \left|\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^t}(s_1) - V_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1)\right| \leq \frac{\epsilon}{8}.$$

Proof. The argument follows directly by Hoeffding's inequality and a union bound. □

We will denote the event in Lemma G.1 and Lemma G.2 to be the good event $\mathcal{G}$.

Algorithm 4 Nash Coordinate Ascent for Independent Linear Markov Potential Games (Lin-Nash-CA)

```
Input: $\epsilon$, $\delta$, $K = 5mH\epsilon^{-1}$
Initialization: $\pi^1$ to be an arbitrary deterministic policy.
for episode $k = 1, 2, \dots, K$ do
    Execute policy $\pi^k$ for $\widetilde{O}(H^2\epsilon^{-2})$ episodes and obtain $\widehat{V}_{1,i}^{\pi^k}(s_1)$ as the empirical average of the total reward for all player $i \in [m]$.
    for $i \in [m]$ do
        Fix all the players except player $i$ to follow policy $\pi_{-i}^k$ and player $i$ runs LINEARMDP_SOLVER with feature $\phi_i(\cdot,\cdot)$, accuracy $\epsilon/8$ and failure probability $\delta/(2mK)$. Set $\widehat{\pi}_i^{k+1}$ to be the output of LINEARMDP_SOLVER.
        Execute policy $(\widehat{\pi}_i^{k+1}, \pi_{-i}^k)$ for $\widetilde{O}(H^2\epsilon^{-2})$ episodes and obtain $\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1)$ as the empirical average of the total reward.
        Set $\Delta_i \leftarrow \widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) - \widehat{V}_{1,i}^{\pi^k}(s_1)$.
    end for
    if $\max_{i\in[m]} \Delta_i > \epsilon/2$ then
        Set $\pi^{k+1} : \pi_i^{k+1} = \pi_i^k, \pi_j^{k+1} = \widehat{\pi}_j^k$ for $i \neq j$ and $j = \mathrm{argmax}_{i\in[m]} \Delta_i$.
    else
        Output $\pi^{\mathrm{output}} = \pi^k$.
    end if
end for
```

Lemma G.3. *Under the good event $\mathcal{G}$, for any $k \in [K]$, if $\max_{i\in[m]} \Delta_i^k > \epsilon/2$ and $j = \mathrm{argmax}_{i\in[m]} \Delta_i^k$, we have*

$$V_{1,j}^{\pi^{k+1}}(s_1) - V_{1,j}^{\pi^k}(s_1) \geq \epsilon/4.$$

And if $\max_{i\in[m]} \Delta_i^k \leq \epsilon/2$, we have

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

Proof. Under the good event $\mathcal{G}$, if $\max_{i\in[m]} \Delta_i^k > \epsilon/2$ and $j = \mathrm{argmax}_{i\in[m]} \Delta_i^k$, we have

$$\begin{aligned} V_{1,j}^{\pi^{k+1}}(s_1) - V_{1,j}^{\pi^k}(s_1) \geq & \widehat{V}_{1,j}^{\widehat{\pi}_j^{k+1},\pi_{-i}^k}(s_1) - \epsilon/8 - \widehat{V}_{1,j}^{\pi^k}(s_1) - \epsilon/8 && \text{(Lemma G.2)} \\ \geq & \epsilon/4. \end{aligned}$$

On the other hand, if $\max_{i\in[m]} \Delta_i^k = \max_{i\in[m]} \left(\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) - \widehat{V}_{1,i}^{\pi^k}(s_1) \right) \leq \epsilon/2$, for all $i \in [m]$ we have

$$\begin{aligned} V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1) \leq & V_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^t}(s_1) + \frac{\epsilon}{8} + O(\nu) - V_{1,i}^{\pi^k}(s_1) \\ \leq & \widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) + \frac{\epsilon}{8} + O(\nu) + \epsilon/8 - \widehat{V}_{1,i}^{\pi^k}(s_1) + \epsilon/8 && \text{(Lemma G.1 and Lemma G.2)} \\ \leq & \epsilon + O(\nu), \end{aligned}$$

completing the proof. □

Theorem 5.3. *For ν-misspecified independent linear Markov potential games with $\Pi^{\mathrm{estimate}} = \{\pi^k\}_{k=1}^K$, with probability at least $1-\delta$, Algorithm 4 will output an $(\epsilon + O(\nu))$-approximate pure Markov NE. The sample complexity is $O(m^2H\epsilon^{-1} \cdot \mathrm{LinearMDP\_SC}(\epsilon/8, \delta/(10m^2H\epsilon^{-1}), d_{\max}))$.*

Proof. Suppose Algorithm 4 does not output a policy, then it ends due to $k = K$. Then under the good event $\mathcal{G}$, by the first argument of Lemma G.3, for all $k \in [K]$, and $j^k = \mathrm{argmax}_{i\in[m]} \Delta_i^k$, we have

$$\Phi(\pi^{k+1}) - \Phi(\pi^k) = V_{1,j^t}^{\pi^{k+1}}(s_1) - V_{1,j^k}^{\pi^k}(s_1) \geq \epsilon/4.$$

Protocol 2 Adversarial Bandit Algorithm

```
Initialize: Action set $\mathcal{B}$, and $p_1$ to be the uniform distribution over $\mathcal{B}$.
for $t = 1, 2, \ldots, T$ do
    Adversary chooses loss $l_t$.
    Player take action $b_t \sim p_t$ and observe noisy bandit-feedback $\widetilde{l}_t(b_t)$.
    Update $p_{t+1} \leftarrow \textsc{Adv\_Bandit\_Update}(b_t, \widetilde{l}_t(b_t))$.
end for
```

As we set $K = 5mH/\epsilon$, we have $\Phi(\pi^{K+1}) > mH \geq \Phi_{\max}$, which is a contradiction. So Algorithm 4 will output a policy π^{output}. As the LINEARMDP_SOLVER always outputs a deterministic policy, π^{output} is a deterministic policy. Then by the second argument of Lemma G.3, when Algorithm 4 terminates, it will output an ϵ-approximate pure NE π^{output}. □

H. Missing Parts in Section 6

H.1. Adversarial Multi-armed Bandit Oracle

ADV_BANDIT_UPDATE subroutine. Consider the adversarial multi-armed bandit problem with B arms. At round t, the adversary chooses some loss l_t and the learner chooses some action $b_t \sim p_t$, where $p_t \in \Delta(\mathcal{B})$ is the policy at round t. Then the learner observes a noisy bandit-feedback $\widetilde{l}_t(b_t) \in [0, 1]$ such that $\mathbb{E}[\widetilde{l}_t(b_t) \mid l_t, b_t] = l_t(b_t)$. The player will update the policy to p_{t+1} for round $t + 1$, which is denoted as $p_{t+1} \leftarrow \textsc{Adv\_Bandit\_Update}(b_t, \widetilde{l}_t(b_t))$.

H.2. Proofs for Algorithm 5

We will set the parameters for Algorithm 5 to be

- $T_{\text{Trig}} = 12 \log(8K_{\max} HS/\delta)$
- $K_{\max} = 9HS \log(N_{\max})$
- $N_{\max} = \widetilde{O}(H^4 S A_{\max} \epsilon^{-2})$ for Markov CCE and $N_{\max} = \widetilde{O}(H^4 S A_{\max}^2 \epsilon^{-2})$ for Markov CE
- $\beta_n = \sqrt{\frac{8H^2 T_{\text{Trig}} \log(2mK_{\max} HS/\delta)}{n \vee T_{\text{Trig}}}}$.

We will use subscript k, t to denote the variables in episode k and inner loop t, and subscript h, i to denote the variables at step h and for player i. We will use K to denote the episode that the Algorithm 5 ends (Line 30 is triggered or $n^{\text{tot}} = N_{\max}$ or $K = K_{\max}$) and N to denote n^{tot} when Algorithm 5 ends. Immediately we have $K \leq K_{\max} \leq N_{\max}$.

By the definition of the adversarial multi-armed bandit oracles (Assumption 6.1 and Assumption 6.2), we have the following two lemmas.

Lemma H.1. *For all $k \in [K]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\begin{aligned}
&\frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \\
&\geq \max_{a_i \in \mathcal{A}_i} \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a}_{-i} \sim \pi_{h,-i}^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) - \frac{n_h^k(s)}{H} \cdot \text{BReg}(n_h^k(s)).
\end{aligned}$$

Lemma H.2. *For all $k \in [K]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\begin{aligned}
&\frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \\
&\geq \max_{\psi_{h,i}} \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \psi_{h,i} \diamond \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) - \frac{n_h^k(s)}{H} \cdot \text{BSwapReg}(n_h^k(s)).
\end{aligned}$$

Algorithm 5 Policy **Re**ply with **B**andit **O**racle in Tabular Markov Games (**PReBO**)

```
Input: $\epsilon$, $\delta$, $\beta$, $T_{\mathrm{Trig}}$, $K_{\max}$, $N_{\max}$
Initialization: Policy Cover $\Pi = \emptyset$. $n^{\mathrm{tot}} = 0$.
for episode $k = 1, 2, \ldots, K_{\max}$ do
    Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$, $n^k = 0$, $n^k_h(s) = 0$ for all $h \in [H]$ and $s \in \mathcal{S}$.
    for $h = H, H-1, \ldots, 1$ do                              ▷ Retrain policy with the current policy cover
        Initialize $\pi^{k,1}_{h,i}$ to be uniform policy for all player $i$. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.
        Each player $i$ initializes an adversarial bandit instance (Protocol 2) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for which we will use $\textsc{No\_Regret\_Update}_{h,i,s}(\cdot)$ to denote the update.
        for $t = 1, 2, \ldots, \sum_{j=1}^{k-1} n^j$ do
            Sample $\pi^l \in \Pi$ with probability $n^l / \sum_{j=1}^{k-1} n^j$.
            Draw a joint trajectory $(s_1, \mathbf{a}_1, \mathbf{r}_1, \ldots, s_h, \mathbf{a}_h, \mathbf{r}_h, s_{h+1})$ from $\pi^l_{1:h-1} \circ \pi^{k,t}_h$,which is the policy that follows $\pi^l$ for the first $h-1$ steps and follows $\pi^{k,t}_h$ for step $h$.
            Update $n^k_h(s_h) \leftarrow n^k_h(s_h) + 1$.
            Update the adversarial bandit instance for player $i$ at step $h$ and state $s_h$: $\pi^{k,t+1}_{h,i}(\cdot|s_h) \leftarrow \textsc{Adv\_Bandit\_Update}_{h,i,s_h}(a_{h,i}, 1 - \left(r_{h,i} + \overline{V}^k_{h+1,i}(s_{h+1})\right)/H)$.
            Update policy $\pi^{k,t+1}_{h,i}(\cdot|s) \leftarrow \pi^{k,t+1}_{h,i}(\cdot|s)$ for $s \neq s_h$.
            Update $\overline{V}^k_{h,i}(s_h) \leftarrow \frac{n^k_h(s_h)-1}{n^k_h(s_h)}\overline{V}^k_{h,i}(s_h) + \frac{1}{n^k_h(s_h)}(r_{h,i} + \overline{V}^k_{h+1,i}(s_{h+1}))$.
            Update $\underline{V}^k_{h,i}(s_h) \leftarrow \frac{n^k_h(s_h)-1}{n^k_h(s_h)}\underline{V}^k_{h,i}(s_h) + \frac{1}{n^k_h(s_h)}(r_{h,i} + \underline{V}^k_{h+1,i}(s_{h+1}))$.
        end for
        Set $\overline{V}^k_{h,i}(s) \leftarrow \mathrm{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot \mathrm{B(Swap)Reg}(n^k_h(s_h)) + \beta_{n^k_h(s)}\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.
        Set $\underline{V}^k_{h,i}(s) \leftarrow \mathrm{proj}_{[0,H+1-h]}\left(\underline{V}^k_{h,i}(s) - \beta_{n^k_h(s)}\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.
    end for
    Set $\pi^k$ to be the Markov joint policy such that $\pi^k_h(\mathbf{a}|s) = \frac{1}{n^k_h(s)} \sum_{j=1}^{n^k_h(s)} \prod_{i\in[m]} \pi^{k,t^k_h(j;s)}_{h,i}(a_i|s)$, where $t^k_h(j;s)$ is the time $t$ such that state $s$ is visited for the $j$-th time in episode $k$ at step $h$.
    if $\max_{i\in[m]} \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le \epsilon$ then                ▷ Policy certification
        Output: $\pi^{\mathrm{output}} = \pi^t$.
    end if
    Set $T^k_h(s) = 0$ for all $h \in [H]$, $s \in \mathcal{S}$.
    repeat                                                      ▷ Update policy cover
        Reset $s = s_1$, $n^k = n^k + 1$, $n^{\mathrm{tot}} = n^{\mathrm{tot}} + 1$.
        for $i \in [m]$ do
            for $h = 1, 2, \ldots, H$ do
                Play $\mathbf{a}_h = \pi^k_h(\cdot|s)$.
                $T^k_h(s_h) \leftarrow T^k_h(s_h) + 1$.
                Get next state $s'$, $s \to s'$.
            end for
        end for
    until $\exists h \in [H]$ such that $T^k_h(s_h) = n^k_h(s_h) \vee T_{\mathrm{Trig}}$ or $n^{\mathrm{tot}} = N_{\max}$.
    Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.
end for
```

H.2.1. Concentration

Lemma H.3. *With probability at least $1-\delta/2$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\left| \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} (r_{h,i}^{k,t_h^k(j;s)} + \overline{V}_{h+1,i}^k(s_{h+1}^{k,t_h^k(j;s)})) - \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k,t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \right|$$
$$\leq \beta_{n_h^k(s)},$$

$$\left| \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} (r_{h,i}^{k,t_h^k(j;s)} + \underline{V}_{h+1,i}^k(s_{h+1}^{k,t_h^k(j;s)})) - \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k,t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s,\mathbf{a}) + \underline{V}_{h+1,i}^k(s')) \right|$$
$$\leq \beta_{n_h^k(s)},$$

where

$$\beta_{n_h^k(s)} = \sqrt{\frac{8H^2 T_{\mathrm{Trig}} \log(2mK_{\max}HS/\delta)}{n_h^k(s) \vee T_{\mathrm{Trig}}}}.$$

Proof. If $n_h^k(s) \leq T_{\mathrm{Trig}}$, we have $\beta_{n_h^k(s)} \geq H$ and the arguments hold directly. If $n_h^k(s) \geq T_{\mathrm{Trig}}$, we have

$$\beta_{n_h^k(s)} = \sqrt{\frac{8H^2 T_{\mathrm{Trig}} \log(2mK_{\max}HS/\delta)}{n_h^k(s) \vee T_{\mathrm{Trig}}}} \geq \sqrt{\frac{8H^2 \log(2mK_{\max}HS/\delta)}{n_h^k(s)}},$$

and by Hoeffding's inequality and union bound, we can prove that the arguments hold with probability at least $1-\delta/2$. □

Lemma H.4. *With probability at least $1-\delta/2$, for all $k \in [K_{\max}]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$n_h^k(s) \vee T_{\mathrm{Trig}} \geq \frac{1}{2} \left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s) \right) \vee T_{\mathrm{Trig}}, n^k d_h^{\pi^k}(s) \leq 2 \left(n_h^k(s) \vee T_{\mathrm{Trig}} \right).$$

In addition, if $T_h^k(s) = n_h^k(s) \vee T_{\mathrm{Trig}}$ is triggered, we have

$$n^k d_h^{\pi^k}(s) \geq \frac{1}{2} \left(n_h^k(s) \vee T_{\mathrm{Trig}} \right).$$

Proof. $n_h^k(s)$ is the sum of $\sum_{l=1}^{k-1} n^l$ independent Bernoulli random variables such that there are n^l random variables with mean $d_h^{\pi^l}(s)$ for $l \in [k-1]$. By Lemma I.3 and union bound, with probability at least $1-\delta/4$, for all $k \in [K_{\max}]$, $h \in [H]$, $s \in \mathcal{S}$, we have

$$n_h^k(s) \vee T_{\mathrm{Trig}} \geq \frac{1}{2} \left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s) \right) \vee T_{\mathrm{Trig}},$$

where $T_{\mathrm{Trig}} \geq 12 \log(8K_{\max}HS/\delta)$.

$T_h^k(s)$ is the sum of n^k i.i.d. Bernoulli random variables with mean n_h^k. For the second argument, by Lemma I.2 and union bound, with probability at least $1-\delta/4$, for all $k \in [K_{\max}]$, $h \in [H]$, $s \in \mathcal{S}$, we have

$$n^k d_h^{\pi^k}(s) \leq 2(n_h^k(s) \vee T_{\mathrm{Trig}}),$$

and if $T_h^k(s) = n_h^k(s) \vee T_{\mathrm{Trig}}$ is triggered, we have

$$n^k d_h^{\pi^k}(s) \geq \frac{1}{2} T_h^k(s) = \frac{1}{2} \left(n_h^k(s) \vee T_{\mathrm{Trig}} \right).$$

□

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma H.3 and Lemma H.4 hold, which holds with probability at least $1-\delta$.

H.2.2. Proofs for Learning Markov CCE with Algorithm 5

Lemma H.5. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\overline{V}_{h,i}^k(s) \geq V_{h,i}^{\dagger,\pi_{-i}^k}(s).$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have

$$\begin{aligned}
&\overline{V}_{h,i}^k(s)\\
=&\operatorname{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\left(r_{h,i}^{k,t_h^k(j;s)}+\overline{V}_{h+1,i}^k\left(s_{h+1}^{k,t_h^k(j;s)}\right)\right)+\frac{H}{n_h^k(s)}\cdot\mathrm{BReg}(n_h^k(s))+\beta_{n_h^k(s)}\right)\\
\geq&\operatorname{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\overline{V}_{h+1,i}^k(s'))+\frac{H}{n_h^k(s)}\cdot\mathrm{BReg}(n_h^k(s))\right) && \text{(Lemma H.3)}\\
\geq&\operatorname{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi_{h,-i}^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\overline{V}_{h+1,i}^k(s'))\right) && \text{(Lemma H.1)}\\
\geq&\operatorname{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi_{h,-i}^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+V_{h+1,i}^{\dagger,\pi_{-i}^k}(s'))\right) && \text{(Induction basis)}\\
=&\operatorname{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi_{h,-i}^k(\cdot|s)}(r_{h,i}(s,\mathbf{a})+V_{h+1,i}^{\dagger,\pi_{-i}^k}(s'))\right)\\
\geq&V_{h,i}^{\dagger,\pi_{-i}^k}(s).
\end{aligned}$$

□

Lemma H.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\underline{V}_{h,i}^k(s) \leq V_{h,i}^{\pi^k}(s).$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have

$$\begin{aligned}
\underline{V}_{h,i}^k(s)=&\operatorname{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\left(r_{h,i}^{k,t_h^k(j;s)}+\underline{V}_{h+1,i}^k\left(s_{h+1}^{k,t_h^k(j;s)}\right)\right)-\beta_{n_h^k(s)}\right)\\
\leq&\operatorname{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\underline{V}_{h+1,i}^k(s'))\right) && \text{(Lemma H.3)}\\
\leq&\operatorname{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+V_{h+1,i}^{\pi^k}(s'))\right) && \text{(Induction basis)}\\
=&\operatorname{proj}_{[0,H+1-h]}\left(\mathbb{E}_{\mathbf{a}\sim\pi_{h,-i}^k(\cdot|s)}(r_{h,i}(s,\mathbf{a})+V_{h+1,i}^{\dagger,\pi_{-i}^k}(s'))\right)\\
\leq&V_{h,i}^{\pi^k}(s).
\end{aligned}$$

□

Lemma H.7. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^k(s_1)\leq\widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^H\sqrt{\frac{H^2A_iT_{\mathrm{Trig}}}{n_h^k(s_h)\vee T_{\mathrm{Trig}}}}\right]\right).$$

Proof. We bound $\overline{V}^k_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1)$ and $V^{\pi^k}_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)$ separately.

$$\begin{aligned}
&\overline{V}^k_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1) \\
&= \mathrm{proj}_{[0,H+1-h]} \left(\frac{1}{n^k_1(s_1)} \sum_{j=1}^{n^k_1(s_1)} (r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s^{k,t^k_h(j;s_1)}_2)) + \frac{H}{n^k_1(s_1)} \cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)} \right) \\
&\quad - V^{\pi^k}_{1,i}(s_1) \\
&\le \frac{1}{n^k_1(s_1)} \sum_{t=1}^{n^k_1(s)} (r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s^{k,t^k_h(j;s_1)}_2)) + \frac{H}{n^k_1(s_1)} \cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)} - V^{\pi^k}_{1,i}(s_1) \\
&\le \frac{1}{n^k_1(s_1)} \sum_{t=1}^{n^k_1(s_1)} \mathbb{E}_{\mathbf{a}_1 \sim \pi_1^{k,t^k_h(j;s_1)}(\cdot|s)} (r_{1,i}(s_1, \mathbf{a}_1) + \overline{V}^k_{2,i}(s_2)) + \frac{H}{n^k_1(s_1)} \cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} \\
&\quad - V^{\pi^k}_{1,i}(s_1) \qquad \text{(Lemma H.3)} \\
&= \mathbb{E}_{\mathbf{a}_1 \sim \pi^k_1(\cdot|s)} (r_{1,i}(s_1, \mathbf{a}_1) + \overline{V}^k_{2,i}(s_2)) + \frac{H}{n^k_1(s_1)} \cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} - V^{\pi^k}_{1,i}(s_1) \\
&= \mathbb{E}_{\pi^k_1} \left[\overline{V}^k_{2,i}(s_2) - V^{\pi^k}_{2,i}(s_2) \right] + \frac{H}{n^k_1(s_1)} \cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} \\
&= \mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} \frac{H}{n^k_h(s_h)} \cdot \mathrm{BReg}(n^k_h(s_h)) + 2\beta_{n^k_h(s_h)} \right],
\end{aligned}$$

where the first inequality is from

$$\frac{1}{n^k_1(s_1)} \sum_{t=1}^{n^k_1(s)} (r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s^{k,t^k_h(j;s_1)}_2)) + \frac{H}{T} \cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)} \ge 0.$$

In addition, we have

$$\begin{aligned}
&V^{\pi^k}_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \\
&= V^{\pi^k}_{1,i}(s_1) - \mathrm{proj}_{[0,H+1-h]} \left(\frac{1}{n^k_1(s)} \sum_{j=1}^{n^k_1(s_1)} (r^{k,t^k_1(j;s)}_{1,i} + \underline{V}^k_{2,i}(s^{k,t^k_1(j;s)}_2)) - \beta_{n^k_1(s_1)} \right) \\
&\le V^{\pi^k}_{1,i}(s_1) - \frac{1}{n^k_1(s_1)} \sum_{j=1}^{n^k_1(s_1)} (r^{k,t^k_1(j;s)}_{1,i} + \underline{V}^k_{2,i}(s^{k,t^k_1(j;s)}_2)) + \beta_{n^k_1(s_1)} \\
&\le V^{\pi^k}_{1,i}(s_1) - \frac{1}{n^k_1(s_1)} \sum_{j=1}^{n^k_1(s_1)} \left(\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^{k,t^k_h(j;s_1)(\cdot|s_1)}} (r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}^k_{2,i}(s_2)) \right) + 2\beta_{n^k_1(s_1)} \qquad \text{(Lemma H.3)} \\
&= V^{\pi^k}_{1,i}(s_1) - \mathbb{E}_{\mathbf{a}_1 \sim \pi^k_1(\cdot|s_1)} (r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}^k_{2,i}(s_2)) + 2\beta_{n^k_1(s_1)} \\
&= \mathbb{E}_{\mathbf{a}_1 \sim \pi^k_1} (V^{\pi^k}_{2,i}(s_2) - \underline{V}^k_{2,i}(s_2)) + 2\beta_{n^k_1(s_1)} \\
&\le \mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} 2\beta_{n^k_h(s_h)} \right],
\end{aligned}$$

where the first inequality is from

$$\frac{1}{n^k_1(s)} \sum_{j=1}^{n^k_1(s_1)} (r^{k,t^k_1(j;s)}_{1,i} + \underline{V}^k_{2,i}(s^{k,t^k_1(j;s)}_2)) - \beta_{n^k_1(s_1)} \le H + 1 - h.$$

Then we have

$$\begin{aligned}\overline{V}^{\pi^k}_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le & \mathbb{E}_{\pi^k}\left[\sum_{h=1}^H \frac{H}{n^k_h(s_h)} \cdot \mathrm{BReg}(n^k_h(s_h)) + 4\beta_{n^k_h(s_h)}\right] \\ \le & \widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^H \sqrt{\frac{H^2 A_i}{n^k_h(s_h) \vee 1}} + \sqrt{\frac{H^2 T_{\mathrm{Trig}}}{n^k_h(s_h) \vee T_{\mathrm{Trig}}}}\right]\right) \\ \le & \widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^H \sqrt{\frac{H^2 A_i T_{\mathrm{Trig}}}{n^k_h(s_h) \vee T_{\mathrm{Trig}}}}\right]\right).\end{aligned}$$

□

Lemma H.8. *Under the good event $\mathcal{G}$, for all $i \in [m]$, we have*

$$\sum_{k=1}^K n^k \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^{\pi^k}_{1,i}(s_1)\right) \le \widetilde{O}\left(H^2\sqrt{SA_{\max}T_{\mathrm{Trig}}N}\right).$$

Proof. Under the good event $\mathcal{G}$, for all $i \in [m]$, we have

$$\begin{aligned}&\sum_{k=1}^K n^k \mathbb{E}_{\pi^k}\sqrt{\frac{1}{n^k_h(s_h) \vee T_{\mathrm{Trig}}}} \\ =&\sum_{k=1}^K n^k \sum_{s\in\mathcal{S}} d^{\pi^k}_h(s)\sqrt{\frac{1}{n^k_h(s) \vee T_{\mathrm{Trig}}}} \\ \le&\sum_{s\in\mathcal{S}}\sum_{k=1}^K n^k d^{\pi^k}_h(s)\sqrt{\frac{2}{(\sum_{l=1}^{k-1} n^l d^{\pi^l}_h(s)) \vee T_{\mathrm{Trig}}}} && \text{(Lemma H.4)} \\ \le&\sum_{s\in\mathcal{S}}\sqrt{32\sum_{k=1}^K n^k d^{\pi^k}_h(s)} && \text{(Lemma H.4 and Lemma I.7)} \\ \le&\sqrt{32S\sum_{k=1}^K n^k}. && (\textstyle\sum_{s\in\mathcal{S}}\sum_{k=1}^K n^k d^{\pi^k}_h(s) = S\sum_{k=1}^K n^k)\end{aligned}$$

Plugging it into Lemma H.7, we can prove the lemma. □

Lemma H.9. *Under the good event $\mathcal{G}$, we have*

$$K \le 9HS\log(N_{\max}),$$

which means $K < K_{\max}$ and Algorithm 5 ends due to either Line 21 ($\max_{i\in[m]} \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le \epsilon$) or Line 34 ($n^{\mathrm{tot}} = N_{\max}$).

Proof. By Lemma H.4, for any $h \in [H]$ and $s \in \mathcal{S}$, whenever $T^k_h(s) = n^k_h(s) \vee T_{\mathrm{Trig}}$ is triggered, we have

$$n^k d^{\pi^k}_h(s) \ge \frac{1}{2}(n^k_h(s) \vee T_{\mathrm{Trig}}) \ge \frac{1}{4}\left(\sum_{l=1}^{k-1} n^l d^{\pi^l}_h(s)\right).$$

Thus for any $h \in [H]$ and $s \in \mathcal{S}$, whenever $T^k_h(s) = n^k_h(s) \vee T_{\mathrm{Trig}}$ is triggered, we have

$$\sum_{l=1}^{k} n^l d^{\pi^l}_h(s) \ge \frac{5}{4}\left(\sum_{l=1}^{k-1} n^l d^{\pi^l}_h(s)\right).$$

In addition, for any $h \in [H]$ and $s \in \mathcal{S}$, for the first time $T_h^k(s) = n_h^k(s) \vee T_{\mathrm{Trig}}$ is triggered, we have

$$\sum_{l=1}^{k} n^l d_h^{\pi^l}(s) \geq n^k d_h^{\pi^k}(s) \geq \frac{1}{2}(n_h^k(s) \vee T_{\mathrm{Trig}}) \geq T_{\mathrm{Trig}}.$$

As $\sum_{l=1}^{k} n^l d_h^{\pi^l}(s)$ is non-decreasing and upper bounded by $N_{\max}$, the number of triggering for any $h \in [H]$ and $s \in \mathcal{S}$ is bounded by $\log(N_{\max}/T_{\mathrm{Trig}})/\log(5/4) \leq 8\log(N_{\max})$, and the total number of triggering is bounded by $8HS\log(N_{\max}) + 1$, where 1 is from the last triggering $n^{\mathrm{tot}} = N_{\max}$. □

Theorem 6.3. *Suppose Algorithm 5 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 6.1. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 5 will output an ϵ-approximate Markov CCE with sample complexity $\widetilde{O}(H^6 S^2 A_{\max}\epsilon^{-2})$.*

Proof. Suppose under the good event $\mathcal{G}$, the algorithm does not end with Line 21 ($\max_{i\in[m]} \overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \leq \epsilon$). Then by Lemma H.9, the algorithm ends by $N = N_{\max}$. By Lemma H.8, under the good event $\mathcal{G}$, we have

$$\begin{aligned}\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \leq& \frac{1}{N}\sum_{k=1}^{K} n^k \max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right)\\ \leq& \widetilde{O}\left(H^2\sqrt{SA_{\max}T_{\mathrm{Trig}}/N_{\max}}\right).\end{aligned}$$

Let $N_{\max} = \widetilde{O}(H^4 S A_{\max}\epsilon^{-2})$ we can have

$$\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \leq \epsilon,$$

which contradicts with Line 21. Thus Algorithm 5 will end at episode k such that

$$\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \leq \epsilon.$$

By Lemma H.5 and Lemma H.6, we have

$$\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) \leq \max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \leq \epsilon,$$

completing the proof. □

H.2.3. Proofs for Learning Markov CE with Algorithm 5

Lemma H.10. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\overline{V}_{h,i}^k(s) \geq \max_{\psi_i} V_{h,i}^{\psi_i \diamond \pi^k}(s).$$

Proof. We prove the lemma by mathematical induction on h. The argument holds for $h = H+1$ as both sides are 0. Suppose the argument holds for $h+1$. By the update rule of $\overline{V}_{h,i}^k(s)$, we have

$$\begin{aligned}&\overline{V}_{h,i}^k(s)\\ =&\mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\left(r_{h,i}^{k,t_h^k(j;s)} + \overline{V}_{h+1,i}^k(s_{h+1}^{k,t_h^k(j;s)})\right) + \frac{H}{n_h^k(s)}\mathrm{BSwapReg}(n_h^k(s)) + \beta_{n_h^k(s)}\right)\\ \geq&\mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}\left(r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')\right) + \frac{H}{n_h^k(s)}\mathrm{BSwapReg}(n_h^k(s))\right) \quad \text{(Lemma H.3)}\end{aligned}$$

$$
\begin{aligned}
&\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}} \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^{k,t_h^k(j;s)}(\cdot|s)}\left(r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s')\right)\right) && \text{(Lemma H.2)}\\
&= \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}} \mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^{k}(\cdot|s)}\left(r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s')\right)\right)\\
&\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}} \mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^{k}(\cdot|s)}\left(r_{h,i}(s,\mathbf{a}) + \max_{\psi_i} V^{\psi_i\diamond\pi^k}_{h+1,i}(s')\right)\right) && \text{(Induction basis)}\\
&\geq \max_{\psi_i} V^{\psi_i\diamond\pi^k}_{h,i}(s).
\end{aligned}
$$

□

Lemma H.11. *Under the good event $\mathcal{G}$, for all $k\in[K]$, $i\in[m]$, we have*

$$
\overline{V}^k_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1) \leq \widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^{H}\sqrt{\frac{H^2A_i^2T_{\mathrm{Trig}}}{n_h^k(s_h)\vee T_{\mathrm{Trig}}}}\right]\right).
$$

Proof. The proof is the same as the proof of Lemma H.7 and we replace BReg with $\mathrm{BSwapReg}$. □

Lemma H.12. *Under the good event $\mathcal{G}$, for all $k\in[K]$, $h\in[H]$, $i\in[m]$, $s\in\mathcal{S}$, we have*

$$
\sum_{k=1}^{K} n^k \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^{\pi^k}_{1,i}(s_1)\right) \leq \widetilde{O}\left(H^2\sqrt{SA_{\max}^2T_{\mathrm{Trig}}N}\right).
$$

Proof. The proof is the same as the proof of Lemma H.8 and we replace Lemma H.7 with Lemma H.11 in the proof. □

Theorem 6.4. *Suppose Algorithm 5 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 6.2. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 5 will output an ϵ-approximate Markov CE with sample complexity is $\widetilde{O}(H^6S^2A_{\max}^2\epsilon^{-2})$.*

Proof. Suppose under the good event $\mathcal{G}$, the algorithm does not end with Line 21 ($\max_{i\in[m]}\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\leq\epsilon$). Then by Lemma H.9, the algorithm ends by $N=N_{\max}$. By Lemma H.12, under the good event $\mathcal{G}$, we have

$$
\begin{aligned}
\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \leq& \frac{1}{N}\sum_{k=1}^{K} n^k \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right)\\
\leq& \widetilde{O}\left(H^2\sqrt{SA_{\max}^2T_{\mathrm{Trig}}/N_{\max}}\right).
\end{aligned}
$$

Let $N_{\max}=\widetilde{O}(H^4SA_{\max}^2\epsilon^{-2})$ we can have

$$
\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \leq \epsilon,
$$

which contradicts with Line 21. Thus Algorithm 5 will end at episode k such that

$$
\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \leq \epsilon.
$$

By Lemma H.10 and Lemma H.6, we have

$$
\max_{i\in[m]}\left(\max_{\psi_i} V^{\psi_i\diamond\pi^k}_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1)\right) \leq \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \leq \epsilon.
$$

□

I. Technical Tools

Lemma I.1. *(Theorem 4 in (Maurer and Pontil, 2009)) For $n \geq 2$, let $X_1, \cdots, X_n$ be i.i.d. random variables with values in $[0, 1]$ and let $\delta > 0$. Define $\widehat{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\widehat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \widehat{X})$. Then we have*

$$\mathbb{P}\left[\left|\widehat{X} - \mathbb{E}[X]\right| > \sqrt{\frac{2\widehat{\sigma}\log(4/\delta)}{n}} + \frac{7\log(4/\delta)}{3(n-1)}\right] \leq \delta.$$

Lemma I.2. *Consider i.i.d. random variables $X_1, X_2, \ldots$ with support in $[0, 1]$ and $\widehat{S}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Suppose $\overline{n} = \min_n\{n : \sum_{i=1}^{n} X_i \geq T_{\mathrm{Trig}}\}$ with $T_{\mathrm{Trig}} \geq 64\log(4n_{\max}/\delta)$. Then if $\overline{n} \leq n_{\max}$, with probability at least $1 - \delta$, we have*

$$\frac{1}{2}\widehat{S}_{\overline{n}} \leq \mathbb{E}[X] \leq \frac{3}{2}\widehat{S}_{\overline{n}},$$

and in addition, for $n \leq \min\{\overline{n}, n_{\max}\}$, we have

$$\mathbb{E}[X] \leq \frac{2T_{\mathrm{Trig}}}{n}.$$

Proof. Define the empirical variance to be

$$\widehat{\sigma}_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \widehat{S}_n)^2.$$

By Lemma I.1, we have that for any fixed $n \geq 2$,

$$\mathbb{P}\left[\left|\widehat{S}_n - \mathbb{E}[X]\right| \leq \sqrt{\frac{2\log(4n_{\max}/\delta)\widehat{\sigma}_n}{n}} + \frac{7\log(4n_{\max}/\delta)}{3(n-1)}\right] \geq 1 - \frac{\delta}{n_{\max}}.$$

Thus we have

$$\mathbb{P}\left[\left|\widehat{S}_n - \mathbb{E}[X]\right| \leq \sqrt{\frac{2\log(4n_{\max}/\delta)\widehat{\sigma}_n}{n}} + \frac{7\log(4n_{\max}/\delta)}{3(n-1)}, \forall 2 \leq n \leq n_{\max}\right] \geq 1 - \delta. \tag{11}$$

The empirical variance can be bounded by

$$\widehat{\sigma}_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \widehat{S}_n)^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^2 - n\widehat{X}^2\right) \leq \frac{1}{n-1}\sum_{i=1}^{n} X_i \leq 2\widehat{S}_n.$$

Thus for $T_{\mathrm{Trig}} \geq 64\log(4n_{\max}/\delta)$, we have $\overline{n}\widehat{S}_{\overline{n}} \geq T_{\mathrm{Trig}} \geq 64\log(4n_{\max}/\delta)$ and

$$\sqrt{\frac{2\log(4\overline{n}/\delta)\widehat{\sigma}_{\overline{n}}}{\overline{n}}} + \frac{7\log(4\overline{n}/\delta)}{3(\overline{n}-1)} \leq \sqrt{\frac{4\log(4\overline{n}/\delta)\widehat{S}_{\overline{n}}}{\overline{n}}} + \frac{7\log(4\overline{n}/\delta)}{3(\overline{n}-1)} \leq \frac{\widehat{S}_{\overline{n}}}{2}.$$

Plugging it into (11), we can prove the first argument.

For $n \leq \min\{\overline{n}, N\}$, we have $\sum_{i=1}^{n} X_i \leq T_{\mathrm{Trig}} + 1 \leq 2T_{\mathrm{Trig}}$, which means

$$\widehat{\sigma}_n \leq 2\widehat{S}_n \leq \frac{4T_{\mathrm{Trig}}}{n}.$$

Plugging it into (11), and with $T_{\mathrm{Trig}} \geq 64\log(4n_{\max}/\delta)$, we can prove the second argument. □

Lemma I.3. *Suppose $X_1, X_2, \cdots, X_n$ are i.i.d. Bernoulli random variables with $\mathbb{E}[X] = p$ and $N = \sum_{i=1}^{n} X_i$. For any $a \geq 12\log(2/\delta)$ with probability at least $1 - \delta$, we have*

$$\frac{1}{2}\left(N \vee a\right) \leq np \vee a \leq 2\left(N \vee a\right).$$

Proof. By the multiplicative Chernoff bound, we have

$$\mathbb{P}\left[|N-np| \geq \frac{1}{2}np\right] \leq 2\exp\left(-\frac{np}{12}\right).$$

Thus if $np \geq 12\log(2/\delta)$, we have

$$\mathbb{P}\left[\frac{1}{2}np \leq N \leq 2np\right] \leq \delta.$$

If $np < 12\log(2/\delta)$, by Bernstein inequality, with probability $1-\delta$ we have

$$\mathbb{P}\left[N-np > t\right] \leq \exp\left(-\frac{t^2/2}{np+t/3}\right).$$

Let $t = a \geq np$ and we have

$$\mathbb{P}\left[N > 2a\right] \leq \exp\left(-\frac{a^2/2}{np+a/3}\right) \leq \exp(-3a/8) \leq \delta.$$

Note that if $N \leq 2a$, we directly have

$$\frac{1}{2}\left(N \vee a\right) \leq np \vee a \leq 2\left(N \vee a\right).$$

□

Lemma I.4. *(Lemma 20.1 in (Lattimore and Szepesvári, 2020)) The Euclidean sphere $S^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$. There exists a set $\mathcal{C}_\epsilon \subset \mathbb{R}^d$ with $|\mathcal{C}_\epsilon| \leq (3/\epsilon)^d$ such that for all $x \in S^{d-1}$ there exists $y \in C_\epsilon$ with $\|x-y\|_2 \leq \epsilon$.*

Lemma I.5. *Let $\Sigma \succeq \lambda I$ be a positive definite matrix and M be a positive semidefinite matrix with eigenvalue upper-bounded by 1. Let $\Sigma' = \Sigma + M$. Then we have*

$$\log\det(\Sigma') \geq \log\det(\Sigma) + \mathrm{Tr}(\Sigma^{-1}M).$$

Proof.

$$\begin{aligned}\det(\Sigma') &= \det(\Sigma + M)\\ &= \det(\Sigma)\det(I + \Sigma^{-1/2}M\Sigma^{-1/2}).\end{aligned}$$

Denote $\lambda_1, \ldots, \lambda_d$ as the eigenvalues of $\Sigma^{-1/2}M\Sigma^{-1/2}$. Then we have

$$x^\top \Sigma^{-1/2}M\Sigma^{-1/2}x \leq \left\|\Sigma^{-1/2}x\right\|_2^2 = x^\top\Sigma^{-1}x \leq \lambda^{-1},$$

which means $\lambda_i \in [0, \lambda^{-1}]$ for all $i \in [d]$. Thus, we have

$$\begin{aligned}&\log\det(\Sigma') = \log\det(\Sigma) + \sum_{i=1}^d \log(1+\lambda_i) \geq \log\det(\Sigma) + \sum_{i=1}^d \frac{\lambda}{\lambda+1}\lambda_i\\ &= \log\det(\Sigma) + \frac{\lambda}{\lambda+1}\mathrm{Tr}(\Sigma^{-1}M),\end{aligned}$$

completing the proof. □

Lemma I.6. *(Lemma 11 in (Zanette and Wainwright, 2022)) For any random vector $\phi \in \mathbb{R}^d$, scalar $\alpha > 0$ and positive definite matrix Σ, we have*

$$\frac{\alpha}{L}\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2 \leq \log\frac{\det(\Sigma + \alpha\mathbb{E}[\phi\phi^\top])}{\det(\Sigma)} \leq \alpha\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2,$$

whenever $\alpha\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2 \leq L$ for some $L \geq e-1$.

Lemma I.7. *Let $b > 0$ and $a_1, a_2, \cdots, a_n > 0$ such that $a_{n+1} \le c \cdot \left(\sum_{l=1}^{n-1} a_l \vee b\right)$ for all $n \ge 1$ and some constant c. Then we have*

$$\sum_{i=1}^{\infty} a_i \sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \le 2\sqrt{(c+1)\sum_{l=1}^{n} a_l}.$$

Proof. Note that for any $i \ge 1$ we have

$$\sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \le \sqrt{\frac{c+1}{(\sum_{l=1}^{i} a_l) \vee b}}.$$

Let $f(x) = \sqrt{\frac{c+1}{x \vee b}}$ for $x \ge 0$ and immediately we have $f(x)$ is non-increasing. Then we have

$$\begin{aligned}
\sum_{i=1}^{n} a_i \sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \le & \sum_{i=1}^{\infty} a_i \sqrt{\frac{c+1}{(\sum_{l=1}^{i} a_l) \vee b}} \\
= & \sum_{i=1}^{n} a_i f(\sum_{l=1}^{i} a_l) \\
\le & \int_0^{\sum_{l=1}^{n} a_l} f(x) \\
\le & 2\sqrt{(c+1)\sum_{l=1}^{n} a_l}.
\end{aligned}$$

□

Lemma I.8. *(Lemma 4 in (Zanette and Wainwright, 2022)) Let $X \in \mathbb{R}^d$ be a random vector and Y be a random variable such that $\|X\|_2 \le 1$, $|Y| \le Y_{\max}$, $(X, Y) \sim \mathbb{P}$ for some distribution $\mathbb{P}$. Let $\{(x_i, y_i)\}_{i=1}^n$ be n i.i.d. samples from $\mathbb{P}$. Then we define*

$$\beta^* := \operatorname*{argmin}_{\|\beta\|_2 \le W} \mathbb{E}_{(X,Y)\sim\mathbb{P}}(Y - \langle X, \beta \rangle)^2,$$

$$\widehat{\beta} := \operatorname*{argmin}_{\|\beta\|_2 \le W} \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle x_i, \beta \rangle)^2.$$

Then with probability at least $1 - \delta$, we have

$$\left\|\beta^* - \widehat{\beta}\right\|_{n\mathbb{E}[XX^\top]+\lambda I} \le 8(W + Y_{\max})\sqrt{d \log(32Wn(W + Y_{\max})) + \log(1/\delta) + \lambda}.$$

Lemma I.9. *(Covariance Concentration) (Proposition 1 in (Zanette and Wainwright, 2022)) Suppose $\{Z_k\}_{k=1^K}$ is a sequence of independent, symmetric and positive definite random matrices of dimension d such that*

$$0 \le \lambda_{\min}(Z_k) \le \lambda_{\max}(Z_k) \le 1, \forall k \in [K].$$

Let $\widehat{\Sigma} = \lambda I + \sum_{k=1}^{K} Z_k$ and $\Sigma = \mathbb{E}[\widehat{\Sigma}]$ for some $\lambda \ge 0$. For any $\delta \in (0, 1)$ and $\lambda > 2\frac{\log(2d/\delta)}{\log(36/35)}$, with probability at least $1 - \delta$ we have

$$\frac{1}{2}\widehat{\Sigma} \preceq \Sigma \preceq \frac{3}{2}\widehat{\Sigma}.$$