# RL in Markov Games with Independent Function Approximation: Improved Sample Complexity Bound under the Local Access Model

**Junyi Fan**[*†] **Yuxuan Han**[*†] **Jialin Zeng**[*†] **Jianfeng Cai**[†]
**Yang Wang**[†‡] **Yang Xiang**[†§] **Jiheng Zhang**[†‡]
† Department of Mathematics, HKUST
‡Department of Industrial Engineering and Decision Analytics, HKUST
§HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute
∗Equal Contribution, Correspondence to: maxiang@ust.hk, jiheng@ust.hk

## Abstract

Efficiently learning equilibria with large state and action spaces in general-sum Markov games while overcoming the curse of multi-agency is a challenging problem. Recent works have attempted to solve this problem by employing independent linear function classes to approximate the marginal $Q$-value for each agent. However, existing sample complexity bounds under such a framework have a suboptimal dependency on the desired accuracy $\varepsilon$ or the action space. In this work, we introduce a new algorithm, Lin-Confident-FTRL, for learning coarse correlated equilibria (CCE) with local access to the simulator, i.e., one can interact with the underlying environment on the visited states. Up to a logarithmic dependence on the size of the state space, Lin-Confident-FTRL learns $\epsilon$-CCE with a provable optimal accuracy bound $O(\epsilon^{-2})$ and gets rids of the linear dependency on the action space, while scaling polynomially with relevant problem parameters (such as the number of agents and time horizon). Moreover, our analysis of Linear-Confident-FTRL generalizes the virtual policy iteration technique in the single-agent local planning literature, which yields a new computationally efficient algorithm with a tighter sample complexity bound when assuming random access to the simulator.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. 

## 1 Introduction

As a flourishing subfield of reinforcement learning, multi-agent reinforcement learning (MARL) systems have demonstrated impressive success across a variety of modern artificial intelligence tasks, such as chess and GO games (Silver et al., 2017), Poker (Brown and Sandholm, 2019), autonomous self-driving (Shalev-Shwartz et al., 2016), and multi-robot controls(Matignon et al., 2012). MARL investigates how multiple agents interact in an unknown shared environment and learn to take actions that maximize their individual reward. Compared to single-agent RL, where an agent only needs to optimize its own behavior by interacting with the environment, the presence of complex interactions among multiple players in MARL poses some novel challenges.

MARL encounters similar challenges as single-agent reinforcement learning in dealing with large state and action spaces, which are further compounded in the multi-agent scenario. In single-agent RL, function approximation is widely employed to tackle the challenges arising from large state and action spaces that cannot be exhaustively explored (Cisneros-Velarde and Koyejo, 2023; Wen and Van Roy, 2017; Jiang et al., 2017; Du et al., 2019; Yang and Wang, 2020; Jin et al., 2020; Wang et al., 2020; Zanette et al., 2020; Jin et al., 2021a; Du et al., 2021; Yin et al., 2022; Foster et al., 2021). However, applying function approximation to MARL using a global function approximation that captures the joint $Q$-value of all agents results in the curse of multi-agency, where the sample complexity scales exponentially with the number of agents (Xie et al., 2020; Huang et al., 2021; Chen et al., 2021; Jin et al., 2022; Chen et al., 2022; Ni et al., 2022). To address this problem, decentralized, or independent linear function approximation has been proposed in Wang et al. (2023); Cui et al. (2023) for learning equilibrium in multi-agent

general-sum Markov games, where the linear function class only models the marginal $Q$-value for each agent. Specifically, Wang et al. (2023) combine new policy replay mechanisms with $V$-learning that can learn $\varepsilon$-coarse correlated equilibirum (CCE) with $O(\varepsilon^{-2})$ sample complexity. However, the sample complexity of their algorithm still depends polynomially on the size of the largest action space $\max_{i=1}^{m} A_i$, which is affected by the large action space issue and does not fully utilize the advantage of function approximation. Cui et al. (2023) also employ policy replay techniques but with on-policy samples that eliminate the dependency on the number of actions. However, this approach yields a sub-optimal sample complexity of $O(\varepsilon^{-4})$ for finding $\varepsilon$-CCE.

While both Wang et al. (2023) and Cui et al. (2023) have utilized the online access model, it is reasonable to believe that compared to the online access model, more flexible sampling protocols, such as local access or random access models can lead to an improved sample complexity. This observation raises the following open question:

*Can we design more sample-efficient algorithms for MARL with independent linear function approximation under stronger access models?*

In this paper, we make an effort to answer this question by designing an algorithm that achieves sharper dependency under the local access model and random access model. Random access model, also known as generative model, allows the player to query any state-action pair. Recently, the local access model has gained popularity in the single-agent RL with function approximation both theoretically (Weisz et al., 2022; Yin et al., 2022; Hao et al., 2022; Li et al., 2021) and empirically (Tavakoli et al., 2020; Lan et al., 2023; Yin et al., 2023). This model allows the agent to query the simulator with previously visited states, providing more versatility than the random access model and accommodating many realistic scenarios. For example, in many video games, players can revisit previously recorded states. We summarize our key contributions and technical innovations under these two models below.

### 1.1 Our Contribution

**Independent linear Markov Games under the local access model.** We propose a more efficient algorithm, Linear-Confident-FTRL, for independent linear Markov games with local access to a simulator. To leverage accumulated information and prevent unnecessary revisits, the algorithm maintains a distinct core set of state-action pairs for each agent, which then determine a common confident state set. Then each agent performs policy learning over his own core set. Whenever a new state outside the confident state set is detected during the learning, the core set is expanded, and policy learning is restarted for all agents. To conduct policy learning, the algorithm employs a decentralized Follow-The-Regularized-Leader (FTRL) subroutine, which is executed by each agent over their own core sets, utilizing an adaptive sampling strategy extended from the tabular and the random model setting (Li et al., 2022). This adaptive sampling strategy effectively mitigates the curse of multi-agency, which is caused by uniform sampling over all state-action pairs.

**Sample complexity bound under the local access model.** By querying from the local access model, the Linear-Confident-FTRL algorithm is provable to learn an $\varepsilon$-CCE with $\tilde{O}(\min\{\frac{\log(S)}{d}, \max_i A_i\}d^3H^6m^2\varepsilon^{-2})$ samples for independent linear Markov Games. Here, $d$ denotes the dimension of the linear function, $S$ is the size of the state space, $m$ represents the number of agents, $H$ stands for the time horizon, and $A_i$ is the number of actions for player $i$. When $S \lesssim e^{d \max_i A_i}$, we get rid of the dependency on action space and achieve near-optimal dependency on $\varepsilon$. For possibly infinite $S$, our algorithm achieves $\tilde{O}(\varepsilon^{-2}d^3H^6m^2 \max_i A_i)$ sample complexity, which is similar to Wang et al. (2023) but sharpens the dependency on $\max_i A_i$ and $d$. We make detailed comparisons with prior works in table 1.

**Sample complexity bound under the random access model.** Our analysis of Linear-Confident-FTRL generalizes the virtual policy iteration technique in the single-agent local planning literature (Hao et al., 2022; Yin et al., 2022), in which a virtual algorithm is constructed and used as a bridge to analyze the performance of the main algorithm. In particular, our construction of the virtual algorithm also yields a new algorithm with a tighter sample complexity bound $\tilde{O}(\min\{\varepsilon^{-2}dH^2, \frac{\log(S)}{d}, \max_i A_i\}d^2H^6m^2\varepsilon^{-2})$ when the random access to the simulator is available. It is worth noting that the minimax lower bound in the tabular case is $\Omega(S \max_i A_i H^4\varepsilon^{-2})$ (Li et al., 2022). Since the independent linear approximation recovers the tabular case with $d = S \max_i A_i$, a lower bound of $\Omega(dH^4\varepsilon^{-2})$ can be derived within this framework. By comparing our sample complexity bound to this lower bound, we can demonstrate that when $S$ is not exponentially large, our proposed algorithm under the random access model achieves optimal dependency on $d$ and $\varepsilon$. On the other hand, for possibly infinity $S$, our sample complexity bound achieves the minimum over the $\tilde{O}(\varepsilon^{-4})$ result in Cui et al. (2023) and the $\tilde{O}(\varepsilon^{-2}A)$ result in Wang et al. (2023), with all other problem-relevant parameters are sharpened.

| Result | Sample Complexity | Tabular Case Complexity | Sampling Protocol |
|---|---|---|---|
| Theorem 6, Jin et al. (2021b) | $\tilde{O}(H^6 S \max_i A_i \varepsilon^{-2})$ | N.A. | Online Access |
| Theorem 3.3, Zhang et al. (2020a) | $\tilde{O}(H^3 S A_1 A_2 \varepsilon^{-2})$ | | Random Access |
| Theorem 2, Li et al. (2022) | $\tilde{O}(H^4 S \sum_{i=1}^m A_i \varepsilon^{-2})$ | | Random Access |
| Theorem 2, Xie et al. (2020) | $\tilde{O}(d^3 H^4 \varepsilon^{-2})$ | $d = SA_1A_2$ | Online Access |
| Theorem 5.2, Chen et al. (2021) | $\tilde{O}(d^2 H^3 \varepsilon^{-2})$ | $d = S^2A_1A_2$ | Online Access |
| Theorem 5, Wang et al. (2023) | $\tilde{O}(d^4 H^6 m^2 \max_i A_i^5 \varepsilon^{-2})$ | $d = S \max_i A_i$ | Online Access |
| Theorem 1, Cui et al. (2023) | $\tilde{O}(d^4 H^{10} m^4 \varepsilon^{-4})$ | | Online Access |
| Theorem 4, Dai et al. (2024)‡ | $\tilde{O}(m^4 d^5 H^6 \log S \varepsilon^{-2})$ | | Online Access |
| Theorem 1 **(This Paper)**† | $\tilde{O}(\min\{\frac{\log(S)}{d}, \max_i A_i\} d^3 H^6 m^2 \varepsilon^{-2})$ | | Local Access |
| Theorem 2 **(This Paper)** | $\tilde{O}(\min\{\varepsilon^{-2} dH^2, \frac{\log(S)}{d}, \max_i A_i\} d^2 H^5 m \varepsilon^{-2})$ | | Random Access |

Table 1: Comparison of different algorithms, where in $\tilde{O}(\cdot)$ we omit $\text{polylog}(A, H, m, d, \varepsilon)$ terms. Results in Zhang et al. (2020a); Chen et al. (2021); Xie et al. (2020) are for learning the $\varepsilon$-Nash Equilibrium(NE) in two player zero-sum Markov Games while other results are for learning $\varepsilon$-CCE for $m$-player general-sum Markov Games.
‡ See the last paragraph of section 1.2.
† When $\min\{d^{-1} \log S, \max_i A_i\} \geq \varepsilon^{-2}$, Theorem 1 also ensures a sample complexity bound independent of $\log S$ and $A$, details are presented in section 3.

### 1.2 Related Work

**Multi-Agent Markov Game.** There exist plenty of prior works in Multi-Agent Games, which offer wide exploration of different algorithms under different settings. Zhang et al. (2020b); Liu et al. (2021) provide model-based algorithms under different sampling protocols, while exponential growth on the number of agents ($\Pi_{i\in[m]} A_i$) are induced in the sample complexity. Bai et al. (2020); Song et al. (2021); Jin et al. (2021b); Mao et al. (2022) circumvent the curse of multi-agency via decentralized algorithms but return the non-Markov policies. Daskalakis et al. (2022) propose an algorithm producing Markov policies, which only depend on current state information, but at the cost of higher sample complexity. In the tabular multi-agent game, Li et al. (2022) provide the first algorithm for learning the $\varepsilon$-NE in two players zero-sum game and $\varepsilon$-CCE in multi-player general-sum game with minimax optimal sample complexity bound under the random access model.

**Function Approximation in RL.** The function approximation framework has been widely applied in single-agent RL with large state and action spaces (Zanette et al., 2020; Jin et al., 2020; Yang and Wang, 2020; Jin et al., 2021a; Wang et al., 2020; Du et al., 2021; Foster et al., 2021). The same framework has also been generalized to Markov games (Xie et al., 2020; Chen et al., 2021; Jin et al., 2022; Huang et al., 2021; Ni et al., 2022) in a centralized manner, i.e., they approximate the joint $Q$ function defined on $\mathcal{S} \times \prod_{i\in[m]} \mathcal{A}_i$, which results in the complexity of the considered function class inherently depend on $\prod_{i\in[m]} A_i$. In contrast, we consider the function approximation in a decentralized manner as in Cui et al. (2023); Wang et al. (2023) to get rid of the curse of the multi-agency.

**RL under Local Access Model.** Single-agent RL with linear function approximation under the local access model has been well investigated in previous works (Li et al., 2021; Wang et al., 2021; Weisz et al., 2021; Yin et al., 2022; Hao et al., 2022; Weisz et al., 2022). Yin et al. (2022); Hao et al. (2022) propose provably efficient algorithms for single-agent learning under the linear realizability assumption. Their algorithm design and analysis rely on the concept of the core set and the construction of virtual algorithms, which we have generalized in our paper to the multi-agent setting using a decentralized approach. The only work considering multi-agent learning under local access, to our knowledge, is Tkachuk et al. (2023). They consider the *cooperative* multi-agent learning but with *global* linear function approximation. They focus on designing sample efficient algorithms for learning the globally optimal policy with computational complexity scales in $\text{Poly}(\max_i A_i, d)$ instead of $\text{Poly}(\Pi_i A_i, d)$ under the additive decomposition assumption on the global $Q$ function. In contrast, our work addresses the learning CCE of *general-sum* Markov games with *independent* function approximation in a *decentralized* manner. It's important to highlight that while the general-sum game encompasses the cooperative game as a particular instance, the CCE policy might not always align with the global optimal policy. This distinction complicates a direct comparison between our results and those presented in Tkachuk et al. (2023). Lastly, we would like

to point out that the computational complexity of our algorithm also scales in $\text{Poly}(\max_i A_i, d)$. This is directly inferred from our algorithm design detailed in Section 3.

**Recent Refined Sample Complexity Bounds under Online Access** After the submission of our paper, a recent and independent work by Dai et al. (2024) studied the same problem in the online access setting and achieved significant improvements in the sample complexity bounds originally presented by Cui et al. (2023) and Wang et al. (2023). By utilizing tools developed for the single-agent setting in Dai et al. (2023) and refining the AVLPR scheme of Wang et al. (2023), Dai et al. (2024) demonstrated that it is possible to obtain a sample complexity bound of the order $\tilde{O}(\frac{m^4 d^5 H^6 \log S}{\varepsilon^2})$. Notably, Dai et al. (2024) achieved a similar sample complexity bound as our results under a weaker access model than ours, where the dependency on $\varepsilon$ is optimal, no polynomial dependency in $A$ is incurred, and only logarithmic dependency on $S$ is present. Refining both our results and those of Dai et al. (2024) to achieve bounds completely independent of the state space size $S$, while maintaining favorable dependencies on $A$ and $\varepsilon$, remains a challenging task and is left as a valuable direction for future research.

## 2 Preliminaries

**Notation** For a positive integer $m$, we use $[m]$ to denote $\{1, \ldots, m\}$. We write $a \lesssim b$ or $a = \tilde{O}(b)$ to denote $a \leq C\text{polylog}\big(A, m, \varepsilon^{-1}, H, \log(1/\delta)\big) \cdot b$ for some absolute constant $C$. We use $\|\cdot\|_2$ and $\|\cdot\|_\infty$ to denote the $\ell_2$ and $\ell_\infty$ norm. Given a finite set $I$, we denote $\text{Unif}(I)$ the uniform distribution over $I$.

### 2.1 Markov Games

We consider the finite horizon general-sum Markov games $(\mathcal{S}, H, \{\mathcal{A}_i\}_{i=1}^m, \{\mathbb{P}_h\}_{h=1}^H, \{r_{h,i}\}_{h,i=1}^{H,m})$. Here, $\mathcal{S}$ is the state space, $H$ denotes the time horizon, and $\mathcal{A}_i$ stands for the action space of the $i$-th player. We let $\mathcal{A} = \prod_{i=1}^m \mathcal{A}_i$ be the joint action space and $\boldsymbol{a} = (a_1, a_2, \cdots, a_m) \in \mathcal{A}$ represent the joint action. Given $s \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$, $\mathbb{P}_h(\cdot|s, \boldsymbol{a})$ denotes the transition probability and $r_{h,i}(s, \boldsymbol{a}) \in [0, 1]$ denotes the deterministic reward received by the $i$-th player at timestep $h$. We denote $S := |\mathcal{S}|, A_i := |\mathcal{A}_i|, A := \max_i A_i$ the cardinality of state and action spaces. Throughout the paper, we assume the considered Markov games always start at some fixed initial state $s_1$.[1]

[1]This assumption can be easily generalized to the setting where the initial state is sampled from some fixed distribution $\mu$, as in Cui et al. (2023); Jin et al. (2021b).

**Markov Policy.** In this work, we consider the learning of Markov policies. A Markov policy selects action depending on historical information only through the current state $s$ and time step $h$. The Markov policy of player $i$ can be represented as $\pi_i := \{\pi_{h,i}\}_{h\in[H]}$ with $\pi_{h,i} : \mathcal{S} \to \Delta(\mathcal{A}_i)$. The joint policy of all agents is denoted by $\pi = (\pi_1, \ldots, \pi_m)$. For a joint policy $\pi$, we denote $\pi_{-i}$ the joint policy excluding the one of player $i$. For $\pi_i' : \mathcal{S}\times[H] \to \Delta(\mathcal{A}_i)$, we use $\pi_i' \times \pi_{-i}$ to describe the policy where all players except player $i$ execute the joint policy $\pi_{-i}$ while player $i$ independently deploys policy $\pi_i'$.

**Value function.** For a policy $\pi$, the value function $V_{h,i}^\pi : \mathcal{S} \to \mathbb{R}$ of the $i$-th player under a Markov policy $\pi$ at step $h$ is defined as

$$V_{h,i}^\pi(s) = \mathbb{E}\big[\sum_{t=h}^H r_{t,i}(s_t, \boldsymbol{a}_t)|s_h = s\big], \quad \forall s \in \mathcal{S}, \tag{1}$$

where the expectation is taken over the state transition and the randomness of policy $\pi$. The $V_{h,i}^\pi$ satisfies the *Bellman equation*:

$$\begin{aligned} V_{h,i}^\pi(s) &= \mathbb{E}_{\boldsymbol{a}\sim\pi}[Q_{h,i}^\pi(s, \boldsymbol{a})], \\ Q_{h,i}^\pi(s, \boldsymbol{a}) &:= r_{h,i}(s, \boldsymbol{a}) + \mathbb{P}_h V_{h+1,i}^\pi(s, \boldsymbol{a}), \end{aligned} \tag{2}$$

where $\mathbb{P}_h V_{h+1,i}^\pi(s, \boldsymbol{a}) := \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,\boldsymbol{a})}[V_{h+1,i}^\pi(s')]$.

Given other players acting according to $\pi_{-i}$, the *best response policy* of the $i$-th player is the policy independent of the randomness of $\pi_{-i}$ achieving $V_{h,i}^{\dagger,\pi_{-i}}(s) := \max_{\pi_i'} V_{h,i}^{\pi_i'\times\pi_{-i}}(s)$. With the dynamic satisfied similar to (2),

$$V_{h,i}^{\dagger,\tilde{\pi}_{-i}}(s) = \max_a \big\{r_{h,i}^{\tilde{\pi}_{h,-i}}(s, a) + \mathbb{P}_h^{\tilde{\pi}_{-i}} V_{h+1,i}^{\dagger,\tilde{\pi}_{-i}}(s, a)\big\},$$

and $\mathbb{P}_h^{\tilde{\pi}_{-i}} V(s, a) := \mathbb{E}_{\boldsymbol{a}_{-i}\sim\tilde{\pi}_{-i}}[\mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,\boldsymbol{a}_{-i})}[V(s')]]$.

**Nash equilibrium(NE).** A product Markov policy $\pi = \pi_1 \times \cdots \times \pi_m$ is a Markov Nash equilibrium at state $s_1$ if $V_{1,i}^\pi(s_1) = V_{1,i}^{\dagger,\pi_{-i}}(s_1), \forall i \in [m]$.

**Coarse correlated equilibrium(CCE).** A joint Markov policy $\pi$ is a Markov CCE at a state $s_1$ if $V_{1,i}^\pi(s_1) \geq V_{1,i}^{\dagger,\pi_{-i}}(s_1), \forall i \in [m]$. In this paper, we study the efficient learning of an $\varepsilon$-Markov CCE policy $\pi$ satisfying:

$$\max_{i\in[m]}\{V_{1,i}^{\dagger,\pi_{-i}}(s_1) - V_{1,i}^\pi(s_1)\} \leq \varepsilon. \tag{3}$$

Obviously, for general-sum Markov games, a Markov NE is also a Markov CCE. Further more, in two player zero-sum games, NE and CCE are equivalent. For multi-player general-sum Markov games, computing the NE is statistically intractable. Therefore, we resort to the weaker and more relaxed equilibrium CCE,

which can be calculated in polynomial computational time for general-sum Markov games Papadimitriou and Roughgarden (2008). Still, it might be challenging for finding such an optimal relaxed equilibrium. We consider the approximated sub-optimal notation, $\varepsilon$-Markov CCE. In this work, our goal is to compute an $\varepsilon$-Markov CCE for the game with as few samples as possible.

### 2.2 RL with Different Sampling Protocols

Given Markov Games, the learner does not have access to the underlying transition probabilities $\{\mathbb{P}_h\}$ and the reward functions $\{r_{h,i}\}$, but is assumed access to a random simulator. Whenever the learner queries the simulator with $(s, \boldsymbol{a}, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, he receives an independent sample $s'$ drawn from $\mathbb{P}_h(\cdot|s, \boldsymbol{a})$. Based on the accessible range of state-action pairs using a simulator, we clarify three different sampling protocols typically used in RL as in Yin et al. (2022):

**Online Access.** The learner can only interact with the simulator (environment) in real-time, and the state can be either reset to an initial state or transit to the next state given the current state and an action.

**Local Access.** The learner can query the simulator with any previously visited state paired with an arbitrary action.

**Random Access.** The learner can query the simulator with arbitrary state-action pairs. Note that the random access model is often referred to as the generative model in the RL literature (Zhang et al., 2020a; Li et al., 2022, 2020).

The online access protocol imposes the least stringent requirement for accessing the simulator, whereas random access is the most restrictive assumption. The local access assumption, which is the central focus of this paper, is stronger than the online access protocol but more practical than the random access assumption. It has been successfully applied in the design of large-scale RL algorithms for practical problems, as demonstrated by previous studies (Yin et al., 2023; Tavakoli et al., 2020; Ecoffet et al., 2019; Lan et al., 2023). In this paper, we show that the local access assumption can lead to improved sample complexity bounds compared to the online access setting.

### 2.3 Independent Function Approximation

Throughout this paper, we make the following assumption about the Markov Games:

**Assumption 1** ($\nu$-misspecified independent linear MDP)**.** *Given a policy class $\Pi$ of interest, each player $i$ is able to access a feature map $\phi_i : \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}^d$ with $\max_{s\in\mathcal{S}, a\in\mathcal{A}_i} \|\phi_i(s, a)\|_2 \leq 1$. And there exists some $\nu > 0$ so that for any $h \in [H]$ and $V : \mathcal{S} \to [0, H + 1 - h]$,*

$$\sup_{\pi\in\Pi} \min_{\|\theta\|_2 \leq H\sqrt{d}} \left\| Q_{h,i}^{\pi_{-i},V}(\cdot, \cdot) - \phi_i(\cdot, \cdot)^\top \theta \right\|_\infty \leq \nu. \quad (4)$$

*where $Q_{h,i}^{\pi_{-i},V}(s, a) := \mathbb{E}_{\boldsymbol{a}_{-i}\sim\pi_{h,-i}(\cdot|s)}\big[r_{h,i}(s, a, \boldsymbol{a}_{-i}) + \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,\boldsymbol{a}_{-i})}[V(s')]\big]$ is the marginal Q function associated with $V$.*

Assumption 1 asserts that for any $i \in [m]$ and $\pi \in \Pi$, if all the other players act according to $\pi_{-i}$, then the $i$-th player's environment is approximately linear MDP. This assumption extends the widely used linear MDP assumption in single-agent RL to multi-agent settings.

Compared to the centralized approximation approach used in prior works (Chen et al., 2021; Xie et al., 2020; Cisneros-Velarde and Koyejo, 2023), which employs a $d \propto S\prod_{i=1}^m A_i$-dimensional linear function class to approximate a global $Q$-function for tabular Markov games, the independent approximation framework presented in Assumption 1 allows for the representation of the same environment with individual $Q$-functions of dimensions $d \propto SA$. This assumption avoids the need for the considered function class to have complexity proportional to the exponential of the number of agents.

As in Cui et al. (2023); Wang et al. (2023), we restrict (4) to a particular policy $\Pi$. As discussed in Appendix D of Wang et al. (2023), if (4) holds with $\nu = 0$ for all $\Pi$, then the MG is essentially tabular. Since our algorithm design does not require prior knowledge of $\Pi$ , we defer the discussion of the policy class $\Pi$ considered in this paper in Appendix A.

## 3 Algorithm and Guarantees for Independent Linear Markov Games

In this section, we present the Lin-Confident-FTRL algorithm for learning $\varepsilon$-CCE with local access to the simulator. We then provide the sample complexity guarantee for this algorithm.

### 3.1 The Lin-Confident-FTRL Algorithm

We now describe the Lin-Confident-FTRL algorithm (Algorithm 2).

Our algorithm design is based on the idea that each agent maintains a core set of state-action pairs. The algorithm consists of two phases: the **policy learning phase** and the **rollout checking phase**. In the policy learning phase, each agent performs decentralized policy learning based on his own core set. In the

**Algorithm 1:** Explore$(s, h)$

**Input:** state $s$, time-step $h$
**for** $i = 1$ **to** $m$ **do**
  **while** $\max_{a\in\mathcal{A}_i} \phi_i(s,a)^\top \Lambda_{h,i}^{-1}\phi_i(s,a) > \tau$ **do**
    $\hat{a}_i = \operatorname{argmax}_{a\in A_i}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(s,a)$
    $\mathcal{D}_{h,i} \leftarrow \mathcal{D}_{h,i} \cup \{(s,\hat{a}_i)\}$
    $\Lambda_{h,i} \leftarrow \Lambda_{h,i} + \phi_i(s,\hat{a}_i)\phi_i(s,\hat{a}_i)^\top$
  **end**
  $\mathcal{C}_{h,i} \leftarrow \{s \in S : \max_{a\in\mathcal{A}_i}\phi(s,a)^\top\Lambda_{h,i}^{-1}\phi(s,a) \le \tau\}$// the well-covered state set
**end**
$\mathcal{C}_h \leftarrow \cap_i \mathcal{C}_{h,i}$

rollout checking phase, the algorithm performs rollout with the learned policy to ensure the trajectory of the policy is well covered within the core set of each player. We will provide a detailed explanation of these two phases in Sections 3.2 and 3.3, respectively.

Before the policy learning phase, the $m$ players draw a joint trajectory of states $s_1, \ldots, s_H$ of length $H$ by independently sampling actions following a uniform policy. Each player then initializes distinct core sets $\{\mathcal{D}_{h,i}\}_{h=1}^H$ with this trajectory through an exploration subroutine described below (Algorithm 1).

**Core set expansion through an exploration subroutine.** During the Explore subroutine at time $h$ with input state $s$, each agent $i$ iteratively appends state action pairs $(s,a)$ to its core set $\mathcal{D}_{h,i}$ and update $\Lambda_{h,i}$ until the coverage condition $\max_{a\in\mathcal{A}_i}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(s,a) \le \tau$ is met at state $s$. Here $\tau$ is a predetermined threshold and $\Lambda_{h,i}^{-1}$ is the precision matrix corresponding to $\mathcal{D}_{h,i}$.

Given $\Lambda_{h,i}$, we define

$$\mathcal{C}_{h,i} := \{s \in \mathcal{S} : \max_{a\in\mathcal{A}_i}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(s,a) \le \tau\} \quad (5)$$

as the set of well-covered states for agent $i$ at step $h$. We refer $\mathcal{C}_h := \cap_i\mathcal{C}_{h,i}$ as the confident state set for all agents. Note that for the implementation of the algorithm, it is not necessary to compute $\mathcal{C}_h$. We introduced it merely for the sake of describing the algorithm conventionally. In fact, the only operation that involves $\mathcal{C}_h$ is to determine whether a state $s$ belongs to it, and this can be done using solely the information from $\Lambda_{h,i}$.

During the subsequent policy learning and rollout checking phases, whenever a state outside the confident set is encountered, the exploration subroutine will be triggered to expand the core set and the learning process will be restarted. Actually we have the following result regarding the cardinality of $\mathcal{D}_{h,i}$:

**Lemma 1** (Yin et al. (2022)). *For each $i$ and $h$, the size of the core set $\mathcal{D}_{h,i}$ will not exceed*

$$C_{\max} := \frac{e}{e-1}\frac{1+\tau}{\tau}d\big(\log(1+\frac{1}{\tau}) + \log(1+\frac{1}{\lambda})\big)$$

As a corollary, both the number of calls to the Explore subroutine and the number of restarts are upper-bounded by $mHC_{\max}$.

**Algorithm 2:** Lin-Confident-FTRL

**Initialize Global variables:**
$\mathcal{C}_{H+1} = \mathcal{S}, \mathcal{C}_h = \emptyset, \forall h \in [H]$ and

$$\widehat{V}_{h,i} = H+1-h, \hat{V}_{h,i}^\dagger = H+1-h, \mathcal{D}_{h,i} = \emptyset,$$
$$\Lambda_{h,i} = \lambda I, \quad \pi_{h,i}^1(\cdot|s) = \mathrm{Unif}(A_i), \quad \forall s, i, h.$$

Sample a trajectory $\{s_1, \cdots, s_H\}$ of length $H$ with policy $\pi_{h,i}^1$
**for** $h = 1$ **to** $H$ **do**
  Explore$(s_h, h)$ //See Algorithm 1
**end**
//Policy Learning Phase
**for** $h = H$ **to** $1$ **do**
  Success $\leftarrow$ Multi-Agent-Learning$(h)$ .//See Algorithm 3
  **if** *Success = False* **then**
    Back to Line 5 //Restart the loop from $h = H$.
  **end**
**end**
$\hat{\pi}_h \leftarrow \frac{1}{K}\sum_{k=1}^K \pi_{h,1}^k \times \cdots \times \pi_{h,m}^k, \forall h \in [H]$.
//Rollout Checking Phase
Success $\leftarrow$ Policy Rollout$(\hat{\pi}, s_1, N)$//See Algorithm 5
**if** *Success = False* **then**
  Return to Line 5
**end**
**for** $i \in [m]$ **do**
  **for** $h = H$ **to** $1$ **do**
    Success $\leftarrow$ Single-Agent-Learning$(h, i, \hat{\pi}_{h,-i})$.//See Algorithm 6
    **if** *Success = False* **then**
      Back to Line 5
    **end**
  **end**
**end**
**for** $i \in [m]$ **do**
  Success $\leftarrow$ Policy Rollout$(\hat{\pi}_i^\dagger \times \hat{\pi}_{-i}, s_1, N)$
  **if** *Success = False* **then**
    Return to Line 5
  **end**
**end**
**return** $\{\hat{\pi}_h\}_{h\in[H]}$

**Algorithm 3:** Multi-Agent-Learning

**Input:** time-step $h$
**for** $k = 1$ **to** $K$ **do**
  **for** $i = 1$ **to** $m$ **do**
    **for** $(\bar{s}, \bar{a}) \in \mathcal{D}_{h,i}$ **do**
      $(r, s') \leftarrow$ local sampling$(h, i, \bar{s}, \bar{a}, \pi^k_{h,-i})$
      //See Algorithm 4
      **if** $s' \notin \mathcal{C}_{h+1}$ **then**
        Explore$(s', h+1)$
        **return** *False*
      **end**
      Compute $q^k_{h,i}(\bar{s}, \bar{a}) = r + \hat{V}_{h+1,i}(s')$.
    **end**
    Update $Q^k_{h,i}(s, a)$ as in (6).
    $\bar{Q}^k_{h,i}(s, a) \leftarrow \frac{k-1}{k} \bar{Q}^{k-1}_{h,i}(s, a) + \frac{1}{k} Q^k_{h,i}(s, a)$.
    $\pi^{k+1}_{h,i}(a|s) \leftarrow \frac{\exp(\eta_k \bar{Q}^k_{h,i}(s,a))}{\sum_{a'} \exp(\eta_k \bar{Q}^k_{h,i}(s,a'))}$.
  **end**
**end**
//Value estimation of $V^{\hat{\pi}}_{h,i}$ with $\hat{\pi}_h = \frac{1}{K} \sum_{k=1}^K \pi^k_{h,1} \times \cdots \times \pi^k_{h,m}$
**for** $i = 1$ **to** $m$ **do**
  Update $\hat{V}_{h,i}(s)$ as in (7)
**end**
**return** *True*

**Algorithm 4:** Local Sampling$(h, i, s, a, \pi_{-i})$

Draw an independent sample from the simulator:

$$s' \sim \mathbb{P}_h(\cdot|s, a, \boldsymbol{a}_{-i}),$$

where $\boldsymbol{a}_{-i} \sim \pi_{h,-i}$
**return** $(r_{h,i}(s, a, \boldsymbol{a}_{-i}), s')$ // the reward & transition pair given the sampled actions.

### 3.2 Policy Learning Phase

After all agents have constructed the initial core sets based on the sampled trajectory, they proceed to the policy learning phase by executing a multi-agent learning subroutine(Algorithm 3) recursively from $h = H$ to $h = 1$. To address the issue of multi-agency, we have incorporated the adaptive sampling strategy proposed in Li et al. (2022), which operates under random access, into this subroutine. We have modified this approach by restricting the sampling to the core set of each agent $i$ instead of all $\mathcal{S} \times \mathcal{A}_i$ pairs. This is because our algorithm operates under local access and the core set provides enough information for efficient learning without revisiting unnecessary states and actions.

**Multi-Agent Learning Subroutine.** At the $k$-th iteration of Algorithm 3, each agent $i$ employs the local sampling subroutine (Algorithm 4) over his core set, which returns a reward-state pair $(r, s')$. This design ensures that $q^k_{h,i} := r + \hat{V}_{h+1,i}(s')$ provides a one-step estimation of $Q^{\pi^k_{-i}, \hat{V}_{h+1,i}}_{h,i}(s, a)$. If the estimators $q^k_{h,i}(\tilde{s}, \tilde{a})$ are collected for all $(\bar{s}, \bar{a}) \in \mathcal{D}_{h,i}$ without restart, we proceed to update $Q^k_{h,i}$ via least square regression over the collected data:

$$Q^k_{h,i}(s, a) = \phi_i(s, a)^\top \Lambda^{-1}_{h,i} \sum_{(\tilde{s}, \tilde{a}) \in \mathcal{D}_{h,i}} \phi_i(\tilde{s}, \tilde{a}) q^k_{h,i}(\tilde{s}, \tilde{a}) \quad (6)$$

and take policy iteration using the FTRL update (Lattimore and Szepesvári, 2020), which has been widely adopted in the multi-agent game to break the curse of multi-agency (Li et al., 2022; Jin et al., 2021b; Song et al., 2021). After $K$ epochs, we obtain final policy $\{\pi^k_{h,i}\}_{k=1}^K$ and the estimated value

$$\hat{V}_{h,i}(s) = \min \left\{ \frac{1}{K} \sum_{k=1}^K \langle \pi^k_{h,i}, Q^k_{h,i}(s, \cdot) \rangle, H - h + 1 \right\} \quad (7)$$

under $\pi$ correspondingly.

**Algorithm 5:** Policy Rollout

**Input:** rollout policy $\pi$, initial state $s_1$, rollout times $N$
**for** $n \in [N]$ **do**
  Set $s' = s_1$
  **for** $h = 1, \ldots, H$ **do**
    Sample $\boldsymbol{a} \sim \pi_h(s')$, $s' \sim \mathbb{P}_h(\cdot|s', \boldsymbol{a})$.
    **if** $s' \notin \mathcal{C}_{h+1}$ **then**
      Explore$(s', h+1)$
      **return** *False*
    **end**
  **end**
**end**
**return** *True*

### 3.3 Rollout Checking Phase

If the policy $\hat{\pi}$ is learned without any restarts, then the Algorithm 2 will execute the final rollout checking procedure (Algorithm 5) to determine whether to output the learned policy $\hat{\pi}$ or not. Given any joint policy $\pi$, the rollout subroutine draws $N$ trajectories by employing $\pi$ for $N$ epochs. Whenever an uncertain state is met during the rollout routine, the algorithm will restart the policy learning phase with the updated confident set.

**Necessity of rollout checking.** The rollout checking is necessary because the policy learning phase only considers information within $\mathcal{C}_h$, while the performance of a policy is determined by all the states encountered in its trajectory. Intuitively, the rollout subroutine ensures that the trajectory generated by

**Algorithm 6:** Single-Agent-Learning

**Input :** time-step $h$,agent $i$, policy $\pi_{-i}$
**for** $(\bar{s}, \bar{a}) \in \mathcal{D}_{h,i}$ **do**
  **for** $k = 1$ **to** $K$ **do**
    $(r, s') \leftarrow$ local sampling$(h, i, \bar{s}, \bar{a}, \pi_{-i})$ //See Algorithm 4
    **if** $s' \notin \mathcal{C}_{h+1}$ **then**
      Explore$(s', h+1)$
      **return** *False*
    **end**
    Compute $q^k_{h,i}(\bar{s}, \bar{a}) = r + \hat{V}^{\dagger}_{h+1,i}(s')$.
  **end**
**end**
$\hat{Q}^{\dagger}_{h,i}(s,a) \leftarrow$
$\frac{1}{K}\phi_i(s,a)^\top \Lambda^{-1}_{h,i} \sum_{k=1}^{K} \sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}} \phi_i(\tilde{s},\tilde{a}) q^k_{h,i}(\tilde{s},\tilde{a})$
$\hat{\pi}^{\dagger}_{h,i}(a|\bar{s}) \leftarrow \mathbf{1}\{a = \text{argmax} Q^{\dagger}_{h,i}(s,\cdot)\}$
$\hat{V}^{\dagger}_{h,i}(s) \leftarrow \max_a \hat{Q}^{\dagger}_{h,i}(s,a)$
**return** *True*

the learned policy only contains states that are well covered by core sets, with a high probability.

Although the aforementioned rollout operation ensures that the trajectory of the joint policy $\hat{\pi}$ lies within well-covered states, this may not hold for best response policies. Specifically, for every player $i$ and their best response policy $\pi^{\dagger}_i$ given $\hat{\pi}_{-i}$, the trajectory of $\pi^{\dagger}_i \times \hat{\pi}_{-i}$ may lie outside of $\{\mathcal{C}_h\}$ with non-negligible probability. This motivates us to perform additional rollout for $\pi^{\dagger}_i \times \hat{\pi}_{-i}$. Since $\pi^{\dagger}_i$ is unknown without knowledge of the underlying transition kernels, we perform a single-agent learning subroutine to obtain an *approximate best response* policy $\hat{\pi}^{\dagger}_i = \{\hat{\pi}^{\dagger}_{h,i}\}_{h\in[H]}$ and then take rollout for $\hat{\pi}^{\dagger}_i \times \hat{\pi}_{-i}$. As shown in the proof, dealing with these learned approximated best response policies is sufficient to provide the CCE guarantee for Algorithm 2.

**Single-Agent Learning Subroutine.** To learn the best response for each agent $i$, we fix the other agents' policies and reduce the problem to a single-agent learning task. Specifically, we use Algorithm 6 to perform least squared value iteration backward in $h$. This subroutine can be seen as a finite-horizon version of the *Confident-LSVI* algorithm proposed in Hao et al. (2022) for single-agent learning under the local access model. Similar to other routines, the learning process restarts when encountering a new uncertain state.

### 3.4 Theoretical Results

Now we state the theoretical result of Algorithm 2, whose proof is deferred to Appendix C.

**Theorem 1.** *Under Assumption 1, Algorithm 2 with* $N, K, \tau, \lambda = Poly(\log(S), d, H, \varepsilon^{-1}, A)$, $\eta_k = \tilde{O}(k \min\{\sqrt{\log(S)/d} + \nu, 1\})]^{-1} Poly(K, d, H))$ *returns an* $(\varepsilon + 3\nu\sqrt{d}H)$*-Markov CCE policy with probability at least* $1-\delta$ *with*

*i)*

$$\tilde{O}\left(\frac{m^2 d^3 H^6}{\varepsilon^2} \min\left\{d^{-1}\log S, A\right\}\right)$$

*query of samples under the local access model when* $\min\left\{d^{-1}\log S, A\right\} \le \varepsilon^{-2}$,

*ii)*

$$\tilde{O}\left(m^2 d^5 H^{14} \varepsilon^{-6}\right)$$

*query of samples under the local access model when* $\min\left\{d^{-1}\log S, A\right\} > \varepsilon^{-2}$.

*The detail of all parameter settings are leaved in Appendix C.*

When compared to previous works operating under online access, Cui et al. (2023) attains $(\varepsilon + \nu H)$-CCE with $\tilde{O}(d^4 H^{10} m^4 \varepsilon^{-4})$ samples. Meanwhile, Wang et al. (2023) achieves $\varepsilon$-CCE with $\tilde{O}(d^4 H^6 m^2 \max_i A_i^5 \varepsilon^{-2})$ samples under the realizability assumption ($\nu = 0$). In the scenario where $\min\{d^{-1}\log S, A\} \le \varepsilon^{-4} H^6$, our result improves the dependency on parameters $d, H, m, \max_i A_i$, and $\varepsilon$. Conversely, in scenarios with extremely large $S, A$ values, our bound $\tilde{O}(m^2 d^5 H^{14} \varepsilon^{-6})$ fall short of those in Cui et al. (2023).

**Tighter Complexity Bound with Random Access.** Since the policy output by Lin-Confident-FTRL is only updated based on the information of the shared confident state set, in the analysis of the algorithm, we need to construct virtual algorithms that connect to Lin-Confident-FTRL on the confident state and have strong guarantees outside the set. Note that the virtual algorithms are solely intended for analytical purposes and will not be implemented under the local access model. Unlike prior works on single-agent RL (Yin et al., 2022; Hao et al., 2022), where an ideal virtual algorithm is constructed using population values of $Q$ functions, we develop our virtual algorithms in an implementable manner under the random access model. As a bonus of our virtual algorithm analysis, we derive an algorithm that can operate directly under the random access model with a tighter sample complexity. We would state the result formally as follows:

**Theorem 2.** *Under Assumption 1, there exists a decentralized algorithm under random access model that returns a joint policy achieving* $(\varepsilon + 3\nu\sqrt{d}H)$*-CCE with probability at least* $1-\delta$ *and* $\tilde{O}(\min\{\varepsilon^{-2}dH^2, \frac{\log(S)}{d}, A\}d^2H^5m\varepsilon^{-2})$ *sample complexity bound. The details of the algorithm design are leaved in Appendix D.*

Under the more restrictive random access protocol, Theorem 2 suggests that there is an algorithm with a

more precise dependency on all parameters compared to previous results. This proposed algorithm can be viewed as an analogue to Algorithm 2. However, due to the relaxed sampling protocol, there's no need for restarts, which results in a savings of a factor $mdH$ when $\{\log(S)/d, A\} < \varepsilon^{-2}$ and a savings of $\varepsilon^{-2}mdH$ when $\{\log(S)/d, A\} \geq \varepsilon^{-2}$. We conjecture that even under the local access protocol, a more refined algorithmic design can avoid this additional restart cost. One possible approach might be to incorporate the *Confident Approximate Policy Iteration* method from Weisz et al. (2022) into our setting. This remains an avenue for future exploration. Lastly, note that in the tabular case, with modifications to the FTRL step-size and value estimation formula, the algorithm proposed in Appendix D coincides with the algorithm proposed in Li et al. (2022), which achieves the minimax optimal sample complexity.

**Sample Complexity without the knowledge of $\nu$.** While our selection of the FTRL stepsize $\eta_k$ in Theorem 1 requires prior knowledge of the misspecification error $\nu$. When $\nu$ is unknown, we can select $\eta_k = \tilde{O}(kK^{-1/2}H\sqrt{d})$ in Algorithm 2. Then the algorithm is still guaranteed to output a $(\varepsilon + 3\nu\sqrt{d}H)$-Markov CCE with $\tilde{O}(m^2d^3H^6\varepsilon^{-2})$ samples.

**Decentralized Implementation and Communication Cost.** While we describe Algorithm 2 and its subroutines in a centralized manner, we remark that it can be implemented in a decentralized manner with limited communication. More precisely, during the running of the algorithm, each agent only need to observe its own rewards and actions. And communication between agents only occurs during the initialization procedure and every time a restart occurs. We will discuss the decentralized implementation in Appendix B and show that the total communication complexity is bounded by $\tilde{O}(mdH)$, which is identical to that of the *PReFI* algorithm proposed in Cui et al. (2023) and the *AVPLR* algorithm proposed in Wang et al. (2023).[2]

## 4 Conclusion

In this work, we have considered multi-agent Markov games with independent linear function approximation within both the random and local access models. Our proposed algorithm, *Linear-Confident-FTRL*, effectively mitigates the challenges associated with multi-agency and circumvents the dependency on the action space for regimes where $S \lesssim e^{d \max_i A_i}$. Additionally, our theoretical analysis has lead to the development of a novel algorithm that offers enhanced sample complexity bounds for independent linear Markov games in the random access model. Several compelling questions remain open for exploration:

The first is to investigate the independent function approximation setting under weaker realizability assumptions. Second, designing an algorithm to attain $O(\varepsilon^{-2})$ sample complexity without polynomial dependency on the action space $A$ and logarithmic dependency on the state space $S$ remains an unresolved challenge and an interesting direction for future research.

[2]We remark here both Cui et al. (2023) and Wang et al. (2023) also propose other fully decentralized algorithms, but with worse sample complexity bound.

## Acknowledgements

Jian-Feng Cai is partially supported by Hong Kong Research Grant Council(RGC) GRFs 16310620, 16306821, and 16307023, and Hong Kong Innovation and Technology Fund MHP/009/20. Jiheng Zhang is supported by RGC GRF 16214121. Yang Xiang is supported by the Project of Hetao Shenzhen-HKUST Innovation Cooperation Zone HZQB-KCZYB-2020083. Yang Wang is supported by RGC CRF 8730063 and Hong Kong Center of AI, Robotics and Electronics (HK CARE) for Prefabricated Construction.

## References

Bai, Y., Jin, C., and Yu, T. (2020). Near-optimal reinforcement learning with self-play. *arXiv preprint arXiv:2006.12007*.

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

Chen, F., Mei, S., and Bai, Y. (2022). Unified algorithms for rl with decision-estimation coefficients: No-regret, pac, and reward-free learning. *arXiv preprint arXiv:2209.11745*.

Chen, Z., Zhou, D., and Gu, Q. (2021). Almost optimal algorithms for two-player markov games with linear function approximation. *arXiv e-prints*, pages arXiv–2102.

Cisneros-Velarde, P. and Koyejo, O. (2023). Finite-sample guarantees for nash q-learning with linear function approximation. *ArXiv*, abs/2303.00177.

Cui, Q., Zhang, K., and Du, S. S. (2023). Breaking the curse of multiagents in a large state space: Rl in markov games with independent linear function approximation. *arXiv preprint arXiv:2302.03673*.

Dai, Y., Cui, Q., and Du, S. S. (2024). Refined sample complexity for markov games with independent linear function approximation. *arXiv preprint arXiv:2402.07082*.

Dai, Y., Luo, H., Wei, C.-Y., and Zimmert, J. (2023). Refined regret for adversarial mdps with linear function approximation. *arXiv preprint arXiv:2301.12942*.

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

Du, S., Kakade, S., Lee, J., Lovett, S., Mahajan, G., Sun, W., and Wang, R. (2021). Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2019). Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. (2019). Go-explore: a new approach for hard-exploration problems. *ArXiv*, abs/1901.10995.

Foster, D. J., Kakade, S. M., Qian, J., and Rakhlin, A. (2021). The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

Gao, B. and Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*.

Hao, B., Lazic, N., Yin, D., Abbasi-Yadkori, Y., and Szepesvari, C. (2022). Confident least square value iteration with local access to a simulator. In *International Conference on Artificial Intelligence and Statistics*, pages 2420–2435. PMLR.

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

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. (2017). Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR.

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

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

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

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Lan, L.-C., Zhang, H., and Hsieh, C.-J. (2023). Can agents run relay race with strangers? generalization of RL to out-of-distribution trajectories. In *The Eleventh International Conference on Learning Representations*.

Lattimore, T. and Szepesvári, C. (2020). Bandit algorithms.

Li, G., Chen, Y., Chi, Y., Gu, Y., and Wei, Y. (2021). Sample-efficient reinforcement learning is feasible for linearly realizable mdps with limited revisiting. *Advances in Neural Information Processing Systems*, 34:16671–16685.

Li, G., Chi, Y., Wei, Y., and Chen, Y. (2022). Minimax-optimal multi-agent rl in markov games with a generative model. In *Advances in Neural Information Processing Systems*.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. (2020). Breaking the sample size barrier in model-based reinforcement learning with a generative model. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12861–12872. Curran Associates, Inc.

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

Mao, W., Yang, L. F., Zhang, K., and Baar, T. (2022). On improving model-free algorithms for decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:2110.05707*.

Matignon, L., Jeanpierre, L., and Mouaddib, A.-I. (2012). Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 2017–2023.

Ni, C., Song, Y., Zhang, X., Jin, C., and Wang, M. (2022). Representation learning for general-sum low-rank markov games. *arXiv preprint arXiv:2210.16976*.

Papadimitriou, C. H. and Roughgarden, T. (2008). Computing correlated equilibria in multi-player games. *J. ACM*, 55(3).

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving (2016). *arXiv preprint arXiv:1610.03295*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

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

Tavakoli, A., Levdik, V., Islam, R., Smith, C. M., and Kormushev, P. (2020). Exploring restart distributions. *arXiv: Learning*.

Tkachuk, V., Bakhtiari, S. A., Kirschner, J., Jusup, M., Bogunovic, I., and Szepesvári, C. (2023). Efficient planning in combinatorial action spaces with applications to cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2302.04376*.

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

Wang, Y., Liu, Q., Bai, Y., and Jin, C. (2023). Breaking the curse of multiagency: Provably efficient decentralized multi-agent rl with function approximation. *arXiv preprint arXiv:2302.06606*.

Wang, Y., Wang, R., and Kakade, S. (2021). An exponential lower bound for linearly realizable mdp with constant suboptimality gap. *Advances in Neural Information Processing Systems*, 34:9521–9533.

Weisz, G., Amortila, P., Janzer, B., Abbasi-Yadkori, Y., Jiang, N., and Szepesvári, C. (2021). On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Conference on Learning Theory*, pages 4355–4385.

Weisz, G., György, A., Kozuno, T., and Szepesvári, C. (2022). Confident approximate policy iteration for efficient local planning in q-realizable mdps. *arXiv preprint arXiv:2210.15755*.

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

Xie, Q., Chen, Y., Wang, Z., and Yang, Z. (2020). Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, pages 3674–3682. PMLR.

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

Yin, D., Hao, B., Abbasi-Yadkori, Y., Lazić, N., and Szepesvári, C. (2022). Efficient local planning with linear function approximation. In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR.

Yin, D., Thiagarajan, S., Lazic, N., Rajaraman, N., Hao, B., and Szepesvari, C. (2023). Sample efficient deep reinforcement learning via local planning. *ArXiv*, abs/2301.12579.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. (2020). Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR.

Zhang, K., Kakade, S., Basar, T., and Yang, L. (2020a). Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1166–1178. Curran Associates, Inc.

Zhang, K., Kakade, S., Basar, T., and Yang, L. (2020b). Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *Advances in Neural Information Processing Systems*, 33:1166–1178.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# A Discussion on Policy Class

As pointed in Appendix D of Wang et al. (2023), if the Assumption 1 holds for all possible policy $\pi$, then the underlying game must be essentially tabular game. Thus it would be necessary to discuss the range of policies where our assumption holds. Actually, to ensure Theorem 2 holds, we need only the Assumption 1 holds for the policy class defined as following:

$$\Pi = \left\{ \prod_{i=1}^{m} \pi_i : \pi_i(s,a) \propto \exp(-\eta \phi_i(s,a)^\top \theta_i) : \eta \geq \eta_0, \theta_i \in \mathbb{R}^d \right\} \tag{8}$$

The above soft-max policy class is similar to those considered in Cui et al. (2023), and it also contains the argmax policy considered in Wang et al. (2023) as $\eta \to +\infty$. However, we would remark that while the considered policy class are similar, the independent linear MDP assumption made in our Assumption 1 is strictly stronger than the assumptions made in Cui et al. (2023) and Wang et al. (2023).

On the other hand, to ensure the result in Theorem 1 holds, we need Assumption 1 holds for the following class, which is a bit complex than (8):

$$\Pi = \left\{ \prod_{i=1}^{m} \pi_i : \pi_i(s,a) \propto \begin{cases} \exp(-\eta \phi_i(s,a)^\top \theta_i), & s \in \mathcal{C}_i \\ \exp(-\eta \phi_i(s,a)^\top \theta_i'), & s \notin \mathcal{C}_i \end{cases} : \eta \geq \eta_0, \theta_i, \theta_i' \in \mathbb{R}^d, \mathcal{C}_i \subset \mathcal{S} \right\} \tag{9}$$

The class defined in (9) can be seen as an extension of (8) in the sense that when we divide the state space into two non-overlapping subsets, the policy is a soft-max policy over each subset. Although the policies generated by our main algorithm always lie in (8), our results in the local access setting require Assumption 1 to hold over (9) due to technical reasons in our analysis. We believe that this assumption can be weakened, which we leave as a future direction.

# B Discussion on the Communication Cost

To discuss the communication cost, we would present the decentralized implementation of Algorithm 2. During the implementation, only the knowledge of $\{|\mathcal{D}_{h,i}|\}_{h\in[H],i\in[m]}$ are need to known to each learner.

- **At the beginning of the algorithm,** each agent independently keep the coreset $\mathcal{D}_{h,i} = \emptyset$ and share the same random seed.
- **When the algorithm is restarted,** each agent will share the coreset size $\{\mathcal{D}_{h,i}\}_{h\in[H],i\in[m]}$. Then for each agent $j$, until some agent $j$ meets a new state $s' \notin \mathcal{C}_{h,j}$ at some $h$, each agent can play action only with the knowledge of $\{|\mathcal{D}_{h,i}|\}_{i\in[m],h\in[H]}$ as the following:
  - To independently implement the line 7 to line 12 of the Algorithm 2, it is sufficient to implement Algorithm 3 independently. In the inner loop of Algorithm 3 with the loop-index $k, i, \bar{s}, \bar{a}$, the $j$-th agent play $\pi_{h,j}^k(\cdot|\bar{s})$ when $i \neq j$ and play $\bar{a}$ when $i = j$. When $i = j$, the $j$-th agent will also update his policy and $Q, V$ functions as line 12 to line 19 in Algorithm 3.
  - Line 13 can be implemented independently since they have communicated the shared random seed.
  - To implement Line 15 of the algorithm, each agent $j$ just needs to play $\frac{1}{K}\sum_{k=1}^{K} \pi_{h,i}^k(\cdot|s')$ with the shared random seed for $N$ epoches.
  - To implement the loop in Line 19 to Line 26 with loop index $i$ and inner loop index $\bar{s}, \bar{a}, k$ in Algorithm 6, the agent $j$ play action $\frac{1}{K}\sum_{k'=1}^{K} \pi(\cdot|\bar{s})$ if $i \neq j$ and play $\bar{a}$ if $i = j$. When $i = j$, the $j$-th agent will also update his policy and $Q, V$ functions as line 11 to line 13 in Algorithm 5.
  - The policy rollout loop in Line 28 of Algorithm 2 can be implemented in a similar way as in Line 15.
- During the algorithm, **if some agent $j$ firstly meet some $s' \notin \mathcal{C}_{h+1,j}$, he will send the restarting signal to each agents**, after receiving such signal, each agent take the explore procedure in Algorithm 1 independently, take restart the learning procedure.

In the above procedure, the communication only occurs in the initialization and restarting, thus is at most $\tilde{O}(mdH)$ times.

**Algorithm 7:** Lin-Confident-FTRL-Virtual

**Initialize Global variables:** FirstMeet = True, $\mathcal{C}_{H+1} = \mathcal{S}, \mathcal{C}_h = \emptyset, \forall h \in [H]$ and

$$\tilde{V}_{h,i} = H + 1 - h, \tilde{V}^{\dagger}_{h,i} = H + 1 - h, \mathcal{D}_{h,i} = \emptyset,$$
$$\Lambda_{h,i} = \lambda I, \quad \forall s, i, h.$$

Sample the same trajectory $\{s_1, \cdots, s_H\}$ of length $H$ and obtain the same initialized core sets as Line 2–4 from Algorithm 2

//Policy Learning Phase

**for** $l = 1$ **to** $mHC_{max}$ **do**

  FirstMeet = True //The $l$-th epoch correspondes the $l$-th restart of Algorithm 2

  **for** $h = H$ **to** 1 **do**

    Multi-Agent-Learning-Virtual($h$) .//See Algorithm 8

  **end**

  $\tilde{\pi}_h \leftarrow \frac{1}{K} \sum_{k=1}^{K} \tilde{\pi}^k_{h,1} \times \cdots \times \tilde{\pi}^k_{h,m}, \forall h \in [H]$.

  //Rollout Checking Phase

  Policy-Rollout-Virtual($\tilde{\pi}, s_1, N$)//See Algorithm 9

  **for** $i \in [m]$ **do**

    **for** $h = H$ **to** 1 **do**

      Single-Agent-Learning-Virtual($h$, $i$, $\tilde{\pi}_{h,-i}$).//See Algorithm 10

    **end**

  **end**

  **for** $i \in [m]$ **do**

    Policy-Rollout-Virtual($\tilde{\pi}^{\dagger}_i \times \tilde{\pi}_{-i}, s_1, N$)

  **end**

  **return** $\{\tilde{\pi}_h\}_{h \in [H]}$

**end**

# C Proof of Theorem 1

## C.1 The Virtual Algorithm

As demonstrated by Hao et al. (2022) in the single-agent case, the core-set-based update only guarantees performance within the core sets. However, it is essential to consider the information outside the sets for comprehensive analysis. To address this, we introduce a virtual algorithm, Lin-Confident-FTRL-Virtual (Algorithm 7), which employs the same update as Algorithm 2 within the core sets and also provides good performance guarantees outside them. It is important to note that this virtual algorithm is solely for analytical purposes and will not be implemented in practice. We explain how the virtual algorithm is coupled with the main algorithm and provide additional comments below.

**No restart but update of the coreset** We run Algorithm 7 for $mHC_{\max}$ epochs. During each epoch, the virtual algorithm employs similar subroutines to the Algorithm 2 except that it does not halt and restart when encountering a new state $s'$ not in $\mathcal{C}_h$ during the iteration. Instead, it continues the $K$-step iteration and returns a policy. Upon initially encountering such a state $s'$, the algorithm explores the state and add a subset of $\{s'\} \times A$ to the core sets for use in the next epoch.

**Coupled Simulator** The virtual algorithm is coupled with the main algorithm in the following way: before the discovery of a new state in each epoch, the simulator in the virtual algorithm generates the same action from random policies and the same trajectory of transition as those of Algorithm 2. This coupled dynamic, combined with the condition that core sets are updated only upon the initial encounter with a new state, ensures that at the start of the $l$-th restart, the core sets of Algorithm 2 are identical to those of the $l$-th epoch of the virtual algorithm. Additionally, since the virtual $Q$ function is updated in the same manner as the main algorithm for states in core sets, the virtual policy in the $l$-th epoch is equivalent to the main policy in core sets at the $l$-th restart before encountering the first uncertain state in that epoch. In particular, there exists some $1 \leq \tau \leq C_{\max}$ such that the main policy is identical to the virtual policy for every $h$ and $s \in \mathcal{C}_h$.

**Virtual policy iteration outside the core sets** Besides the core sets, the virtual algorithm maintains in addition a collection of complementary sets $\tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}$ satisfying the confident state set of $(\tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}) \cup \mathcal{D}_{h,i}$ is $S$. Obviously $\tilde{\mathcal{D}}_{h,i}$ is also measurable with respect to the information collected up to finishing $l$-th epoch. The virtual algorithm can query the unvisited states on $\tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}$, which implies that the virtual algorithm has random access to the simulator. The virtual algorithm samples over $\tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}$ and perform least square regression to update $Q$ functions and virtual policies for states outside $\mathcal{C}_h$. Note again that although the virtual algorithm is assumed random access to the simulator, it serves only as a means for analysis and is never implemented.

**Algorithm 9:** Policy-Rollout-Virtual

**Input:** rollout policy $\pi$, initial state $s_1$, rollout times $N$

**for** $n \in [N]$ **do**
  Set $s' = s_1$
  **for** $h = 1, \dots, H$ **do**
    **if** *FirstMeet = True* **then**
      Obtain the same $(\boldsymbol{a}, s')$ as the sampled pair from Line 5 of Algorithm 5 within the same restarting epoch of Algorithm 2
      **if** $s' \notin \mathcal{C}_{h+1}$ **then**
        Explore$(s', h+1)$
        *FirstMeet = False*
      **end**
    **end**
    **else**
      Sample $\boldsymbol{a} \sim \pi_h(s')$, $\quad s' \sim \mathbb{P}_h(\cdot|s', \boldsymbol{a})$.
    **end**
  **end**
**end**

### C.2 Analysis of the Virtual Algorithm

For any $1 \le \ell \le mHC_{\max}$, denote $\mathcal{F}^{\ell-1}$ the $\sigma$-algebra generated by all actions and transitions before the $\ell$-th epoch. If it holds that conditioned on $\mathcal{F}^{\ell-1}$, the policy $\{\tilde{\pi}_h^\ell\}_{h\in[H]}$ outputted by the $\ell$-th epoch satisfies

$$\mathbb{P}\big(V_{1,i}^{\dagger,\tilde{\pi}^\ell_{-i}} - V_{1,i}^{\tilde{\pi}^\ell} \gtrsim H\nu\sqrt{d} + H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)}\big) + \gamma H^2\sqrt{\frac{2\log A_i}{K}}\big) \le \frac{\delta}{mHC_{\max}}. \tag{10}$$

Then it holds that

$$\begin{aligned}
&\mathbb{P}\big(V_{1,i}^{\dagger,\tilde{\pi}^\ell_{-i}} - V_{1,i}^{\tilde{\pi}^\ell} \gtrsim H\nu\sqrt{d} + H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)} + \big) + \gamma H^2\sqrt{\frac{2\log A_i}{K}}, \exists 1 \le \ell \le mHC_{\max}\big)\\
\le&\mathbb{E}[\sum_{\ell=1}^{mHC_{\max}} \mathbf{1}\{V_{1,i}^{\dagger,\tilde{\pi}^\ell_{-i}} - V_{1,i}^{\tilde{\pi}^\ell} \gtrsim H\nu\sqrt{d} + H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)}\big) + \gamma H^2\sqrt{\frac{2\log A_i}{K}}\}]\\
=&\mathbb{E}[\sum_{\ell=1}^{mHC_{\max}} \mathbb{E}[\mathbf{1}\{V_{1,i}^{\dagger,\tilde{\pi}^\ell_{-i}} - V_{1,i}^{\tilde{\pi}^\ell} \gtrsim H\nu\sqrt{d} + H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)}\big) + \gamma H^2\sqrt{\frac{2\log A_i}{K}}\}|\mathcal{F}^{\ell-1}]] \le \delta.
\end{aligned}$$

Then Let $H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} \lesssim \varepsilon$, we have with probability at least $1-\delta$,

$$V_{1,i}^{\dagger,\tilde{\pi}^\ell_{-i}} - V_{1,i}^{\tilde{\pi}^\ell} \le \varepsilon + 3H\nu\sqrt{d}, \quad 1 \le l \le mHC_{\max}.$$

where the coefficient 3 for $H\nu\sqrt{d}$ is from combining (11) and (20). And we select corresponding parameters $K, \tau$ here for $H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} \lesssim \varepsilon$.

**Algorithm 8:** Multi-Agent-Learning-Virtual

**Input:**time-step $h$
**Initialize:** $\tilde{\mathcal{D}}_{h,i}, \tilde{\Lambda}_{h,i} = \Lambda_{h,i}, i \in [m]$
**for** $i = 1$ **to** $m$ **do**
  **while** $\max_{(\tilde{s},a)\in\mathcal{S}\times\mathcal{A}_i} \phi_i(\tilde{s},a)^\top \tilde{\Lambda}_{h,i}^{-1}\phi_i(\tilde{s},a) > \tau$ **do**
    $(\hat{s},\hat{a}_i) = \text{argmax}_{(\tilde{s},a)\in\mathcal{S}\times A_i} \phi_i(\tilde{s},a)^\top \tilde{\Lambda}_{h,i}^{-1}\phi_i(\tilde{s},a)$
    $\tilde{\mathcal{D}}_{h,i} \leftarrow \tilde{\mathcal{D}}_{h,i} \cup \{(\hat{s},\hat{a}_i)\}$
    $\tilde{\Lambda}_{h,i} \leftarrow \tilde{\Lambda}_{h,i} + \phi_i(\hat{s},\hat{a}_i)\phi_i(\hat{s},\hat{a}_i)^\top$
  **end**
**end**
**for** $k = 1$ **to** $K$ **do**
  **for** $i = 1$ **to** $m$ **do**
    **for** $(\bar{s},\bar{a}) \in \mathcal{D}_{h,i}$ **do**
      **if** *FirstMeet* $=$ *True* **then**
        Obtain the same $(r, s')$ as the sampled pair from Line 5 of Algorithm 3 within the same restarting epoch of Algorithm 2
        **if** $s' \notin \mathcal{C}_{h+1}$ **then**
          Explore$(s', h+1)$
          *FirstMeet* $=$ *False*
        **end**
      **end**
      **else**
        $(r, s') \leftarrow$ local sampling$(h, i, \bar{s}, \bar{a}, \tilde{\pi}^k_{h,-i})$
      **end**
      Compute $q^k_{h,i}(\bar{s},\bar{a}) = r + \tilde{V}_{h+1,i}(s')$.
    **end**
    $\tilde{Q}^k_{h,i}(s,a) \leftarrow \phi_i(s,a)^\top \Lambda^{-1}_{h,i} \sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}} \phi_i(\tilde{s},\tilde{a}) q^k_{h,i}(\tilde{s},\tilde{a})$.
    $\bar{Q}^k_{h,i}(s,a) \leftarrow \frac{k-1}{k}\bar{Q}^{k-1}_{h,i}(s,a) + \frac{1}{k}\tilde{Q}^k_{h,i}(s,a)$.
    $\tilde{\pi}^{k+1}_{h,i}(a|s) \leftarrow \frac{\exp(\eta_k \bar{Q}^k_{h,i}(s,a))}{\sum_{a'}\exp(\eta_k \bar{Q}^k_{h,i}(s,a'))}$.
  **end**
  **for** $i = 1$ **to** $m$ **do**
    **for** $(\bar{s},\bar{a}) \in \tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}$ **do**
      $(r, s') \leftarrow$ local sampling$(i, \bar{s}, \bar{a}, \tilde{\pi}^k_h)$ //See Algorithm 4
      Compute $q^k_{h,i}(\bar{s},\bar{a}) = r + \tilde{V}_{h+1,i}(s')$.
    **end**
    $\tilde{Q}^k_{h,i}(s,a) \leftarrow \phi_i(s,a)^\top \Lambda^{-1}_{h,i} \sum_{(\tilde{s},\tilde{a})\in\tilde{\mathcal{D}}_{h,i}} \phi_i(\tilde{s},\tilde{a}) q^k_{h,i}(\tilde{s},\tilde{a})$ for $s \in S \setminus \mathcal{C}_h$.
    $\bar{Q}^k_{h,i}(s,a) \leftarrow \frac{k-1}{k}\bar{Q}^{k-1}_{h,i}(s,a) + \frac{1}{k}\tilde{Q}^k_{h,i}(s,a)$. for $s \in S \setminus \mathcal{C}_h$.
    $\tilde{\pi}^{k+1}_{h,i}(a|s) \leftarrow \frac{\exp(\eta_k \bar{Q}^k_{h,i}(s,a))}{\sum_{a'}\exp(\eta_k \bar{Q}^k_{h,i}(s,a'))}$ for $s \in S \setminus \mathcal{C}_h$.
  **end**
**end**
//Value estimation of $\tilde{V}^{\tilde{\pi}}_{h,i}$ with $\tilde{\pi}_h = \frac{1}{K}\sum_{k=1}^K \tilde{\pi}^k_{h,1} \times \cdots \times \tilde{\pi}^k_{h,m}$
**for** $i = 1$ **to** $m$ **do**
  $\tilde{V}_{h,i}(s) \leftarrow \min\left\{\frac{1}{K}\sum_{k=1}^K \langle \tilde{\pi}^k_{h,i}, \tilde{Q}^k_{h,i}(s,\cdot)\rangle, H-h+1\right\}$
**end**

**Algorithm 10:** Single-Agent-Learning-Virtual

**Input :** time-step $h$,agent $i$, policy $\pi_{-i}$

// Also inherit the coresets $\mathcal{D}_{h,i}, \tilde{\mathcal{D}}_{h,i}$ from Algorithm 8

**for** $(\bar{s},\bar{a}) \in \mathcal{D}_{h,i}$ **do**
  **for** $k = 1$ **to** $K$ **do**
    **if** *FirstMeet* = *True* **then**
      Obtain the same $(r, s')$ as the sampled pair from Line 3 of Algorithm 6 within the same restarting epoch of Algorithm 2
      **if** $s' \notin \mathcal{C}_{h+1}$ **then**
        Explore$(s', h+1)$
        *FirstMeet* = *False*
      **end**
    **end**
    **else**
      $(r, s') \leftarrow$ local sampling$(h, i, \bar{s}, \bar{a}, \pi_{-i})$ //See Algorithm 4
    **end**
    Compute $q^k_{h,i}(\bar{s},\bar{a}) = r + \tilde{V}^{\dagger}_{h+1,i}(s')$.
  **end**
**end**

$\tilde{Q}^{\dagger}_{h,i}(s,a) \leftarrow \frac{1}{K}\phi_i(s,a)^{\top}\Lambda^{-1}_{h,i}\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})q^k_{h,i}(\tilde{s},\tilde{a})$

$\tilde{\pi}^{\dagger}_{h,i}(a|\tilde{s}) \leftarrow \mathbf{1}\{a = \text{argmax}\tilde{Q}^{\dagger}_{h,i}(s,\cdot)\}$

$\tilde{V}^{\dagger}_{h,i}(s) \leftarrow \max_a \tilde{Q}^{\dagger}_{h,i}(s,a)$

**repeat** the loop in line 2 with $\tilde{\mathcal{D}}_{h,i} \setminus \mathcal{D}_{h,i}$ and update $\tilde{Q}^{\dagger}_{h,i}, \tilde{\pi}^{\dagger}_{h,i}, \tilde{V}^{\dagger}_{h,i}$ using the collected data as in line 17 to line 19 over $s \in S \setminus \mathcal{C}_h$.

When $\min\{d^{-1}\log S, A\} \le \varepsilon^{-2}$, select $K = \tilde{O}\big(H^4 d\varepsilon^{-2}\min\{d^{-1}\log S, A\}\big)$, $\tau = 1$, then it holds

$$H^2\sqrt{\tau}\Big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} \lesssim \varepsilon + cH\nu\sqrt{d}.$$

The corresponding query of samples is $m^2H^2KC^2_{\max} = \tilde{O}\big(m^2H^6d^3\varepsilon^{-2}\min\{d^{-1}\log S, A\}\big)$.

When $\min\{d^{-1}\log S, A\} > \varepsilon^{-2}$, select $K = \tilde{O}\big(H^4 d\varepsilon^{-2}\big)$, $\tau = \tilde{O}\big(H^{-4}\varepsilon^2 d^{-1}\big)$, then it holds

$$\begin{aligned} H^2\sqrt{\tau}\Big(\sqrt{\frac{\log(S)}{K}} \wedge \sqrt{d(\frac{A}{K}\wedge 1)} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} &\le H^2\sqrt{\tau d} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} \\ &\lesssim H^2\sqrt{\tau d} + H^2\sqrt{\frac{d}{K}} \\ &\lesssim \varepsilon. \end{aligned}$$

And the corresponding query of samples is $m^2H^2KC^2_{\max} = \tilde{O}\big(m^2H^{14}d^5\varepsilon^{-6}\big)$.

Thus it is sufficient to prove (10) for every fixed $\ell$. For simplicity of the notation, we omit the index $\ell$ in the followed analysis.

### C.2.1 Proof of (10)

We recall the following notations:

$$\begin{aligned} V^{\tilde{\pi}}_{h,i} &= \mathbb{E}_{\boldsymbol{a}\sim\tilde{\pi}}[Q^{\tilde{\pi}}_{h,i}(s,\boldsymbol{a})], \\ V^{\dagger,\tilde{\pi}_{-i}}_{h,i}(s) &= \max_a\big\{r^{\tilde{\pi}_{h,-i}}_{h,i}(s,a) + \mathbb{P}^{\tilde{\pi}_{-i}}_h V^{\dagger,\tilde{\pi}_{-i}}_{h+1,i}(s,a)\big\}, \quad \text{with } V^{\dagger,\tilde{\pi}_{-i}}_{H+1,i} = 0 \\ P^{\tilde{\pi}_{-i}}_h V(s,a) &= \mathbb{E}_{\boldsymbol{a}_{-i}\sim\tilde{\pi}_{-i}}[\mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,\boldsymbol{a}_{-i})}[V(s')]] \\ r^{\tilde{\pi}_{h,-i}}_{h,i}(s,a) &= \mathbb{E}_{\boldsymbol{a}_{-i}\sim\tilde{\pi}_{h,-i}(\cdot|s)}\big[r_{h,i}(s,a,\boldsymbol{a}_{-i})\big], \end{aligned}$$

$$\tilde{V}_{h,i}(s) = \min\left\{\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{a}\sim\tilde{\pi}^k_{h,i}}[\tilde{Q}^k_{h,i}(s,\boldsymbol{a})], H-h+1\right\},$$
$$Q^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(s,a) = \mathbb{E}_{\boldsymbol{a}_{-i}\sim\tilde{\pi}^k_{h,-i}(\cdot|s)}\big[r_{h,i}(s,a,\boldsymbol{a}_{-i}) + \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a,\boldsymbol{a}_{-i})}[\tilde{V}_{h+1,i}(s')]\big]$$

We begin with the following decomposition for each $h\in[H]$:

$$\begin{aligned}V^{\dagger,\tilde{\pi}_{-i}}_{h,i} - V^{\tilde{\pi}}_{h,i} &= V^{\dagger,\tilde{\pi}_{-i}}_{h,i} - V^{\tilde{\pi}}_{h,i} \pm \tilde{V}_{h,i}\\ &= \underbrace{V^{\dagger,\tilde{\pi}_{-i}}_{h,i} - \tilde{V}_{h,i}}_{I^{(1)}_{h,i}} + \underbrace{\tilde{V}_{h,i} - V^{\tilde{\pi}}_{h,i}}_{I^{(2)}_{h,i}}.\end{aligned}$$

We deal with $I^{(1)}_{h,i}$ and $I^{(2)}_{h,i}$ separately in the following two subsections:

**Lemma 2.** *For every $i\in[m], h\in[H]$, we have, with high probability,*

$$I^{(1)}_{1,i} \lesssim H^2\sqrt{\frac{\tau\log(1/\delta)}{K}\min\{d,\log S\}} + 2H\nu\sqrt{d} + \gamma H^2\sqrt{\frac{2\log A_i}{K}}. \tag{11}$$

**Lemma 3.** *For every $i\in[m], h\in[H]$, we have, with high probability,*

$$I^{(2)}_{1,i} \lesssim H\nu\sqrt{d} + H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}}\wedge\sqrt{d(\frac{A}{K}\wedge 1)}\big). \tag{12}$$

### C.2.2 Proof of Lemma 2

*Proof.* We would bound $I^{(1)}_{h,i}$ by taking backward induction on $h$:

Firstly we have $I^{(1)}_{H+1,i}=0$ by definition, now suppose it holds for $h+1$ that with probability at least $1-\delta_{h+1}$,

$$I^{(1)}_{h+1,i}\le z_{h+1,i}$$

for some $z_{h+1,i}\ge 0$,

then we have

$$\begin{aligned}V^{\dagger,\tilde{\pi}_{-i}}_{h,i}(s) &= \max_a\big\{r^{\tilde{\pi}_{h,-i}}_{h,i}(s,a) + \mathbb{P}^{\pi_{-i}}_h V^{\dagger,\tilde{\pi}_{-i}}_{h+1,i}(s,a)\big\}\\ &\le \max_a\big\{r^{\tilde{\pi}_{h,-i}}_{h,i}(s,a) + \mathbb{P}^{\pi_{-i}}_h \tilde{V}_{h+1,i}(s,a)\big\} + z_{h+1,i}\end{aligned}$$

Now if we denote

$$\zeta_{h,i}(s,a) := \big|r^{\tilde{\pi}_{h,-i}}_{h,i}(s,a) + \mathbb{P}_h\tilde{V}_{h+1,i}(s,a) - \frac{1}{K}\sum_{k=1}^{K}\tilde{Q}^k_{h,i}(s,a)\big|,$$

then it holds by induction assumption that with probability at least $1-\delta_{h+1}$

$$\begin{aligned}V^{\dagger,\tilde{\pi}_{-i}}_{h,i}(s) &\le \max_a\frac{1}{K}\sum_{k=1}^{K}\tilde{Q}^k_{h,i}(s,a) + \max_a\zeta_{h,i}(s,a) + z_{h+1,i}\\ &\le \tilde{V}_{h,i}(s) + Reg(FTRL) + \max_a\zeta_{h,i}(s,a) + z_{h+1,i},\end{aligned}$$

with

$$Reg(FTRL) := \frac{1}{K}\max_{a'}\big(\sum_{k=1}^{K}\tilde{Q}^k_{h,i}(s,a') - \sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\tilde{Q}^k_{h,i}(s,a)]\big)$$

For $Reg(FTRL)$ we have the following lemma:

**Lemma 4.** *For $\gamma := \min\{1+\sqrt{\tau\log(SA/\delta)}+\nu\sqrt{d}, \sqrt{d}\}$, we have with probability at least $1-\delta$, it holds for any $s \in \mathcal{S}$ that*

$$Reg(FTRL) \lesssim \gamma H\sqrt{\frac{2\log A_i}{K}}.$$

Now it remains to bound $\zeta_{h,i}$:

When $s \in \mathcal{C}_h$, we have

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\tilde{Q}_{h,i}^{k}(s,a)\\
&=\frac{1}{K}\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})q_{h,i}^{k}(\tilde{s},\tilde{a})\\
&=\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[\sum_{k=1}^{K}Q_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a}))\big]+\underbrace{J_1}_{\text{Martingale Concentration}}\\
&=\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\phi_i(\tilde{s},\tilde{a})^\top\sum_{k=1}^{K}\theta_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}+\underbrace{J_1}_{\text{Martingale Concentration}}+\underbrace{J_2}_{\text{Misspecfic Error}}\\
&=\frac{1}{K}\sum_{k=1}^{K}\phi_i(s,a)^\top\theta_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}+\underbrace{J_1}_{\text{Martingale Concentration}}+\underbrace{J_2}_{\text{Misspecfic Error}}+\underbrace{J_3}_{\text{Incurred by }\lambda}\\
&=\frac{1}{K}\sum_{k=1}^{K}Q_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}(s,a)+O(\nu)+\underbrace{J_1}_{\text{Martingale Concentration}}+\underbrace{J_2}_{\text{Misspecfic Error}}+\underbrace{J_3}_{\text{Incurred by }\lambda}
\end{aligned}$$

where

$$\begin{aligned}
J_1&=\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[\sum_{k=1}^{K}(q_{h,i}^{k}(\tilde{s},\tilde{a})-Q_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a}))\big]\\
J_2&=\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[\sum_{k=1}^{K}(Q_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a}))-\phi_i(\tilde{s},\tilde{a})^\top\theta_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}})\big]\\
J_3&=-\frac{1}{K}\sum_{k=1}^{K}\lambda\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\theta_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}},
\end{aligned}$$

with

$$\theta_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}=\mathrm{argmin}_{\|\theta\|_2\le H\sqrt{d}}\|Q_{h,i}^{\tilde{\pi}_{h,-i}^{k},\tilde{V}_{h+1,i}}-\phi_i(\cdot,\cdot)^\top\theta\|_\infty.$$

Now we aim to control $J_1, J_2, J_3$ separately:

**Bounding $J_1$** For $J_1$, we have the following Lemma:

**Lemma 5.** *With probability at least $1-\delta$, it holds that*

$$J_1 \lesssim H\sqrt{\frac{\tau\log(1/\delta)}{K}\min\{d,\log S\}}.$$

**Bounding $J_2$** For $J_2$, we have by the Assumption 1,

$$|J_2| \le |\phi_i(s,a)^\top\sum_{\tilde{s},\tilde{a}\in\mathcal{D}_{h,i}}\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\cdot\nu|$$

$$\leq \sqrt{|\mathcal{D}_{h,i}| \sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}} |\phi_i(s,a)^\top \Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})|^2} \cdot \nu$$
$$\lesssim \nu\sqrt{d}.$$

**Bounding $J_3$** For $J_3$, we have it holds straightforwardly that

$$|J_3| \leq H\sqrt{\tau\lambda d}$$

In addition, noticing that for $s \notin \mathcal{C}_h$, by our design of virtual algorithm , it holds that

$$\sqrt{|\tilde{D}_{h,i}|} \lesssim \sqrt{\frac{d}{\tau}}, \quad \|\phi_i(s,a)\|^2_{\tilde{\Lambda}^{-1}_{h,i}} \leq \tau, \forall s \notin \mathcal{C}_h$$

thus our arguement when $s \in \mathcal{C}_h$(including the proof of Lemma 5) still holds by replacing $\mathcal{D}_{h,i}, \Lambda_h^{-1}$ by $\tilde{\mathcal{D}}_{h,i}, \tilde{\Lambda}_h^{-1}$. Finally, selecting $\lambda = \lambda_0 := 1/KdH^2$, and by the induction assumption on $(h+1)$-th step, we have with probability at least $1-(\delta_{h+1}+\delta)$,

$$z_h = z_{h+1} + O(H\sqrt{\frac{\tau\log(1/\delta)}{K}\min\{d,\log S\}} + \gamma H\sqrt{\frac{2\log A_i}{K}}) + 2\nu\sqrt{d} \tag{13}$$

Thus (13) recursively with $z_{H+1} = \delta_{H+1} = 0$ leads to with probability at least $1-H\delta$,

$$I_{1,i}^{(1)} \lesssim H^2\sqrt{\frac{\tau\log(1/\delta)}{K}\min\{d,\log S\}} + \gamma H^2\sqrt{\frac{2\log A_i}{K}} + 2H\nu\sqrt{d}$$

□

### C.2.3 Proof of Lemma 3

We would also bound $I_{h,i}^{(2)}$ via backward induction on $h$:

Firstly, we have $I_{H+1,i}^{(2)} = 0$ by definition, now suppose it holds for $h+1$ that with probability at least $1-\delta_{h+1}$,

$$\|I_{h+1,i}^{(2)}\|_\infty \leq \xi_{h+1},$$

then we have for $h$-th time-step, for every $s \in \mathcal{C}_h$,

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^K \mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}[\tilde{Q}^k_{h,i}(s,a_i)] \\
&= \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}\left[\left\langle\phi_i(s,a_i), \Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big(r^k_{i,h}(\tilde{s},\tilde{a}) + \tilde{V}_{h+1,i}(s^k_{\tilde{s},\tilde{a}})\big)\right\rangle\right] \\
&= \underbrace{\frac{1}{K}\sum_{k=1}^K \mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}\left[\phi_i(s,a_i)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})Q_{h,i}^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\right]}_{G_1} \\
&\quad + \underbrace{\frac{1}{K}\sum_{k=1}^K \mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}\left[\phi_i(s,a_i)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\varepsilon_k(\tilde{s},\tilde{a})\right]}_{G_2}.
\end{aligned}$$

with

$$\varepsilon_k(\tilde{s},\tilde{a}) := \big(r^k_{i,h}(\tilde{s},\tilde{a}) + \tilde{V}_{h+1,i}(s^k_{\tilde{s},\tilde{a}})\big) - Q_{h,i}^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big).$$

where $r^k_{i,h}(\tilde{s},\tilde{a}), s^k_{\tilde{s},\tilde{a}}$ denote the $r, s'$ obtained by local sampling$(h, i, \tilde{s}, \tilde{a}, \tilde{\pi}^k_{h,-i})$.

Now we would discuss $G_1, G_2$ separately:

**Bounding $G_1$:**
By Assumption 1 and the induction assumption on $h+1$, we have

$$\begin{aligned} G_1 &= \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}\left[\phi_i(s,a_i)^\top\Lambda^{-1}_{h,i}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})Q^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(\tilde{s},\tilde{a}))\right] \\ &= \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}[Q^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(s,a_i)] + \tilde{O}(\nu\sqrt{d}). \end{aligned}$$

Where in the last equation we used the similar argument as in bounding $J_2, J_3$ in previous section.

Now noticing that with probability at least $1-\delta_{h+1}$,

$$\begin{aligned} |Q^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(s,a) - Q^{\tilde{\pi}^k_{h,-i},V^{\tilde{\pi}}_{h+1,i}}_{h,i}(s,a)| &\le \int|\tilde{V}_{h+1,i}(s') - V^{\tilde{\pi}}_{h+1,i}(s')|d\mathbb{P}^{\tilde{\pi}^k_{h,-i}}_h(s'|s,a) \\ &\le \xi_{h+1}, \end{aligned}$$

we have with probability at least $1-\delta_{h+1}$,

$$\begin{aligned} |G_1 - V^{\tilde{\pi}}_{h,i}(s)| &\le \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{a_i\sim\tilde{\pi}^k_{h,i}}[|Q^{\tilde{\pi}^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(s,a_i) - Q^{\tilde{\pi}^k_{h,-i},V^{\tilde{\pi}}_{h+1,i}}_{h,i}(s,a_i)|] + \tilde{O}(\nu\sqrt{d}) \\ &\le \xi_{h+1} + \tilde{O}(\nu\sqrt{d}). \end{aligned}$$

**Bounding $G_2$:**

We have following lemma regarding the upper bound of $|G_2|$ :

**Lemma 6.** *With probability at least $1-\delta$, it holds that*

$$|G_2| \lesssim H\sqrt{\tau\min\left\{\frac{\log(S/\delta)}{K}, d\big(1\wedge A/K\big)\cdot\right\}}.$$

*Proof of Lemma 6.* For any fixed $\tilde{s}\in\mathcal{C}_h$ we have denoted $\mathcal{F}_k(\tilde{s},\tilde{a})$ the filtration generated by the information before taking the $k$-th sampling on $\tilde{s},\tilde{a}$, then for

$$\tilde{Z}_k(\tilde{s},\tilde{a}) := \mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})q^k_{h,i}(\tilde{s},\tilde{a}),$$

it holds that

$$\mathbb{E}[\tilde{Z}_k(\tilde{s},\tilde{a})|\mathcal{F}_k(\tilde{s},\tilde{a})] = \mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big(r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a}) + \mathbb{P}^{\tilde{\pi}^k_{h,-i}}_h\tilde{V}_{h+1,i}(\tilde{s},\tilde{a})\big),$$

and $|\tilde{Z}_k(\tilde{s},\tilde{a})| \le H\max_a|\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})| \le H\tau$ almost surely. Moreover, notice that

$$\mathrm{Var}[\tilde{Z}_k(\tilde{s},\tilde{a})|\mathcal{F}_k(\tilde{s},\tilde{a})] \le \mathbb{E}[\tilde{Z}_k(\tilde{s},\tilde{a})^2|\mathcal{F}_k(\tilde{s},\tilde{a})] \le H^2|\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big|^2,$$

thus applying Freedman's inequality as in Li et al. (2020) leads to with probability at least $1-\delta$,

$$\begin{aligned} &\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a}) - r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a}) - \mathbb{P}^{\tilde{\pi}^k_{h,-i}}_h\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big] \\ =&\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\big(\tilde{Z}_k(\tilde{s},\tilde{a}) - \mathbb{E}[\tilde{Z}_k(\tilde{s},\tilde{a})|\mathcal{F}_k(\tilde{s},\tilde{a})]\big) \end{aligned}$$

$$\lesssim \frac{H}{K}\Bigg(\sqrt{\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\big|\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big|^2\log(Kd/\delta)}+\tau\log(Kd/\delta)\Bigg)$$
$$\lesssim H\sqrt{\frac{\tau}{K}\log(Kd/\delta)}+\frac{H\tau}{K}\log(Kd/\delta)$$

the last line is by

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\big|\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big|^2\\
=&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\phi_i(\tilde{s},\tilde{a})^\top\Lambda_{h,i}^{-1}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]\\
\le&\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]\\
\le&K\tau
\end{aligned}$$

Now taking union bound over all $s\in\mathcal{C}_h$, we get with probability at least $1-\delta$,

$$|G_2|\lesssim H\sqrt{\frac{\tau\log(S/\delta)}{K}}. \tag{14}$$

On the other hand, we have

$$\begin{aligned}
&\big|\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\big|\\
\le&\frac{\sqrt{\tau}}{K}\sum_{k=1}^{K}\Big\|\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\Big\|_{\Lambda^{-1}}.
\end{aligned}$$

Now for each $k$, consider the $\epsilon_0$-net $\mathcal{N}_\epsilon$ of $\mathbb{B}_d$, then it holds that with probability at least $1-\delta$,

$$\begin{aligned}
&\Big\|\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\pi^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\pi^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\Big\|_{\Lambda^{-1}}\\
\le&\sup_{g\in\mathcal{N}_\epsilon}g^\top\Lambda^{-1/2}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\pi^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\pi^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]+\epsilon_0H\sqrt{\tau|\mathcal{D}_{h,i}|}\\
\le&H\sqrt{\log(|\mathcal{N}_{\epsilon_0}|/\delta)}+\epsilon_0H\sqrt{\tau|\mathcal{D}_{h,i}|}.
\end{aligned}$$

Now letting $\epsilon_0=O(\sqrt{\frac{1}{K\tau|\mathcal{D}_{h,i}|}})$ and noticing that $\log|\mathcal{N}_{\epsilon_0}|=\tilde{O}(d)$, we have it holds with probability at least $1-\delta$ that

$$|G_2|\lesssim H\sqrt{\tau\big[\log(|\mathcal{N}_{\epsilon_0}|/\delta)+\frac{1}{K}\big]}=\tilde{O}(H\sqrt{\tau d}) \tag{15}$$

Finally, if we consider the metric over $\mathcal{S}$:

$$D(s,s'):=\max_a\|\phi(s,a)-\phi(s',a)\|$$

then if we consider the minimal $\epsilon$ cover $\mathcal{N}_\epsilon$ of $\mathcal{F}$, it holds trivially that

$$|\mathcal{N}_D(\mathcal{S};\epsilon)|\le|\mathcal{N}_\epsilon|^A \tag{16}$$

by $\{\phi(s,a)_{a\in\mathcal{A}}:s\in\mathcal{S}\}\subset\mathcal{F}^A$ and the fact $\big|\mathcal{N}_{\|\cdot\|_{2,\infty}}(\mathcal{F}^A;\epsilon)\big|\le|\mathcal{N}_\epsilon|^A$.

Now for any $s \in \mathcal{C}_h$, consider its best approximation $\bar{s} \in \mathcal{N}_D(\mathcal{S};\epsilon)$, then it holds that

$$
\begin{aligned}
&\big|\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\big| \\
\leq&\big|\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(\bar{s},a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\big| \\
+&\big|\frac{1}{K}\sum_{k=1}^{K}(\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(\bar{s},a)]-\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)])\Lambda^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\big|.
\end{aligned}
\tag{17}
$$

In particular, noticing that by $D(s,\bar{s})\leq\epsilon$, we have

$$
\begin{aligned}
&\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)] \\
=&\sum_a\tilde{\pi}^k_{h,i}(a|s)\phi_i(s,a) \\
=&\sum_a\tilde{\pi}^k_{h,i}(a|\bar{s})\phi_i(\bar{s},a)+[\sum_a\tilde{\pi}^k_{h,i}(a|s)-\sum_a\tilde{\pi}^k_{h,i}(a|\bar{s})]\phi_i(s,a)+\sum_a\tilde{\pi}^k_{h,i}(a|\bar{s})[\phi_i(\bar{s},a)-\phi_i(s,a)].
\end{aligned}
$$

Now by

$$
\|\sum_a\tilde{\pi}^k_{h,i}(a|\bar{s})[\phi_i(\bar{s},a)-\phi_i(s,a)]\|\leq D(s,\bar{s})\leq\epsilon
$$

and

$$
\begin{aligned}
&\|[\sum_a\tilde{\pi}^k_{h,i}(a|s)-\sum_a\tilde{\pi}^k_{h,i}(a|\bar{s})]\phi_i(s,a)\| \\
\leq&\max_a\|\phi_i(s,a)\|\cdot\sum_a|\tilde{\pi}^k_{h,i}(a|s)-\tilde{\pi}^k_{h,i}(a|\bar{s})| \\
\leq&\|\mathrm{SoftMax}\big(\bar{Q}^k_{h,i}(\cdot,s),\eta_k\big)-\mathrm{SoftMax}\big(\bar{Q}^k_{h,i}(\cdot,\bar{s}),\eta_k\big)\|_1 \\
\leq&\eta_k\|\bar{Q}^k_{h,i}(\cdot,s)-\bar{Q}^k_{h,i}(\cdot,\bar{s})\|_\infty,
\end{aligned}
$$

where in the last line we have used the explicit formula of Jacobian of Soft-Max function in Gao and Pavel (2017) and the following inequality:

$$
\begin{aligned}
\|\sigma_\eta\big(z\big)-\sigma_\eta\big(z'\big)\|_1&=\|\langle D\sigma_\lambda(\xi)(z-z')\|_1 \\
&\leq\|z-z'\|_\infty\sum_{i,j}|D\sigma_\lambda(\xi)|_{i,j} \\
&\leq\eta\|z-z'\|_\infty.
\end{aligned}
$$

and

$$
\begin{aligned}
\|\bar{Q}^k_{h,i}(\cdot,s)-\bar{Q}^k_{h,i}(\cdot,\bar{s})\|_\infty&\leq\max_k\|Q^k_{h,i}(\cdot,s)-Q^k_{h,i}(\cdot,\bar{s})\|_\infty \\
&\leq\max_a\|\phi_i(s,a)-\phi_i(\bar{s},a)\|_2\|\hat{\theta}\|_2 \\
&\leq D(s,\bar{s})O(Hd/\lambda)
\end{aligned}
$$

we get

$$
\|\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(\bar{s},a)]-\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(s,a)]\|_{\Lambda^{-1}}\lesssim\epsilon\sqrt{K/\lambda}.
$$

with proper choice of $\epsilon$.

Applying Freedman's inequality in the last line of (17), we have then with probability at least $1-\delta$,

$$
\big|\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}^k_{h,i}}[\phi_i(\bar{s},a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[(q^k_{h,i}(\tilde{s},\tilde{a})-r^{\tilde{\pi}^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}^k_{h,-i}}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\big|
$$

$$+O\big(\epsilon H\sqrt{\tau/\lambda}\big).$$

Thus it suffice to control the deviation over the first term with fixed $\bar{s}\in\mathcal{N}_\epsilon^D$. Using exactly the same arguement as in establishing (14), we have for every $\bar{s}$, it holds that

$$\Big|\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\mathbb{E}_{a\sim\tilde{\pi}_{h,i}^k}[\phi_i(\bar{s},a)]^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})\big[(q_{h,i}^k(\tilde{s},\tilde{a})-r_{h,i}^{\tilde{\pi}_{h,-i}^k}(\tilde{s},\tilde{a})-\mathbb{P}_h^{\tilde{\pi}_{h,-i}^k}\tilde{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\Big|\lesssim H\sqrt{\frac{\tau}{K}\log(1/\delta)},$$

then taking union bound over $\bar{s}\in\mathcal{N}_\epsilon^D$ and let $\epsilon=\sqrt{\lambda/K}$ leads to

$$|G_2|\lesssim H\sqrt{\frac{\tau dA}{K}}. \tag{18}$$

Now combining (14),(15),(18) leads to the desired result.

□

Combining our bounds on $I_{h,i}^{(1)}$ and $I_{h,i}^{(2)}$, we get then

$$\xi_h\le\xi_{h+1}+\tilde{O}\bigg(\nu\sqrt{d}+H\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}}\wedge\sqrt{d(\frac{A}{K}\wedge 1)}\big)\bigg) \tag{19}$$

Applying (19) recursively with $\xi_{H+1}=0$ leads to

$$\xi_1\lesssim H\nu\sqrt{d}+H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}}\wedge\sqrt{d(\frac{A}{K}\wedge 1)}\big) \tag{20}$$

Combining (11) and (20) together leads to

$$|V_{1,i}^{\dagger,\tilde{\pi}_{-i}}-V_{1,i}^{\tilde{\pi}}|\lesssim H\nu\sqrt{d}+H^2\sqrt{\tau}\big(\sqrt{\frac{\log(S)}{K}}\wedge\sqrt{d(\frac{A}{K}\wedge 1)}\big)$$

That provides the CCE guarantee of every epoch of the virtual algorithm.

### C.3 Analysis of the Single Agent Learning Subroutine

We would show that for each agent $i$, the single agent learning subroutine is provable to output an approximation of the best-response policy when other agents are playing according to $\tilde{\pi}_{-i}$:

**Lemma 7.** *With probability at least $1-\delta$, we have Algorithm 10 returns a policy $\tilde{\pi}_i^\dagger$ so that*

$$V_{h,i}^{\dagger,\tilde{\pi}_{-i}}-V_{h,i}^{\tilde{\pi}_i^\dagger\times\tilde{\pi}_{-i}}\lesssim H^2\sqrt{\frac{\tau\min\{\log S,d\}}{K}}+\nu H\sqrt{d}.$$

*Proof.* As in proof of (10), we have for every $h$,

$$V_{h,i}^{\dagger,\tilde{\pi}_{-i}}-V_{h,i}^{\tilde{\pi}_i^\dagger\times\tilde{\pi}_{-i}}=\underbrace{V_{h,i}^{\dagger,\tilde{\pi}_{-i}}-\tilde{V}_{h,i}^\dagger}_{:=J_{h,i}^{(1)}}+\underbrace{\tilde{V}_{h,i}^\dagger-V_{h,i}^{\tilde{\pi}_i^\dagger\times\tilde{\pi}_{-i}}}_{:=J_{h,i}^{(2)}}$$

For $J_{h,i}^{(1)}$, we have

$$\begin{aligned}V_{h,i}^{\dagger,\tilde{\pi}_{-i}}(s)&=\max_a\big\{r_{h,i}^{\tilde{\pi}_{-i}}(s,a)+\mathbb{P}_{h,i}^{\pi_{-i}}V_{h+1,i}^{\dagger,\tilde{\pi}_{-i}}(s,a)\big\}\\&\le\max_a\big\{r_{h,i}^{\tilde{\pi}_{-i}}(s,a)+\mathbb{P}_{h,i}^{\pi_{-i}}\tilde{V}_{h+1,i}^\dagger(s,a)\big\}+\max_s\big(V_{h+1,i}^{\dagger,\tilde{\pi}_{-i}}(s)-\tilde{V}_{h+1,i}^\dagger(s,a)\big)\end{aligned}$$

$$\leq \tilde{V}^{\dagger}_{h,i}(s) + \big(\max_a \{r^{\tilde{\pi}_{-i}}_{h,i}(s,a) + \mathbb{P}^{\pi_{-i}}_{h,i}\tilde{V}^{\dagger}_{h+1,i}(s)\} - \tilde{V}^{\dagger}_{h,i}(s)\big) + \max_s \big(V^{\dagger,\tilde{\pi}_{-i}}_{h+1,i}(s) - \tilde{V}^{\dagger}_{h+1,i}(s)\big)$$
$$\leq \tilde{V}^{\dagger}_{h,i}(s) + \max_a \{r^{\tilde{\pi}_{-i}}_{h,i}(s,a) + \mathbb{P}^{\pi_{-i}}_{h,i}\tilde{V}^{\dagger}_{h+1,i}(s,a) - \tilde{Q}^{\dagger}_{h,i}(s,a)\} + \max_s \big(V^{\dagger,\tilde{\pi}_{-i}}_{h+1,i}(s,a) - \tilde{V}^{\dagger}_{h+1,i}(s)\big)$$

Now we can bound

$$\max_{s,a} \{r^{\tilde{\pi}_{-i}}_{h,i}(s,a) + \mathbb{P}^{\pi_{-i}}_{h,i}\tilde{V}^{\dagger}_{h+1,i}(s,a) - \tilde{Q}^{\dagger}_{h,i}(s,a)\}$$

using the property of coreset and the martingale concentration argument as in bounding $J_1, J_2, J_3$ in section C.2.2, we have then with probability at least $1-\delta$,

$$\max_s J^{(1)}_{h,i}(s) \leq \max_s J^{(1)}_{h+1,i}(s) + \tilde{O}(H\sqrt{\frac{\tau \min\{d, \log S\}}{K}} + \nu\sqrt{d}) \lesssim H^2\sqrt{\frac{\tau \min\{d, \log S\}}{K}} + \nu H\sqrt{d}, \quad \forall h \in [H].$$

For $J^{(2)}_{h,i}$, noticing that

$$|\tilde{V}^{\dagger}_{h,i}(s) - V^{\tilde{\pi}^{\dagger}_i \times \tilde{\pi}_{-i}}_{h,i}(s)| = \tilde{V}^{\dagger}_{h,i}(s) - \big(r^{\tilde{\pi}_{-i}}_{h,i}(s, \tilde{\pi}^{\dagger}_i(s)) + \mathbb{P}^{\tilde{\pi}_{-i}}_{h,i} V^{\tilde{\pi}^{\dagger}_i \times \tilde{\pi}_{-i}}_{h+1,i}(s, \tilde{\pi}^{\dagger}_i(s))\big)$$
$$\leq \big|\tilde{Q}^{\dagger}_{h,i}(s, \tilde{\pi}^{\dagger}_i(s)) - \big(r^{\tilde{\pi}_{-i}}_{h,i}(s, \tilde{\pi}^{\dagger}_i(s)) + \mathbb{P}^{\tilde{\pi}_{-i}}_{h,i}\tilde{V}^{\dagger}_{h+1,i}(s, \tilde{\pi}^{\dagger}_i(s))\big)\big| + \max_s |\tilde{V}^{\dagger}_{h+1,i}(s) - V^{\tilde{\pi}^{\dagger}_i \times \tilde{\pi}_{-i}}_{h+1,i}(s)|$$
$$\leq \max_{s,a} \big|\tilde{Q}^{\dagger}_{h,i}(s,a) - \big(r^{\tilde{\pi}_{-i}}_{h,i}(s,a) + \mathbb{P}^{\tilde{\pi}_{-i}}_{h,i}\tilde{V}^{\dagger}_{h+1,i}(s,a)\big)\big| + \max_s |\tilde{V}^{\dagger}_{h+1,i}(s) - V^{\tilde{\pi}^{\dagger}_i \times \tilde{\pi}_{-i}}_{h+1,i}(s)|.$$

Then similar to $J^{(1)}_{h,i}$, we get

$$\max_s J^{(2)}_{h,i}(s) \leq \max_s J^{(2)}_{h+1,i}(s) + \tilde{O}(H\sqrt{\frac{\tau \min\{d, \log S\}}{K}} + \nu\sqrt{d}) \lesssim H^2\sqrt{\frac{\tau \min\{d, \log S\}}{K}} + \nu H\sqrt{d}, \quad \forall h \in [H].$$

□

### C.4 Analysis of the Main Algorithm

Now we would derive the $\varepsilon$-CCE guarantee of the main algorithm by bridging its performance to the virtual algorithm.

Firstly, at the last round of the main algorithm(means it succesfully returns the policy without restarting), the outputed policy $\hat{\pi}$ **at every** $s \in \mathcal{C}^{\ell}_h$ has the same performance as the $\tau$-th virtual algorithm for some $1 \leq \tau \leq L$, for which we denote by $\tilde{\pi}$. In particular, since $(s_0, a) \in \mathcal{C}^{\tau}_1$ for all $a \in A$ by definition, we have then by the union bound arguement, with high probability

$$V^{\dagger,\tilde{\pi}_{-i}}_{1,i}(s_0) - V^{\tilde{\pi}}_{1,i}(s_0) \leq \varepsilon + c\nu H\sqrt{d}, \quad \forall i \in [m].$$

Now we still need to bridge $V^{\dagger,\tilde{\pi}_{-i}}_{1,i}(s_0)$ to $V^{\dagger,\hat{\pi}_{-i}}_{1,i}(s_0)$ and $V^{\dagger,\tilde{\pi}}_{1,i}(s_0)$ to $V^{\hat{\pi}}_{1,i}(s_0)$. To do that, we define another virtual algorithm(called quasi algorithm) that nearly same as the algorithm in section C.1, except that for all $s \in S$, we learn all $\bar{\pi}$ from previously defined $\hat{V}_{h+1,i}$ for every $h, i$. Since this algorithm also coupled with the main algorithm, we have its output policy at $\tau$-th epoch is same as the main algorithm **at every** $s \in S$. Thus for every $h, i, s, a$ we have

$$Q^{\hat{\pi}}_{h,i}(s,a) = Q^{\bar{\pi}}_{h,i}(s,a).$$

Thus we need only provide the guarantee of the quasi algorithm:

At every epoch of the Quasi algorithm, for each agent $i$ and its core-set $\mathcal{D}_{h,i}$, consider the $N$ reward paths $\{\bar{r}_{h,i,n}(\tilde{s}, \tilde{a})\}$ generated in the second last uncertainty check loop, then we have by i.i.d. concentration it holds w.h.p.

$$|V^{\bar{\pi}}_{1,i}(s_0) - \frac{1}{N}\sum_{n=1}^{N} \bar{r}_{h,i,n}(\tilde{s}, \tilde{a})| \lesssim \frac{H}{\sqrt{N}}.$$

Taking union bounds over epochs, we have the result holds for every epoch. By the same argument, such result also holds for the quasi algorithm.

Now if we consider the $\tau$-th epoch, we have then

$$\left|V_{1,i}^{\hat{\pi}}(s_0) - V_{1,i}^{\tilde{\pi}}(s_0)\right| \lesssim \frac{H}{\sqrt{N}}. \tag{21}$$

Combining (21) with the $\varepsilon$-CCE guarantee of $\tilde{\pi}$, we have

$$|V_{1,i}^{\hat{\pi}}(s_0) - V_{1,i}^{\dagger,\tilde{\pi}_{-i}}(s_0)| \leq \varepsilon + c\big(\nu H\sqrt{d} + \frac{H}{\sqrt{N}}\big)$$

Now it is sufficient to bridge $V_{1,i}^{\dagger,\tilde{\pi}_{-i}}(s_0)$ with $V_{1,i}^{\dagger,\hat{\pi}_{-i}}(s_0)$, to do that, we consider another virtual algorithm, called the virtual-II algorithm, its operation on learning $\hat{\pi}$ is same as the quasi algorithm, the main difference is its single-agent learning procedure: for each $i, h$, it maintains a complementary core dataset $\tilde{D}_{h,i}$ in the same way as the virtual I algorithm, and taking LSVI using the collected complementary data for $s \notin \mathcal{C}_h$ while do the same LSVI as the main algorithm for $s \in \mathcal{C}_h$. Applying Lemma 7 leads to the following error guarantee for every epoch output policy of the virtual-I, virtual-II algorithm:

$$|V_{1,i}^{\tilde{\pi}_i^\dagger \times \tilde{\pi}_{-i}}(s_0) - V_{1,i}^{\dagger,\tilde{\pi}_{-i}}(s_0)| \leq \varepsilon + c\nu H\sqrt{d}, \quad |V_{1,i}^{\hat{\pi}^\dagger \times \hat{\pi}_{-i}}(s_0) - V_{1,i}^{\dagger,\hat{\pi}_{-i}}(s_0)| \leq \varepsilon + c\nu H\sqrt{d}. \tag{22}$$

On the other hand, the last rollout procedure for every $i$ guarantees at the $\tau$-th epoch,

$$|V_{1,i}^{\hat{\pi}^{\dagger,i}}(s_0) - V_{1,i}^{\tilde{\pi}^{\dagger,i}}(s_0)| \leq \frac{H}{\sqrt{N}} \tag{23}$$

Combining (22) and (23), we have $|V_{1,i}^{\dagger,\tilde{\pi}_{-i}}(s_0) - V_{1,i}^{\dagger,\hat{\pi}_{-i}}(s_0)| \lesssim \varepsilon + \frac{H}{\sqrt{N}}$. That then leads to the desired bound $|V_{1,i}^{\hat{\pi}}(s_0) - V_{1,i}^{\dagger,\hat{\pi}_{-i}}(s_0)| \lesssim \varepsilon + \frac{H}{\sqrt{N}} + c\nu H\sqrt{d}$. Finally letting $N \asymp H^2/\varepsilon^2$ leads to the desired result.

# D Results under the Random Access Model

## D.1 Algorithm under the Random Access Model

We propose the algorithm under the random access model in Algorithm 11 and make several remarks.

**Remark 1.** *When letting $\beta_{h,i} = 0$ and $\alpha_k = \frac{1}{k}$, the update formulas of $\hat{Q}_{h,i}$ and $\hat{V}_{h,i}$ is same as those in Algorithm 3 and Algorithm 8.*

**Remark 2.** *Compared with the algorithm under the local access model, Algorithm 11 doesn't contain the Policy-Rollout subroutine(Line 15 to Line 18 and Line 27 to Line 32) in Algorithm 2. The main reason is that the random access protocol makes the algorithm easy have high confidence to all states after the exploration phase in line 2.*

**Remark 3.** *When we consider the tabular case, i.e. $\phi_i(s,a) = \boldsymbol{e}_{s,a} \in \mathbb{R}^{SA_i}$, Algorithm 11 with*

$$\lambda = 0, \quad \tau = 1, \quad \alpha_k = \frac{c_\alpha \log K}{k - 1 + c_\alpha \log K}, \quad \beta_{i,h} = c_b\sqrt{\frac{\log^3(\frac{KS\sum_i A_i}{\delta}}{KH}} \sum_{k=1}^{K} \alpha_k^K \left\{ Var_{\pi_{i,h}^k(\cdot|s}\big(q_{i,h}^k(s,\cdot)\big) + H \right\}$$

*with*

$$\alpha_i^k = \alpha_i \prod_{j=i+1}^{k} (1-\alpha_j) \text{ if } 0 < i < k, \quad \alpha_i^k = \alpha_k \text{ if } i = k$$

*recovers the algorithm proposed in Li et al. (2022). They have shown that such selection of parameter allows the algorithm to learn a $\varepsilon$-CCE with $\tilde{O}(\frac{H^4 S\sum_i A_i}{\varepsilon^2})$ sample complexity.*

**Algorithm 11:** Linear Game Random Access

**Input :** learning rates $\{\alpha_k\}$ and $\{\eta_{k+1}\}$

**for** $i = 1$ **to** $m$ **do**

select $\mathcal{D}_i \subset S \times A_i, i \in [m]$ such that

$$\max_{s,a} \phi_i(s,a)\big(\sum_{\bar{s}\in\mathcal{C}_i}\sum_{\bar{a}\in\mathcal{A}_i}\phi_i(\bar{s},\bar{a})\phi_i(\bar{s},\bar{a})^\top + \lambda I\big)^{-1}\phi_i(s,a) < \tau, \quad \forall i \in [m].$$

**end**

**for** $h = H$ **to** $1$ **do**

**for** $k = 1$ **to** $K$ **do**

**for** $i = 1$ **to** $m$ **do**

**for** $(s,a) \in \mathcal{D}_i$ **do**

$(r, s') \leftarrow$ local sampling$(i, s, a, \pi^k_{h,-i})$ Compute $q^k_{h,i}(s,a) = r + \hat{V}_{h+1,i}(s')$.

**end**

$$\theta^k = \text{argmin}_\theta \sum_{(s,a)\in\mathcal{D}_i} |q^k_{h,i}(s,a) - \langle\phi_i(s,a),\theta\rangle|_2^2 + \lambda\|\theta\|_2^2$$

$$Q^k_{h,i}(s,a) = \langle\phi_i(s,a), (1-\alpha_k)\theta^{k-1} + \alpha_k\theta^k\rangle.$$

$$\pi^{k+1}_{h,i}(a_i|s) = \frac{\exp(\eta_{k+1}Q^k_{h,i}(s,a_i))}{\sum_{a'}\exp(\eta_{k+1}Q^k_{h,i}(s,a'))}, \quad \forall s \in S.$$

**end**

**end**

**for** $i = 1$ **to** $m$ **do**

$$\hat{V}_{h,i}(s) = \min\left\{\sum_{k=1}^K \alpha_k^K \langle\pi^{k'}_{h,i}, q^{k'}_{h,i}(s,\cdot)\rangle + \beta_{h,i}(s), H-h+1\right\}, \forall s \in S.$$

**end**

**end**

**return** $\hat{\pi}_{h,i} := \sum_{k=1}^K \alpha_k^K \pi^k_{h,i}$.

### D.2 Proof of Theorem 2

Firstly, we would note that when $\beta = 0$, Algorithm 11 is nearly same as the Algorithm 8 despite a slight difference on the construction of the coreset $\mathcal{D}_{h,i}$. And during the proof of Lemma 2 and Lemma 3, the only property we have required for the $\mathcal{D}_{h,i}$ can be summarized as the following:

$$|\mathcal{D}_{h,i}| \lesssim \frac{d(1+\tau)}{\tau}, \quad \sup_{s,a}\phi_i(s,a)(\sum_{\bar{s},\bar{a}\in\mathcal{D}_{\langle,\rangle}}\phi(\bar{s},\bar{a})\phi(\bar{s},\bar{a})^\top + \lambda I)^{-1}\phi_i(s,a) < \tau.$$

And such property is straightforward to verify for $\mathcal{D}_{h,i} \equiv \mathcal{D}_i$. Thus the analysis in Appendix C.2 and the result in (10) can be applied for Algorithm 11. To prove Theorem 2, it suffice to specify the selection of parameters based on (10):

1. When $\min\{d^{-1}\log S, A\} \le \varepsilon^{-2}$: letting

$$K = \tilde{O}(H^4 d\varepsilon^{-2}\min\{d^{-1}\log S, A\}), \tau = 1$$

leads to the $\varepsilon + c\nu\sqrt{d}H$-CCE guarantee, in this case, the total sample complexity is given by $\tilde{O}(KmdH) = \tilde{O}(\varepsilon^{-2}H^5d^2m\min\{d^{-1}\log S, A\})$.

2. When $\min\{d^{-1}\log S, A\} > \varepsilon^{-2}$ : letting

$$K = \tilde{O}(H^4 d\varepsilon^{-2}), \tau = \tilde{O}(H^{-4}\varepsilon^2 d^{-1})$$

leads to the $\varepsilon + cH\nu\sqrt{d}$-CCE guarantee, in this case, the total sample complexity is given by $\tilde{O}(KmdH/\tau) = \tilde{O}(\varepsilon^{-4}H^9d^3)$.

Combining the sample complexity of this two cases leads to the $\tilde{O}(\min\{\varepsilon^{-2}dH^4, d^{-1}\log S, A\}d^2H^5m\varepsilon^{-2})$ sample complexity result.

# E Proof of Auxiliary Results

## E.1 Proof of Lemma 4

*Proof.* We recall the following standard regret result of FTRL Lattimore and Szepesvári (2020):

**Lemma 8.** *For a sequence $\{y_t\}_{t=1}^T \in [0,1]^A$ and the policy sequence generated by*

$$\pi_{t+1,a} \propto \exp(-\eta \sum_{k=1}^{t} y_{ka})$$

*with $\eta = \sqrt{2\log(A)/T}$, it holds that*

$$\max_a \frac{1}{T}\sum_{t=1}^{T}\big(\langle \pi_t, y_t\rangle - y_{ta}\big) \le \sqrt{2\frac{\log A}{T}}$$

Now since it holds the following lemma regarding the bound of $Q_{h,i}^k$:

**Lemma 9.** *With probability at least $1-\delta$, we have*

$$\max_{s,a}|\tilde{Q}_{h,i}^k(s,a)| \lesssim H\min\{\sqrt{d}, 1+\sqrt{\log(SA/\delta)}+\nu\sqrt{d}\}.$$

*Proof of Lemma 9.* Denote

$$\theta_{h,i}^{\pi_{h,-i}^k, \tilde{V}_{h+1,i}} := \mathrm{argmin}_{\|\theta\|_2 \le H\sqrt{d}} \|\phi_i(s,a)^\top\theta - Q_{h,i}^{\tilde{\pi}_{-i,h}^k, \tilde{V}_{h+1,i}}\|_\infty$$

**1. When $s \in \mathcal{C}_h$:**

$$\begin{aligned}
\tilde{Q}_{h,i}^k(s,a) &= \phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})q_{h,i}^k(\tilde{s},\tilde{a}) \\
&= \phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})[q_{h,i}^k(\tilde{s},\tilde{a}) \pm \phi_i(\tilde{s},\tilde{a})^\top\theta_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}] \\
&= Q_{h,i}^{\pi_{-i,h}^k,\tilde{V}_{h+1,i}}(s,a) + \phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})[q_{h,i}^k(\tilde{s},\tilde{a}) - \phi_i(\tilde{s},\tilde{a})^\top\theta_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}] + O(\nu + H\sqrt{\tau\lambda d}) \\
&= Q_{h,i}^{\pi_{-i,h}^k,\tilde{V}_{h+1,i}}(s,a) + O(\nu\sqrt{d\log d} + H\sqrt{\tau\log(1/\delta)}) \\
&= O\big(H(1+\sqrt{\tau\log(1/\delta)}) + \nu\sqrt{d\log d}\big),
\end{aligned}$$

Where the last second line is by

$$\begin{aligned}
&\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})[q_{h,i}^k(\tilde{s},\tilde{a}) - \phi_i(\tilde{s},\tilde{a})^\top\theta_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}] \\
=&\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\big[\underbrace{Q_{h,i}^{\pi_{-i,h}^k}(\tilde{s},\tilde{a}) - \phi_i(\tilde{s},\tilde{a})^\top\theta_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}}_{\nu_{h,i}^k(\tilde{s},\tilde{a})} + \underbrace{q_{h,i}^k(\tilde{s},\tilde{a}) - Q_{h,i}^{\pi_{-i,h}^k}(\tilde{s},\tilde{a})}_{\mu_{h,i}^k(\tilde{s},\tilde{a})}\big]
\end{aligned}$$

and

$$|\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\nu_{h,i}^k(\tilde{s},\tilde{a})| \le \sqrt{\sum_{\tilde{s},\tilde{a}\in\mathcal{D}_{h,i}}[\phi_i(s,a)^\top\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})]^2}\sqrt{|\mathcal{D}_{h,i}|}\nu$$

$$\lesssim \sqrt{\tau} \cdot \sqrt{\frac{d \log d}{\tau}} \nu.$$

and with probability at least $1-\delta$,

$$\begin{aligned} |\phi_i(s,a)^\top \Lambda_{h,i}^{-1} \sum_{(\tilde{s},\tilde{a}) \in \mathcal{D}_{h,i}} \phi_i(\tilde{s},\tilde{a}) \mu_{h,i}^k(\tilde{s},\tilde{a})| &\leq H \sqrt{\sum_{\tilde{s},\tilde{a} \in \mathcal{D}_{h,i}} [\phi_i(s,a)^\top \Lambda_{h,i}^{-1} \phi_i(\tilde{s},\tilde{a})]^2 \log(1/\delta)} \\ &\lesssim H \sqrt{\tau \log(1/\delta)}. \end{aligned}$$

That for any fixed $(s,a) \in \mathcal{C}_h \times \mathcal{A}_i$, with probability at least $1-\delta$,

$$|\tilde{Q}_{h,i}^k(s,a)| \leq c\big[H(1+\sqrt{\tau \log(SA/\delta)}) + \nu \sqrt{d \log d}\big]$$

On the other hand, we have the following determinstic bound:

$$\begin{aligned} \tilde{Q}_{h,i}^k(s,a) &= \phi_i(s,a)^\top \Lambda_{h,i}^{-1} \sum_{(\tilde{s},\tilde{a}) \in \mathcal{D}_{h,i}} \phi_i(\tilde{s},\tilde{a}) q_{h,i}^k(\tilde{s},\tilde{a}) \\ &\leq H \sqrt{|\mathcal{D}_{h,i}|} \cdot \sqrt{\sum_{\tilde{s},\tilde{a} \in \mathcal{D}_{h,i}} [\phi_i(s,a)^\top \Lambda_{h,i}^{-1} \phi_i(\tilde{s},\tilde{a})]^2} \\ &\lesssim H\sqrt{d}. \end{aligned}$$

Thus for any fixed $(s,a) \in \mathcal{C}_h \times \mathcal{A}_i$, with probability at least $1-\delta$,

$$|\tilde{Q}_{h,i}^k(s,a)| \lesssim H \min\{1+\sqrt{\tau \log(1/\delta)}) + \nu\sqrt{d}, \sqrt{d}\}$$

**2. When** $s \notin \mathcal{C}_h$:

By our construction of $\tilde{\mathcal{D}}_{h,i}, \tilde{\Lambda}_h^{-1}$ in the virtual algorithm, it holds that

$$\sqrt{|\tilde{D}_{h,i}|} \lesssim \sqrt{d/\tau}, \quad \|\phi_i(s,a)\|_{\tilde{\Lambda}_{h,i}^{-1}} \leq \tau, \forall s \notin \mathcal{C}_h$$

thus our argument when $s \in \mathcal{C}_h$ still holds by replacing $\mathcal{D}_{h,i}, \Lambda_h^{-1}$ by $\tilde{\mathcal{D}}_{h,i}, \tilde{\Lambda}_h^{-1}$.

i.e. for any fixed $(s,a) \notin \mathcal{C}_h \times \mathcal{A}_i$, with probability at least $1-\delta$,

$$|Q_{h,i}^k(s,a)| \lesssim H \min\{1+\sqrt{\tau \log(1/\delta)}) + \nu\sqrt{d}, \sqrt{d}\}$$

Now, taking union bound on $(s,a)$, we have with probability at least $1-\delta$,

$$|Q_{h,i}^k(s,a)| \lesssim H \min\{1+\sqrt{\tau \log(SA/\delta)} + \nu\sqrt{d}, \sqrt{d}\}.$$

□

Denote $\gamma := \min\{1+\sqrt{\tau \log(SA/\delta)} + \nu\sqrt{d}, \sqrt{d}\}$, by Lemma 9 we have there exists some absolute number $c > 0$ so that with probability at least $1-\delta$,

$$\tilde{y}_k(s,a) := \frac{cH\gamma - Q_{h,i}^k(s,a)}{2cH\gamma} \in [0,1] \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}_i.$$

Now by Lemma 8, for the policy sequence generated by

$$\pi_{k,a}(a|s) \propto \exp(-\eta \sum_{t=1}^k \tilde{y}_t(s,a)) \propto \exp(\frac{\eta}{2cH\gamma} \sum_{t=1}^k \tilde{Q}_{h,i}^k(s,a)) \propto \exp(\eta_k \tilde{\tilde{Q}}_{h,i}^k(s,a)) \tag{24}$$

with $\eta = \sqrt{2\log(A_i)/K}$, it holds that

$$\max_a \frac{1}{K}\sum_{k=1}^{K}\big(\langle \pi_k(\cdot|s), \tilde{y}_k(s,a)\rangle - \tilde{y}_k(s,a)\big) \le \sqrt{2\frac{\log A_i}{K}} \tag{25}$$

Multiplying $2c\gamma$ to both sides of (25) and noticing that the iteration formula in (25) is exactly the formula for updating $\tilde{\pi}^k_{h,i}$ in Algorithm 8, we get with probability at least $1-\delta$,

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{a\sim\pi^k_{h,i}}[\tilde{Q}^k_{h,i}(s,a)] \le \frac{1}{K}\min_a\sum_{k=1}^{K}\tilde{Q}^k_{h,i}(s,a) + 2c\gamma H\sqrt{\frac{2\log A_i}{K}},$$

That leads to the desired result. □

### E.2 Proof of Lemma 5

*Proof of Lemma.* For any $s, a$ we have denote $\mathcal{F}_k(\tilde{s},\tilde{a})$ the filtration genearted by the information before taking the $k$-th time sampling on $\tilde{s},\tilde{a}$, then for

$$Z_k(\tilde{s},\tilde{a}) := \phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\bar{q}^k_{h,i}(\tilde{s},\tilde{a}),$$

it holds that $\mathbb{E}[Z_k(\tilde{s},\tilde{a})|\mathcal{F}_k(\tilde{s},\tilde{a})] = \phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big(r^{\pi^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a}) + \mathbb{P}^{\pi^k_{h,-i}}_h\widehat{V}_{h+1,i}(\tilde{s},\tilde{a})\big)$, and $|Z_k(\tilde{s},\tilde{a})| \le H|\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})|$ a.s., thus applying Azuma-Hoeffding's inequality leads to with probability at least $1-\delta$,

$$\begin{aligned}
&\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big[\sum_{k=1}^{K}(\bar{q}^k_{h,i}(\tilde{s},\tilde{a}) - r^{\pi^k_{h,-i}}_{h,i}(\tilde{s},\tilde{a}) - \mathbb{P}^{\pi^k_{h,-i}}_h\widehat{V}_{h+1,i}(\tilde{s},\tilde{a}))\big]\\
=&\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\big(Z_k(\tilde{s},\tilde{a}) - \mathbb{E}[Z_k(\tilde{s},\tilde{a})|\mathcal{F}_k(\tilde{s},\tilde{a})]\big)\\
\lesssim&\frac{H}{K}\sqrt{\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}|\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})|^2\log(1/\delta)}.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}|\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})|^2 =& \sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\phi_i(\tilde{s},\tilde{a}))^\top\Lambda^{-1}_{h,i}\phi_i(s,a)\\
\le&\sum_{k=1}^{K}\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(s,a)\\
\le& K\tau.
\end{aligned}$$

Taking union bound over $\mathcal{C}_h\times\mathcal{A}_i$ leads to with probability at least $1-\delta$,

$$J_1 \lesssim H\sqrt{\frac{\tau\log(SA/\delta)}{K}}, \quad \forall(s,a)\in\mathcal{S}\times\mathcal{A}_i. \tag{26}$$

On the other hand, we have it holds for all $(s,a)$ that

$$J_1 = \frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(s,a)^\top\Lambda^{-1}_{h,i}\phi_i(\tilde{s},\tilde{a})\big[\sum_{k=1}^{K}(q^k_{h,i}(\tilde{s},\tilde{a}) - Q^{\pi^k_{h,-i},\tilde{V}_{h+1,i}}_{h,i}(\tilde{s},\tilde{a}))\big]$$

$$\leq \frac{1}{K}\|\phi_i(s,a)\|_{\Lambda_{h,i}^{-1}}\Big\|\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\Big\|_{\Lambda_{h,i}^{-1}}$$
$$\leq \frac{\sqrt{\tau}}{K}\Big\|\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\Big\|_{\Lambda_{h,i}^{-1}}.$$

Now noticing that

$$\begin{aligned}
&\Big\|\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\sum_{k=1}^{K}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\Big\|_{\Lambda_{h,i}^{-1}}\\
=&\Big\|\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\Big\|_2\\
=&\sup_{\|v\|_2=1}\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}v^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big).
\end{aligned}$$

Denote $\mathbb{S}_{\mathcal{H}}=\{g\in\mathbb{R}^d:\|g\|_2=1\}$ , then for any fixed $g\in\mathbb{S}_{\mathcal{H}}$, we have by Azuma-Hoeffding inequality, with probability at least $1-\delta$,

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}g^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\\
\lesssim&H\sqrt{\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}(g^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a}))^2\log(1/\delta)}\leq H\sqrt{K\log(1/\delta)}.
\end{aligned}$$

If we consider the minimal $\epsilon$-net $\mathcal{N}_\epsilon$ of $\mathbb{S}_{\mathcal{H}}$, i.e.

$$\forall g\in\mathbb{S}_{\mathcal{H}},\exists g_0\in\mathcal{N}_\epsilon \text{ such that } \|g-g_0\|_2\leq\epsilon.$$

In particular for any $g,g'\in\mathbb{S}_{\mathcal{H}}$, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}(g-g')^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\\
\lesssim&\|g-g'\|_{\mathcal{H}}H\sqrt{Kd\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}\phi_i(\tilde{s},\tilde{a})\Lambda_{h,i}^{-1}\phi_i(\tilde{s},\tilde{a})}\\
\lesssim&\|g-g'\|_{\mathcal{H}}HK\sqrt{d}.
\end{aligned}$$

That implies for any $g\in\mathbb{S}_{\mathcal{H}}$, there exists some $g_0\in\mathcal{N}_\epsilon$ so that

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}g^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)\\
=&\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}g_0^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)+O(\epsilon HK\sqrt{d}).
\end{aligned}$$

Thus setting $\epsilon=\epsilon_0:=\frac{1}{\sqrt{K}}$ and taking union bound over $\mathcal{N}_\epsilon$ leads to with probability at least $1-\delta$,

$$\sup_{v\in\mathbb{S}^{d-1}}\sum_{k=1}^{K}\sum_{(\tilde{s},\tilde{a})\in\mathcal{D}_{h,i}}v^\top\Lambda_{h,i}^{-1/2}\phi_i(\tilde{s},\tilde{a})\big(q_{h,i}^k(\tilde{s},\tilde{a})-Q_{h,i}^{\pi_{h,-i}^k,\tilde{V}_{h+1,i}}(\tilde{s},\tilde{a})\big)$$

$$\lesssim H\sqrt{K\big[\log(|\mathcal{N}_{\epsilon_0}|/\delta)+d\big]}.$$
$$\lesssim H\sqrt{K\big[d\log(1/\delta)\big]}$$

That leads to another bound of $J_1$: with probability at least $1-\delta$,

$$J_1 \lesssim H\sqrt{\tau\frac{d\log(1/\delta)}{K}} \tag{27}$$

Combining (27) and (26) together leads to the desired result for $s \in \mathcal{C}_h$. □