Learning Pareto-Optimal Policies in Low-Rank Cooperative Markov Games

Abhimanyu Dubey [1] **Alex Pentland** [1]

Abstract

We study cooperative multi-agent reinforcement learning in episodic Markov games with n agents. While the global state and action spaces typically grow exponentially in n in this setting, in this paper, we introduce a novel framework that combines function approximation and a graphical dependence structure that restricts the decision space to $o(dn)$ for each agent, where d is the ambient dimensionality of the problem. We present a multi-agent value iteration algorithm that, under mild assumptions, provably recovers the set of Pareto-optimal policies, with finite-sample guarantees on the incurred regret. Furthermore, we demonstrate that our algorithm is *no-regret* even when there are only $\mathcal{O}(1)$ episodes with communication, providing a scalable and provably no-regret algorithm for multi-agent reinforcement learning with function approximation. Our work provides a tractable approach to multi-agent decision-making that is provably efficient and amenable to large-scale collaborative systems.

Cooperative multi-agent reinforcement learning (MARL) is becoming increasingly prevalent in applications such as robotics (Ding et al., 2020), power grid management (Yu et al., 2014), traffic control (Bazzan, 2009) and team games (Zhao et al., 2019). In this setting, a group of n agents, each with their own state and action spaces, interact simultaneously to maximize their cumulative rewards. The foundational challenge in these multi-agent environments (also known as multi-agent MDPs (Boutilier, 1996) or *cooperative* Markov games (Shapley, 1953)) is that despite having small individual state and action spaces, the joint space grows exponentially in n, introducing a curse of dimensionality that makes standard approaches intractable. Furthermore, designing a *globally* optimal policy is difficult owing to communication and computational constraints.

[1]Media Lab and Institute for Data, Systems and Society, Massachusetts Institute of Technology. Correspondence to: Abhimanyu Dubey <dubeya@mit.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

In single-agent *tabular* reinforcement learning (RL), algorithms exist that provably incur a regret over T episodes that scales as $\widetilde{\mathcal{O}}(H\sqrt{|\mathcal{S}||\mathcal{A}|T})$[1], where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, and H denotes the length of each episode. Such settings normally are agnostic to the low-rank structure present in many environments, and recent work (Jin et al., 2020; Wang et al., 2020; Yang et al., 2020) has explored a *low-rank* linear formulation of MDPs, where the transition kernels and reward functions are assumed to be linear functions of a known $d-$dimensional feature of the state and action. Under this assumption, algorithms have been proposed that provably incur a regret of $\widetilde{\mathcal{O}}(H^2\sqrt{d^3T})$, and when $d^3 \ll |\mathcal{S}||\mathcal{A}|$, this low-rank structure can be exploited effectively in many environments. Concurrently, the multi-agent RL literature has focused on establishing *local dependence structures* (Qu and Li, 2019; Qu et al., 2020), where the dynamics are assumed to be a function of only a subset of agents, effectively reducing the dependence on n from exponential to polynomial, providing *localized* algorithms with provable asymptotic convergence. This complements the approaches based on *factored* MDPs (Guestrin et al., 2002; 2001; Roth et al., 2007), where the rewards incurred by any agent is decomposed into a sum of several latent reward functions.

In this paper, we unify these two perspectives of low-rank function approximation and local dependence structures to present a scalable, provably efficient approach to cooperative multi-agent reinforcement learning. Specifically, we seek to answer the following open question - *can we design tractable, scalable and provably efficient cooperative MARL algorithms with function approximation?*

Contributions. We answer the question affirmatively under mild environmental conditions. First, we present a characterization of cooperative Markov games based on a graphical influence model, where a known (connected, undirected) graph G determines the structure of influence (i.e., an edge (i, j) exists in G if agents i and j influence each other). We extend the single-agent low-rank MDP (Jin et al., 2020) environment to multi-player MDPs and provide a set of weak assumptions, titled *clique-dominance*, that are sufficient to reduce the effective size of the joint state-action space from $\mathcal{O}((|\mathcal{S}||\mathcal{A}|)^n)$ to $o(dn)$, where d is the dimensionality of the

[1]The $\widetilde{\mathcal{O}}$ notation ignores polylogarithmic factors.

approximating function class. Next, we generalize the cooperative MARL objective from maximizing total reward to a broader class of *Pareto-optimal* policies, and characterize conditions in which this class of policies can be efficiently recovered by the *method of scalarization* (Knowles, 2006) by minimizing *Bayes* regret. Thirdly, we introduce **MultOVI**, a decentralized *vector-valued* optimistic value iteration algorithm that even under partial observability conditions, obtains a cumulative *Bayes regret* of $\widetilde{\mathcal{O}}(\theta(G)H^2\sqrt{d^3T})$ over T episodes, where $\theta(G)$ denotes the *clique covering* number of G. **MultOVI** runs in polynomial time and only requires a communication budget of $o(nd^2 \log T)$ rounds per agent, which can be much smaller for sparse G. This ensures that **MultOVI** is scalable to very large environments and adapts to the sparsity of influence as well. Furthermore, in contrast to the existing work in cooperative MARL that converges to the global optimal policy (i.e., maximizing total reward), **MultOVI** can, under mild conditions, recover any subset of policies in the Pareto frontier, additionally enabling *adaptive* load-balancing (Schaerf et al., 1994). Moreover, a direct corollary of our analysis also provides the first no-regret algorithm for multi-objective RL (Mossalam et al., 2016) with function approximation.

Organization. Section 2 presents assumptions about the Markov game considered. Section 3 presents our performance objective and recovery guarantees. Section 4 presents our algorithm and associated regret upper and lower bounds. We defer full proofs, a survey of related work, additional remarks, and experiments to the Appendix for brevity.

2. Preliminaries

Notation. We denote vectors by lowercase solid letters, i.e., $\boldsymbol{x}$, matrices by uppercase solid letters $\mathbf{X}$, and sets by calligraphic letters, i.e., $\mathcal{X}$, the ellipsoid norm of a vector $\boldsymbol{x}$ as $\|\boldsymbol{x}\|_{\mathbf{S}} = \sqrt{\boldsymbol{x}^\top \mathbf{S}\boldsymbol{x}}$ for some matrix $\mathbf{S}$. We denote the interval $a, ..., b$ for $b \geq a$ by $[a, b]$ and as $[b]$ when $a = 1$.

Cooperative Markov Games. We consider the simultaneous-move Markov game (Xie et al., 2020), which is an extension of an MDP to multiple agents, and is also known as a multi-agent MDP (Boutilier, 1996). A Markov game (MG) can be formally described as $\text{MG}(\mathcal{S}, \mathcal{M}, \mathcal{A}, H, \mathbb{P}, \mathbf{R})$, where the set of agents $\mathcal{M}$ is finite and countable with size n, the state and action spaces are factorized as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \mathcal{S}_n$ and $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \mathcal{A}_n$, where $\mathcal{S}_\nu$ and $\mathcal{A}_\nu$ denote the individual state and action space for agent ν respectively. The transition matrix $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}, \mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ determines how the joint state evolves given an existing joint state-action, and the reward function $\mathbf{R} = \{\boldsymbol{r}_h\}_{h=1}^H, \boldsymbol{r}_h = \{r_{\nu,h}\}_{\nu=1}^n, r_{\nu,h} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward obtained by each agent ν in the MG. We further denote, for any subset $\mathcal{Z} \subseteq \mathcal{M}$ of agents, the marginal transition probability for the subset as $\mathbb{P}^{\mathcal{Z}} = \{\mathbb{P}_h^{\mathcal{Z}}\}_{h=1}^H$ such that $\mathbb{P}_h^{\mathcal{Z}} : \mathcal{S} \times \mathcal{A} \times \left(\prod_{i\in\mathcal{Z}} \mathcal{S}_i\right) \to [0, 1]$. Next, we consider a graphical model of influence in order to remove the exponential dependence on n (a generalization of prior work, e.g., Qu and Li (2019); Qu et al. (2020)), summarized below.

Assumption 1 (Local Influence)**.** *Let $G = (\mathcal{M}, \mathcal{E})$ denote an undirected network of influence between agents in $\mathcal{M}$, i.e., $\mathcal{E}$ contains an edge (i, j) if the reward of agent i is a function of agent j (and vice-versa), and let $\mathcal{N}^+(\nu)$ denote, for any agent ν its neighborhood in G (including itself). Alternatively, this implies that the reward for any agent ν obeys $r_{\nu,h} = r_{\nu,h}(\widetilde{\boldsymbol{x}}_\nu, \widetilde{\boldsymbol{a}}_\nu)$ where $\widetilde{\boldsymbol{x}}_\nu = \{x_j\}_{j\in\mathcal{N}^+(\nu)}$ and $\widetilde{\boldsymbol{a}}_\nu = \{x_j\}_{j\in\mathcal{N}^+(\nu)}$ denote the local state-action for ν.*

Ref. Remark 4 in Appendix B. We are not quite yet equipped with a feasible learning model. This is evident as Assumption 1 in the worst case still leads to an exponential dependence on n, and combinatorial state-action spaces have been known to be intractable (Blondel and Tsitsiklis, 1999; Papadimitriou and Tsitsiklis, 1987). As a consequence, recent work has suggested additional conditions bounding the strength of interactions between agents to develop efficient policies (Qu and Li, 2019; Qu et al., 2020). We now describe a similar assumption to characterize dynamics.

Definition 1 (Clique covering number)**.** *A k-clique cover $\mathcal{C} = \{C_1, ..., C_k\}$ of any graph G is a partition of G into k non-overlapping subgraphs such that each subgraph $C_i, i \in [k]$ is strongly connected. The clique covering number $\theta(G)$ is the size of the smallest clique covering $\mathcal{C}^\star$ of G.*

Assumption 2 (Clique-Dominant Dynamics)**.** *For the network G defined in Assumption 1 let $\mathcal{C} = \cup_{l\in[k]} C_l$ be a known k-clique cover. For any $V \subseteq G$ and joint state-action pair $(\boldsymbol{x}, \boldsymbol{a})$, let $\boldsymbol{x}_V = \{\cup_{i\in V} x_i\}$ and $\boldsymbol{a}_V = \{\cup_{i\in V} a_i\}$ denote the joint state and action of all agents in V, and $\bar{\boldsymbol{x}}_V, \bar{\boldsymbol{a}}_V$ denote the joint state and action of all agents not in V. We assume that for each $C \in \mathcal{C}$ and $h \in [H]$ there exists an unknown kernel $\widetilde{\mathbb{P}}_h^C : \left(\prod_{i\in C} \mathcal{S}_i\right) \times \left(\prod_{i\in C} \mathcal{A}_i\right) \times \left(\prod_{i\in C} \mathcal{S}_i\right) \to [0, 1]$, unknown functions $\{\widetilde{r}_{\nu,h}\}_{\nu=1}^C$ and a known nondecreasing function $\varepsilon(\cdot) : [1, n] \to [0, 1]$ such that for any joint state-action $(\boldsymbol{x}, \boldsymbol{a}) = (\{\boldsymbol{x}_C, \bar{\boldsymbol{x}}_C\}, \{\boldsymbol{a}_C, \bar{\boldsymbol{a}}_C\})$, we have for any $\nu \in C$,*

$$|r_{\nu,h}(\boldsymbol{x}, \boldsymbol{a}) - \widetilde{r}_{\nu,h}(\boldsymbol{x}_C, \boldsymbol{a}_C)| \leq \varepsilon(k) \text{ and },$$
$$\left\|\mathbb{P}_h^C(\cdot|\boldsymbol{x}, \boldsymbol{a}) - \widetilde{\mathbb{P}}_h^C(\cdot|\boldsymbol{x}_C, \boldsymbol{a}_C)\right\|_{\mathsf{TV}} \leq \varepsilon(k).$$

Remark 1 (Feasibility of Clique-Dominance)**.** Assumption 2 assumes that if any group of agents C is strongly-connected (i.e., all influence each other), their joint information suffices to "approximately" explain the individual reward and joint marginal transition dynamics up to a factor ε for all agents in C. Naturally, for a smaller clique-covering, a lower approximation error ε can be expected. Similar assumptions for local regularity have been made in prior

work: Qu and Li (2019) introduce the $(c, \rho)-$exponential decay property, see Remarks 5, 6 in Appendix B.

Setting. The game proceeds as follows. In each episode $t \in [T]$ each agent ν fixes a policy $\boldsymbol{\pi}_\nu(t) = \{\pi_\nu^h(t)\}_{h=1}^H$ in a (joint) initial state $\boldsymbol{x}_1(t) = \{x_\nu^1(t)\}_{\nu=1}^n$ picked arbitrarily by the environment. For each step $h \in [H]$ of the episode, each agent observes the local state $\widetilde{\boldsymbol{x}}_\nu^h(t)$, selects an individual action $a_\nu^h(t) \sim \pi_\nu^h(\cdot|\widetilde{\boldsymbol{x}}_\nu^h(t))$ (collectively the joint action $\boldsymbol{a}_h(t) = \{a_\nu^h(t)\}_{\nu=1}^n$), and obtains a reward $r_\nu^h(\widetilde{\boldsymbol{x}}_\nu^h(t), \widetilde{\boldsymbol{a}}_\nu^h(t))$ (collectively the joint reward $\boldsymbol{r}_h(\boldsymbol{x}_h(t), \boldsymbol{a}_h(t)) = \{r_\nu^h(\widetilde{\boldsymbol{x}}_\nu^h(t), \widetilde{\boldsymbol{a}}_\nu^h(t))\}_{\nu=1}^n$). All agents transition subsequently to a new joint state $\boldsymbol{x}_{h+1}(t) = \{x_\nu^{h+1}\}_{\nu=1}^n$ sampled according to $\mathbb{P}_h(\cdot|\boldsymbol{x}_h(t), \boldsymbol{a}_h(t))$. The episode terminates at step $H+1$ where all agents receive no reward. The agents can then (optionally) communicate by sharing messages to neighbors in G after each episode. Let $\boldsymbol{\pi} = \{\boldsymbol{\pi}_\nu\}_{\nu=1}^n$ denote a joint policy for all n agents. We can define the vector-valued value function over all joint states $\boldsymbol{x} \in \mathcal{S}$ for a policy $\boldsymbol{\pi}$ and step h as $\mathbf{V}_h^{\boldsymbol{\pi}}(\boldsymbol{x}) \triangleq \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{i=h}^H \boldsymbol{r}_i(\boldsymbol{x}_i, \boldsymbol{a}_i) \,\middle|\, \boldsymbol{x}_h = \boldsymbol{x}\right]$. Analogously, we define the vector-valued Q-function for a policy $\boldsymbol{\pi}$ and any $\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}, h \in [H], \mathbf{Q}_h^{\boldsymbol{\pi}}(\boldsymbol{x}, \boldsymbol{a}) \triangleq \boldsymbol{r}_h(\boldsymbol{x}, \boldsymbol{a}) + \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{i=h+1}^H \boldsymbol{r}_i(\boldsymbol{x}_i, \boldsymbol{a}_i) \,\middle|\, \boldsymbol{x}_h = \boldsymbol{x}, \boldsymbol{a}_h = \boldsymbol{a}\right]$.

3. Beyond Team-Average Rewards

Cooperative MARL focuses primarily on global objectives, most commonly that of *team-average* reward. While this objective is indeed valid in many environments, we aim to recover the richer class of *Pareto-optimal* objectives (Buchanan, 1962). Formally,

Definition 2 (Pareto optimality, Paria et al. (2020))**.** *A policy $\boldsymbol{\pi}$ Pareto-dominates policy $\boldsymbol{\pi}'$ iff $\mathbf{V}_1^{\boldsymbol{\pi}}(\boldsymbol{x}) \succeq \mathbf{V}_1^{\boldsymbol{\pi}'}(\boldsymbol{x}) \forall\, \boldsymbol{x} \in \mathcal{S}$. A policy is Pareto-optimal if it is not Pareto-dominated by another policy. We denote the set of all policies by $\boldsymbol{\Pi}$, and the Pareto-optimal policies by $\boldsymbol{\Pi}^\star$.*

It is evident that *joint* policies that maximize any agent's individual reward as well as the average reward are all elements of $\boldsymbol{\Pi}^\star$. The motivation to consider recovering the Pareto frontier is indeed derived from applications, e.g., in EV charging (Marinescu et al., 2014), smart grids (Chiu et al., 2019), and workflow optimization (Wang et al., 2019).

Random Scalarizations. To recover $\boldsymbol{\Pi}^\star$, our approach is to utilize the method of *random scalarizations* (Knowles, 2006). The key idea in the method of scalarization is to observe that if the Pareto frontier $\boldsymbol{\Pi}^\star$ is convex, then there is a bijective mapping of each policy in $\boldsymbol{\Pi}^\star$ to the optimal policy of a *scalarized* MDP. Consider a *scalarization function* $\mathfrak{s}_{\boldsymbol{\upsilon}}(\boldsymbol{x}) = \boldsymbol{\upsilon}^\top \boldsymbol{x} : \mathbb{R}^n \to \mathbb{R}$ parameterized by $\boldsymbol{\upsilon}$ belonging to the set $\boldsymbol{\Upsilon} \subseteq \Delta^n$ (unit simplex in n dimensions). We then have the *scalarized* value function $V_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}}(\boldsymbol{x}) : \mathcal{S} \to \mathbb{R}$ and $Q-$function $Q_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for some joint policy $\boldsymbol{\pi}$ as $V_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}}(\boldsymbol{x}) \triangleq \mathfrak{s}_{\boldsymbol{\upsilon}}(\mathbf{V}_h^{\boldsymbol{\pi}}(\boldsymbol{x})) = \boldsymbol{\upsilon}^\top \mathbf{V}_h^{\boldsymbol{\pi}}(\boldsymbol{x})$, and $Q_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}}(\boldsymbol{x}, \boldsymbol{a}) \triangleq \mathfrak{s}_{\boldsymbol{\upsilon}}(\mathbf{Q}_h^{\boldsymbol{\pi}}(\boldsymbol{x}, \boldsymbol{a})) = \boldsymbol{\upsilon}^\top \mathbf{Q}_h^{\boldsymbol{\pi}}(\boldsymbol{x}, \boldsymbol{a})$. Since both $\mathcal{A} = \prod_i \mathcal{A}_i$ and H are finite, there exists an optimal multi-agent policy for any fixed scalarization $\boldsymbol{\upsilon}$, which gives the value $V_{\boldsymbol{\upsilon},h}^\star = \sup_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} V_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$ and $h \in [H]$. This policy coincides with the optimal policy for an MDP over the space $\mathcal{S} \times \mathcal{A}$, defined as follows.

Proposition 1. *For the scalarized value function given above, the Bellman optimality conditions are given as, for all $h \in [H], \boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}, \boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}, Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}) = \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_h(\boldsymbol{x}, \boldsymbol{a}) + \mathbb{P}_h V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}), V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}) = \max_{\boldsymbol{a} \in \mathcal{A}} Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a})$, and $V_{\boldsymbol{\upsilon},H+1}^\star(\boldsymbol{x}) = 0$.*

The optimal policy for any fixed $\boldsymbol{\upsilon}$ is given by the greedy policy with respect to the Bellman-optimal scalarized $Q-$values. We denote this (unique) optimal policy by $\pi_{\boldsymbol{\upsilon}}^\star$. The next result claims that by "projecting" a cooperative Markov game to an MDP via scalarization, one can recover a policy on the Pareto frontier. Indeed, when the set $\boldsymbol{\Pi}^\star$ is convex, then the set of policies $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star = \{\boldsymbol{\pi}_{\boldsymbol{\upsilon}}^\star | \boldsymbol{\upsilon} \in \Delta^n\}$ spans $\boldsymbol{\Pi}^\star$, and one can recover $\boldsymbol{\Pi}^\star$ by simply learning $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star$.

Theorem 1. *For any Markov game with finite $\mathcal{A}$ and H, $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star \subseteq \boldsymbol{\Pi}^\star$. If $\boldsymbol{\Pi}^\star$ is convex, $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star = \boldsymbol{\Pi}^\star$.*

Remark 2 (Limits of Scalarization)**.** This approach suffers from the drawback that convexity assumptions on the scalarization function limit algorithms to only recover policies within the convex regions of $\boldsymbol{\Pi}^\star$ (Vamplew et al., 2008), which is exact when $\boldsymbol{\Pi}^\star$ is convex. Subsequently, our algorithm is limited in this sense as it relies on scalarizations, however, we leave the extension to non-convex regions as future work, and assume $\boldsymbol{\Pi}^\star$ to be convex for simplicity.

Bayes Regret. As mentioned earlier, in many applications, we may require learning policies that prioritize an agent over others. Hence, we consider a general notion of *Bayes regret*. Our objective is to approximate $\boldsymbol{\Pi}^\star$ by learning a set of T policies $\widehat{\boldsymbol{\Pi}}_T$ that minimize the Bayes regret, given by,

$$\mathfrak{R}_B(T) \triangleq \mathop{\mathbb{E}}_{\boldsymbol{\upsilon} \sim p_{\boldsymbol{\Upsilon}}}\left[\max_{\boldsymbol{x} \in \mathcal{S}}\left[V_{\boldsymbol{\upsilon},1}^\star(\boldsymbol{x}) - \max_{\boldsymbol{\pi} \in \widehat{\boldsymbol{\Pi}}_T} V_{\boldsymbol{\upsilon},1}^{\boldsymbol{\pi}}(\boldsymbol{x})\right]\right]. \tag{1}$$

Here $p_{\boldsymbol{\Upsilon}}$ is a distribution over $\boldsymbol{\Upsilon}$ that characterizes the nature of policies we wish to recover. For example, if we set $p_{\boldsymbol{\Upsilon}}$ as the uniform distribution over Δ^n then we can expect the policies recovered to prioritize all agents equally[2]. The advantage of minimizing Bayes regret can be understood as follows. For any $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$, if $\boldsymbol{\pi}_{\boldsymbol{\upsilon}}^\star \in \widehat{\boldsymbol{\Pi}}_T$, then the regret incurred is 0. Hence, by collecting policies that minimize Bayes regret, we are effectively searching for policies

[2]One may consider minimizing the regret for a fixed scalarization $\boldsymbol{\upsilon}' = \mathbb{E}_{p_{\boldsymbol{\Upsilon}}} \boldsymbol{\upsilon}$, however, that will also recover only one policy in $\boldsymbol{\Pi}^\star$, whereas we desire to capture regions of $\boldsymbol{\Pi}^\star$.

that span dense regions of $\mathbf{\Pi}^\star$ (assuming convexity, see Remark 2). Consider now the *cumulative regret*:

$$\mathfrak{R}_C(T) \triangleq \sum_{t\in[T]} \mathop{\mathbb{E}}_{\boldsymbol{\upsilon}_t\sim p_{\boldsymbol{\Upsilon}}} \left[\max_{\boldsymbol{x}\in\mathcal{S}} \left[V^\star_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x})\right]\right]. \tag{2}$$

Where $\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_T \sim p_{\boldsymbol{\Upsilon}}$ are sampled i.i.d. from $p_{\boldsymbol{\Upsilon}}$, and $\boldsymbol{\pi}_t$ refers to the joint policy at episode t. Under suitable conditions on $\mathfrak{s}$ and $\boldsymbol{\Upsilon}$, we can bound the two quantities.

Proposition 2. *For $\mathfrak{s}$ that is Lipschitz and bounded $\boldsymbol{\Upsilon}$, we have that $\mathfrak{R}_B(T) \leq \frac{1}{T}\mathfrak{R}_C(T) + o(1)$.*

4. An Efficient Algorithm

We now present our algorithm **`MultOVI`** (Multiagent Optimistic Value Iteration) that provides a polynomial sample complexity for environments with low-rank structure.

Assumption 3 (Clique-dominant Linear Markov Game)**.** *Let $\mathcal{C}$ be a clique covering of G, and for any clique $C \in \mathcal{C}$, let $\mathcal{S}_C = \prod_{\nu\in C}\mathcal{S}_\nu$ and $\mathcal{A}_C = \prod_{\nu\in C}\mathcal{A}_\nu$ denote the joint state and action space of agents within C. A Markov Game MG$(\mathcal{S}, \mathcal{M}, \mathcal{A}, H, \mathbb{P}, \mathbf{R})$ is a clique-dominant linear Markov game if (a) it is clique-dominant (i.e., obeys Assumption 2), and (b) for every $C \in \mathcal{C}, h \in [H]$, for a set of $|C|+1$ features $\{\boldsymbol{\phi}_\nu\}_{\nu\in C}, \boldsymbol{\phi}_v : \mathcal{S}_C \times \mathcal{A}_C \to \mathbb{R}^d$ and $\boldsymbol{\psi}_C : \mathcal{S}_C \times \mathcal{A}_C \to \mathbb{R}^d$, there exist d unknown measures $\boldsymbol{\mu}^C_h(\cdot) = \{\mu^1_{C,h}(\cdot), ..., \mu^d_{C,h}(\cdot)\}$ over $\mathcal{S}_C$ and an unknown vector $\boldsymbol{\theta}^C_h \in \mathbb{R}^d$ such that $\forall(\boldsymbol{x},\boldsymbol{a}) \in \mathcal{S}_C \times \mathcal{A}_C$ and $\nu \in C$,*

$$\widetilde{\mathbb{P}}^C_h(\cdot|\boldsymbol{x},\boldsymbol{a}) = \left\langle \boldsymbol{\psi}_C(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{\mu}^C_h(\cdot)\right\rangle, \text{ and}$$
$$\widetilde{r}_{\nu,h}(\boldsymbol{x},\boldsymbol{a}) = \left\langle \boldsymbol{\phi}_\nu(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{\theta}^C_h\right\rangle.$$

We denote the overall clique feature vector as $\boldsymbol{\Phi}_C(\cdot) \in \mathbb{R}^{d\times|C|}$, where, for any $\boldsymbol{x} \in \mathcal{S}_C, \boldsymbol{a} \in \mathcal{A}_C$, $\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a}) = [[\boldsymbol{\phi}_1(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{\psi}_C(\boldsymbol{x},\boldsymbol{a})]^\top, ..., [\boldsymbol{\phi}_{|C|}(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{\psi}_C(\boldsymbol{x},\boldsymbol{a})]^\top]^\top$, and the overall approximate clique reward $\widetilde{\boldsymbol{r}}^C_h(\boldsymbol{x},\boldsymbol{a}) = [\widetilde{r}_{1,h}(\boldsymbol{x},\boldsymbol{a}), ..., \widetilde{r}_{|C|,h}(\boldsymbol{x},\boldsymbol{a})]^\top$. Under this representation, we have that for any $\boldsymbol{x} \in \mathcal{S}_C, \boldsymbol{a} \in \mathcal{A}_C, h \in [H]$,

$$\widetilde{\boldsymbol{r}}^C_h(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})^\top \begin{bmatrix}\boldsymbol{\theta}^C_h \\ \mathbf{0}_d\end{bmatrix}, \text{ and,}$$
$$\mathbf{1}_{|C|}\cdot\widetilde{\mathbb{P}}^C_h(\cdot|\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})^\top \begin{bmatrix}\mathbf{0}_d \\ \boldsymbol{\mu}^C_h(\cdot)\end{bmatrix}.$$

Assume WLOG $\forall C \in \mathcal{C}$, $\|\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\| \leq \sqrt{|C|}$ $\forall\,(\boldsymbol{x},\boldsymbol{a}) \in \mathcal{S}_C \times \mathcal{A}_C$, $\|\boldsymbol{\theta}^C_h\| \leq \sqrt{d}$, $\|\boldsymbol{\mu}^C_h(\mathcal{S}_C)\| \leq \sqrt{d}$.

Essentially, this assumption requires that once we are provided a clique covering, and the Markov game obeys the *clique-dominance* property (Assumption 2), the approximate rewards $\widetilde{r}_{\nu,h}$ are linear functions of a known feature vector $\boldsymbol{\phi}_C$ evaluated on the joint state-action of the agents within its clique. Additionally, it assumes that the approximate marginal transitions $\widetilde{\mathbb{P}}^C_h$ are linear functions of a known feature $\boldsymbol{\psi}_C$. This, in fact, is a straightforward extension of the single-agent linear MDP parameterization (see Assumption A in Jin et al. (2020)) to *clique-dominant* Markov games, see Remark 7 in Appendix B to compare.

4.1. Algorithm Design

The first step in our approach is to compute a k-clique covering $\mathcal{C}$ of the influence graph G. Recall that by Remark 6 that this can be done in polynomial time with a 1.25 approximation of $\mathcal{C}^\star$. Since the game is clique-dominant (Assumption 2), we can learn k decentralized policies $\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_k$, one corresponding to each clique of agents in $\mathcal{C}$ without incurring too much approximation error. Now, to motivate the design, we first observe that Assumption 3 implies that for each clique $C \in \mathcal{C}$, there exist a set of weights such that the scalarized $Q-$values for any parameter $\boldsymbol{\upsilon}_C$ are *almost* linear projections of the overall clique features $\boldsymbol{\Phi}_C(\cdot)$, where the total error is no larger than $2H\varepsilon(k)$.

Lemma 1 (Almost linear weights in Markov Games)**.** *Under Assumption 3 for graph G with k cliques ordered from $1, ..., k$, we have, for any fixed decentralized policy $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_k\}$ and $\boldsymbol{\upsilon} = \{\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_k\} \in \boldsymbol{\Upsilon}$, there exist weights $\{\boldsymbol{w}^{\boldsymbol{\pi}_\tau}_{\boldsymbol{\upsilon},h}\}^{H,k}_{h=1,\tau=1}$ such that $\left|Q^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},h}(\boldsymbol{x},\boldsymbol{a}) - \sum_{\tau=1}^k \boldsymbol{\upsilon}_\tau^\top \boldsymbol{\Phi}_\tau(\boldsymbol{x}_\tau,\boldsymbol{a}_\tau)^\top \boldsymbol{w}^{\boldsymbol{\pi}_\tau}_{\boldsymbol{\upsilon},h}\right| \leq 2H\varepsilon(k)$ $\forall(\boldsymbol{x},\boldsymbol{a},h)$, where $\|\boldsymbol{w}^{\boldsymbol{\pi}_k}_{\boldsymbol{\upsilon},h}\|_2 \leq 2H\sqrt{d}$.*

Armed with this observation, we design a policy using *vector-valued* linear least-squares regression, as the optimal policy is only at most $2H\varepsilon$ away from the best least-squares fit. In a nutshell, our approach can be summed up in two steps: (a) first, we approximate the Pareto frontier $\mathbf{\Pi}^\star$ with the set of policies $\mathbf{\Pi}^\star_{\boldsymbol{\Upsilon}}$ recoverable by scalarization (see Remark 2), (b) next, we empirically approximate $\mathbf{\Pi}^\star_{\boldsymbol{\Upsilon}}$ with a collection of T policies (one for each episode), such that the *Bayes Regret* is minimized (Proposition 2). In each episode $t \in [T]$, we sample a scalarization parameter $\boldsymbol{\upsilon}_t \sim p_{\boldsymbol{\Upsilon}}$, and run k *vector-valued* decentralized linear least-squares regressions to approximate the optimal policy $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}_t}$ with k policies $\boldsymbol{\pi}_1(t), ..., \boldsymbol{\pi}_k(t)$ such that the resulting $Q-$values overestimate $Q^\star_{\boldsymbol{\upsilon}_t,h}$ with high probability. Then, each agent in clique τ selects the corresponding greedy action with respect to $\boldsymbol{\pi}_\tau(t)$. This approach is carried out via *value iteration with optimism*, as described below.

We describe the policy for any clique $C \in \mathcal{C}$ of size n_C. For any scalarization $\boldsymbol{\upsilon}(t) \in \mathbb{R}^n$, the n_C values corresponding to agents in $\mathcal{C}$ is denoted by $\boldsymbol{\upsilon}_C(t)$. Now, consider the MDP $\widetilde{\text{MDP}}_C$ formed by scalarizing the Markov game corresponding to the approximate rewards $\widetilde{\boldsymbol{r}}^C_h$ and transition dynamics $\widetilde{\mathbb{P}}^C_h$ with the parameter $\boldsymbol{\upsilon}_{C,t}$ (i.e., the reward function in $\widetilde{\text{MDP}}_C$ is given by $\boldsymbol{\upsilon}_C(t)^\top \boldsymbol{r}^C_h$, transition by $\widetilde{\mathbb{P}}_h$ and state-action spaces as $\mathcal{S}_C$ and $\mathcal{A}_C$ respectively). For each clique C, we will use value iteration to recover the optimal

$$V_C^{h+1}(t)(\boldsymbol{x}) \leftarrow \arg\max_{\boldsymbol{a}\in\mathcal{A}} \left[\boldsymbol{v}_C(t)^\top \left(\mathbf{Q}_C^{h+1}(t)^\top \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\right)\right] \ \forall\, \boldsymbol{x}\in\mathcal{S}_C,$$

$$\widehat{\mathbf{Q}}_C^h(t) \leftarrow \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \left[\sum_{\tau\in[s_t^C]} \left\|\boldsymbol{y}_C^h(\tau) - \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(\tau),\boldsymbol{a}_C^h(\tau))^\top \boldsymbol{w}\right\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2\right],$$

$$Q_C^h(t)(\boldsymbol{x},\boldsymbol{a}) \leftarrow \boldsymbol{v}_C(t)^\top \left(\widehat{\mathbf{Q}}_C^h(t)^\top \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\right) + \beta_C^h(t)\cdot \left\|\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})^\top \boldsymbol{\Lambda}_C^h(t)^{-1} \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\right\|_2.$$

Figure 1. Vector-valued least squares regression update rule for `MultOVI`.

policy for this $\widetilde{\text{MDP}}_C$ (let us call it $\widetilde{\boldsymbol{\pi}}_C^\star(t)$). The algorithm is a distributed variant of least-squares value iteration with UCB exploration. Following Proposition 1, the key idea is to make sure that each agent in C acts according to the joint policy that is aiming to mimic $\widetilde{\boldsymbol{\pi}}_C^\star(t)$. Therefore, we must ensure that the local estimate for the *joint* policy obtained by any agent must be identical, such that the *joint* action is in accordance with $\widetilde{\boldsymbol{\pi}}_C^\star(t)$. To achieve this we will obtain the approximated (scalar) $Q-$values for $\widetilde{\boldsymbol{\pi}}_C^\star(t)$ by recursively applying the Bellman equation and solving the resulting equations via a *vector-valued* regression. Since the policy variables are designed to be identical each agent in C at all times, we describe the procedure for any agent within C. For any episode t, let us assume that the last round of synchronization between agents in C occured at time s_t^C. Each agent within the clique C obtains an *identical* sequence of value functions $\{Q_C^h(t)\}_{h\in[H]}$ by iteratively performing linear least-squares ridge regression from the history available from the previous s_t^C episodes by first learning a vector $Q-$function $\widehat{\mathbf{Q}}_C^h(t)$ over $\mathbb{R}^{n_C}$, which is scalarized by $\boldsymbol{v}_C(t)$ to obtain the $Q-$values as $Q_C^h(t) = \boldsymbol{v}_C(t)^\top \widehat{\mathbf{Q}}_C^h(t)$. Each agent m first sets $\widehat{\mathbf{Q}}_C^{H+1}(t)$ to be a zero vector in $\mathbb{R}^{n_C}$, and for any $h\in[H]$, solves the following sequence of regressions to obtain $Q-$values described in Figure 1. Where the last equation holds for any $(\boldsymbol{x},\boldsymbol{a})\in(\mathcal{S}_C\times\mathcal{A}_C)$ and the targets $\boldsymbol{y}_C^h(\tau) = \boldsymbol{r}_h^C(\boldsymbol{x}_C^h(\tau),\boldsymbol{a}_C^h(\tau)) + \mathbf{1}_{n_C}\cdot V_C^{h+1}(t)(\boldsymbol{x}_C^{h+1}(\tau))$, $\mathbf{1}_{n_C}$ denotes the all-ones vector in $\mathbb{R}^{n_C}$, $\beta_C^h(t)$ is selected such that with high probability the estimated Q-values overestimate the require $Q-$values, and $\boldsymbol{\Lambda}_C^h(t)$ is described subsequently. Once all of these quantities are computed, each agent $\nu\in C$ selects the action $a_\nu^h(t) = \left[\arg\max_{\boldsymbol{a}\in\mathcal{A}_C} Q_C^h(t)(\boldsymbol{x}_C^h(t),\boldsymbol{a})\right]_\nu$ for each $h\in[H]$. Hence, the joint clique action $\boldsymbol{a}_C^h(t) = \{a_\nu^h(t)\}_{\nu=1}^{n_C} = \arg\max_{\boldsymbol{a}\in\mathcal{A}_C} Q_C^h(\boldsymbol{x}_C^h(t),\boldsymbol{a})$. Observe that while the computation of the policy is decentralized, the policies executed for all agents $\nu\in C$ coincide at all times by the modeling assumption and the periodic synchronizations between agents. We now present the closed form of $\widehat{\mathbf{Q}}_C^h(t)$. Consider the contraction $\boldsymbol{z}_C^h(\tau) = (\boldsymbol{x}_C^h(\tau),\boldsymbol{a}_C^h(\tau))$ and the map $\widehat{\boldsymbol{\Phi}}_C^h(t):\mathbb{R}^d\to\mathbb{R}^{tn_C}$ such that for any $\boldsymbol{\theta}\in\mathbb{R}^d$,

$$\boldsymbol{\Phi}_C^h(t)\boldsymbol{\theta} \triangleq \left[(\boldsymbol{\Phi}_C(\boldsymbol{z}_C^h(1))^\top\boldsymbol{\theta})^\top, ..., (\boldsymbol{\Phi}(\boldsymbol{z}_C^h(t))^\top\boldsymbol{\theta})^\top\right]^\top.$$

Now, consider $\boldsymbol{\Lambda}_C^h(t) = \boldsymbol{\Phi}_C^h(s_t^C)^\top\boldsymbol{\Phi}_C^h(s_t^C) + \lambda\mathbf{I}_d \in \mathbb{R}^{d\times d}$, and $\mathbf{U}_C^h(t) = \sum_{\tau=1}^{s_t^C} \boldsymbol{\Phi}_C(\boldsymbol{z}_C^h(\tau))\boldsymbol{y}_C^h(\tau)$. Then, we have by a multi-task concentration (see Appendix B of Chowdhury and Gopalan (2020)),

$$\widehat{\mathbf{Q}}_C^h(t)(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})^\top \boldsymbol{\Lambda}_C^h(t)^{-1}\mathbf{U}_C^h(t).$$

The algorithm is presented in Algorithm 1. The algorithm is essentially learning k multi-agent policies by solving a vector-valued regression, one for each clique in the covering $\mathcal{C}$, such that the group of agents in each clique can learn the approximate clique-based MG (ref. Assumption 2).

4.2. Regret Analysis

Theorem 2. *Algorithm 1 on a game with n agents satisfying Assumptions 1, 2, 3 with error $\varepsilon_\star$, approximate clique covering $\widehat{\mathcal{C}}$, and $\kappa\cdot dH\cdot\theta(G)$ rounds of communication for some $\kappa>1$, obtains, with probability at least $1-\alpha$, regret:*

$$\widetilde{\mathcal{O}}\left(\theta(G)\cdot d^{\frac{3}{2}}H^2(2T\cdot n_{\max})^{\frac{2}{\kappa}}\left(\sqrt{T\log\left(\tfrac{1}{\alpha}\right)} + 2Tn_{\max}\cdot\varepsilon_\star\right)\right).$$

Where $\theta(G)$ denotes the clique covering number of G, and $n_{\max}$ is the size of the largest clique in $\widehat{\mathcal{C}}$.

Remark 3 (Regret Bound)**.** *The above Theorem suggests that our algorithm is no-regret (up to a factor $\varepsilon_\star$), as long as there are a sufficient (logarithmic) rounds of communication. Further, if the clique-dominance error $\varepsilon_\star = o(T^{-\gamma})$ for $\gamma>0$ then the algorithm is no-regret regardless. Additionally, we see that* ***`MultOVI`*** *can be simulated on a single agent with n objectives, where $S=1$ and G is complete, which provides, to the best of our knowledge, the first no-regret algorithm for multi-objective reinforcement learning (Mossalam et al., 2016). We present further clarifications, discussions and lower bounds in Remarks 8, 9, 10 and 11 in Section B of the Appendix.*

Conclusion. We presented the first (to the best of our knowledge) no-regret algorithm for partially-observable cooperative Markov games, with competitive experimental performance (experiments deferred to Appendix for brevity). We generalize several concepts in the cooperative MARL literature, and we believe our results will be important for further work in cooperative MARL.

References

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017.

A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207, 2005.

A. L. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3):342, 2009.

V. D. Blondel and J. N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3):479–489, 1999.

C. Boutilier. Planning, learning and coordination in multiagent decision processes. Citeseer, 1996.

J. M. Buchanan. The relevance of pareto optimality. *Journal of conflict resolution*, 6(4):341–354, 1962.

G. D. Canas and L. Rosasco. Learning probability measures with respect to optimal transport metrics. *arXiv preprint arXiv:1209.1077*, 2012.

M. R. Cerioli, L. Faria, T. O. Ferreira, C. A. Martinhon, F. Protti, and B. Reed. Partition into cliques for cubic graphs: Planar case, complexity and approximation. *Discrete Applied Mathematics*, 156(12):2270–2278, 2008.

W.-Y. Chiu, J.-T. Hsieh, and C.-M. Chen. Pareto optimal demand response based on energy costs and load factor in smart grid. *IEEE Transactions on Industrial Informatics*, 16(3):1811–1822, 2019.

S. R. Chowdhury and A. Gopalan. No-regret algorithms for multi-task bayesian optimization. *arXiv preprint arXiv:2008.08885*, 2020.

G. Ding, J. J. Koh, K. Merckaert, B. Vanderborght, M. M. Nicotra, C. Heckman, A. Roncone, and L. Chen. Distributed reinforcement learning for cooperative multi-robot object manipulation. *arXiv preprint arXiv:2003.09540*, 2020.

A. Ghosh, S. R. Chowdhury, and A. Gopalan. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

M. Grötschel, L. Lovász, and A. Schrijver. Stable sets in graphs. In *Geometric Algorithms and Combinatorial Optimization*, pages 272–303. Springer, 1988.

H. Gu, X. Guo, X. Wei, and R. Xu. Q-learning for mean-field controls. *arXiv preprint arXiv:2002.04131*, 2020.

C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored mdps. In *NIPS*, volume 1, pages 1523–1530, 2001.

C. Guestrin, S. Venkataraman, and D. Koller. Context-specific multiagent coordination and planning with factored mdps. In *AAAI/IAAI*, pages 253–259, 2002.

E. Hillel, Z. Karnin, T. Koren, R. Lempel, and O. Somekh. Distributed exploration in multi-armed bandits. *arXiv preprint arXiv:1311.0800*, 2013.

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

S. Kar, J. M. Moura, and H. V. Poor. Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus innovations. *IEEE Transactions on Signal Processing*, 61(7):1848–1862, 2013.

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

A. Marinescu, I. Dusparic, A. Taylor, V. Cahill, and S. Clarke. Decentralised multi-agent reinforcement learning for dynamic and uncertain environments. *arXiv preprint arXiv:1409.4561*, 2014.

M. Molloy. The list chromatic number of graphs with small clique number. *Journal of Combinatorial Theory, Series B*, 134:264–284, 2019.

H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson. Multi-objective deep reinforcement learning. *arXiv preprint arXiv:1610.02707*, 2016.

A. Pacchiano, M. Phan, Y. Abbasi-Yadkori, A. Rao, J. Zimmert, T. Lattimore, and C. Szepesvari. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*, 2020.

C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

B. Paria, K. Kandasamy, and B. Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2020.

G. Qu and N. Li. Exploiting fast decaying and locality in multi-agent mdp with tree dependence structure. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6479–6486. IEEE, 2019.

G. Qu, Y. Lin, A. Wierman, and N. Li. Scalable multi-agent reinforcement learning for networked systems with average reward. *arXiv preprint arXiv:2006.06626*, 2020.

M. Roth, R. Simmons, and M. Veloso. Exploiting factored representations for decentralized execution in multiagent teams. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–7, 2007.

A. Schaerf, Y. Shoham, and M. Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *Journal of artificial intelligence research*, 2:475–500, 1994.

D. Shah, V. Somani, Q. Xie, and Z. Xu. On reinforcement learning for turn-based zero-sum markov games. *arXiv preprint arXiv:2002.10620*, 2020.

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

H. Thadakamaila, U. N. Raghavan, S. Kumara, and R. Albert. Survivability of multiagent-based supply networks: a topological perspect. *IEEE Intelligent Systems*, 19(5): 24–31, 2004.

P. Vamplew, J. Yearwood, R. Dazeley, and A. Berry. On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In *Australasian joint conference on artificial intelligence*, pages 372–378. Springer, 2008.

R. Wang, R. Salakhutdinov, and L. F. Yang. Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*, 2020.

X. F. Wang and T. Sandholm. Learning near-pareto-optimal conventions in polynomial time. 2003.

Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie. Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning. *IEEE Access*, 7:39974–39982, 2019.

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

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

Z. Yang, C. Jin, Z. Wang, M. Wang, and M. I. Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/9fa04f87c9138de23e92582b4ce549ec-Abstract.html.

T. Yoshikawa. Decomposition of dynamic team decision problems. *IEEE Transactions on Automatic Control*, 23 (4):627–632, 1978.

T. Yu, H. Wang, B. Zhou, K. Chan, and J. Tang. Multi-agent correlated equilibrium q (λ) learning for coordinated smart generation control of interconnected power grids. *IEEE transactions on power systems*, 30(4):1669–1679, 2014.

K. Zhang, Z. Yang, and T. Basar. Networked multi-agent reinforcement learning in continuous spaces. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2771–2776. IEEE, 2018a.

K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018b.

K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Y. Zhao, I. Borovikov, J. Rupert, C. Somers, and A. Beirami. On multi-agent learning in team sports games. *arXiv preprint arXiv:1906.10124*, 2019.

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

A. Related Work

It is difficult to summarize the rich literature on cooperative multi-agent reinforcement learning, being examined by various perspectives from the AI (Lauer and Riedmiller, 2000; Boutilier, 1996), control (Yoshikawa, 1978; Wang and Sandholm, 2003) and statistical learning communities (Xie et al., 2020). While there has been extensive recent work on provably efficient algorithms for *competitive* multiplayer RL (Xie et al., 2020; Zhao et al., 2021; Shah et al., 2020), our work is placed in the *cooperative* MARL setting, with the objective being to efficiently find *globally* optimal policies, where recent work has focused on *locality* assumptions in order to reduce the policy search space (Qu and Li, 2019; Qu et al., 2020). However, the more general *heterogeneous* reward setting considered in our work, where each agent may have unique rewards, corresponds to the *team average* games studied previously (Kar et al., 2013; Zhang et al., 2018b;a). While some of these approaches do provide tractable algorithms that are decentralized and convergent, none provide finite-time regret guarantees, and moreover, focus only on maximizing the *team average* reward. In our paper, however, we study a more general form of regret in order to recover a set of policies on the *Pareto frontier*. For a detailed overview of algorithms in cooperative MARL, we refer the readers to the illuminating survey by Zhang et al. (2019). Our work builds on the increasingly relevant line of work in (single-agent) reinforcement learning with function approximation (Wang et al., 2020; Yang et al., 2020; Yang and Wang, 2020; Jin et al., 2020), however, our environment suffers from several additional challenges not present in single-agent settings, such as communication costs, scalability issues and decentralized multi-agent planning, which are the key contributions of this paper.

B. Omitted Remarks

Remark 4 (Feasibility of Local Influence)**.** Networked influence assumptions similar to Assumption 1 have been explored extensively in the literature (Gu et al., 2020; Qu and Li, 2019; Qu et al., 2020; Guestrin et al., 2001), and is commonly present in many real-world environments such as supply-chain networks (Thadakamaila et al., 2004) and social networks (Barabasi, 2005). However, in contrast to prior work, which assume the individual reward functions to be functions only of the *local* state and action, we consider a broader model where even *local* rewards are functions of the neighborhood.

Remark 5 (Comparison with Exponential Decay)**.** Compared to the $(c, \rho)-$decay of (Qu and Li, 2019), our assumptions are both weaker and stronger in some aspects. First, we do not require any knowledge of *pairwise* interactions, and make assumptions at the subgraph level, and second, we do not require an exponential decay: simply an upper bound on the error suffices. Consequently, our guarantee only utilizes local neighborhoods (i.e., agents at distance 1), whereas $(c, \rho)-$exponential decay utilizes *all* interactions. In this regard, we remark that our clique-dominance assumption can incorporate further neighbors by partitioning the $\kappa-$power of G and introducing state-action communication between agents (as any agent can only observe its neighbors, hence information about distant neighbors must be communicated), which we omit for simplicity.

Remark 6 (Complexity of clique covering)**.** Assumption 2 requires a clique covering of G, which is NP-hard (Karp, 1972), however, for special cases, can be found in polynomial time (e.g., triangle-free graphs (Molloy, 2019) and perfect graphs (Grötschel et al., 1988)). Cerioli et al. (2008) provide a polynomial-time algorithm that gives a 1.25 approximation of the minimal clique covering, therefore, we can replace $\mathcal{C}^\star$ with an approximate covering $\widehat{\mathcal{C}}$ such that $|\widehat{\mathcal{C}}| \leq 1.25|\mathcal{C}^\star|$ for any G in our approach.

Remark 7 (Multi-agent modeling assumptions)**.** In contrast to the typical linear MDP assumption, here we model the rewards and dynamics for each clique of agents separately, each with d linear dimensions each. In the single-agent setting, identical assumptions on the reward and transition kernels will lead to a model with complexity d, whereas in our formulation we have a complexity of $2d$, implying that our fomulation incurs an overhead of $2\sqrt{2}$ in the regret if applied to the single-agent setting, compared to the model presented in Jin et al. (2020). Furthermore, observe that in the *fully-cooperative* setting (where agents share the reward function), i.e., $r_{1,h} = ... = r_{n,h} \forall h \in [H]$, we have that assuming, for all agents that $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_2 = ... = \boldsymbol{\phi}_n$ satisfies the modeling requirement.

Remark 8 (Regret Bound)**.** Theorem 2 claims in conjunction with Proposition 2 that **`MultOVI`** obtains Bayes regret of $\widetilde{\mathcal{O}}(\theta(G) \cdot \sqrt{T})$ even with limited communication. Note that for complete G, $\theta(G) = 1$ and the dependence on T matches that of MDP algorithms exactly (e.g., Jin et al. (2020)), demonstrating that our analysis is tight. Additionally, we see that this algorithm can easily be applied to an MDP by simply selecting $p_{\boldsymbol{\Upsilon}}$ to be a point mass at the appropriate $\boldsymbol{\upsilon}$, with no increase in regret. Thirdly, we see that **`MultOVI`** can be simulated on a single agent with n objectives, where $S = 1$ and G is complete, which provides, to the best of our knowledge, the first no-regret algorithm for multi-objective reinforcement learning (Mossalam et al., 2016).

Remark 9 (Lower Bounds)**.** For tabular multi-agent reinforcement learning, in the Appendix, we demonstrate that a collection of $\theta(G)$ episodic MDPs (one corresponding to each clique $C \in \mathcal{C}$, with state-action spaces $\mathcal{S}_C$ and $\mathcal{A}_C$) can be constructed such that the cumulative Bayes regret incurred by any algorithm is $\Omega(H\sqrt{dT})$ for each C, and therefore the total regret in the Markov game is $\Omega(\theta(G)H\sqrt{dT})$ where $n_{\min}$ is the size of the smallest clique of G, demonstrating that the $\theta(G)$ term is unavoidable in general. Further, the utilization of "Bernstein-type" confidence bonuses can shave off an additional factor of $\sqrt{H}$ in our regret (see discussion in Jin et al. (2020)). Regarding the optimal dependence on communication, we conjecture that our bound is almost-optimal, as similar lower bounds have been demonstrated for distributed exploration in multi-armed bandits (Hillel et al., 2013).

Remark 10 (Modeling influence and unknown dynamics)**.** For arbitrary influence graphs G, the misspecification ε incurred by using a k-clique covering $\mathcal{C}$ of G (Assumption 2) can be unknown in general, and may be unique for each $\mathcal{C}$. In this setting, we conjecture that a corraling-type algorithm (Pacchiano et al., 2020; Agarwal et al., 2017) that adaptively selects the best clique covering $\mathcal{C}$ can provide regret close to our algorithm without knowing the misspecification $\varepsilon(k)$.

Remark 11 (Communication complexity)**.** We can control the communication budget by adjusting the threshold parameter S. Note that when $S = 1$, communication will occur each round, as the threshold will be satisfied trivially by the rank-1 update to the Gram matrix. If the horizon T is known in advance, one can set $S = (1 + n_{\max}T/d)^{1/D}$ for some independent constant $D > 1$, to ensure that the total rounds of communication is a fixed constant $\theta(G)(dD + 1)H$, which provides us a group regret of $\widetilde{\mathcal{O}}(\theta(G) \cdot n^{\frac{1}{2D}} \cdot T^{\frac{1}{2}+\frac{1}{2D}})$. A balance can be obtained by setting $S = C'$ for some absolute constant C', leading to a total $\mathcal{O}(\theta(G) \cdot \log(n_{\max}T))$ rounds with $\widetilde{\mathcal{O}}(\theta(G)\sqrt{T})$ regret.

C. Algorithm Design: Extended

The first step in our approach is to compute a k-clique covering $\mathcal{C}$ of the influence graph G. Recall that by Remark 6 that this can be done in polynomial time with a 1.25 approximation of $\mathcal{C}^\star$. Since the game is clique-dominant (Assumption 2), we can learn k decentralized policies $\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_k$, one corresponding to each clique of agents in $\mathcal{C}$ without incurring too much approximation error. Now, to motivate the design, we first observe that Assumption 3 implies that for each clique $C \in \mathcal{C}$, there exist a set of weights such that the scalarized $Q-$values for any parameter $\boldsymbol{\upsilon}_C$ are *almost* linear projections of the overall clique features $\boldsymbol{\Phi}_C(\cdot)$, where the total error is no larger than $2H\varepsilon(k)$.

Lemma 2 (Almost linear weights in Markov Games)**.** *Under Assumption 3 for graph G with k cliques ordered from $1, ..., k$, we have, for any fixed decentralized policy $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_k\}$ and $\boldsymbol{\upsilon} = \{\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_k\} \in \boldsymbol{\Upsilon}$, there exist weights $\{\boldsymbol{w}^{\boldsymbol{\pi}_\tau}_{\boldsymbol{\upsilon},h}\}_{h=1,\tau=1}^{H,k}$ such that $\left|Q^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},h}(\boldsymbol{x}, \boldsymbol{a}) - \sum_{\tau=1}^{k} \boldsymbol{\upsilon}_\tau^\top \boldsymbol{\Phi}_\tau(\boldsymbol{x}_\tau, \boldsymbol{a}_\tau)^\top \boldsymbol{w}^{\boldsymbol{\pi}_\tau}_{\boldsymbol{\upsilon},h}\right| \leq 2H\varepsilon(k)\ \forall(\boldsymbol{x}, \boldsymbol{a}, h)$, where $\|\boldsymbol{w}^{\boldsymbol{\pi}_k}_{\boldsymbol{\upsilon},h}\|_2 \leq 2H\sqrt{d}$.*

Armed with this observation, we design a policy using *vector-valued* linear least-squares regression, as the optimal policy is only at most $2H\varepsilon$ away from the best least-squares fit. In a nutshell, our approach can be summed up in two steps: (a) first, we approximate the Pareto frontier $\boldsymbol{\Pi}^\star$ with the set of policies $\boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$ recoverable by scalarization (see Remark 2), (b) next, we empirically approximate $\boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$ with a collection of T policies (one for each episode), such that the *Bayes Regret* is minimized (Proposition 2). In each episode $t \in [T]$, we sample a scalarization parameter $\boldsymbol{\upsilon}_t \sim p_{\boldsymbol{\Upsilon}}$, and run k *vector-valued* decentralized linear least-squares regressions to approximate the optimal policy $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}_t}$ with k policies $\boldsymbol{\pi}_1(t), ..., \boldsymbol{\pi}_k(t)$ such that the resulting $Q-$values overestimate $Q^\star_{\boldsymbol{\upsilon}_t,h}$ with high probability. Then, each agent in clique τ selects the corresponding greedy action with respect to $\boldsymbol{\pi}_\tau(t)$. This approach is carried out via *value iteration with optimism*, as described below.

We describe the policy for any clique $C \in \mathcal{C}$ of size n_C. For any scalarization $\boldsymbol{\upsilon}(t) \in \mathbb{R}^n$, the n_C values corresponding to agents in $\mathcal{C}$ is denoted by $\boldsymbol{\upsilon}_C(t)$. Now, consider the MDP $\widetilde{\text{MDP}}_C$ formed by scalarizing the Markov game corresponding to the approximate rewards $\widetilde{\boldsymbol{r}}^C_h$ and transition dynamics $\widetilde{\mathbb{P}}^C_h$ with the parameter $\boldsymbol{\upsilon}_{C,t}$ (i.e., the reward function in $\widetilde{\text{MDP}}_C$ is given by $\boldsymbol{\upsilon}_C(t)^\top \boldsymbol{r}^C_h$, transition by $\widetilde{\mathbb{P}}_h$ and state-action spaces as $\mathcal{S}_C$ and $\mathcal{A}_C$ respectively). For each clique C, we will use value iteration to recover the optimal policy for this $\widetilde{\text{MDP}}_C$ (let us call it $\widetilde{\boldsymbol{\pi}}^\star_C(t)$). The algorithm is a distributed variant of least-squares value iteration with UCB exploration. Following Proposition 1, the key idea is to make sure that each agent in C acts according to the joint policy that is aiming to mimic $\widetilde{\boldsymbol{\pi}}^\star_C(t)$. Therefore, we must ensure that the local estimate for the *joint* policy obtained by any agent must be identical, such that the *joint* action is in accordance with $\widetilde{\boldsymbol{\pi}}^\star_C(t)$. To achieve this we will obtain the approximated (scalar) $Q-$values for $\widetilde{\boldsymbol{\pi}}^\star_C(t)$ by recursively applying the Bellman equation and solving the resulting equations via a *vector-valued* regression. Since the policy variables are designed to be identical each agent in C at all times, we describe the procedure for any agent within C.

For any episode t, let us assume that the last round of synchronization between agents in C occured at time s^C_t. Each

Algorithm 1 `MultOVI`: Decentralized Learning in Low-Rank Cooperative Markov Games

Input: $T, \mathbf{\Phi}, H, S$, sequence $\beta_h = \{(\beta_h^t)_t\}$.
Initialize: $\mathbf{\Lambda}_h^C(t) = \lambda \mathbf{I}_d, \delta\mathbf{\Lambda}_C^h(t) = \mathbf{0}, \mathcal{U}_\nu^h, \mathcal{W}_\nu^h = \emptyset$ for each $\nu \in G$, clique cover $\widehat{\mathcal{C}}$ of G.
for episode $t = 1, 2, ..., T$ **do**
  Sample $\boldsymbol{\upsilon}_t \sim p_{\boldsymbol{\Upsilon}}$ using public randomness.
  for clique $C \in \widehat{\mathcal{C}}$ **do**
    for agent $\nu \in C$ **do**
      Set $V_C^{H+1}(t)(\cdot) \leftarrow 0$.
      for step $h = H, ..., 1$ **do**
        Compute $Q_C^h(t)(\cdot, \cdot)$ using vector-valued least-squares regression on $\mathcal{U}_\nu^h$.
        Set $V_C^{h+1}(t)(\cdot) \leftarrow \max_{\boldsymbol{a} \in \mathcal{A}_C} Q_C^h(t)(\cdot, \boldsymbol{a})$.
      end for
      for step $h = 1, ..., H$ **do**
        Take action $a_\nu^h(t) \leftarrow [\arg\max_{a \in \mathcal{A}_C} Q_C^h(t)(\boldsymbol{x}_C^h(t), \boldsymbol{a})]_\nu$.
        Observe $r_\nu^h(t), \widetilde{\boldsymbol{x}}_\nu^{h+1}$.
        Update $\delta\mathbf{\Lambda}_C^h(t) \leftarrow \delta\mathbf{\Lambda}_C^h(t-1) + \mathbf{\Phi}_C(\boldsymbol{z}_C^h(t))\mathbf{\Phi}_C(\boldsymbol{z}_C^h(t))^\top$.
        Update $\mathcal{W}_\nu^h \leftarrow \mathcal{W}_\nu^h \cup (\nu, x_\nu^h(t), r_\nu^h(t))$.
        if $\log \frac{\det(\mathbf{\Lambda}_C^h(t) + \delta\mathbf{\Lambda}_C^h(t) + \lambda\mathbf{I})}{\det(\mathbf{\Lambda}_C^h(t) + \lambda\mathbf{I})} > S$ **then**
          SYNCHRONIZE$\leftarrow$ TRUE.
        end if
      end for
    end for
    if SYNCHRONIZE **then**
      Assign arbitrary agent in C as the SERVER AGENT.
      for step $h = H, ..., 1$ **do**
$\forall$ AGENTS Send $\mathcal{W}_\nu^h \rightarrow$SERVER AGENT.
SERVER AGENT Aggregate $\mathcal{W}^h \leftarrow \cup_{\nu \in C} \mathcal{W}_\nu^h$.
SERVER AGENT Communicate $\mathcal{W}^h$ to each agent.
$\forall$ AGENTS Set $\delta\mathbf{\Lambda}_C^h(t+1) \leftarrow 0, \mathcal{W}_\nu^h \leftarrow \emptyset$.
$\forall$ AGENTS Set $\mathbf{\Lambda}_C^h(t+1) \leftarrow \mathbf{\Lambda}_C^h(t) + \sum_{(\boldsymbol{x}, \boldsymbol{a}) \in \mathcal{W}^h} \mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})\mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})^\top$.
$\forall$ AGENTS Set $\mathcal{U}_\nu^h \leftarrow \mathcal{U}_\nu^h \cup \mathcal{W}^h$
      end for
    end if
  end for
end for

agent within the clique C obtains an *identical* sequence of value functions $\{Q_C^h(t)\}_{h \in [H]}$ by iteratively performing linear least-squares ridge regression from the history available from the previous s_t^C episodes by first learning a vector $Q-$function $\widehat{\mathbf{Q}}_C^h(t)$ over $\mathbb{R}^{n_C}$, which is scalarized by $\boldsymbol{\upsilon}_C(t)$ to obtain the $Q-$values as $Q_C^h(t) = \boldsymbol{\upsilon}_C(t)^\top \widehat{\mathbf{Q}}_C^h(t)$. Each agent m first sets $\widehat{\mathbf{Q}}_C^{H+1}(t)$ to be a zero vector in $\mathbb{R}^{n_C}$, and for any $h \in [H]$, solves the following sequence of regressions to obtain $Q-$values. For each $h = H, ..., 1$, for each agent computes,

$$V_C^{h+1}(t)(\boldsymbol{x}) \leftarrow \arg\max_{\boldsymbol{a} \in \mathcal{A}} \left[\boldsymbol{\upsilon}_C(t)^\top \left(\mathbf{Q}_C^{h+1}(t)^\top \mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})\right)\right] \ \forall\, \boldsymbol{x} \in \mathcal{S}_C,$$

$$\widehat{\mathbf{Q}}_C^h(t) \leftarrow \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \left[\sum_{\tau \in [s_t^C]} \left\|\boldsymbol{y}_C^h(\tau) - \mathbf{\Phi}_C(\boldsymbol{x}_C^h(\tau), \boldsymbol{a}_C^h(\tau))^\top \boldsymbol{w}\right\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2\right],$$

$$Q_C^h(t)(\boldsymbol{x}, \boldsymbol{a}) \leftarrow \boldsymbol{\upsilon}_C(t)^\top \left(\widehat{\mathbf{Q}}_C^h(t)^\top \mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})\right) + \beta_C^h(t) \cdot \left\|\mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})^\top \mathbf{\Lambda}_C^h(t)^{-1} \mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})\right\|_2.$$

Where the last equation holds for any $(\boldsymbol{x}, \boldsymbol{a}) \in (\mathcal{S}_C \times \mathcal{A}_C)$ and the targets $\boldsymbol{y}_C^h(\tau) = \boldsymbol{r}_h^C(\boldsymbol{x}_C^h(\tau), \boldsymbol{a}_C^h(\tau)) + \mathbf{1}_{n_C} \cdot V_C^{h+1}(t)(\boldsymbol{x}_C^{h+1}(\tau))$, $\mathbf{1}_{n_C}$ denotes the all-ones vector in $\mathbb{R}^{n_C}$, $\beta_C^h(t)$ is selected such that with high probability the estimated Q-values overestimate the require $Q-$values, and $\mathbf{\Lambda}_C^h(t)$ is described subsequently. Once all of these quantities are computed, each agent $\nu \in C$ selects the action $a_\nu^h(t) = \left[\arg\max_{\boldsymbol{a} \in \mathcal{A}_C} Q_C^h(t)(\boldsymbol{x}_C^h(t), \boldsymbol{a})\right]_\nu$ for each $h \in [H]$. Hence, the joint clique action $\boldsymbol{a}_C^h(t) = \{a_\nu^h(t)\}_{\nu=1}^{n_C} = \arg\max_{\boldsymbol{a} \in \mathcal{A}_C} Q_C^h(\boldsymbol{x}_C^h(t), \boldsymbol{a})$. Observe that while the computation of the policy is decentralized, the policies executed for all agents $\nu \in C$ coincide at all times by the modeling assumption and the periodic synchronizations between agents. We now present the closed form of $\widehat{\mathbf{Q}}_C^h(t)$. Consider the contraction

$\boldsymbol{z}_C^h(\tau) = (\boldsymbol{x}_C^h(\tau), \boldsymbol{a}_C^h(\tau))$ and the map $\widehat{\boldsymbol{\Phi}}_C^h(t) : \mathbb{R}^d \to \mathbb{R}^{tn_C}$ such that for any $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\boldsymbol{\Phi}_C^h(t)\boldsymbol{\theta} \triangleq \left[(\boldsymbol{\Phi}_C(\boldsymbol{z}_C^h(1))^\top \boldsymbol{\theta})^\top, ..., (\boldsymbol{\Phi}(\boldsymbol{z}_C^h(t))^\top \boldsymbol{\theta})^\top\right]^\top.$$

Now, consider $\boldsymbol{\Lambda}_C^h(t) = \boldsymbol{\Phi}_C^h(s_t^C)^\top \boldsymbol{\Phi}_C^h(s_t^C) + \lambda \mathbf{I}_d \in \mathbb{R}^{d\times d}$, and $\mathbf{U}_C^h(t) = \sum_{\tau=1}^{s_t^C} \boldsymbol{\Phi}_C(\boldsymbol{z}_C^h(\tau))\boldsymbol{y}_C^h(\tau)$. Then, we have by a multi-task concentration (see Appendix B of Chowdhury and Gopalan (2020)),

$$\widehat{\mathbf{Q}}_C^h(t)(\boldsymbol{x}, \boldsymbol{a}) = \boldsymbol{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})^\top \boldsymbol{\Lambda}_C^h(t)^{-1} \mathbf{U}_C^h(t).$$

The algorithm is presented in Algorithm 1. The algorithm is essentially learning k multi-agent policies by solving a vector-valued regression, one for each clique in the covering $\mathcal{C}$, such that the group of agents in each clique can learn the approximate clique-based MG (ref. Assumption 2). Since these approximate games themselves are bounded close to the true Markov game (by clique-dominance), this ensures that the agents incur low regret. We next present an analysis of communication cost.

Communication. Note that within a clique, the common state $\boldsymbol{x}_C$ is visible to all agents (Assumption 1), and hence the agents only require communication of rewards within a clique. To limit rounds in which communication occurs, we consider a synchronization criterion that is triggered whenever any agent in the clique explores a sufficiently novel part of the environment. Specifically, whenever $\det(\boldsymbol{\Lambda}_C^h(t)) \geq S \cdot \det(\boldsymbol{\Lambda}_C^h(s_t^C))$, for any $h \in [H]$ where S is a fixed constant determined in advance, the agents synchronize their rewards within their correponding clique C. The synchronization can be done in $\mathcal{O}(n)$ messages by designating one agent per clique as the SERVER to aggregate messages.

Lemma 3 (Communication complexity)**.** *Let the clique covering number be $\theta(G)$ and let $n_{\max} \leq n$ denote the size of the largest clique of G. If we use threshold $S > 1$, then the total number of episodes with communication $\gamma \leq dH \cdot \theta(G) \cdot \log_S\left(1 + \frac{Tn_{\max}}{d}\right)) + \theta(G) \cdot H$. When $S \leq 1$, $\gamma = T$.*

D. Full Proofs

D.1. Proof of Proposition 1

We restate the Proposition for clarity.

Proposition 1. For the scalarized value function, the Bellman optimality conditions are given as, for all $h \in [H], \boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}$ for any fixed $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$,

$$Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}) = \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_h(\boldsymbol{x}, \boldsymbol{a}) + \mathbb{P}_h V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}), V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}) = \max_{\boldsymbol{a}\in\mathcal{A}} Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}), \text{ and } V_{\boldsymbol{\upsilon},H+1}^\star(\boldsymbol{x}) = 0.$$

Proof. We prove the above result by reducing the scalarized MMDP to an equivalent MDP. Observe that for any fixed $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$, the (vector-valued) rewards can be scalarized to a scalar reward. For any step $h \in [H]$, for any *fixed* $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$, consider the MDP with state space $\mathcal{S} = \mathcal{S}_1 \times ... \times \mathcal{S}_n$, action space $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_n$ and reward function r_h' such that for all $(\boldsymbol{x}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}, r_h'(\boldsymbol{x}, \boldsymbol{a}) = \boldsymbol{\upsilon}^\top \boldsymbol{r}_h(\boldsymbol{x}, \boldsymbol{a})$. Therefore $r_h'(\boldsymbol{x}, \boldsymbol{a}) \in [0, 1]$ (since $\boldsymbol{r}_h$ lies on the $n-$dimensional simplex). Therefore, if the group of agents cooperate to optimize the scalarized reward (for any *fixed* scalarization parameter), the optimal (joint) policy coincides with the optimal policy for the aforementioned MDP defined over the *joint* state and action spaces. The optimal policy for the scalarized MDP is given by the greedy policy with respect to the following parameters:

$$Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}) = r_h'(\boldsymbol{x}, \boldsymbol{a}) + \mathbb{P}_h V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}), V_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}) = \max_{\boldsymbol{a}\in\mathcal{A}} Q_{\boldsymbol{\upsilon},h}^\star(\boldsymbol{x}, \boldsymbol{a}), \text{ and } V_{\boldsymbol{\upsilon},H+1}^\star(\boldsymbol{x}) = 0. \tag{3}$$

Replacing the reward function with the vector-valued reward in terms of $\boldsymbol{\upsilon}$ provides us the result. □

D.2. Proof of Theorem 1

We first restate the Theorem for clarity.

Theorem 1. For any Markov game with finite $\mathcal{A}$ and H, $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star \subseteq \boldsymbol{\Pi}^\star$. If $\boldsymbol{\Pi}^\star$ is convex, $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star = \boldsymbol{\Pi}^\star$.

Proof. First, we prove the forward direction, i.e., that $\boldsymbol{\Pi}_{\boldsymbol{\Upsilon}}^\star \subseteq \boldsymbol{\Pi}^\star$. The proof proceeds by contradiction. Assume that $\boldsymbol{\pi}_{\boldsymbol{\upsilon}}^\star$ does not lie in the Pareto frontier, then there exists a policy $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ such that $\mathbf{V}_1^{\boldsymbol{\pi}'}(\boldsymbol{x}) \succeq \mathbf{V}_1^{\boldsymbol{\pi}_{\boldsymbol{\upsilon}}^\star}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$ and $\boldsymbol{\pi} \neq \boldsymbol{\pi}_{\boldsymbol{\upsilon}}^\star$.

Consider the final step H. Then, for any state $\boldsymbol{x} \in \mathcal{S}$, we have that if $\mathbf{V}_H^{\boldsymbol{\pi}'}(\boldsymbol{x}) \succeq \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x})$, then,

$$\boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{\pi}'(\boldsymbol{x})) \succeq \boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) \implies \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{\pi}'(\boldsymbol{x})) \geq \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})). \tag{4}$$

However, this is only true with equality if $\boldsymbol{\pi}'(\boldsymbol{x}) = \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$, as for any $\boldsymbol{x} \in \mathcal{S}$, $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}(\boldsymbol{x}) = \arg\max[\mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{a})] \geq \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_H(\boldsymbol{x}, \boldsymbol{a}')$ for any other $\boldsymbol{a}' \in \mathcal{A}$. Therefore, we have that $\boldsymbol{\pi}'_H(\boldsymbol{x}) = \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{S}$, and that $\mathbf{V}_H^{\boldsymbol{\pi}'}(\boldsymbol{x}) = \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x})$. This implies that $\mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}'}(\boldsymbol{x}, \boldsymbol{a}) = \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}, \boldsymbol{a})$ for all $\boldsymbol{x} \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$. Now, if $\mathbf{V}_{H-1}^{\boldsymbol{\pi}'}(\boldsymbol{x}) \succeq \mathbf{V}_{H-1}^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x})$, then we have that,

$$\begin{aligned}
&\boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{x}' \sim \mathbb{P}_H(\cdot|\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x}))} \left[\mathbf{V}_H^{\boldsymbol{\pi}'}(\boldsymbol{x}') \right] \\
&\qquad \succeq \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) \\
&\implies \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{x}' \sim \mathbb{P}_H(\cdot|\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x}))} \left[\mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}') \right] \\
&\qquad \succeq \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) \\
&\implies \mathfrak{s}_{\boldsymbol{\upsilon}} \left(\boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{x}' \sim \mathbb{P}_H(\cdot|\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x}))} \left[\mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}') \right] \right) \\
&\qquad \geq \mathfrak{s}_{\boldsymbol{\upsilon}} \left(\boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) \right) \\
&\implies \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{x}' \sim \mathbb{P}_H(\cdot|\boldsymbol{x}, \boldsymbol{\pi}'_{H-1}(\boldsymbol{x}))} \left[\mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}') \right] \\
&\qquad \geq \mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x}, \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}(\boldsymbol{x})).
\end{aligned}$$

This is true only if $\boldsymbol{\pi}'_{H-1}(\boldsymbol{x}) = \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{S}$, as $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}$ is the greedy policy with respect to $\mathfrak{s}_{\boldsymbol{\upsilon}} \boldsymbol{r}_{H-1}(\boldsymbol{x}, \boldsymbol{a}) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}}(\boldsymbol{x}, \boldsymbol{a})$. Continuing this argument inductively for $h = H-2, H-3, ..., 1$ we obtain that $\mathbf{V}_1^{\boldsymbol{\pi}'}(\boldsymbol{x}) \succeq \mathbf{V}_1^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{S}$ only if $\boldsymbol{\pi}' = \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}$. This is a contradiction as we assumed that $\boldsymbol{\pi}' \neq \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}$, and hence $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}$ lies in $\boldsymbol{\Pi}^\star$.

We now prove the other direction for convex $\boldsymbol{\Pi}^\star$, i.e., that if $\boldsymbol{\Pi}^\star$ is convex, then $\boldsymbol{\Pi}^\star \subseteq \boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$ for $\boldsymbol{\Upsilon} = \Delta^n$. This proof proceeds by contradiction as well. Let us assume that there exists a policy $\boldsymbol{\pi}$ in $\boldsymbol{\Pi}^\star$ that is not present in $\boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$. Therefore, there does not exist any $\boldsymbol{\upsilon} \in \Delta^n$ such that $\boldsymbol{\pi}$ maximizes the value function of the scalarized MDP. Alternatively stated, for each $\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon}$, there exists another policy $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}} \in \boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$ such that $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}} \neq \boldsymbol{\pi}$ and it maximizes the scalarized value function $\mathbf{V}^\star_{\boldsymbol{\upsilon},1}$. Now, observe that since $\boldsymbol{\pi} \in \boldsymbol{\Pi}^\star$, it must be that for all $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$, $\mathbf{V}_1^{\boldsymbol{\pi}} \succeq \mathbf{V}_1^{\boldsymbol{\pi}'}$. Additionally, since $\boldsymbol{\Pi}^\star$ is convex and the scalarization function $\boldsymbol{\upsilon}^\top(\cdot)$ is linear, each scalarization function $\mathfrak{s}_{\boldsymbol{\upsilon}}(\cdot)$ for $\boldsymbol{\upsilon} \in \Delta^n$ is convex over $\boldsymbol{\Pi}^\star$. Therefore, each policy that maximizes the scalarized value function corresponding to any $\boldsymbol{\upsilon}$ is a global optimum in $\boldsymbol{\Pi}^\star$.

Now, consider the scalarization $\boldsymbol{\upsilon}_\star$ where $[\boldsymbol{\upsilon}_\star]_i = \frac{[\mathbf{V}_1^{\boldsymbol{\pi}}]_i}{\|\mathbf{V}_1^{\boldsymbol{\pi}}\|_1} \in \Delta^n$. Now, by our assumption, there must exist an alternative policy $\boldsymbol{\pi}' \neq \boldsymbol{\pi}$ in $\boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$, such that (by the convexity of scalarization), $\boldsymbol{\upsilon}_\star^\top(\mathbf{V}_1^{\boldsymbol{\pi}'} - \mathbf{V}_1^{\boldsymbol{\pi}}) \leq 0$. This implies that $[\mathbf{V}_1^{\boldsymbol{\pi}}]_i^2 \leq [\mathbf{V}_1^{\boldsymbol{\pi}'}]_i [\mathbf{V}_1^{\boldsymbol{\pi}}]_i \implies [\mathbf{V}_1^{\boldsymbol{\pi}'}]_i \geq [\mathbf{V}_1^{\boldsymbol{\pi}}]_i \implies \mathbf{V}_1^{\boldsymbol{\pi}'} \succeq \mathbf{V}_1^{\boldsymbol{\pi}}$. This is a contradiction as $\boldsymbol{\pi}$ is Pareto-optimal, and hence $\boldsymbol{\pi} \in \boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$. □

D.3. Proof of Proposition 2

We first restate Proposition 2 for clarity.

Proposition 2. For any scalarization $\mathfrak{s}$ that is Lipschitz and bounded $\boldsymbol{\Upsilon}$, we have that $\mathfrak{R}_B(T) \leq \frac{1}{T}\mathfrak{R}_C(T) + o(1)$.

Proof. We will follow the approach in (Paria et al., 2020) (Appendix B.3). Recall that $\boldsymbol{\Upsilon}$ is a bounded subset of $\mathbb{R}^n$. Now, we have that since $\mathfrak{s}_{\boldsymbol{\upsilon}}(\cdot) = \boldsymbol{\upsilon}^\top(\cdot)$, we have that $\mathfrak{s}_{\boldsymbol{\upsilon}}$ is Lipschitz with constant n with respect to the ℓ_1-norm, i.e., for any $\boldsymbol{y} \in \mathbb{R}^n$,

$$|\mathfrak{s}_{\boldsymbol{\upsilon}}(\boldsymbol{y}) - \mathfrak{s}_{\boldsymbol{\upsilon}'}(\boldsymbol{y})| \leq n \|\boldsymbol{\upsilon} - \boldsymbol{\upsilon}'\|_1. \tag{5}$$

Now, consider the Wasserstein distance conditioned on the history $\mathcal{H}$ between the sampling distribution $p_{\boldsymbol{\Upsilon}}$ on $\boldsymbol{\Upsilon}$ and the empirical distribution $\widehat{p}_{\boldsymbol{\Upsilon}}$ corresponding to $\{\boldsymbol{\upsilon}_t\}_{t=1}^T$,

$$W_1(p_{\boldsymbol{\Upsilon}}, \widehat{p}_{\boldsymbol{\Upsilon}}) = \inf_q \left\{ \mathbb{E}_q \|X - Y\|_1, q(X) = p_{\boldsymbol{\Upsilon}}, q(Y) = \widehat{p}_{\boldsymbol{\Upsilon}} \right\}, \tag{6}$$

where q is a joint distribution on the RVs X, Y with marginal distributions equal to $p_{\boldsymbol{\Upsilon}}$ and $\widehat{p}_{\boldsymbol{\Upsilon}}$. We therefore have for some randomly drawn samples $\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}_2, ..., \boldsymbol{\upsilon}_T$ and for any arbitrary sequence of (joint) policies $\widehat{\boldsymbol{\Pi}}_T = \{\boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_T\}$, for any state $\boldsymbol{x} \in \mathcal{S}$,

$$\frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\upsilon}\in\boldsymbol{\Upsilon}} \left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\boldsymbol{x}) \right] \right] \tag{7}$$

$$\leq \frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\upsilon}\in\boldsymbol{\Upsilon}} \left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\boldsymbol{x}) \right] \right] \tag{8}$$

$$\leq \mathbb{E}_{q(X,Y)} \left[\max_{\boldsymbol{x}\in\mathcal{S}} \left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{X,1}(\boldsymbol{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{Y,1}(\boldsymbol{x}) \right] \right] \tag{9}$$

$$\leq n \cdot \mathbb{E}_{q(X,Y)} \left[\|X - Y\|_1 \right]. \tag{10}$$

Taking an expectation with respect to $\mathcal{H} = \{\boldsymbol{\upsilon}_1, .., \boldsymbol{\upsilon}_T\}$, we have,

$$\mathfrak{R}_B(T) - \frac{1}{T}\mathfrak{R}_C(T) \tag{11}$$

$$= \mathbb{E}_{\boldsymbol{\upsilon}\in\boldsymbol{\Upsilon}} \left[\max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon},1}(\boldsymbol{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\boldsymbol{x}) \right] \right] - \mathbb{E}_{\mathcal{H}} \left[\frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) \right] \right] \tag{12}$$

$$= \mathbb{E}_{\mathcal{H}} \left[\frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) \right] \right] - \mathbb{E}_{\mathcal{H}} \left[\frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) \right] \right] \tag{13}$$

$$\leq \mathbb{E}_{\mathcal{H}} \left[\frac{1}{T}\sum_{t=1}^{T} \max_{\boldsymbol{x}\in\mathcal{S}} \left[V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\upsilon}\in\boldsymbol{\Upsilon}} \left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\boldsymbol{x}) \right] \right] \right] \tag{14}$$

$$\leq n \cdot \mathbb{E}_{q(X,Y)} \left[\|X - Y\|_1 \right]. \tag{15}$$

The penultimate inequality follows from $\max$ being a contraction mapping in bounded domains, and the final inequality follows from the previous analysis. To complete the proof, we first take an infimum over q and observe that the subsequent RHS converges at a rate of $\widetilde{\mathcal{O}}(T^{-\frac{1}{n}})$ under mild regulatory conditions, as shown by Canas and Rosasco (2012). □

D.4. Proof of Lemma 2

Follows from Lemma 11.

D.5. Proof of Theorem 2

The proof for this result is to essentially solve the approximate scalarized MDP for each clique We first present a vector-valued concentration result.

Lemma 4. *Select any clique C in a clique covering $\widehat{\mathcal{C}}$ such that $|C| = M$. For any $m \in [M], h \in [H]$ and $t \in [T]$, let k_t denote the episode after which the last local synchronization has taken place, and $\mathbf{S}^h_C(t)$ and $\boldsymbol{\Lambda}^h_C(t)$ be defined as follows.*

$$\mathbf{S}^h_C(t) = \sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) \left[\boldsymbol{v}^t_{\boldsymbol{\upsilon},h+1}(\boldsymbol{x}^{h+1}_C(\tau)) - (\widetilde{\mathbb{P}}^C_h \boldsymbol{v}^t_{\boldsymbol{\upsilon},h+1})(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) \right],$$
$$\boldsymbol{\Lambda}^h_C(t) = \lambda \mathbf{I}_d + (\boldsymbol{\Phi}^h_C(k_t))^\top (\boldsymbol{\Phi}^h_C(k_t)).$$

Where $\boldsymbol{v}^t_{\boldsymbol{\upsilon},h+1}(\boldsymbol{x}) = \mathbf{1}_M \cdot V^t_{\boldsymbol{\upsilon},h+1}(\boldsymbol{x}) \; \forall \; \boldsymbol{x} \in \mathcal{S}_C$, $\mathbf{1}_M$ denotes the all-ones vector in $\mathbb{R}^M$, and C_β is the constant such that $\beta^h_C(t) = C_\beta \cdot dH\sqrt{\log(TMH)}$. Then, there exists a constant B such that with probability at least $1 - \delta$,

$$\sup_{\boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C} \left\| \mathbf{S}^h_C(t) \right\|_{(\boldsymbol{\Lambda}^t_h)^{-1}} \leq B \cdot dH \sqrt{2 \log\left(\frac{(C_\beta + 2) dMTH}{\delta'} \right)}.$$

Proof. The proof is done in two steps. The first step is to bound the deviations in $\mathbf{S}$ for any fixed function V by a martingale concentration. The second step is to bound the resulting concentration over all functions V by a covering argument. Finally, we select appropriate constants to provide the form of the result required.

Step 1. Recall that $\mathbf{S}_C^h(t) = \sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau))[V_{\boldsymbol{v},h+1}^t(\boldsymbol{x}_C^{h+1}(\tau)) - (\widetilde{\mathbb{P}}_h^C V_{\boldsymbol{v},h+1}^t)(\boldsymbol{z}_C^h(\tau))]$, where $\boldsymbol{v}_{\boldsymbol{v},h+1}^t$ is the vector with each entry being $V_{\boldsymbol{v},h+1}^t$. We have that $V_{\boldsymbol{v},h+1}^t(\boldsymbol{x}_C^{h+1}(\tau)) - (\widetilde{\mathbb{P}}_h^C V_{\boldsymbol{v},h+1}^t)(\boldsymbol{z}_C^h(\tau)) = \boldsymbol{v}_{\boldsymbol{v},h+1}^t - \widetilde{\mathbb{P}}_h^C \boldsymbol{v}_{\boldsymbol{v},h+1}^t$. Consider the following distance metric $\text{dist}_{\mathbf{\Upsilon}_C}$,

$$\text{dist}_{\mathbf{\Upsilon}_C}(\boldsymbol{v}, \boldsymbol{v}') = \sup_{\boldsymbol{x} \in \mathcal{S}_C, \boldsymbol{v} \in \mathbf{\Upsilon}_C} \|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}(\boldsymbol{x}')\|_1 . \tag{16}$$

Let $\mathcal{V}_{\mathbf{\Upsilon}_C}$ be the family of all vector-valued UCB value functions that can be produced by the algorithm on clique C, and now let $\mathcal{N}_\epsilon$ be an $\epsilon-$covering of $\mathcal{V}_{\mathbf{\Upsilon}_C}$ under $\text{dist}_{\mathbf{\Upsilon}_C}$, i.e., for every $\boldsymbol{v} \in \mathcal{V}_{\mathbf{\Upsilon}}$, there exists $\boldsymbol{v}' \in \mathcal{N}_\epsilon$ such that $\text{dist}_{\mathbf{\Upsilon}_C}(\boldsymbol{v}, \boldsymbol{v}') \leq \epsilon$. Now, here again, we adopt a similar strategy as the independent case. To bound the RHS, we decompose $\mathbf{S}_C^h(t)$ in terms of the covering described earlier. We know that since $\mathcal{N}_\epsilon$ is an $\epsilon-$covering of $\mathcal{V}_{\mathbf{\Upsilon}_C}$, there exists a $\boldsymbol{v}' \in \mathcal{N}_\epsilon$ and $\mathbf{\Delta} = \boldsymbol{v}_{\boldsymbol{v},h+1}^t - \boldsymbol{v}'$ such that,

$$\mathbf{S}_C^h(t) \;=\; \sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \left[\boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) - \widetilde{\mathbb{P}}_h^C \boldsymbol{v}'(\boldsymbol{z}_C^h(\tau))\right] \;+\; \sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \left[\mathbf{\Delta}(\boldsymbol{x}_C^{h+1}(\tau)) - \widetilde{\mathbb{P}}_h^C \mathbf{\Delta}(\boldsymbol{z}_C^h(\tau))\right].$$

Now, observe that by the definition of the covering, we have that $\|\mathbf{\Delta}\|_1 \leq \epsilon$. Therefore, we have that $\|\mathbf{\Delta}(\boldsymbol{x})\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq \epsilon/\sqrt{\lambda}$, and $\left\|\widetilde{\mathbb{P}}_h^C \mathbf{\Delta}(\boldsymbol{z})\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq \epsilon/\sqrt{\lambda}$ for all $\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathcal{S}, h \in [H]$. Therefore, since $\|\mathbf{\Phi}_C(\boldsymbol{z})\|_2 \leq \sqrt{M}$,

$$\left\|\mathbf{S}_C^h(t)\right\|^2_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq 2 \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \left[\boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) - \widetilde{\mathbb{P}}_h^C \boldsymbol{v}'(\boldsymbol{z}_C^h(\tau))\right]\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} + \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C^\tau \bar{\boldsymbol{\varepsilon}}_\tau\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} + \frac{8Mt^2\epsilon^2}{\lambda}.$$

Here $\bar{\boldsymbol{\varepsilon}}_\tau$ denotes the maximum misspecification incurred from observing $\widetilde{\mathbb{P}}_C^h$ instead of the true $\mathbb{P}_h$. By a standard argument from misspecified bandits (see, e.g., (Ghosh et al., 2017)), using the fact that $\mathbf{\Phi}_C$ has maximum norm $\sqrt{M}$ and the misspecification is bounded by $2\varepsilon(k)$, we can bound the second term by $\varepsilon(k) \cdot \sqrt{dtM \log\left(\frac{\det(\mathbf{\Lambda}_C^h(t))}{\lambda \mathbf{I}_d}\right)}$. To bound the first term on the RHS, we consider the substitution $\boldsymbol{\varepsilon}_{\tau,h}^t = \boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) - \widetilde{\mathbb{P}}_h^C \boldsymbol{v}'(\boldsymbol{z}_h^\tau)$. To bound the first term on the RHS, we consider the filtration $\{\mathcal{F}_\tau\}_{\tau=0}^\infty$ where $\mathcal{F}_0$ is empty, and $\mathcal{F}_\tau = \sigma\left(\left\{\bigcup \left(\boldsymbol{x}_{h+1}^i, \mathbf{\Phi}_C(\boldsymbol{z}_h^i)\right)\right\}_{i \leq \tau}\right)$, and σ denotes the $\sigma-$algebra generated by a finite set. Then, we have that,

$$\begin{aligned}
&\left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \left[\boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) - \widetilde{\mathbb{P}}_h^C \boldsymbol{v}'(\boldsymbol{z}_C^h(\tau))\right]\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} \\
&= \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \left[\boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) - \mathbb{E}\left[\boldsymbol{v}'(\boldsymbol{x}_C^{h+1}(\tau)) | \mathcal{F}_{\tau-1}\right]\right]\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} = \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \boldsymbol{\varepsilon}_{\tau,h}^t\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}}.
\end{aligned}$$

Note that for each $\boldsymbol{\varepsilon}_{\tau,h}^t$, each entry is bounded by $2H$, and therefore we have that the vector $\boldsymbol{\varepsilon}_{\tau,h}^t$ is $H-$sub-Gaussian. Then, applying Lemma 14, we have that,

$$\left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\boldsymbol{z}_C^h(\tau)) \boldsymbol{\varepsilon}_{\tau,h}^t\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq H^2 \log\left(\frac{\det\left(\mathbf{\Lambda}_h^t\right)}{\det\left(\lambda \mathbf{I}_d\right) \delta^2}\right) \leq H^2 \log\left(\frac{\det\left(\bar{\mathbf{\Lambda}}_h^t\right)}{\det\left(\lambda \mathbf{I}_d\right) \delta^2}\right). \tag{17}$$

Replacing this result for each $\boldsymbol{v} \in \mathcal{N}_\epsilon$, we have by a union bound over each $t \in [T], h \in [H]$, we have with probability at least $1 - \delta$, simultaneously for each $t \in [T], h \in [H]$,

$$\sup_{\boldsymbol{v}_t \in \mathbf{\Upsilon}, \boldsymbol{v} \in \mathcal{V}_{\mathbf{\Upsilon}}} \left\|\mathbf{S}_C^h(t)\right\|_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq 2H \sqrt{\log\left(\frac{\det\left(\bar{\mathbf{\Lambda}}_h^t\right)}{\det\left(\lambda \mathbf{I}_d\right)}\right) + \log\left(\frac{HT|\mathcal{N}_\epsilon|}{\delta}\right) + \frac{2Mt^2\epsilon^2}{H^2\lambda}} \tag{18}$$

$$\leq 2H\sqrt{d\log\left(\frac{Mt+\lambda}{\lambda}\right)+\log\left(\frac{|\mathcal{N}_\epsilon|}{\delta}\right)+\log(HT)+\frac{2Mt^2\epsilon^2}{H^2\lambda}}. \tag{19}$$

The last step follows once again by first noticing that $\|\mathbf{\Phi}_C(\cdot)\| \leq \sqrt{M}$ and then applying an AM-GM inequality, and then using the determinant-trace inequality.

Step 2. Here $\mathcal{N}_\epsilon$ is an $\epsilon-$covering of the function class $\mathcal{V}_{\mathbf{\Upsilon}_C}$ for any $h \in [H], m \in [M]$ or $t \in [T]$ under the distance function $\text{dist}_{\mathbf{\Upsilon}_C}(\boldsymbol{v}, \boldsymbol{v}') = \sup_{\boldsymbol{x}\in\mathcal{S},\boldsymbol{v}\in\mathbf{\Upsilon}} \|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}(\boldsymbol{x}')\|_1$. To bound this quantity by the appropriate covering number, we first observe that for any $V \in \mathcal{V}_{\mathbf{\Upsilon}_C}$, we have that the policy weights are bounded as $2HM\sqrt{dT/\lambda}$ (Lemma 12). Therefore, by Lemma 10 we have for any constant B such that $\beta_h^t \leq B$,

$$\log\left(\mathcal{N}_\varepsilon\right) \leq d\cdot\log\left(1+\frac{8HM^3}{\varepsilon}\sqrt{\frac{dT}{\lambda}}\right)+d^2\log\left(1+\frac{8Md^{1/2}B^2}{\lambda\varepsilon^2}\right). \tag{20}$$

Recall that we select the hyperparameters $\lambda = 1$ and $\beta = \mathcal{O}(dH\sqrt{\log(TMH)}$, and to balance the terms in $\bar{\beta}_C^h(t)$ we select $\epsilon = \epsilon^\star = dH/\sqrt{MT^2}$. Finally, we obtain that for some absolute constant C_β, by replacing the above values,

$$\log\left(\mathcal{N}_\varepsilon\right) \leq d\cdot\log\left(1+\frac{8M^{7/2}T^{3/2}}{d^{1/2}}\right)+d^2\log\left(1+8C_\beta d^{1/2}MT^2\log(TMH)\right). \tag{21}$$

Therefore, for some absolute constant C' independent of M, T, H, d and C_β, we have,

$$\log|\mathcal{N}_\varepsilon| \leq C'd^2\log\left(C_\beta\cdot dMT\log(TMH)\right). \tag{22}$$

Replacing this result in the result from Step 1, we have that with probability at least $1-\delta'/2$ for all $t \in [T], h \in [H]$ simultaneously,

$$\left\|\mathbf{S}_C^h(t)\right\|^2_{(\mathbf{\Lambda}_C^h(t))^{-1}} \leq 2H\Bigg((d+2+\varepsilon(k)dMT)\log\frac{MT+\lambda}{\lambda}+2\log\left(\frac{1}{\delta'}\right) + C'd^2\log\left(C_\beta\cdot dMT\log(TMH)\right)+2+4\log(TH)\Bigg).$$

This implies that there exists an absolute constant B independent of M, T, H, d and C_β, such that, with probability at least $1-\delta'/2$ for all $t \in [T], h \in [H], \boldsymbol{v}_C \in \mathbf{\Upsilon}_C$ simultaneously,

$$\left\|\mathbf{S}_C^h(t)\right\|_{(\mathbf{\Lambda}_h^t)^{-1}} \leq B\cdot(dH+\varepsilon(k)H\sqrt{dMT})\sqrt{2\log\left(\frac{(C_\beta+2)dMTH}{\delta'}\right)}. \tag{23}$$

□

Next, we present the key result for cooperative value iteration, which demonstrates that for any agent the estimated $Q-$values have bounded error for any policy π.

Lemma 5. *Fix a clique $C \in \widehat{\mathcal{C}}$ such that $|C| = M$. For each C, there exists an absolute constant c_β such that for $\beta_C^h(t) = c_\beta\cdot(dH+\varepsilon(k)\cdot\sqrt{dtM})\sqrt{\log(2dMHt/\delta')}$ for any policy π, there exists a constant C'_β such that for each $x \in \mathcal{S}, a \in \mathcal{A}$ we have for all $m \in C, t \in [T], h \in [H]$ simultaneously, with probability at least $1-\delta'/2$,*

$$\left|\langle\mathbf{\Phi}_C(\boldsymbol{x}_C,\boldsymbol{a}_C),\boldsymbol{w}_{m,h}^t-\boldsymbol{w}_h^\pi\rangle\right| \leq \mathbb{P}_h(V_{m,h+1}^t-V_{m,h+1}^\pi)(\boldsymbol{x},\boldsymbol{a})+4H\varepsilon(k) + C'_\beta\cdot dH\cdot\|\mathbf{\Phi}_C(\boldsymbol{z}_C)\|_{(\mathbf{\Lambda}_C^h(t))^{-1}}\cdot\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

Proof. By the Bellman equation and Assumptions 1, 2, 3, we have that for any policy $\boldsymbol{\pi}$, and $\boldsymbol{v}_C \in \mathbf{\Upsilon}_C$, there exist weights $\boldsymbol{w}_{\boldsymbol{v}_C,h}^\pi$ such that, for all $\boldsymbol{z} = \{\boldsymbol{z}_C, \bar{\boldsymbol{z}}_C\} \in \mathcal{Z} = \mathcal{S}\times\mathcal{A}$,

$$\boldsymbol{v}_C^\top\mathbf{\Phi}_C(\boldsymbol{z}_C)^\top\boldsymbol{w}_{\boldsymbol{v}_C,h}^\pi = \boldsymbol{v}_C^\top\widetilde{\boldsymbol{r}}_h^C(\boldsymbol{z}_C)+\widetilde{\mathbb{P}}_h^C V_{\boldsymbol{v}_C,h+1}^\pi(\boldsymbol{z}) = \boldsymbol{v}_C^\top\left(\widetilde{\boldsymbol{r}}_h^C(\boldsymbol{z})+\mathbf{1}_M\cdot\widetilde{\mathbb{P}}_h^C V_{\boldsymbol{v}_C,h+1}^\pi(\boldsymbol{z})\right).$$

We have,

$$\boldsymbol{w}^t_{\boldsymbol{\nu}_C,h} - \boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h} \tag{24}$$

$$= (\boldsymbol{\Lambda}^h_C(t))^{-1} \sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\boldsymbol{r}_h(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) + \mathbf{1}_M \cdot V^t_{\boldsymbol{\nu}_C,h+1}(\boldsymbol{x}_\tau)]\right] - \boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h} \tag{25}$$

$$= (\boldsymbol{\Lambda}^h_C(t))^{-1} \left\{-\lambda \boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h} + \sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\mathbf{1}_M \cdot (V^t_{\boldsymbol{\nu}_C,h+1}(\boldsymbol{x}'_\tau) - \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{\nu}_C,h+1}(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)))]\right]\right\}. \tag{26}$$

$$\begin{aligned}
\boldsymbol{w}^t_{\boldsymbol{\nu}_C,h} - \boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h} = \underbrace{-\lambda(\boldsymbol{\Lambda}^h_C(t))^{-1}\boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h}}_{\boldsymbol{v}_1} \\
\underbrace{+ (\boldsymbol{\Lambda}^h_C(t))^{-1} \left\{\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\mathbf{1}_M \cdot (V^t_{\boldsymbol{\nu}_C,h+1}(\boldsymbol{x}'_\tau) - \widetilde{\mathbb{P}}^C_h V^t_{\boldsymbol{\nu}_C,h+1}(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)))]\right]\right\}}_{\boldsymbol{v}_2} \\
\underbrace{+ (\boldsymbol{\Lambda}^h_C(t))^{-1} \left\{\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\mathbf{1}_M \cdot (\widetilde{\mathbb{P}}^C_h V^t_{\boldsymbol{\nu}_C,h+1} - \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{\nu}_C,h+1})(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))]\right]\right\}}_{\boldsymbol{v}_3} + \\
\underbrace{+ (\boldsymbol{\Lambda}^h_C(t))^{-1} \left\{\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\mathbf{1}_M \cdot (\mathbb{P}_h V^t_{\boldsymbol{\nu}_C,h+1} - \widetilde{\mathbb{P}}^C_h V^t_{\boldsymbol{\nu}_C,h+1})(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))]\right]\right\}}_{\boldsymbol{v}_4} \\
\underbrace{+ (\boldsymbol{\Lambda}^h_C(t))^{-1} \left\{\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\boldsymbol{r}_{[C]}(\boldsymbol{x}_h(t), \boldsymbol{a}_h(t)) - \widetilde{\boldsymbol{r}}_C(\boldsymbol{x}^h_C(t), \boldsymbol{a}^h_C(t))]\right]\right\}}_{\boldsymbol{v}_5}.
\end{aligned} \tag{27}$$

Now, we know that for any $\boldsymbol{z} \in \mathcal{Z}$ for any policy π,

$$\begin{aligned}
&\|\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), \boldsymbol{v}_1\rangle\|_2 \\
&\leq \lambda \|\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^h_C(t))^{-1}\boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h}\rangle\|_2 \leq \lambda \cdot \|\boldsymbol{w}^\pi_{\boldsymbol{\nu}_C,h}\| \|\boldsymbol{\Phi}_C(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \leq 2HM\lambda\sqrt{d} \cdot \|\boldsymbol{\Phi}_C(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}}
\end{aligned}$$

Here the last inequality follows from Lemma 11. For the second term, we have by Lemma 4 that there exists an absolute constant C_β, independent of M, T, H, d such that, with probability at least $1 - \delta'/2$ for all $t \in [T], h \in [H], \boldsymbol{\nu}_C \in \boldsymbol{\Upsilon}_C$ simultaneously,

$$\|\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), \boldsymbol{v}_2\rangle\|_2 \leq \|\boldsymbol{\Phi}_C(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot C_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}. \tag{28}$$

For the third term, note that,

$$\langle \boldsymbol{\Phi}_C(\boldsymbol{x}, \boldsymbol{a}), \boldsymbol{v}_3\rangle \tag{29}$$

$$= \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \left\{\sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))[\mathbf{1}_M \cdot (\mathbb{P}_h V^t_{\boldsymbol{\nu}_C,h+1} - \mathbb{P}_h V^\pi_{\boldsymbol{\nu}_C,h+1})(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))]\right\}\right\rangle \tag{30}$$

$$= \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \left\{\sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))^\top \int (V^t_{\boldsymbol{\nu}_C,h+1} - V^\pi_{\boldsymbol{\nu}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}')\right\}\right\rangle \tag{31}$$

$$= \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \left\{\sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))^\top \int (V^t_{\boldsymbol{\nu}_C,h+1} - V^\pi_{\boldsymbol{\nu}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}')\right\}\right\rangle \tag{32}$$

$$= \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), \int (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}') \right\rangle - \lambda \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \int (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}') \right\rangle \tag{33}$$

$$= \int (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x}') \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), \boldsymbol{\mu}_h(\boldsymbol{x}') \right\rangle - \lambda \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \int (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}') \right\rangle \tag{34}$$

$$= \mathbf{1}_M \cdot \left(\widetilde{\mathbb{P}}^C_h (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x},\boldsymbol{a}) \right) - \lambda \left\langle \boldsymbol{\Phi}_C(\boldsymbol{z}), (\boldsymbol{\Lambda}^t_h)^{-1} \int (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x}')d\boldsymbol{\mu}_h(\boldsymbol{x}') \right\rangle \tag{35}$$

$$\leq \mathbf{1}_M \cdot \left(\widetilde{\mathbb{P}}^C_h (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x},\boldsymbol{a}) + 2H\sqrt{d\lambda} \|\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \right). \tag{36}$$

For the last two terms, we can bound them by a similar argument of misspecification as Lemma 4. We can bound both terms by $\mathbf{1}_M \cdot \left(\varepsilon(k) \cdot H\sqrt{dMT} \|\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \right)$. Putting it all together, we have that since $\langle \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,h} - \boldsymbol{w}^\pi_{\boldsymbol{\upsilon}_C,h} \rangle = \langle \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{\upsilon}_1 + \boldsymbol{\upsilon}_2 + \boldsymbol{\upsilon}_3 + \boldsymbol{\upsilon}_4 + \boldsymbol{\upsilon}_5 \rangle$, there exists an absolute constant C_β independent of M, T, H, d, such that, with probability at least $1 - \delta'/2$ for all $t \in [T], h \in [H], \boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C$ simultaneously,

$$\begin{aligned} \left| \langle \boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,h} - \boldsymbol{w}^\pi_{\boldsymbol{\upsilon}_C,h} \rangle \right| &\leq \boldsymbol{\upsilon}_C^\top \mathbf{1}_M \cdot \left(\mathbb{P}_h (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x},\boldsymbol{a}) \right) + 4H\varepsilon(k) \\ &+ \|\boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \|\boldsymbol{\upsilon}_C\|_2 \left(2H\sqrt{d\lambda} + C_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 2HM\lambda\sqrt{d} + 2H\varepsilon(k)\sqrt{dMT} \right) \end{aligned} \tag{37}$$

Since $\lambda \leq 1$ and $\|\boldsymbol{\upsilon}_C\|_2 \leq 1$, there exists a constant C'_β that we have the following for any $(x,a) \in \mathcal{S} \times \mathcal{A}$ with probability $1 - \delta'/2$ simultaneously for all $h \in [H], \boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C, t \in [T]$,

$$\begin{aligned} \left| \langle \boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x},\boldsymbol{a}), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,h} - \boldsymbol{w}^\pi_{\boldsymbol{\upsilon}_C,h} \rangle \right| &\leq \mathbb{P}_h (V^t_{\boldsymbol{\upsilon}_C,h+1} - V^\pi_{\boldsymbol{\upsilon}_C,h+1})(\boldsymbol{x},\boldsymbol{a}) + 4H\varepsilon(k) \\ &+ C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT} \right) \cdot \|\Phi(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}. \end{aligned}$$

□

Lemma 6 (UCB in the Multiagent Setting)**.** *For each $C \in \widehat{\mathcal{C}}$, with probability at least $1 - \delta'/2$, we have that for all $(\boldsymbol{x}_C, \boldsymbol{a}_C, h, t, \boldsymbol{\upsilon}_C) \in \mathcal{S}_C \times \mathcal{A}_C \times [H] \times [T] \times \boldsymbol{\Upsilon}_C$,*

$$Q^t_{v,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) \geq Q^\star_{v,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) - 4H(H + 1 - h)\varepsilon(k).$$

Proof. We prove this result by induction. First, for the last step H, note that the statement holds as $Q^t_{\boldsymbol{\upsilon}_C,H}(\boldsymbol{x}_C,\boldsymbol{a}_C) \geq Q^\star_{\boldsymbol{\upsilon}_C,H}(\boldsymbol{x}_C,\boldsymbol{a}_C) - 4H\varepsilon(k)$ for all $\boldsymbol{\upsilon}_C$. Recall that the value function at step $H+1$ is zero. Therefore, by Lemma 5, we have that, for any $\boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C$,

$$\begin{aligned} &\left| \langle \boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x}_C,\boldsymbol{a}_C), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,H} \rangle - Q^\star_{\boldsymbol{\upsilon}_C,H}(\boldsymbol{x}_C,\boldsymbol{a}_C) \right| \\ &\qquad \leq C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT} \right) \cdot \|\boldsymbol{\Phi}_C(\boldsymbol{z}_C)\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 4H\varepsilon(k). \end{aligned}$$

We have $Q^\star_{\boldsymbol{\upsilon}_C,H}(\boldsymbol{x}_C,\boldsymbol{a}_C) \leq \langle \boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x}_C,\boldsymbol{a}_C), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,H} \rangle + C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT} \right) \cdot \|\Phi(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} = Q^t_{\boldsymbol{\upsilon}_C,H}$. Now, for the inductive case, we have by Lemma 5 for any $h \in [H], \boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C$,

$$\begin{aligned} &\left| \langle \boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x}_C,\boldsymbol{a}_C), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,h} - \boldsymbol{w}^\star_{\boldsymbol{\upsilon}_C,h} \rangle - \left(\mathbb{P}_h V^\star_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C) - \mathbb{P}_h V^t_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C) \right) \right| \\ &\leq C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT} \right) \cdot \|\Phi(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}. \end{aligned}$$

By the inductive assumption we have $Q^t_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C) \geq Q^\star_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C)$ implying $\mathbb{P}_h V^\star_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C) - \mathbb{P}_h V^t_{\boldsymbol{\upsilon}_C,h+1}(\boldsymbol{x}_C,\boldsymbol{a}_C) \geq 0$. Substituting the appropriate Q value formulations we have,

$$Q^{\star}_{\boldsymbol{\upsilon}_C,h} \leq \langle \boldsymbol{\upsilon}_C^{\top} \boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C), \boldsymbol{w}^t_{\boldsymbol{\upsilon}_C,h} \rangle + 4H\varepsilon(k) + C'_{\beta} \cdot \left(dH + \varepsilon(k) H \sqrt{dMT} \right) \cdot \|\Phi(\boldsymbol{z})\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot \sqrt{2 \log\left(\frac{dMTH}{\delta'} \right)} = Q^t_{\boldsymbol{\upsilon}_C,h}(\boldsymbol{x}_C, \boldsymbol{a}_C).$$

□

Lemma 7 (Recursive Relation in Multiagent MDP Settings)**.** *Fix a clique $C \in \widehat{\mathcal{C}}$ of size M. For any $\boldsymbol{\upsilon}_C \in \boldsymbol{\Upsilon}_C$, let $\delta^t_{\boldsymbol{\upsilon}_C,h} = V^t_{\boldsymbol{\upsilon}_C,h}(\boldsymbol{x}^h_C(t)) - V^{\pi_t}_{\boldsymbol{\upsilon}_C,h}(\boldsymbol{x}^h_C(t))$, and $\boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h+1} = \mathbb{E}\left[\delta^t_{\boldsymbol{\upsilon}_C,h} | \boldsymbol{x}^h_C(t), \boldsymbol{a}^h_C(t)\right] - \delta^t_{\boldsymbol{\upsilon}_C,h}$. Then, with probability at least $1-\alpha$, for all $(t,h) \in [T] \times [H]$ simultaneously,*

$$\delta^t_{\boldsymbol{\upsilon}_C,h} \leq \delta^t_{\boldsymbol{\upsilon}_C,h+1} + \boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h+1} + 4H\varepsilon(k) + 2 \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(t), \boldsymbol{a}^h_C(t)) \right\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot C'_{\beta} \cdot \left(dH + \varepsilon(k) H \sqrt{dMT} \right) \cdot \sqrt{2 \log\left(\frac{dMTH}{\alpha} \right)}.$$

Proof. By Lemma 5, we have that for any $(\boldsymbol{x}_C, \boldsymbol{a}_C, h, \boldsymbol{\upsilon}_C, t) \in \mathcal{S}_C \times \mathcal{A}_C \times [H] \times \boldsymbol{\Upsilon}_C \times [T]$ with probability at least $1 - \alpha/2$,

$$Q^t_{\boldsymbol{\upsilon}_C,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) - Q^{\pi_t}_{\boldsymbol{\upsilon}_C,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) \leq \mathbb{P}_h(V^t_{\boldsymbol{\upsilon}_C,h+1} - V^{\pi_t}_{\boldsymbol{\upsilon}_C,h})(\boldsymbol{x}_C, \boldsymbol{a}_C) + 4H\varepsilon(k) + 2 \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C) \right\|_{(\boldsymbol{\Lambda}^h_C(t))^{-1}} \cdot C_{\beta} \cdot \left(dH + \varepsilon(k) H \sqrt{dMT} \right) \cdot \sqrt{2 \log\left(\frac{dMTH}{\alpha} \right)}.$$

Replacing the definition of $\delta^t_{\boldsymbol{\upsilon}_C,h}$ and $V^{\pi_t}_{\boldsymbol{\upsilon}_C,h}$ finishes the proof. □

Lemma 8. *For each clique $C \in \widehat{\mathcal{C}}$ and each $\boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h}$ as defined earlier and any $\delta \in (0,1)$, we have that with probability at least $1 - \delta/2$,*

$$\sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{C \in \widehat{\mathcal{C}}} \boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h} \leq \sqrt{2H^3 T |\widehat{\mathcal{C}}| \log\left(\frac{2}{\alpha} \right)}. \tag{38}$$

Proof. Observe that following the reasoning in Theorem 3.1 of Jin et al. (2020), we can see that $\{\boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h}\}_{h,t,C}$ is a martingale difference sequence (computation within each clique at any instant is independent of the current state of other cliques). Furthermore, since $|\boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h}| \leq H$ regardless of $\boldsymbol{\upsilon}_C$, which allows us to apply Azuma-Hoeffding inequality. We have, for any $t > 0$,

$$\mathbb{P}\left(\sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{C \in \widehat{\mathcal{C}}} \boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_C,h} > t \right) \leq \exp\left(-\frac{t^2}{2T|\widehat{\mathcal{C}}|H^2} \right).$$

Rearranging provides us the final result. □

We are now ready to prove Theorem 2. We first restate the Theorem for completeness.

Theorem 2. Algorithm 1 when run on a game with n agents satisfying Assumptions 1, 2, 3 with error $\varepsilon_\star$, approximate clique covering $\widehat{\mathcal{C}}$, and $\kappa \cdot dH \cdot \theta(G)$ rounds of communication for some $\kappa > 1$, $\beta^h_C(t) = \mathcal{O}(H\sqrt{d \log(ntH)} + \varepsilon_\star \sqrt{dT})\ \forall\ C \in \widehat{\mathcal{C}}$ obtains, with probability at least $1-\alpha$, regret:

$$\mathfrak{R}_C(T) = \widetilde{\mathcal{O}}\left(\theta(G) \cdot d^{\frac{3}{2}} H^2 \cdot \max\left\{1, (2T \cdot n_{\max})^{\frac{2}{\kappa}}\right\} \left(\sqrt{T \log\left(\frac{1}{\alpha}\right)} + 2T \cdot \varepsilon_\star \right) \right).$$

Where $\theta(G)$ denotes the clique covering number of G, and $n_{\max}$ is the size of the largest clique in $\widehat{\mathcal{C}}$.

Proof. We have by the definition of cumulative regret:

$$\mathfrak{R}_C(T) = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\upsilon}_t \sim \boldsymbol{\Upsilon}} \left[\max_{\boldsymbol{x}_1^t \in \mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) - V^{\pi_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) \right] \right] = \mathbb{E}_{\boldsymbol{\upsilon}_t \sim \boldsymbol{\Upsilon}} \left[\sum_{t=1}^{T} \max_{\boldsymbol{x}_1^t \in \mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) - V^{\pi_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) \right] \right]. \tag{39}$$

Our analysis focuses only on the term inside the expectation, which we will bound via terms that are independent of $\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_T$, bounding $\mathfrak{R}_C$. We bound the cumulative regret incurred by each clique, summing over which gives us the cumulative regret.

$$\sum_{t=1}^{T} \max_{\boldsymbol{x}_1^t \in \mathcal{S}} \left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) - V^{\pi_t}_{\boldsymbol{\upsilon}_t,1}(\boldsymbol{x}_1^t) \right] \leq \sum_{C \in \widehat{\mathcal{C}}} \left(\sum_{t=1}^{T} \max_{\boldsymbol{x}_C \in \mathcal{S}_C} \left[V^{\star}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) - V^{\pi_{t,C}}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) \right] \right).$$

We can bound the clique-wise regret for any $C \in \widehat{\mathcal{C}}$ of size M as follows.

$$\begin{aligned}
&\sum_{t=1}^{T} \max_{\boldsymbol{x}_C \in \mathcal{S}_C} \left[V^{\star}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) - V^{\pi_{t,C}}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) \right] \leq \sum_{t=1}^{T} \max_{\boldsymbol{x}_C \in \mathcal{S}_C} \delta^t_{\boldsymbol{\upsilon}_{t,C},1} + 4HT\varepsilon(k) \\
&\leq \sum_{t,h}^{T,H} \boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_{t,C},h} + 2C'_{\beta} \cdot \left(dH + \varepsilon(k) H \sqrt{dMT} \right) \cdot \sqrt{2 \log \left(\frac{dMTH}{\alpha} \right)} \left(\sum_{t,h}^{T,H} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\boldsymbol{\Lambda}_C^h(t))^{-1}} \right) + 4H\varepsilon(k).
\end{aligned}$$

Where the last inequality holds with probability at least $1-\alpha/2$, via Lemma 7 and Lemma 6. To bound the second summation, we can use the technique in Theorem 4 of Abbasi-Yadkori et al. (2011). Assume that the last time synchronization of rewards occured was at instant k_T. We therefore have, by Lemma 12 of Abbasi-Yadkori et al. (2011), for any $h \in [H]$

$$\begin{aligned}
&\sum_{t=1}^{T} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\boldsymbol{\Lambda}_C^h(t))^{-1}} \leq \frac{\det(\bar{\boldsymbol{\Lambda}}_C^h(t))}{\det(\boldsymbol{\Lambda}_C^h(t))} \sum_{t=1}^{T} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\bar{\boldsymbol{\Lambda}}_C^h(t))^{-1}} \\
&\leq \sqrt{S} \sum_{t=1}^{T} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\bar{\boldsymbol{\Lambda}}_C^h(t))^{-1}}
\end{aligned}$$

Here $\bar{\boldsymbol{\Lambda}}_C^h(t) = \sum_{t=1}^{T} \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t))^{\top}$ and the last inequality follows from the algorithms' synchronization condition. Replacing this result, we have that,

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{h=1}^{H} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\boldsymbol{\Lambda}_C^h(t))^{-1}} \leq 2 \sum_{h=1}^{H} \left(\sqrt{S} \sum_{t=1}^{T} \left\| \boldsymbol{\Phi}_C(\boldsymbol{x}_C^h(t), \boldsymbol{a}_C^h(t)) \right\|_{(\bar{\boldsymbol{\Lambda}}_C^h(t))^{-1}} \right) \\
&\leq 2H \sqrt{ST \cdot d \log \frac{MT + \lambda}{\lambda}}.
\end{aligned}$$

Where the last inequality is an application of Lemma 13 and using the fact that $\|\boldsymbol{\Phi}_C(\cdot)\|_2 \leq \sqrt{M}$. Replacing this result, we have that with probability at least $1 - \alpha/2$, by a union bound over all cliques in $\widehat{\mathcal{C}}$,

$$\begin{aligned}
&\sum_{C \in \widehat{\mathcal{C}}} \left(\sum_{t=1}^{T} \max_{\boldsymbol{x}_C \in \mathcal{S}_C} \left[V^{\star}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) - V^{\pi_{t,C}}_{\boldsymbol{\upsilon}_{t,C},1}(\boldsymbol{x}_C) \right] \right) \\
&\leq \sum_{t,h,C}^{T,H,\widehat{\mathcal{C}}} \boldsymbol{\xi}^t_{\boldsymbol{\upsilon}_{t,C},h} + 2C'_{\beta} \cdot H^2 \left(d + \varepsilon(k) \sqrt{dMT} \right) \cdot \sqrt{2ST \log \left(\frac{dMTH|\widehat{\mathcal{C}}|}{\alpha} \right) \cdot d \log \frac{MT + \lambda}{\lambda}} + 4HT|\widehat{\mathcal{C}}|\varepsilon(k).
\end{aligned}$$

We can bound the second term via Lemma 8. Taking expectation of the RHS over $\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_T$ and rewriting S in terms of κ via Lemma 3 gives us the final result (the $\widetilde{\mathcal{O}}$ notation hides polylogarithmic factors). □

D.6. Proof of Lemma 3

Proof. Let the total rounds of communication triggered by the threshold condition in any step $h \in [H]$ in any clique C of size M be given by $n_h(C)$. Then, we have, by the communication criterion,

$$S^{n_h(C)} < \frac{\det\left(\mathbf{\Lambda}_C^h(t)\right)}{\det\left(\lambda \mathbf{I}_d\right)} \leq (1 + MT/d)^d. \tag{40}$$

Where the last inequality follows from Lemma 13 and the fact that $\|\mathbf{\Phi}\| \leq \sqrt{M} \leq \sqrt{n_{\max}}$. This gives us that $n_h(C) \leq d \log_S (1 + n_{\max} T/d)) + 1$. Furthermore, by noticing that $\gamma \leq \sum_{C \in \widehat{\mathcal{C}}} \sum_{h=1}^{H} n_h(C)$, and that $|\widehat{\mathcal{C}}| \leq 1.25 \cdot \theta(G)$, we have the final result. □

E. Auxiliary Results

E.1. Covering Number Bounds

Lemma 9 (Covering Number of the Euclidean Ball)**.** *For any $\varepsilon > 0$, the $\varepsilon-$covering number of the Euclidean ball in $\mathbb{R}^d$ with radius $R > 0$ is less than $(1 + 2R/\varepsilon)^d$.*

Lemma 10 (Covering number for Markov game UCB-style functions)**.** *Let $\mathcal{V}$ denote a class of functions mapping from $\mathcal{S}$ to $\mathbb{R}$ with the following parameteric form*

$$\boldsymbol{v}_{\boldsymbol{v}}(\cdot) = \mathbf{1}_M \cdot \min\left\{\max_{\boldsymbol{a} \in \mathcal{A}} \left[\langle \boldsymbol{v}, \boldsymbol{v}(\cdot, \boldsymbol{a})\rangle + \beta \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{\Lambda}^{-1} \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|\right], H\right\}, \boldsymbol{v}(\cdot, \boldsymbol{a}) = \boldsymbol{w}^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a})$$

where the parameters $(\boldsymbol{w}, \beta, \mathbf{\Lambda})$ are such that $\boldsymbol{w} \in \mathbb{R}^d$, $\|\boldsymbol{w}\|_2 \leq L$, $\beta \in (0, B]$, $\|\mathbf{\Phi}_C(\boldsymbol{x}, \boldsymbol{a})\| \leq \sqrt{M}\ \forall (\boldsymbol{x}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$, and the minimum eigenvalue of $\mathbf{\Lambda}$ satisfies $\lambda_{\min}(\mathbf{\Lambda}) \geq \lambda$. Let $\mathcal{N}_\varepsilon$ be the $\varepsilon-$covering number of $\mathcal{V}$ with respect to the distance dist$(\boldsymbol{v}, \boldsymbol{v}') = \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{v} \in \mathbf{\Upsilon}} |\boldsymbol{v}_{\boldsymbol{v}}(\boldsymbol{x}) - \boldsymbol{v}'_{\boldsymbol{v}}(\boldsymbol{x})|$. Then,

$$\log\left(\mathcal{N}_\varepsilon\right) \leq d \cdot \log\left(1 + \frac{4LM^2}{\varepsilon}\right) + d^2 \log\left(1 + \frac{8Md^{1/2}B^2}{\lambda \varepsilon^2}\right).$$

Proof. We have that for two matrices $\mathbf{A}_1 = \beta^2 \mathbf{\Lambda}_1^{-1}, \mathbf{A}_2 = \beta^2 \mathbf{\Lambda}_2^{-1}$ and weight matrices $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^d$,

$$\sup_{\boldsymbol{v} \in \mathbf{\Upsilon}, \boldsymbol{x} \in \mathcal{S}} |\boldsymbol{v}_{\boldsymbol{v}}(\boldsymbol{x}) - \boldsymbol{v}'_{\boldsymbol{v}}(\boldsymbol{x})|_1 \tag{41}$$

$$= M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{v} \in \mathbf{\Upsilon}} \left|\boldsymbol{v}^\top \boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}^\top \boldsymbol{v}'(\boldsymbol{x})\right| \tag{42}$$

$$\leq M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}} |\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}'(\boldsymbol{x})|_1 \tag{43}$$

$$\leq M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left|\left[\boldsymbol{w}_1^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a}) + \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_1 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2\right] - \left[\boldsymbol{w}_2^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a}) + \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_2 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2\right]\right|_1 \tag{44}$$

$$\leq M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left|(\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a}) + \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_1 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2 - \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_2 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2\right|_1 \tag{45}$$

$$\leq M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left|(\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right|_1 + M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left|\left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_1 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2 - \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \mathbf{A}_2 \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2\right| \tag{46}$$

$$\leq M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left|(\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right|_1 + M \cdot \sup_{\boldsymbol{x} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}} \left\|\mathbf{\Phi}_C(\cdot, \boldsymbol{a})^\top \left(\mathbf{A}_1 - \mathbf{A}_2\right) \mathbf{\Phi}_C(\cdot, \boldsymbol{a})\right\|_2 \tag{47}$$

$$\leq M^{3/2} \cdot \sup_{\mathbf{\Phi}: \|\mathbf{\Phi}\| \leq \sqrt{M}} \left[\left\|(\boldsymbol{w}_1 - \boldsymbol{w}_2)^\top \mathbf{\Phi}\right\|_2\right] + M \cdot \sup_{\mathbf{\Phi}: \|\mathbf{\Phi}\| \leq \sqrt{M}} \left\|\mathbf{\Phi}^\top \left(\mathbf{A}_1 - \mathbf{A}_2\right) \mathbf{\Phi}\right\|_2 \tag{48}$$

$$\leq M^2 \cdot \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + M^2 \|\mathbf{A}_1 - \mathbf{A}_2\|_2 \tag{49}$$

$$\leq M^2 \cdot \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 + M^2 \|\mathbf{A}_1 - \mathbf{A}_2\|_F \tag{50}$$

$$\tag{51}$$

Now, let $\mathcal{C}_{\boldsymbol{w}}$ be an $\varepsilon/(2M^2)$ cover of $\left\{\boldsymbol{w} \in \mathbb{R}^d \;\middle|\; \|\boldsymbol{w}\|_2 \leq L\right\}$ with respect to the Frobenius-norm, and $\mathcal{C}_{\mathbf{A}}$ be an $\varepsilon^2/4$ cover of $\left\{\mathbf{A} \in \mathbb{R}^{d\times d} | \|\mathbf{A}\|_F \leq (M^2 d)^{1/2} B^2 \lambda^{-1}\right\}$ with respect to the Frobenius norm. By Lemma 9 we have,

$$|\mathcal{C}_{\boldsymbol{w}}| \leq (1 + 4LM^2/\varepsilon)^d, |\mathcal{C}_{\mathbf{A}}| \leq (1 + 8(M^2 d)^{1/2} B^2/(\lambda\varepsilon^2))^{d^2}. \tag{52}$$

Therefore, we can select, for any $\boldsymbol{v}_{\boldsymbol{v}}(\cdot)$, corresponding weight $\boldsymbol{w} \in \mathcal{C}_{\boldsymbol{w}}$, and matrix $\mathbf{A} \in \mathcal{C}_{\mathbf{A}}$. Therefore, $\mathcal{N}_\varepsilon \leq |\mathcal{C}_{\mathbf{A}}| \cdot |\mathcal{C}_{\boldsymbol{w}}|$. This gives us,

$$\log\left(\mathcal{N}_\varepsilon\right) \leq d \cdot \log\left(1 + \frac{4LM^2}{\varepsilon}\right) + d^2 \log\left(1 + \frac{8Md^{1/2}B^2}{\lambda\varepsilon^2}\right). \tag{53}$$

□

Lemma 11 (Linearity of weights in Markov game)**.** *In a game with n agents satisfying Assumptions 1, 2, 3, for any policy π, clique $C \in \widehat{\mathcal{C}}$ of size M, and $\boldsymbol{v}_C \in \boldsymbol{\Upsilon}_C$, there exists weights $\{\boldsymbol{w}^\pi_{\boldsymbol{v}_C,h}\}_{h\in[H]}$ such that $|Q^\pi_{\boldsymbol{v}_C,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) - \boldsymbol{v}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C)^\top \boldsymbol{w}^\pi_{\boldsymbol{v}_C,h}| \leq 2H\varepsilon(k)$ for all $(\boldsymbol{x}_C, \boldsymbol{a}_C, h) \in \mathcal{S}_C \times \mathcal{A}_C \times [H]$, where $\|\boldsymbol{w}^\pi_{\boldsymbol{v}_C,h}\|_2 \leq 2H\sqrt{d}$.*

Proof. By the Bellman equation and Proposition 1, we have that for any MDP corresponding to the scalarization parameter $\boldsymbol{v}_C \in \boldsymbol{\Upsilon}_C$ and any policy $\boldsymbol{\pi}$, state $\boldsymbol{x} \in \mathcal{S}_C$, joint action $\boldsymbol{a} \in \mathcal{A}_C$,

$$\begin{aligned}
&Q^\pi_{\boldsymbol{v}_C,h}(\boldsymbol{x}_C, \boldsymbol{a}_C) && (54)\\
&\leq \boldsymbol{v}_C^\top \widetilde{\boldsymbol{r}}^C_h(\boldsymbol{x}_C, \boldsymbol{a}_C) + \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{v}_C,h+1}(\boldsymbol{x}_C, \boldsymbol{a}_C) + 2H\varepsilon(k) && (55)\\
&\leq \boldsymbol{v}_C^\top \left(\widetilde{\boldsymbol{r}}^C_h(\boldsymbol{x}_C, \boldsymbol{a}_C) + \mathbf{1}_M \cdot \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{v}_C,h+1}(\boldsymbol{x}_C, \boldsymbol{a}_C)\right) + 2H\varepsilon(k) && (56)\\
&\leq \boldsymbol{v}_C^\top \left(\boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C)^\top \begin{bmatrix}\boldsymbol{\theta}_h \\ \mathbf{0}_d\end{bmatrix} + \int V^\pi_{\boldsymbol{v}_C,h+1}(\boldsymbol{x}'_C) \boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C)^\top \begin{bmatrix}\mathbf{0}_d \\ d\boldsymbol{\mu}_h(\boldsymbol{x}'_C)\end{bmatrix} d\boldsymbol{x}'_C\right) + 2H\varepsilon(k) && (57)\\
&\leq \boldsymbol{v}_C^\top \boldsymbol{\Phi}_C(\boldsymbol{x}_C, \boldsymbol{a}_C)^\top \boldsymbol{w}^\pi_{\boldsymbol{v}_C,h} + 2H\varepsilon(k). && (58)
\end{aligned}$$

The first inequality follows from Assumption 2. Here $\boldsymbol{w}^\pi_{\boldsymbol{v}_C,h} = \begin{bmatrix}\boldsymbol{\theta}_h \\ \int V^\pi_{\boldsymbol{v}_C,h+1}(\boldsymbol{x}'_C) d\boldsymbol{\mu}(\boldsymbol{x}'_C) d\boldsymbol{x}'_C\end{bmatrix}$. Therefore, since $\|\boldsymbol{\theta}_h\| \leq \sqrt{d}$ and $\|\int V^\pi_{\boldsymbol{v}_C,h+1}(\boldsymbol{x}'_C) d\boldsymbol{\mu}(\boldsymbol{x}'_C)\| \leq H\sqrt{d}$, the result follows. □

Lemma 12 (Bound on Weights)**.** *For any $C \in \widehat{\mathcal{C}}, |C| = M, t \in [T], h \in [H], \boldsymbol{v} \in \boldsymbol{\Upsilon}$, the weights $\boldsymbol{w}^t_{\boldsymbol{v}_C,h}$ satisfy*

$$\|\boldsymbol{w}^t_{\boldsymbol{v}_C,h}\|_2 \leq 2HM\sqrt{dt/\lambda}.$$

Proof. For any vector $\boldsymbol{v} \in \mathbb{R}^d | \|\boldsymbol{v}\| = 1$,

$$\begin{aligned}
\left|\boldsymbol{v}^\top \boldsymbol{w}^t_{\boldsymbol{v},h}\right| &= \left|\boldsymbol{v}^\top \left(\boldsymbol{\Lambda}^t_h\right)^{-1} \left(\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) \left[\boldsymbol{r}_h(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) + \max_{\boldsymbol{a}\in\mathcal{A}} Q_{\boldsymbol{v},h+1}(\boldsymbol{x}'_\tau, \boldsymbol{a})\right]\right]\right)\right| && (59)\\
&\leq \sqrt{k_t \cdot \sum_{\tau=1}^{k_t} \left(\boldsymbol{v}^\top \left(\boldsymbol{\Lambda}^t_h\right)^{-1} \left[\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) \left[\boldsymbol{r}_h(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau)) + \max_{\boldsymbol{a}\in\mathcal{A}} Q_{\boldsymbol{v},h+1}(\boldsymbol{x}'_\tau, \boldsymbol{a})\right]\right]\right)^2} && (60)\\
&\leq HM\sqrt{k_t \cdot \sum_{\tau=1}^{k_t} \left\|\boldsymbol{v}^\top \left(\boldsymbol{\Lambda}^t_h\right)^{-1} \boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))\right\|_2^2} && (61)\\
&\leq 2HM\sqrt{k_t \cdot \sum_{\tau=1}^{k_t} \|\boldsymbol{v}\|^2_{\left(\boldsymbol{\Lambda}^h_C(t)\right)^{-1}} \|\boldsymbol{\Phi}_C(\boldsymbol{x}^h_C(\tau), \boldsymbol{a}^h_C(\tau))\|^2_{\left(\boldsymbol{\Lambda}^h_C(t)\right)^{-1}}} && (62)\\
&\leq 2HM\|\boldsymbol{v}\|\sqrt{dk_t/\lambda} \leq 2HM\sqrt{dt/\lambda}. && (63)
\end{aligned}$$

The penultimate inequality follows from Lemma 13 and the final inequality follows from the fact that $k_t \leq t$. The remainder of the proof follows from the fact that for any vector $\boldsymbol{w}$, $\|\boldsymbol{w}\| = \max_{\boldsymbol{v}:\|\boldsymbol{v}\|=1} |\boldsymbol{v}^\top \boldsymbol{w}|$. □

Lemma 13 (Elliptical Potential, Lemma 3 of Abbasi-Yadkori et al. (2011)). *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n \in \mathbb{R}^d$ be vectors such that $\|\boldsymbol{x}\|_2 \leq L$. Then, for any positive definite matrix $\boldsymbol{U}_0 \in \mathbb{R}^{d\times d}$, define $\boldsymbol{U}_t := \boldsymbol{U}_{t-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top$ for all t. Then, for any $\nu > 1$,*

$$\sum_{t=1}^{n} \|\boldsymbol{x}_t\|^2_{\boldsymbol{U}_{t-1}^{-1}} \leq 2d \log_\nu \left(\frac{tr(\boldsymbol{U}_0) + nL^2}{d \det^{1/d}(\boldsymbol{U}_0)} \right).$$

E.2. Multi-task concentration bound (Chowdhury and Gopalan, 2020)

We assume the multi-agent kernel $\boldsymbol{\Gamma}$ to be continuous relative to the operator norm on $\mathcal{L}(\mathbb{R}^n)$, the space of bounded linear operators from $\mathbb{R}^n$ to itself (for some $n > 0$). Then the RKHS $\mathcal{H}_{\boldsymbol{\Gamma}}(\mathcal{X}^n)$ associated with the kernel $\boldsymbol{\Gamma}$ is a subspace of the space of continuous functions from $\mathcal{X}^n$ to $\mathbb{R}^n$, and hence, $\boldsymbol{\Gamma}$ is a Mercer kernel. Let μ be a measure on the (compact) set $\mathcal{X}^n$. Since $\boldsymbol{\Gamma}$ is a Mercer kernel on $\mathcal{X}$ and $\sup_{\mathbf{X}\in\mathcal{X}^n} \|\boldsymbol{\Gamma}(\mathbf{X},\mathbf{X})\| < \infty$, the RKHS $\mathcal{H}_{\boldsymbol{\Gamma}}(\mathcal{X}^n)$ is a subspace of $L^2(\mathcal{X}^n, \mu; \mathbb{R}^n)$, the Banach space of measurable functions $g : \mathcal{X}^n \to \mathbb{R}^n$ such that $\int_{\mathcal{X}^n} \|g(\mathbf{X})\|^2 d\mu(\mathbf{X}) < \infty$, with norm $\|g\|_{L^2} = \left(\int_{\mathcal{X}^n} \|g(\mathbf{X})\|^2 d\mu(\mathbf{X}).\right)^{1/2}$. Since $\boldsymbol{\Gamma}(\mathbf{X},\mathbf{X}) \in \mathcal{L}(\mathbb{R}^n)$ is a compact operator, by the Mercer theorem

We can therefore define a feature map $\Phi : \mathcal{X}^M \to \mathcal{L}(\mathbb{R}^n, \ell^2)$ of the multi-agent kernel Γ by

$$\boldsymbol{\Phi}(\mathbf{X})^\top \boldsymbol{y} = \left(\sqrt{\nu_1}\psi_1(\boldsymbol{x}_1)^\top \boldsymbol{y}, \sqrt{\nu_2}\psi_2(\boldsymbol{x}_2)^\top \boldsymbol{y}, ..., \sqrt{\nu_M}\psi_M(\boldsymbol{x}_M)^\top \boldsymbol{y}\right), \ \forall \mathbf{X} \in \mathcal{X}^M, \boldsymbol{y} \in \mathbb{R}^m. \tag{64}$$

We then obtain $F(\mathbf{X}) = \Phi(\mathbf{X})^\top \boldsymbol{\theta}^\star$ and $\Gamma(\mathbf{X},\mathbf{X}') = \Phi(\mathbf{X})^\top \Phi(\mathbf{X}') \ \forall \ \mathbf{X}, \mathbf{X}' \in \mathcal{X}^M$.

Define $\mathbf{S}_t = \sum_{\tau=1}^{t} \Phi(\mathbf{X}_\tau)^\top \varepsilon_\tau$, where $\varepsilon_1, ..., \varepsilon_t$ are the noise vectors in $\mathbb{R}^M$. Now consider $\mathcal{F}_{t-1}$, the σ-algebra generated by the random variables $\{\mathbf{X}_\tau, \varepsilon_\tau\}_{\tau=1}^{t-1}$ and $\mathbf{X}_t$. We can see that $\mathbf{S}_t$ is $\mathcal{F}_t$-measurable, and additionally, $\mathbb{E}[\mathbf{S}_t|\mathcal{F}_{t-1}] = \mathbf{S}_{t-1}$. Therefore, $\{\mathbf{S}_t\}_{t\geqslant 1}$ is a martingale with outputs in ℓ^2 space. Following (Chowdhury and Gopalan, 2020), consider now the map $\Phi_{\mathcal{X}_t} : \ell^2 \to \mathbb{R}^{Mt}$:

$$\Phi_{\mathcal{X}_t}\boldsymbol{\theta} = \left[\left(\Phi(\mathbf{X}_1)^\top\boldsymbol{\theta}\right)^\top, \left(\Phi(\mathbf{X}_1)^\top\boldsymbol{\theta}\right)^\top, ..., \left(\Phi(\mathbf{X}_t)^\top\boldsymbol{\theta}\right)^\top\right]^\top, \ \forall \ \boldsymbol{\theta} \in \ell^2. \tag{65}$$

Additionally, denote $\mathbf{V}_t := \Phi_{\mathcal{X}_t}^\top \Phi_{\mathcal{X}_t}$ be a map from ℓ^2 to itself, with $\mathbf{I}$ being the identity operator in ℓ^2. We have the following result from (Chowdhury and Gopalan, 2020) that provides us with a self-normalized martingale bound.

Lemma 14 (Lemma 3 of (Chowdhury and Gopalan, 2020)). *Let the noise vectors $\{\boldsymbol{\varepsilon}_t\}_{t\geqslant 1}$ be σ-sub-Gaussian. Then, for any $\eta > 0$ and $\delta \in (0, 1]$, with probability at least $1-\delta$, the following holds uniformly over all $t \geqslant 1$:*

$$\|\mathbf{S}_t\|_{(\mathbf{V}_t+\eta\mathbf{I})^{-1}} \leqslant \sigma\sqrt{2\log(1/\delta) + \log\det(\mathbf{I} + \eta^{-1}\mathbf{V}_t)}.$$

Alternatively stated, we have again that with probability at least $1-\delta$, the following holds uniformly over all $t \geqslant 1$:

$$\|\boldsymbol{\mathcal{E}}_t\|^2_{\left((\mathbf{K}_t+\eta\mathbf{I})^{-1}+\mathbf{I}\right)^{-1}} \leqslant 2\sigma^2 \log\left[\frac{\sqrt{\det(\mathbf{I}(1+\eta) + \mathbf{K}_t)}}{\delta}\right].$$

F. Lower Bound

The central observation in this setting is that under the clique-dominance assumption (Assumption 2), it is impossible to obtain regret that avoids the $\theta(G)$ factor. Rather than provide a formal proof, we can provide a straightforward outline to obtain the guarantee. For any influence graph G, we can construct a minimal clique covering $\mathcal{C}_\star$ and we can construct a unique Markov game for each clique in $\mathcal{C}_\star$. For any clique C we construct a Markov game MG_C such that the reward functions for each agent in C are identical functions of the clique state-action (let us call it r_h^c for any agent c and step h), i.e., $r_h^c = r_h^C \forall c \in C$) and the marginal transition probability is only a function of the clique state-action as well. Now, observe that under this criterion, the scalarized reward is independent of the parameter $\boldsymbol{\upsilon}$ and is always r_h^C and hence one can find the regret in $\boldsymbol{\Pi}^\star_{\boldsymbol{\Upsilon}}$ for the MG by simply choosing an arbitrary value of $\boldsymbol{\upsilon}$ and solving the scalarized MDP. Since we are considering the tabular setting, for any clique C, we can set $d_C = |\mathcal{S}_C| \times |\mathcal{A}_C|$. Note that for any MDP we have that the regret obeys $\Omega(H^2\sqrt{|\mathcal{S}||\mathcal{A}|T})$, which gives us the regret within a clique as $\Omega(H^2\sqrt{d_C T})$. Summing over the clique cover we obtain the lower bound for $\mathfrak{R}_C$.

G. Experiments

We run experiments on a basic cooperative multi-agent RL grid-world environment, `GridExplore` described as follows. In the first game, `GridExplore`, the agents are randomly placed in a grid of blank cells. Agents explore the grid by observing cells which are denoted as 'explored'. Each agent obtains a reward for the number of cells they have explored. Each agent has the following actions {`L, R, U, D, LU, LD, RU, RD`} and there are a total of $n = 8$ agents. The visibility of other agents is examined under 3 settings: (a) each agent can see all others (full), (b) each agent can only observe a random half of agents (partial), and (c) each agent can only observe their actions (self). The board is of size `10x10` and $p_{\boldsymbol{\Upsilon}}$ is the uniform distribution over Δ_n. The game runs in episodes of length `200` and $T = 5 \times 10^6$.

For each agent c in clique C, the feature $\boldsymbol{\phi}_c$ is given as the combined action of all visible agents (of dimensionality $8n'$) and $\boldsymbol{\psi}_C$ is the joint state of all agents in C (of dimensionality $100n'$) where $n' = 1$ in the self setting, $n' = \lfloor n/2 \rfloor + 1$ in the partial setting, $n' = n$ in the full setting. For our algorithm, we select $\varepsilon = 0.5 \times 10^{-6}$.

We present the average reward (over all agents) over the last 1000 episodes for 100 repeated trials in the table below. As baselines we consider a group of n individual Q-learning on the agents personal state space `Q-ind`, n individual `DQN` agents using a custom CNN with 3 hidden layers: 2 convolutional layers with filter size 4 and 32 filters each, and 1 fully-connected layer of dimensionality 256; and the `LSVI-UCB` algorithm proposed by Jin et al. (2020) on the same input space as ours.

Baseline	Full	Partial	Self
`Q-ind`	20.885 ± 2.833	16.294 ± 4.239	13.202 ± 4.887
`DQN`	35.932 ± 3.094	27.587 ± 5.059	18.478 ± 2.093
`LSVI-ind`	17.439 ± 2.192	9.847 ± 4.292	7.340 ± 3.778
`MultOVI`	31.294 ± 3.776	22.119 ± 5.882	15.395 ± 3.098

Table 1. Results on `GridExplore` environment.

We observe that our algorithm comfortably outperforms the individual baselines `Q-ind` and `LSVI` in all three settings, however, `DQN` outperforms our algorithm, presumably owing to better feature representations learnt from the deep neural networks. Future work may consider approaches to combine deep neural network based approaches with the multi-agent UCB algorithm as ours.