Safe Cooperative Multi-Agent Reinforcement Learning with Function Approximation

Hao-Lun Hsu HAO-LUN.HSU@DUKE.EDU
Department of Computer Science
Duke University

Miroslav Pajic MIROSLAV.PAJIC@DUKE.EDU
Department of Electrical and Computer Engineering
Duke University

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Cooperative multi-agent reinforcement learning (MARL) has demonstrated significant promise in dynamic control environments, where effective communication and tailored exploration strategies facilitate collaboration. However, ensuring safe exploration remains challenging, as even a single unsafe action from any agent may result in catastrophic consequences. To mitigate this risk, we introduce Scoop-LSVI, a UCB-based cooperative parallel RL framework that achieves low cumulative regret under minimal communication overhead while adhering to safety constraints. Scoop-LSVI enables multiple agents to solve isolated Markov Decision Processes (MDPs) concurrently and share information to enhance collective learning efficiency. We establish a regret bound of $\tilde{O}(\kappa d^{3/2}H^2\sqrt{MK})$, where d is the feature dimension, H is the horizon length, M is the number of agents, K is the number of episodes per agent, and κ quantifies safety constraints. Our result aligns with state-of-the-art findings for unsafe cooperative MARL and matches the regret bound of UCB-based safe single-agent RL algorithms when $M = 1$, highlighting the potential of Scoop-LSVI to support safe and efficient learning in cooperative MARL applications.

Keywords: Reinforcement learning, Multi-agent, Upper confidence bound (UCB)

1. Introduction

Cooperative multi-agent reinforcement learning (MARL) systems have received substantial attention across dynamic and control settings, such as robotics (Ding et al., 2020; Liu et al., 2019), gaming (Tsay et al., 2011; Zhao et al., 2019; Ye et al., 2020), and energy systems (Yeh et al., 2023; Hsu et al., 2024). In these domains, agents can leverage the advantages of cooperation through effective communication and individualized exploration strategies. A core challenge in such systems lies in ensuring that agents not only benefit from shared information but also autonomously acquire new knowledge through their own interactions with the environment, thereby maintaining a careful balance in the exploration-exploitation trade-off.

One prominent approach to address this exploration-exploitation dilemma is the *Optimism in the Face of Uncertainty* (OFU) principle (Abbasi-Yadkori et al., 2011). The OFU framework has inspired a range of upper confidence bound (UCB) algorithms that have been successfully applied to contextual bandits (Chu et al., 2011; Abbasi-Yadkori et al., 2011; Li et al., 2017), single-agent reinforcement learning (RL) (Chu et al., 2011; Abbasi-Yadkori et al., 2011; Li et al., 2017), and, more recently, cooperative MARL (Dubey and Pentland, 2021; Min et al., 2023). Despite these advances, real-world applications often impose stringent safety constraints: unrestricted exploration can lead

to severe, and sometimes catastrophic, failures (Hsu et al., 2022). In multi-agent settings, this risk is compounded, as the simultaneous exploration by multiple agents amplifies the potential for unsafe behaviors. This challenge raises a critical question: **How can we guarantee safe collaboration in MARL systems while preserving exploration efficiency?**

In this work, we propose Scoop-LSVI, a UCB-based cooperative parallel RL framework that achieves low total regret with minimal communication cost, while rigorously respecting safety constraints. Parallel RL extends the capabilities of heterogeneous federated learning (Li et al., 2020) and broadens the scope of transfer learning in RL (Taylor and Stone, 2009) to support multiple sources (Yao and Doretto, 2010). In our approach, multiple agents concurrently operate over isolated Markov Decision Processes (MDPs) and periodically exchange information to accelerate collective learning.

Specifically, Scoop-LSVI attains a regret bound of $\tilde{O}(\kappa d^{3/2}H^2\sqrt{MK})$, where d is the feature dimension, H is the horizon length, M is the number of agents, K is the number of episodes for each agent, and κ is a constant characterizing the safety constraints. Our regret bound matches the state-of-the-art results for unsafe cooperative MAR with regret of $\tilde{O}(d^{3/2}H^2\sqrt{MK})$ (Dubey and Pentland, 2021; Min et al., 2023; Hsu et al., 2024) and recovers existing safe single-agent bounds when $M = 1$ with $\tilde{O}(\kappa d^{3/2}H^2\sqrt{K})$[1] (Amani et al., 2021).

Notations We denote $[n] := \{1, 2, \ldots, n\}$ for any positive integer n. The $d \times d$ identity matrix is denoted by $\mathbf{I}$. Vectors are represented by bold lowercase letters while matrices are denoted by bold uppercase letters. For any vector $\mathbf{x} \in \mathbb{R}^d$ and a positive semidefinite matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, we define $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma}\mathbf{x}}$. The normalized vector is given by $\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. We define the span of the safe feature $\boldsymbol{\phi}(s, a_0(s))$ as $\Theta_s = \text{span}\left(\boldsymbol{\phi}\big(s, a_0(s)\big)\right) := \{\alpha\boldsymbol{\phi}(s, a_0(s)) := \alpha \in \mathbb{R}\}$ and the orthogonal complement of Θ_s as $\Theta_s^\perp := \{\mathbf{y} \in \mathbb{R}^d : \langle \mathbf{y}, \mathbf{x}\rangle = 0, \forall \mathbf{x} \in \Theta_s\rangle\}$. For any $\mathbf{x} \in \mathbb{R}^d$, we define the projection of $\mathbf{x}$ onto Θ_s as $\Phi_0(s, \mathbf{x}) := \left\langle \mathbf{x}, \tilde{\boldsymbol{\phi}}(s, a_0(s))\right\rangle \tilde{\boldsymbol{\phi}}(s, a_0(s))$ and the projection of $\mathbf{x}$ onto the orthogonal subspace $\Theta_s^\perp$ as $\Phi_0^\perp(s, \mathbf{x}) := \mathbf{x} - \Phi_0(s, \mathbf{x})$. For convenience, we introduce the shorthand $\boldsymbol{\phi}_h^k := \boldsymbol{\phi}(s_h^k, a_h^k)$. Additionally, for two positive sequences $\{a_n\}$ and $\{b_n\}$ with $n = 1, 2, \ldots$, we write $a_n = \mathcal{O}(b_n)$ if there exists an absolute constant $C > 0$ such that $a_n \leq Cb_n$ holds for all $n \geq 1$. We also use $\widetilde{\mathcal{O}}$ to further hide polylogarithmic terms.

2. Problem Formulation

In parallel Markov Decision Processes (MDPs), a set of M agents independently interact with their respective discrete-time MDPs. For each agent $m \in \mathcal{M}$, the MDP is defined by the tuple $\text{MDP}(\mathcal{S}, \mathcal{A}, H, \mathbb{P}_m, r_m, c_m)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, and H represents the time horizon. The transition kernels, $\mathbb{P}_m = \{\mathbb{P}_{m,h}\}_{h\in[H]}$, describe the probabilistic state transitions. At each time step $h \in [H]$, the transition kernel $\mathbb{P}_{m,h}(s'|s, a)$ specifies the probability distribution over the next state s', given the current state-action pair (s, a). The reward and safety constraint functions are denoted by $r_m = \{r_{m,h}\}_{h\in[H]}$ and $c_m = \{c_{m,h}\}_{h\in[H]}$, respectively, where each $r_{m,h} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and $c_{m,h} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ are deterministic. While the agents share the same state and action spaces, their reward functions, safety constraint functions, and transition dynamics can vary.

1. Their original result is written as $\tilde{O}(\kappa\sqrt{d^3H^3T}$, where κ, d, H share the same definition as our variables while T is the total number of steps, i.e., $T = HK$.

We consider a setting in which $\mathcal{S}$ and $\mathcal{A}$, are known, but the reward functions $r_{m,h}$, safety constraints $c_{m,h}$, and transition dynamics $\mathbb{P}_{m,h}(s'|s,a)$ are unknown and must be learned online. In each episode $k \in [K]$, the environment assigns an initial state $s^k_{m,1}$ to each agent $m \in \mathcal{M}$. Over the course of the episode, each agent m interacts with its environment by observing its current state $s^k_{m,h}$, selecting an action $a^k_{m,h}$, receiving a reward $r_{m,h}(s^k_{m,h}, a^k_{m,h})$, and observing a noise-perturbed safety measure $z^k_{m,h} = c_{m,h}(s^k_{m,h}, a^k_{m,h}) + \epsilon^k_{m,h}$, where $\epsilon^k_{m,h}$ represents random noise. The agent then transitions to the next state $s^k_{m,h+1}$ based on the probability distribution $\mathbb{P}_{m,h}(s^k_{m,h+1}|s^k_{m,h}, a^k_{m,h})$. The episode terminates at step $H+1$, at which point the reward defaults to 0. A *randomized* policy $\pi_m = \{\pi_{m,h}\}_{h\in[H]}$ is defined as a sequence of decision rules where $\pi_{m,h} : \mathcal{S} \to \Delta_{\mathcal{A}}$ maps states to distributions over actions given agent m and step h.

Safety Constraint Given the safety-critical nature of the environment, each agent must ensure that its actions comply with safety constraints. Specifically, at each time step $h \in [H]$ of episode k, the agent must select a *safe* action $a^k_{m,h}$ such that the expected safety measure satisfies:

$$\mathbb{E}_{a^k_{m,h}\sim\pi^k_{m,h}} c_{m,h}(s^k_{m,h}, a^k_{m,h}) \le \vartheta \tag{2.1}$$

with high probability, where ϑ is a known safety threshold. We accordingly define the unknown set of safe policies by

$$\Pi^{\text{safe}} := \left\{\pi = \left\{\pi_m : \{\pi_{m,h}(s) \in \mathcal{A}^{\text{safe}}_{m,h}(s)\}_{h\in[H]}\right\}_{m\in\mathcal{M}}, \forall (m,s,h) \in \mathcal{M}\times\mathcal{S}\times[H]\right\}, \tag{2.2}$$

where $\mathcal{A}^{\text{safe}}_{m,h}(s) := \left\{\theta \in \Delta_{\mathcal{A}} : \mathbb{E}_{a\sim\theta} c_{m,h}(s,a) \le \vartheta\right\}$. Thus, after observing the current state $s^k_{m,h}$, the agent must choose an action that remains within the set $\mathcal{A}^{\text{safe}}_{m,h}(s^k_{m,h})$, ensuring safety compliance with high probability.

Value Functions and Optimal Policies For each agent $m \in \mathcal{M}$, the cumulative expected reward under a safe policy $\pi \in \Pi^{\text{safe}}$, also known as the value function $V^\pi_{m,h}(s) : \mathcal{S} \to \mathbb{R}$, is defined as:

$$V^\pi_{m,h}(s) := \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r_{m,h'}(s_{m,h'}, a_{m,h'})|s_{m,h} = s\right] \tag{2.3}$$

Similarly, the state-action value function, $Q^\pi_{m,h} : \mathcal{S}\times\mathcal{A}^{\text{safe}}_{m,h}(\cdot) \to \mathbb{R}$, for a safe policy $\pi \in \Pi^{\text{safe}}$ is given by:

$$Q^\pi_{m,h}(s,a) := \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r_{m,h'}(s_{m,h'}, a_{m,h'})|s_{m,h} = s, a_{m,h} = a\right] \tag{2.4}$$

for any $(h,s,a) \in [H]\times\mathcal{S}\times\mathcal{A}$. The expectation in the definition of both value and Q functions for a policy π are over both the environment and the randomness of policy π. The optimal safe policy for each agent $m \in \mathcal{M}$, denoted by π^*_m, maximizes the cumulative reward while adhering to the safety constraints. The corresponding optimal value function is denoted as $V^{\pi^*_m}_{m,h}(s) := V^*_{m,h}(s) = \sup_{\pi_m\in\Pi^{\text{safe}}} V^\pi_{m,h}(s)$. Thus, for all $(m,s,h) \in \mathcal{M}\times\mathcal{S}\times[H]$ and $a \in \mathcal{A}^{\text{safe}}_{m,h}(s)$, the Bellman equations for an arbitrary safe policy $\pi \in \Pi^{\text{safe}}$ and the optimal safe policy are expressed as:

$$Q^\pi_{m,h}(s,a) = r_{m,h}(s,a) + \mathbb{P}_{m,h}V^\pi_{m,h+1}(s,a), \quad V^\pi_{m,h}(s) = \mathbb{E}_{a\sim\pi}[Q^\pi_{m,h}(s,a)] \tag{2.5}$$

$$Q^*_{m,h}(s,a) = r_{m,h}(s,a) + \mathbb{P}_{m,h}V^*_{m,h+1}(s,a), \quad V^*_{m,h}(s) = \max_{\theta \in \mathcal{A}^{\text{safe}}_{m,h}(s)} \mathbb{E}_{a\in\theta}[Q^*_{m,h}(s,a)] \tag{2.6}$$

Here, $V^{\pi}_{m,H+1}(s) = V^*_{m,H+1}(s) = 0$ at the terminal step. The primary goal of each agent is to minimize cumulative regret across K episodes while maintaining adherence to the safety constraints. The regret is defined as the difference between the optimal value function and the achieved value under the chosen policies, summed over all agents and episodes:

$$\text{Regret}(K) = \sum_{m\in\mathcal{M}} \sum_{k=1}^{K} \left[V^*_{m,1}(s^k_{m,1}) - V^{\pi^k_m}_{m,1}(s^k_{m,1})\right].$$

with the constraint that $\pi^k_m \in \Pi^{\text{safe}}$ for all $k \in [K]$ and $m \in \mathcal{M}$ with high probability.

3. Scoop-LSVI

We first present a unified safe cooperative parallel learning framework, *Safe cooperative Least-Square Value Iteration* (Scoop-LSVI, Algorithm 1). Then we discuss the instantiation of our algorithm in the linear structure.

Algorithm 1 Scoop-LSVI

Initialization: set $U^{\text{ser}}_h(k), U^{\text{loc}}_{m,h}(k) = \emptyset$.
for episode $k = 1, ..., K$ **do**
  for agent $m \in \mathcal{M}$ **do**
    Receive initial state $s^k_{m,1}$.
    for step $h = 1, ..., H$ **do**
      Compute $\mathcal{A}^k_{m,h}(s) \forall s \in \mathcal{S}$, $Q^k_{m,h}(s,a) \forall (s,a) \in \mathcal{S} \times \mathcal{A}^k_{m,h}(\cdot)$.
    end for
    for step $h = 1, ..., H$ **do**
      $a^k_{m,h} \leftarrow \arg\max_{\theta \in \mathcal{A}^k_{m,h}(s^k_{m,h})} \mathbb{E}_{a\sim\theta}[Q^k_{m,h}(s^k_{m,h}, a)]$.
      Receive $s^k_{m,h+1}$, $r_{m,h}$ and $z_{m,h}$.
      $U^{\text{loc}}_{m,h}(k) \leftarrow U^{\text{loc}}_{m,h}(k) \bigcup \left(s^k_{m,h}, a^k_{m,h}, s^k_{m,h+1}\right)$.
      if Condition, then SYNCHRONIZE $\leftarrow$ True.
    end for
  end for
  if SYNCHRONIZE **then**
    for step $h = H, ..., 1$ **do**
      $\forall$ *AGENT*: Send $U^{\text{loc}}_{m,h}(k)$ to *SERVER*.
      SERVER: $U^{\text{loc}}_h(k) \leftarrow \bigcup_{m\in\mathcal{M}} U^{\text{loc}}_{m,h}(k)$.
      SERVER: $U^{\text{ser}}_h(k) \leftarrow U^{\text{ser}}_h(k) \bigcup U^{\text{loc}}_h(k)$.
      SERVER: Send $U^{\text{ser}}_h(k)$ to each *AGENT*.
      $\forall$ *AGENT*: Set $U^{\text{loc}}_{m,h}(k) \leftarrow \emptyset$.
    end for
  end if
end for

3.1. Algorithm Interpretation

We present our *Safe cooperative Least- Square Value Iteration* (Scoop-LSVI), as outlined in Algorithm 1. In this setup, each agent executes Least-Square Value Iteration (LSVI) in parallel, making decisions based on both local and globally synchronized data. Communication between agents and a central server plays a pivotal role in ensuring collective decision-making. Prior to discussing the detailed workings of the algorithm, we first define the notations related to the datasets stored locally on each agent and at the central server following the settings in Hsu et al. (2024).

Index notation Let k_s represent the most recent episode before episode k where synchronization occurs. For each episode k at step h, we define three datasets:

$$U_h^{\text{ser}}(k) = \left\{\left(s_{n,h}^{\tau}, a_{n,h}^{\tau}, s_{n,h+1}^{\tau}\right)\right\}_{n\in\mathcal{M},\tau\in[k_s]}, \tag{3.1a}$$

$$U_{m,h}^{\text{loc}}(k) = \left\{\left(s_{m,h}^{\tau}, a_{m,h}^{\tau}, s_{m,h+1}^{\tau}\right)\right\}_{\tau=k_s+1}^{k-1}, \tag{3.1b}$$

$$U_{m,h}(k) = U_h^{\text{ser}}(k) \bigcup U_{m,h}^{\text{loc}}(k). \tag{3.1c}$$

Here, $U_h^{\text{ser}}(k)$ represents the dataset synchronized across all agents following the most recent episode k_s. The dataset $U_{m,h}^{\text{loc}}(k)$ captures the unique data collected by agent m since the last synchronization. The combined dataset $U_{m,h}(k)$ contains all the data available to agent m at the current episode.

We define $\mathcal{K}(k) = |U_{m,h}(k)|$ as the total number of data points, and for clarity, re-order the data points in $U_{m,h}(k)$, re-naming each tuple $(s_{m,h}^{\tau}, a_{m,h}^{\tau}, s_{m,h+1}^{\tau})$ as (s^l, a^l, s'^l). Thus, the dataset can be represented $U_{m,h}(k) = \bigcup_{l=1}^{\mathcal{K}(k)}(s^l, a^l, s'^l)$. This recording is achieved via the following one-to-one mapping:

$$l_{m,k}(n,\tau) = \begin{cases} (\tau-1)M + n & \tau \le k_s, \\ (M-1)k_s + \tau & k_s < \tau \le k-1. \end{cases} \tag{3.2}$$

We will use both indices $(s, a, s') \in U_{m,h}(k)$ and $l \in [\mathcal{K}(k)]$ interchangeably for the summation over set $U_{m,h}(k)$. In our framework, each episode k in Algorithm 1 comprises two stages that are crucial for maintaining both the local autonomy of agents and global synchronization. The first stage (Lines 3-14) involves independent computations by each agent, while the second stage (Lines 15-23) facilitates communication among agents and the server. This structure is designed to promote decentralized learning during exploration while periodically synchronizing agents to maintain a cohesive learning process. A significant distinction between Scoop-LSVI and Coop-LSVI (Dubey and Pentland, 2021) lies in the safety requirement for the actions chosen by agents. In Scoop-LSVI, it is essential that the action $a_{m,h}^k$ selected at any step h during the episode must always belong to the unknown safe set $\mathcal{A}_{m,h}^{\text{safe}}(s)$.

In the first stage (Lines 3-14) of Algorithm 1, each agent m operates independently, carrying out two key tasks. First, every agent computes a set, $\mathcal{A}_{m,h}^k(s)$, for all $s \in \mathcal{S}$ in Line 6, which we will show is guaranteed to be a subset of the unknown safe set, $\mathcal{A}_{m,h}^{\text{safe}}(s)$, in the descriptions of (3.7). Once the safe action set is determined, the agent proceeds to update the Q functions $\{Q_{m,h}^k\}_{h=1}^H$ through LSVI in Line 6. In the second part (Lines 8-13), after determining the safe action sets and obtaining the estimated Q functions, in each step h we execute the greedy policy with respect to $Q_{m,h}^k$ and collect new data points which are added to the local dataset $U_{m,h}^{\text{loc}}(k)$ (Lines 9-11). Then we verify the synchronization condition (Line 12). In this paper, we mainly consider if we

have a feature mapping $\phi(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, based on (3.1), we define the following empirical covariance matrices.

$$
\begin{aligned}
{}^{\text{ser}}\mathbf{\Lambda}_h^k &= \textstyle\sum_{(s^l,a^l,s'^l)\in U_h^{\text{ser}}(k)} \boldsymbol{\phi}^l {\boldsymbol{\phi}^l}^\top, \\
{}^{\text{loc}}\mathbf{\Lambda}_{m,h}^k &= \textstyle\sum_{(s^l,a^l,s'^l)\in U_{m,h}^{\text{loc}}(k)} \boldsymbol{\phi}^l {\boldsymbol{\phi}^l}^\top, \\
\mathbf{\Lambda}_{m,h}^k &= {}^{\text{ser}}\mathbf{\Lambda}_h^k + {}^{\text{loc}}\mathbf{\Lambda}_{m,h}^k + \lambda \mathbf{I}.
\end{aligned}
$$

where $\boldsymbol{\phi}^l = \boldsymbol{\phi}\big(s^l, a^l\big)$. We synchronize as long as the following condition is met,

$$
\log \frac{\det\left({}^{\text{ser}}\mathbf{\Lambda}_h^k + {}^{\text{loc}}\mathbf{\Lambda}_{m,h}^k + \lambda\mathbf{I}\right)}{\det\left({}^{\text{ser}}\mathbf{\Lambda}_h^k + \lambda\mathbf{I}\right)} \geq \frac{S}{(k-k_s)}, \tag{3.3}
$$

where S is a communication control factor.

The second stage (Lines 15-23) is executed only when the synchronization condition is satisfied. Each agent uploads its local transition set $U_{m,h}^{\text{loc}}(k)$, i.e., the newly collected local data after the last synchronization, to the server. The server gathers all information together in $U_h^{\text{ser}}(k)$ and sends it back to all agents. Finally, each agent resets the local transition set $U_{m,h}^{\text{loc}}(k) \leftarrow \emptyset$. Consequently, agent m gains access to an updated dataset $U_{m,h}(k) = U_h^{\text{ser}}(k) \bigcup U_{m,h}^{\text{loc}}(k)$, which contains the global historical data of all agents up to last synchronization along with its local transitions.

3.2. Instantiation in the Linear Function Class

Definition 3.1 (Linear MDP (Jin et al., 2020; Amani et al., 2021)) *An MDP $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$ is a linear MDP with feature map $\boldsymbol{\phi} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown measures $\boldsymbol{\mu}_h = (\mu_h^1, ..., \mu_h^d)$ over $\mathcal{S}$ and unknown vectors $\boldsymbol{\theta}_h, \boldsymbol{\gamma}_h \in \mathbb{R}^d$ such that for any $(s,a) \in \mathcal{S}\times\mathcal{A}$,*

$$
\mathbb{P}_h(\cdot|s,a) = \big\langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h(\cdot) \big\rangle, \quad r_h(s,a) = \big\langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h \big\rangle, \quad c_h(s,a) = \big\langle \boldsymbol{\phi}(s,a), \boldsymbol{\gamma}_h \big\rangle.
$$

Assumption 3.2 (Non-empty safe sets (Amani et al., 2021)) *There exists a known safe action $a_0(s)$ such that $a_0 \in \mathcal{A}_h^{safe}(s)$ with unknown safety measure $\vartheta_h(s) := \left\langle \boldsymbol{\phi}(s, a_0(s)), \boldsymbol{\gamma}_h \right\rangle < \vartheta$ for all $(s,h) \in \mathcal{S}\times[H]$.[2] The notations can be extended to any agent $m \in \mathcal{M}$ with $\mathcal{A}_{m,h}^{safe}(s)$ and $\vartheta_{m,h}(s)$.*

The construction of $\mathcal{A}_{m,h}^k(s)$ in Line 6 requires an appropriate confidence set around the unknown parameter $\boldsymbol{\gamma}_h$, which defines the safety constraints (see Definition 3.1). Given that the agent knows $\vartheta_{m,h}(s)$ (see Definition 3.2), it can compute $z_{m,h,s}^k := \left\langle \Phi_0^\perp(s, \boldsymbol{\phi}_h^k), \Phi_0^\perp(s, \boldsymbol{\gamma}_h) \right\rangle + \epsilon_{m,h}^k = z_{m,h}^k - \frac{\left\langle \boldsymbol{\phi}_h^k, \tilde{\boldsymbol{\phi}}(s,a_0(s)) \right\rangle}{\|\boldsymbol{\phi}(s,a_0(s)\|} \vartheta_{m,h}(s)$, representing the cost incurred by $a_{m,h}^k$ along the subspace $\Theta_s^\perp$, which is orthogonal to $\boldsymbol{\phi}(s,a_0(s))$. Thus, the agent does not need to construct confidence sets around $\boldsymbol{\gamma}_h$ along the normalized safe feature vector, $\tilde{\boldsymbol{\phi}}(s,a_0(s))$. Instead, it only builds the confidence sets around the component $\Phi_0^\perp(s,\boldsymbol{\gamma}_h)$, which lines in the orthogonal direction to $\tilde{\boldsymbol{\phi}}(s,a_0(s))$ (Amani et al., 2021):

$$
C_{m,h}^k(s) := \Big\{ \boldsymbol{\upsilon} \in \mathbb{R}^d : \{\|\boldsymbol{\upsilon} - \boldsymbol{\gamma}_{m,h,s}^k\|\}_{\mathbf{\Lambda}_{m,h,s}^k} \leq \beta \Big\}, \tag{3.4}
$$

2. $\vartheta_h(s)$ can be unknown by playing $a_0(s)$ with adaptive rounds to construct a conservative estimator for the gap $\vartheta - \vartheta_h(s)$ in Appendix A.4 in (Amani et al., 2021).

where $\boldsymbol{\gamma}^k_{m,h,s} := (\boldsymbol{\Lambda}^k_{m,h,s})^{-1}\boldsymbol{r}^k_{m,h,s}$ denotes the regularized least-square estimator of $\Phi_0^{\perp}(s,\boldsymbol{\gamma}_h)$ computed by the inverse of Gram matrix

$$\boldsymbol{\Lambda}^k_{m,h,s} := \lambda\Big(\mathbf{I} - \tilde{\boldsymbol{\phi}}(s,a_0(s))\tilde{\boldsymbol{\phi}}^{\top}(s,a_0(s))\Big) + \sum_{U_{m,h}(k)} \Phi_0^{\perp}\Big(s,\boldsymbol{\phi}^l\Big)\Phi_0^{\perp,\top}\Big(s,\boldsymbol{\phi}^l\Big)$$

and $\boldsymbol{r}^k_{m,h,s} := \sum_{U_{m,h}(k)} z^l_s \Phi_0^{\perp}\Big(s,\boldsymbol{\phi}^l\Big)$. Note that z^l_s follows the one-to-one mapping from $z^k_{m,h,s}$ in (3.2). The exploration factor β, which will be defined in Theorem 4.3, is chosen to ensure that the event

$$\varepsilon_1 := \Big\{\Phi_0^{\perp}(s,\boldsymbol{\gamma}_h) \in C^k_{m,h}(s), \forall (m,s,h,k) \in \mathcal{M}\times\mathcal{S}\times[H]\times[K]\Big\} \tag{3.5}$$

indicating that $\Phi_0^{\perp}(s,\boldsymbol{\gamma}_h)$ belongs to the confidence sets $C^k_{m,h}(s)$ with high probability. In this implementation, β is treated as a tuning parameter. Under the assumption that event ε_1 holds, the agent proceeds to compute the following inner approximations of the true unknown safe sets $\mathcal{A}^{\text{safe}}_{m,h}$ for all $s \in \mathcal{S}$:

$$\begin{aligned}\mathcal{A}^k_{m,h}(s) = \Big\{\theta \in \Delta_A : \mathbb{E}_{a\sim\theta}\Big[\frac{\Big\langle \Phi_0(s,\boldsymbol{\phi}(s,a)), \tilde{\boldsymbol{\phi}}(s,a_0(s))\Big\rangle}{\|\boldsymbol{\phi}(s,a_0(s))\|}\vartheta_{m,h}(s)\Big] \\ + \max_{\boldsymbol{\upsilon}\in C^k_{m,h}(s)}\Big\langle \Phi_0^{\perp}(s,\mathbb{E}_{a\sim\theta}[\boldsymbol{\phi}(s,a)]),\boldsymbol{\upsilon}\Big\rangle \le \vartheta\Big\}\end{aligned} \tag{3.6}$$

$$\begin{aligned}= \Big\{\theta \in \Delta_A : \underbrace{\frac{\Big\langle \Phi_0(s,\boldsymbol{\phi}^{\theta}(s)), \tilde{\boldsymbol{\phi}}(s,a_0(s))\Big\rangle}{\|\boldsymbol{\phi}(s,a_0(s))\|}\vartheta_{m,h}(s)}_{\textbf{(v1)}} \\ + \underbrace{\Big\langle \boldsymbol{\gamma}^k_{m,h,s}, \Phi_0^{\perp}(s,\boldsymbol{\phi}^{\theta}(s))\Big\rangle + \beta\|\Phi_0^{\perp}(s,\boldsymbol{\phi}^{\theta}(s))\|_{(\boldsymbol{\Lambda}^k_{m,h,s})^{-1}}}_{\textbf{(v2)}} \le \vartheta\Big\}\end{aligned} \tag{3.7}$$

Note that (**v1**) in (3.7) represents the known cost of action a at state s along direction $\tilde{\boldsymbol{\phi}}(s,a_0(s))$ and (**v2**) corresponds to its maximum possible cost in the orthogonal space $\Theta^{\perp}_s$. Thus, these terms provide a high probability upper bound on the true unknown cost, implying that $A^k_{m,h}(s) \subset A^{\text{safe}}_{m,h}(s)$.

Proposition 3.3 *Conditioned on ε_1, for all $(m,s,h,k) \in \mathcal{M}\times\mathcal{S}\times[H]\times[K]$, it holds that $\langle \boldsymbol{\phi}(s,a), \boldsymbol{\gamma}_h\rangle \le \vartheta$, $\forall a \in \mathcal{A}^k_{m,h}(s)$ in each own MDP.*

Therefore, conditioned on ε_1, the action $a^k_{m,h}$ drawn from distribution $\mathcal{A}^k_{m,h}(s^k_{m,h})$ in Line 9 in Algorithm 1 does not violate the safety constraint. Note that $\mathcal{A}^k_{m,h}(s)$ is always not empty from Assumption 3.2, so the safe action $a_0(s)$ is always in $\mathcal{A}^k_{m,h}(s)$. Due to the estimation of safe sets $\mathcal{A}^k_{m,h}(s)$ and linear structure of the MDP, we can parameterize $Q^*_{m,h}(s,a)$ by a linear form $\langle \mathbf{w}^*_{m,h}, \boldsymbol{\phi}(s,a)\rangle$. Then for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, each agent m computes

$$Q^k_{m,h}(s,a) = \min\Big\{\big\langle \mathbf{w}^k_{m,h}, \boldsymbol{\phi}(s,a)\big\rangle + \kappa_h(s)\beta\|\boldsymbol{\phi}(s,a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}, H\Big\}, \tag{3.8}$$

where $\mathbf{w}^k_{m,h} := (\mathbf{\Lambda}^k_{m,h})^{-1}\mathbf{b}^k_{m,h}$ is the regularized least-squares estimator of $\mathbf{w}^*_{m,h}$ computed by the Gram matrix $\mathbf{\Lambda}^k_{m,h} := \lambda\mathbf{I} + \sum_{l=1}^{\mathcal{K}(k)} \boldsymbol{\phi}^l \boldsymbol{\phi}^{l\top}$ and $\mathbf{b}^k_{m,h} := \sum_{l=1}^{\mathcal{K}(k)} \boldsymbol{\phi}^l \Big[r^l + \max_{\theta \in \mathcal{A}^k_{m,h+1}(s'^l)} \mathbb{E}_{a\sim\theta}[Q^k_{m,h+1}(s'^l, a)]\Big]$. Here, $\kappa_h(s)\beta\|\boldsymbol{\phi}(s,a)\|_{(\mathbf{\Lambda}^k_{m,h})^{-1}}$ is an exploration bonus, where $\kappa_h(s) > 1$ encourages sufficient exploration regarding the uncertainty about c, while β incentivizes exploration with respect to the uncertainty about r and $\mathbb{P}$.

Communication cost In the linear function class, the closed-form solution for $\mathbf{w}^k_{m,h}$ is given by $(\mathbf{\Lambda}^k_{m,h})^{-1}\mathbf{b}^k_{m,h}$. During synchronization (Lines 15-23) in Algorithm 1, agents only transmit local statistics ${}^{\text{loc}}\mathbf{\Lambda}^k_{m,h}$ and ${}^{\text{loc}}\mathbf{b}^k_{m,h}$ to collaboratively solve the regression problem. The total number of communication rounds between the agents and the server in Algorithm 1 is bounded by $\text{CPX} = \widetilde{\mathcal{O}}((d + K/S)MH)$, where S is a communication control factor introduced in (3.3). The number of transmitted random bits grows logarithmically with episodes K, matching the result of Hsu et al. (2024). This bound leverages interval-based synchronization and matrix determinant properties for tighter estimates. Notably, incorporating safety constraints does not increase communication complexity, as hard constraints are enforced for each agent $m \in \mathcal{M}$.

4. Theoretical Analysis

Assumption 4.1 (Sub-Gaussioan Noise) *For all $(m,h,k) \in \mathcal{M}\times[H]\times[K]$, $\epsilon^k_{m,h}$ is a zero-mean σ-sub-Gaussian random variable.*

Assumption 4.2 (Boundedness) *Without loss of generality, we assume that $\|\boldsymbol{\phi}(s,a)\| \le 1$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ and $\max\{\|\boldsymbol{\mu}_h(\mathcal{S})\|, \|\boldsymbol{\theta}_h\|, \|\boldsymbol{\gamma}_h\|\} \le \sqrt{d}$ for all $h \in [H]$.*

Theorem 4.3 (Regret of Scoop-LSVI) *In Algorithm 1, under Definition 3.1, Assumptions 3.2, 4.1, 4.2 and the determinant synchronization condition (3.3), there exists an absolute constant $c_\beta > 0$ such that for any fixed $p \in (0, 1/3)$, if we set $\beta := c_\beta dH\sqrt{2\log(\frac{dMKH}{p})}$ and $\lambda = 1$, then with probability at least $1-3p$, we obtain the cumulative regret: $\text{Regret}(K) = \tilde{O}(\kappa dH^2(M\sqrt{S} + \sqrt{dMK}))$*

Remark 4.4 *When we choose $S = O(Kd/M)$ in the synchronization condition (3.3), the cumulative regret of Scoop-LSVI becomes $\tilde{O}(\kappa d^{3/2}H^2\sqrt{MK})$, which matches the result of UCB exploration without safety constraints (Dubey and Pentland, 2021). When $M = 1$, the regret becomes $\tilde{O}(\kappa d^{3/2}H^2\sqrt{K})$, which matches the existing safe single-agent result (Amani et al., 2021). Note that if there is no communication at all and agents act independently, with the same number of learning rounds (or samples), the cumulative regrets becomes $\tilde{O}(M\cdot\kappa d^{3/2}H^2\sqrt{K})$. By incorporating communication, our regret bound in Theorem 4.3 is lower than that of the independent setting by a factor of $\sqrt{M}$ with safety constraints.*

Theorem 4.5 ((Abbasi-Yadkori et al., 2011; Jin et al., 2020)) *For any policy π and $\forall(m,a,s,h,k) \in \mathcal{M}\times\mathcal{A}\times\mathcal{S}\times[H]\times[K]$, define the event*

$$\varepsilon_2 := \Big\{\Big|\big\langle \mathbf{w}^k_{m,h}, \phi(s,a)\big\rangle - Q^\pi_{m,h}(s,a) + \mathbb{P}_{m,h}(V^\pi_{m,h+1} - V^k_{m,h+1})(s,a)\Big| \le \beta\|\phi(s,a)\|_{(\mathbf{\Lambda}^k_{m,h})^{-1}}\Big\},$$

and recall the definition of ε_1 in (3.5). Then, under Definition 3.1, Assumptions 3.2, 4.1, 4.2 and the definition of β in Theorem 4.3, there exists an absolute constant $c_\beta > 0$, such that for any fixed $p \in (0, 1/3)$, with probability at least $1 - p$, the event $\varepsilon := \varepsilon_1 \cap \varepsilon_2$ holds.

Lemma 4.6 (Optimism in the face of safety constraint (Amani et al., 2021)) *. Let Definition 3.1, Assumptions 3.2, 4.1, 4.2 hold and $\kappa_h(s) := \frac{2H}{\vartheta - \vartheta_h(s)} + 1$. Then, conditioned on $\varepsilon := \varepsilon_1 \cap \varepsilon_2$, it holds that $V_h^*(s) \leq V_h^k(s), \forall (s, h, k) \in \mathcal{S} \times [H] \times [K]$.*

As the conclusion from Lemma 4.6 and direct extension toward parallel MDPs, we show that $Q^*_{m,h}(s, a) \leq Q^k_{m,h}(s, a), \forall (m, a, s, h, k) \in \mathcal{M} \times \mathcal{A} \times \mathcal{S} \times [H] \times [K]$. This highlights the UCB nature of Scoop-LSVI, allowing us to exploit the standard analysis of unsafe Coop-LSVI (Dubey and Pentland, 2021) to establish the regret bound.

Lemma 4.7 (Variance control via communication in homogeneous factored environments) *Let Algorithm 1 be run for any $K > 0$ and $\mathcal{M} \geq 1$, with S as the communication control factor. Then the following holds for the accumulated variance.*

$$\sum_{m \in \mathcal{M}} \sum_{k=1}^{K} \|\boldsymbol{\phi}(s, a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}} \leq 2 \log\left(\frac{\det(\boldsymbol{\Lambda}^k_h)}{\det(\lambda \mathbf{I})}\right)\left(\frac{M}{log2}\right)\sqrt{S} + 2\sqrt{2MK\left(\frac{\det(\boldsymbol{\Lambda}^k_h)}{\det(\lambda \mathbf{I})}\right)} \tag{4.1}$$

4.1. Proof Sketches of Theorem 4.3

Let $\delta^k_{m,h} := V^k_{m,h}(s^k_{m,h}) - V^{\pi^k_m}_{m,h}(s^k_{m,h})$ and $\zeta^k_{m,h+1} := \mathbb{E}_{a \sim \pi^k_{m,h}}\left[\left(\mathbb{P}_{m,h}\left(V^k_{m,h+1} - V^{\pi^k_m}_{m,h+1}\right)\right)(s^k_{m,h}, a)\right] - \delta^k_{m,h+1}$. Conditioned on ε_2, we can write

$$\begin{aligned}
\delta^k_{m,h} &= V^k_{m,h}(s^k_{m,h}) - V^{\pi^k_m}_{m,h}(s^k_{m,h}) \\
&= \min\left\{\max_{\theta \in \mathcal{A}^k_{m,h}(s^k_{m,h})} \mathbb{E}_{a \sim \theta}[Q^k_{m,h}(s^k_{m,h}, a)], H\right\} - \mathbb{E}_{a \sim \pi^k_{m,h}}[Q^{\pi^k_m}_{m,h}(s^k_{m,h}, a)] \\
&\leq \max_{\theta \in \mathcal{A}^k_{m,h}(s^k_{m,h})} \mathbb{E}_{a \sim \theta}[Q^k_{m,h}(s^k_{m,h}, a)] - \mathbb{E}_{a \sim \pi^k_{m,h}}[Q^{\pi^k_m}_{m,h}(s^k_{m,h}, a)] \\
&= \mathbb{E}_{a \sim \pi^k_{m,h}}[Q^k_{m,h}(s^k_{m,h}, a)] - \mathbb{E}_{a \sim \pi^k_{m,h}}[Q^{\pi^k_m}_{m,h}(s^k_{m,h}, a)] \\
&\leq \mathbb{E}_{a \sim \pi^k_{m,h}}[\langle \mathbf{w}^k_{m,h}, \boldsymbol{\phi}(s, a)\rangle + \kappa_h(s)\beta\|\boldsymbol{\phi}(s^k_{m,h}, a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}] - \mathbb{E}_{a \sim \pi^k_{m,h}}[Q^{\pi^k_m}_{m,h}(s^k_{m,h}, a)] \\
&\leq \mathbb{E}_{a \sim \pi^k_{m,h}}\left[\left(\mathbb{P}_{m,h}\left(V^k_{m,h+1} - V^{\pi^k_m}_{m,h+1}\right)\right)(s^k_{m,h}, a)\right] \\
&\quad + (1 + \kappa_h(s))\beta\mathbb{E}_{a \sim \pi^k_{m,h}}\left[\|\boldsymbol{\phi}(s^k_{m,h}, a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}\right] \\
&= \delta^k_{m,h+1} + \zeta^k_{m,h+1} + (1 + \kappa_h(s))\beta\mathbb{E}_{a \sim \pi^k_{m,h}}\left[\|\boldsymbol{\phi}(s^k_{m,h}, a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}\right]
\end{aligned} \tag{4.2}$$

Now, conditioning on event $\varepsilon := \varepsilon_1 \cap \varepsilon_2$, we bound the cumulative regret as follows:

$$\begin{aligned}
&\text{Regret}(K) = \sum_{m\in\mathcal{M}}\sum_{k=1}^{K} V^{*}_{m,1}(s^k_{m,1}) - V^{\pi^k_m}_{m,1}(s^k_{m,1}) \leq \sum_{m\in\mathcal{M}}\sum_{k=1}^{K}\delta^k_{m,1} \\
&\leq \sum_{m\in\mathcal{M}}\sum_{k=1}^{K}\sum_{h=1}^{H}\zeta^k_{m,h} + \sum_{m\in\mathcal{M}}\sum_{k=1}^{K}\sum_{h=1}^{H}(1+\kappa_h(s))\beta\mathbb{E}_{a\sim\pi^k_{m,h}}\left[\|\boldsymbol{\phi}(s^k_{m,h},a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}\right] \\
&\leq \sqrt{2H^3MK\log(\frac{1}{p})} + (1+\kappa)\beta\sum_{m\in\mathcal{M}}\sum_{k=1}^{K}\sum_{h=1}^{H}\underbrace{\mathbb{E}_{a\sim\pi^k_{m,h}}\left[\|\boldsymbol{\phi}(s^k_{m,h},a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}\right]}_{\textbf{(v3)}} \\
&\leq \sqrt{2H^3MK\log(\frac{1}{p})} + (1+\kappa)\beta\left(\sum_{h=1}^{H}\sum_{m\in\mathcal{M}}\sum_{k=1}^{K}\|\boldsymbol{\phi}(s,a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}} + \sqrt{\frac{2HMK\log(1/p)}{\lambda}}\right) \\
&\leq \sqrt{2H^3MK\log(\frac{1}{p})} + (1+\kappa)\beta \\
&\quad\left(\sum_{h=1}^{H}\left(2\log\left(\frac{\det(\boldsymbol{\Lambda}^k_h)}{\det(\lambda\mathbf{I})}\right)\left(\frac{M}{log2}\right)\sqrt{S} + 2\sqrt{2MK\left(\frac{\det(\boldsymbol{\Lambda}^k_h)}{\det(\lambda\mathbf{I})}\right)}\right) + \sqrt{\frac{2HMK\log(1/p)}{\lambda}}\right) \\
&\leq \sqrt{2H^3MK\log(\frac{1}{p})} + (1+\kappa)c_\beta dH^2\sqrt{2\log\frac{dMKH}{p}} \\
&\quad\left(2\log(dMK)M\sqrt{S} + 2\sqrt{2dMK\log(MK)} + \sqrt{\frac{2MK\log(1/p)}{\lambda}}\right). \qquad (4.3)
\end{aligned}$$

The second inequality follows from the recursive relation in (4.2), with $\{\zeta^k_{m,h}\}$ as a martingale difference sequence satisfying $\|\zeta^k_{m,h}\| \leq 2H$. Azuma-Hoeffding inequality yields the third inequality (Dubey and Pentland, 2021). To bound (**v3**), we define the martingale difference sequence $\iota^k_{m,h} := \mathbb{E}_{a\sim\pi^k_{m,h}}\left[\|\phi(s^k_{m,h},a)\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}\right] - \|\phi(s^k_{m,h},a^k_{m,h})\|_{(\boldsymbol{\Lambda}^k_{m,h})^{-1}}$, where $|\iota^k_{m,h}| \leq 2/\sqrt{\lambda}$ for any $(m,h,k) \in \mathcal{M}\times[H]\times[K]$. Decomposing (**v3**) into two terms, Azuma-Hoeffding inequality is applied again to bound $\iota^k_{m,h}$, yielding the fourth inequality. The fifth inequality follows from Lemma 4.7, using the determinant-trace inequality and $\|\phi(\cdot)\| \leq 1$ to bound the summation. Finally, substituting $\beta = c_\beta dH\sqrt{2\log\frac{dMKH}{p}}$ from Lemma 5 in (Jin et al., 2020) with an additional factor of H for unified bounds, gives $\text{Regret}(K) = \tilde{O}(\kappa dH^2(M\sqrt{S} + \sqrt{dMK}))$ when $\lambda = 1$.

5. Conclusion

We introduced Scoop-LSVI, a safe cooperative multi-agent RL algorithm that achieves sub-linear regret and communication complexity under linear function approximation. Our analysis demonstrates that incorporating communication reduces the regret bound in Theorem 4.3 by a factor of $\sqrt{M}$ with safety constraints compared to the independent setting. Several open directions emerge from this work, including extending Scoop-LSVI to a fully decentralized network topology, addressing scenarios without prior knowledge of safe action for every state, and relaxing the assumption of linear constraint functions. We hope this study provides a valuable stepping stone for advancing safe and cooperative multi-agent RL research.

Acknowledgments

This work is sponsored in part by the ONR under agreement N00014-23-1-2206, AFOSR under the award number FA9550-19-1-0169, and by the NSF under NAIAD Award 2332744 as well as the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562.

References

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

Sanae Amani, Christos Thrampoulidis, and Lin Yang. Safe reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pages 243–253. PMLR, 2021.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

Guohui Ding, Joewie J Koh, Kelly Merckaert, Bram Vanderborght, Marco M Nicotra, Christoffer Heckman, Alessandro Roncone, and Lijun Chen. Distributed reinforcement learning for cooperative multi-robot object manipulation. In *19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1831–1833. IFAAMAS, 2020.

Abhimanyu Dubey and Alex Pentland. Provably efficient cooperative multi-agent reinforcement learning with function approximation. *arXiv preprint arXiv:2103.04972*, 2021.

Hao-Lun Hsu, Qiuhua Huang, and Sehoon Ha. Improving safety in deep reinforcement learning using unsupervised action planning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5567–5573, 2022.

Hao-Lun Hsu, Weixin Wang, Miroslav Pajic, and Pan Xu. Randomized exploration in cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 2024.

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

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2017.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

Boyi Liu, Lujia Wang, and Ming Liu. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters*, 4(4): 4555–4562, 2019.

Yifei Min, Jiafan He, Tianhao Wang, and Quanquan Gu. Cooperative multi-agent reinforcement learning: asynchronous communication and linear function approximation. In *International Conference on Machine Learning*, pages 24785–24811. PMLR, 2023.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(56):1633–1685, 2009.

Jyh-Jong Tsay, Chao-Cheng Chen, and Jyh-Jung Hsu. Evolving intelligent mario controller by reinforcement learning. In *International Conference on Technologies and Applications of Artificial Intelligence*, pages 266–272, 2011.

Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

Christopher Yeh, Victor Li, Rajeev Datta, Julio Arroyo, Nicolas Christianson, Chi Zhang, Yize Chen, Mohammad Mehdi Hosseini, Azarang Golmohammadi, Yuanyuan Shi, Yisong Yue, and Adam Wierman. Sustaingym: A benchmark suite of reinforcement learning for sustainability applications. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. PMLR, 2023.

Yunqi Zhao, Igor Borovikov, Jason Rupert, Caedmon Somers, and Ahmad Beirami. On multi-agent learning in team sports games. *arXiv preprint arXiv:1906.10124*, 2019.