# Private and Provably Efficient Federated Decision-Making

by

Abhimanyu Dubey

B.Tech., Indian Institute of Technology Delhi (2016)
M.Tech., Indian Institute of Technology Delhi (2016)
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Feburary 2022

© Massachusetts Institute of Technology 2022. All rights reserved.

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Program in Media Arts and Sciences,
School of Architecture and Planning
October 17, 2021

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Alex P. Pentland
Toshiba Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Tod Machover
Academic Head, Program in Media Arts and Sciences

**Private and Provably Efficient Federated Decision-Making**

by

Abhimanyu Dubey

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on October 17, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

## Abstract

In this thesis, we study sequential multi-armed bandit and reinforcement learning in the federated setting, where a group of agents collaborates to improve their collective reward by communicating over a network.

We first study the multi-armed bandit problem in a decentralized environment. We study federated bandit learning under several real-world environmental constraints, such as differentially private communication, heavy-tailed perturbations, and the presence of adversarial corruptions. For each of these constraints, we present algorithms with near-optimal regret guarantees and maintain competitive experimental performance on real-world networks. We characterize the asymptotic and minimax rates for these problems via network-dependent lower bounds as well. These algorithms provide substantial improvements over existing work in a variety of real-world and synthetic network topologies.

Next, we study the contextual bandit problem in a federated learning setting with differential privacy. In this setting, we propose algorithms that match the optimal rate (up to polylogarithmic terms) with only a logarithmic communication budget. We extend our approach to heterogeneous federated learning via a kernel-based approach, and also provide a no-regret algorithm for private Gaussian process bandit optimization.

Finally, we study reinforcement learning in both the multi-agent and federated setting with linear function approximation. We propose variants of least-squares value iteration algorithms that are provably no-regret with only a constant communication budget.

We believe that the future of machine learning entails large-scale cooperation between various data-driven entities, and this work will be beneficial to the development of reliable, scalable, and secure decision-making systems.

Thesis Supervisor: Alex P. Pentland
Title: Toshiba Professor of Media Arts and Sciences

This doctoral thesis has been examined by the following committee.

Professor Alex P. Pentland . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Thesis Advisor
Toshiba Professor of Media, Arts, and Sciences,
Massachusetts Institute of Technology

Professor Tommi Jaakkola . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Member, Thesis Committee
Thomas Siebel Professor of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Professor Gauri Joshi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Member, Thesis Committee
Assistant Professor, Department of Electrical and Computer Engineering
Carnegie Mellon University

Professor Arya Mazumdar . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Member, Thesis Committee
Associate Professor, HDSI and Department of Computer Science
University of California, San Diego

# Acknowledgments

When I began my research at MIT, I was guided extensively by my advisor, Sandy, through a number of tough spots. Right from scoping a massive research problem into a manageable research plan, understanding the aspects of a problem that are worthy of pursuit, to making career decisions, he has been instrumental in this process, for which I am extremely grateful. I would also like to acknowledge my gratitude to my thesis committee members Professors Tommi Jaakkola, Gauri Joshi and Arya Mazumdar, for providing me with several intriguing conversations about research, and for their time and effort in reviewing my work.

It goes without saying that my research outputs are the product of several fruitful collaborations with wonderful researchers, including collaborators beyond the work in this thesis: Dhruv Mahajan, Vignesh Ramanathan, Udari Madhushani, Prof. Naomi Leonard, Laurens van der Maaten, Yixuan Li, Zeki Yalniz, Anirudh Goyal, Moitreya Chatterjee and Prof. Narendra Ahuja, thank you all for your insight, and for helping me expand my research skills.

Next, I would like to thank my fellow Pentlandians at the Human Dynamics Lab, and the members of other labs at MIT who have been immensely helpful through discussions, whiteboard sessions and their companionship. Michiel Bakker, Esteban Moro, Dan Calacci, Tara Sowrirajan, Yuan Yuan, Yan Leng, Cyril de Bodt, Morgan Frank, Abdulrahman Alotaibi, Isabella Loaiza-Saa, Pedro Reynolds-Cuéllar and Peaks Krafft: thank you for all the interesting discussions and coffee breaks (back when we could have coffee together). Thanks, of course, to Stephen Buckley for all of his efforts in making the Human Dynamics machine run smoothly, and to Teri Hagen for all her help in making sure I got what I needed. To Ziv Epstein and Matt Groh, two people who have broadened my research acumen in the best ways possible – it's been one heck of a ride since our first year at the Lab, and here's hoping we get to work on more astounding stuff in the future.

I would like to especially thank Dhaval Adjodah, Nikhil Naik, Linda Peterson and Amna Carreiro for being instrumental in helping me navigate the first years of graduate school - it is unquestionable that my graduate school experience would have been markedly different if not for your advice.

My Cambridge crew - Ishaan, Spandan, Mayank and Chetan, thank you for making

Cambridge my home away from home, and for giving me my fondest memories of grad school. One can never spend enough blithe evenings at the Cellar.

I would like to thank my parents, Pramod and Geeta, and my grandfather, Jagdish, for instilling in me the values that have brought me this far, and for always making me believe in myself. I would like to thank my brother, Abhishek, for making sure I always kept my head on my shoulders, and for putting up with my aggresively mediocre sense of humor. Finally, I would like to thank my partner, Surabhi, for being with me through thick and thin, without whom this thesis would merely be a piece of printed paper.

# Contents

# List of Figures

# List of Tables

# Foreword

This thesis is divided into three parts with a total of 10 chapters, each of which can be read independently of the others. This introduces some redundancy within the text at the cost of readability. The first chapter serves as an introduction to the thesis and its contributions, followed by a brief survey of related background. Chapters 2-5 are concerned with federated multi-armed bandits, Chapters 6-8 are about contextual bandits, and Chapters 9 and 10 discuss problems in reinforcement learning.

**Notation**. We denote vectors by lowercase solid letters, i.e., $\mathbf{x}$, matrices by uppercase solid letters $\mathbf{X}$, and sets by calligraphic letters, i.e., $\mathcal{X}$. We denote the ellipsoid norm of a vector $\mathbf{x}$ as $\|\mathbf{x}\|_{\mathbf{S}} = \sqrt{\mathbf{x}^\top \mathbf{S} \mathbf{x}}$ for some matrix $\mathbf{S}$. We denote the interval $a, ..., b$ for $b \geq a$ by $[a, b]$ and simply as $[b]$ when $a = 1$. We denote the $\gamma^{th}$ power of graph $G$ as $G_\gamma$ ($G_\gamma$ has edge $(i, j)$ if the shortest distance between $i$ and $j$ in $G$ is $\leq \gamma$). $N_\gamma(v)$ denotes the set of nodes in $G$ at a distance of at most $\gamma$ (including itself) from node $v \in G$. The norm $\|\cdot\|$ denotes the $\ell_2(L_2)$ norm unless otherwise specified via a subscript.

# Chapter 1

# Introduction

An increasingly popular machine learning paradigm is that of *federated learning*, where a collection of learning agents (e.g., cellphone devices) collaborate to learn a stronger machine learning model without sharing any of their raw data (Kairouz et al., 2019). The need for federated learning is obvious from a practical perspective: many applications involve data that is distributed, i.e., spread across many entities, where it is infeasible to share data (due to privacy or computational concerns), and an approach that allows learning a joint model from on all the data sources combined is a great improvement over the alternative of having individual models for each distributed dataset.

As expected, federated learning has seen a tremendous increase in adoption, from applications in Google's mobile keyboard (Pichai, 2019) to several services in Apple's iOS (Briggs et al., 2021) and medical imaging as well (Rieke et al., 2020). Beyond these, applications have also been proposed or described in a variety of domains including financial risk prediction for reinsurance (Wang et al., 2020b), electronic health records analysis (Vaid et al., 2020), smart manufacturing (Savazzi et al., 2021) and pharmaceuticals discovery as well (Chen et al., 2020a).

The default purpose of federated learning is that of optimization, i.e., given a set of data points distributed across several entities, the general objective is to find an optimal prediction function that fits this data, e.g., learning a neural network. While it goes without saying that this setting has immense practical utility, many intellingent systems are designed for decision-making under uncertainty, e.g., in clinical trials (Sui et al., 2017) or in online recommender systems (Mary et al., 2015). Furthermore, it is well-established that data distributions are non-statitionary, i.e., they evolve over time. Even in standard applications, this would require retraining the model on the incoming new data frequently (Ditzler et al., 2015).

These reasons motivate us to study sequential decision-making in the federated setting. Compared to the task of optimization, sequential decision-making under uncertainty has the challenge that the agent is required to take actions in an environment as well. This is often at odds with the objective of optimization, since it requires the agent to "explore" the environment carefully to understand the best actions, and exploration will inevitably lead to larger (albeit temporary) suboptimality (Bubeck et al., 2009). On the flipside, it is not possible to derive provably efficient algorithms for decision-making without making precise assumptions about the nature of the environment (Lattimore & Szepesvári, 2020). This is in contrast to the problems in optimization, where recent research is largely focused on problems that do not satisfy the nice assumptions made in decision-making environments.

An aspect of particular importance in this thesis is that of *privacy* in federated decision-making. Increasingly, machine learning and federated learning systems are being trained with fine-grained data sources collected from people at scale, and recent research has demonstrated that it is possible to recover this sensitive information at the individual level from the trained models (Dwork et al., 2017). This becomes a greater challenge in the federated setting, where we specifically desire not to reveal any sensitive information in the process of learning machine learning models. While the paradigm of privacy in federated learning may seem to be in its infancy, there is a rich history of academic research on sharing statistics securely that lies at its foundations, from early work in cryptography (Rivest et al., 1978; Yao, 1982) and database systems (Agrawal & Srikant, 2000). Over the past decade the fair use and protection of individual data has become a matter of global concern (Tankard, 2016; Albrecht, 2016; Goddard, 2017), and is an important aspect in algorithm design in this work as well.

Given this background and motivation, we now discuss the precise contributions in this thesis.

## 1.1 Summary of Contributions

As discussed, the broader focus of this thesis is to study problems in sequential decision-making under uncertainty in the federated setting. We consider two forms of federated learning, namely, the more popular *distributed* setting (where a collection of agents communicate via a server) and the more pragmatic *decentralized* setting, where agents must communicate directly via peer-to-peer messages over a communication network $G$[1]. The thesis is divided into three parts based on the complexity of the decision-making environment considered.

### 1.1.1 Multi-Armed Bandits

The first part concerns the study of multi-armed bandit problems in decentralized federated environments. Specifically, we study a group of $M$ agents communicating via a network $G$ by passing messages that persist for $\gamma$ rounds. The objective of the agents is to collectively minimize the cumulative group regret in a $K-$armed bandit environment. Chapter 2 introduces this problem setting, and presents a basic algorithm FedUCB1 which obtains a cumulative regret of $\mathcal{O}\left(\bar{\chi}(G_\gamma) \cdot K \cdot \log(T)\right)$[2]. FedUCB1 a relatively straightforward algorithm that is present primarily for conceptual exposition, and the analysis of demonstrates the basic technical arguments used to prove regret guarantees. Chapter 2 also presents two categories of lower bounds on the networked federated bandit problem.

First we present two instance-dependent asymptotic guarantees, where we demonstrate that any multi-agent policy must incur $\Omega(K \cdot \log(T))$ regret over $T$ rounds in the limit $T \to \infty$, and furthermore, if the (multi-agent) policy satisfies certain consistency constraints (which, e.g., are satisfied by FedUCB1) then it must incur an asymptotic regret of at least $\Omega\left(\alpha(G_{\gamma+1} \cdot K \cdot \log(T)\right)$. We demonstrate that even in sparse networks, the regret obtained by FedUCB1 is within constant factors of the lower bound. Our asymptotic bounds build on the analysis for single-agent bandits that is folklore in the bandit community (Burnetas & Katehakis, 1996; Lattimore & Szepesvári, 2020) but include several novel aspects specific to the federated setting.

[1] In the subsequent chapter we provide more detail about both of these communication environments and highlight their practical merits, however, from a technical point of view, it is well-established that analyzing decentralized algorithms is more challenging.

[2] $\bar{\chi}(G_\gamma)$ denotes the clique-covering number and $\alpha(G_\gamma)$ denotes the indpendence number of the $\gamma$ power graph of $G$, please see Chapter 2 for more details.

Next, we present two minimax-optimal instance-independent guarantees: we show a $\Omega\left(\sqrt{KM(T+\bar{d}(G_\gamma))}\right)$ rate for arbitrary multi-agent policies (where $\bar{d}(G)$ denotes the average degree in $G$), and a $\Omega\left(\sqrt{\alpha_\star(G_\gamma)\cdot KMT}\right)$ rate for policies that once again satisfy certain consistency properties, where $\alpha_\star(G)$ denotes Turán's lower bound on the independence number of $G$ (Turán, 1941). Both these guarantees, while building on existing techniques for demonstrating lower bounds for multi-armed bandits, introduce new arguments and constructions for the federated setting to characterize multi-agent policies. These lower bounds form the scaffolding for deriving lower bounds in subsequent chapters as "plug-in" bounds for special environments, see e.g., Chapter 3.

Chapter 3 studies the federated multi-armed bandit under differentially-private communication. We present a variant of the aforementioned FedUCB1 algorithm that is modified to ensure that communication between agents respects $(\varepsilon, \delta)$-differential privacy with respect to each agents' personal reward sequence. We propose a variant of the previously discussed FedUCB1 algorithm that satisfies the constraints set by differential privacy with only a constant increase of $\mathcal{O}(\frac{M}{\varepsilon})$ in the group regret. From a technical perspective, our analysis relies on bounding the confidence intervals of a sum of Laplace distributions (since we use the Laplace mechanism to ensure privacy), with the key contribution being the selection of the variable per-message privacy threshold that provides an overall $(\varepsilon, \delta)$ guarantee by the advanced composition theorem (Kairouz et al., 2015).

Furthermore, this algorithm provides a glimpse into the *communication-privacy* synergy we see more broadly in differentially-private bandit algorithms (this is discussed in more detail in Chapter 6). We see that in contrast to the standard setting of Chapter 2, the differential privacy constraint requires agents to communicate only in $\mathcal{O}\left(\frac{T}{\varepsilon}\right)$ rounds. While we do not provide a rigorous proof of the minimax regret in this setting, we conjecture that the constant $\mathcal{O}\left(\frac{M}{\varepsilon}\right)$ can perhaps be improved to no less $\Omega\left(\frac{d_{\max}(G_\gamma)}{\varepsilon}\right)$ by an informal argument concerning the single-agent regret in the private setting.

In the subsequent Chapter 4, we discuss federated multi-armed bandits with heavy-tailed losses. In contrast to Chapters 2 and 3 where we assume arms provide sub-Gaussian rewards, in this section we consider rewards drawn from $(1+\varepsilon)-$heavy tailed distributions, i.e., distributions with finite moments of order at most $(1+\varepsilon)$. We present an extension of FedUCB1 which utilizes a robust mean estimator to obtain the optimal rate (in terms of $\Delta$, the minimum arm separation).

We take the opportunity in this chapter to discuss how the leading term in the regret of FedUCB1 can be improved from the clique covering number $\bar{\chi}(G_\gamma)$ to the domination number $\psi(G_\gamma) \leq \bar{\chi}(G_\gamma)$ if agents communicate via messages of size $\mathcal{O}(K \log(MTK))$ bits, instead of the default $\mathcal{O}(\log(MTK))$ bits of communication utilized by FedUCB1. The key technical argument is to improve the diffusion of information within the network $G$ by allowing weakly-connected agents to directly leverage the estimators of the well-connected agents in $G$ (whereas, in FedUCB1, each agent simply constructs their own estimator). Furthermore, we demonstrate that if we relax the individual consistency property of FedUCB1, i.e., we allow agents to directly mimic other agents regardless of their own reward sequence, then we can achieve the $\psi(G_\gamma)$ rate even with $\mathcal{O}(\log(MTK))$ bits per message. Note that while we discuss these extensions within the heavy-tailed context for continuity (and as this is the order in which they are published), these extensions are applicable to the sub-Gaussian setting as well. We close the chapter by establishing asymptotic lower bounds for the heavy-tailed federated bandit problem.

Chapter 5 discusses the federated bandit problem under adversarial corruptions in communication. Specifically, we consider two models of adversarial perturbations: the first is *Huber contamination* (Huber, 1965), where an adversary can corrupt an $\epsilon$-fraction of all rewards at random, so as to replace the original reward distribution (say, $P$) with the contaminated mixture distribution $(1-\epsilon) \cdot P + \epsilon \cdot Q$, where $Q$ can be any arbitrary distribution ($Q$ can potentially be heavy-tailed and unique for each arm), where $\epsilon \leq \frac{1}{2}$ is known to the agents in advance. We show that a simple extension of the FedUCB1 with robust estimation suffices to provide a no-regret algorithm for small $\epsilon$.

While robustness to arbitrary contamination is an appealing objective, the proposed algorithm requires knowledge of $\epsilon$ (or an upper bound thereof). Indeed, one cannot directly extend known single-agent techniques for model selection such as corraling of bandits (Agarwal et al., 2017) to recover a guarantee without knowledge of $\epsilon$ due to delays in communication and the lack of shared randomness in a decentralized setting. Therefore, to handle unknown adversarial perturbations, we consider an alternative *bounded* perturbation model where an adversary can arbitrarily corrupt any individual message passed between two agents by a maximum amount $\epsilon \leq 1$. Under this corruption model we present an algorithm CHARM (Cooperative Hybrid Arm Elimination) that provides a regret of $\mathcal{O}\left(\psi(G_\gamma) \cdot K \cdot \log^2(T) + MTK\epsilon\right)$, i.e., linear in the total amount of corruption, which is no-regret as long as $\epsilon = \mathcal{O}(T^{-\alpha})$ for some $\alpha > 0$. The analysis of CHARM introduces several new technical components: in contrast to traditional robust arm-elimination approaches, we are faced with a decentralized collection of agents that each have their own arm indices, and face delays in communication. To handle these issues, we exploit the nature of corruption by using a hybrid approach, where each agent switches between a federated arm elimination and UCB exploration using only personal observations. This approach mitigates the effect caused by delayed message-passing through the network and provides us with an efficient algorithm.

### 1.1.2 Contextual Bandits

In Part II we switch gears to consider *contextual* decision-making problems, i.e., bandit problems where the reward from an arm is a function of a time-varying context description. In Chapter 6 we study the $d-$dimensional linear contextual bandit in the federated setting with differential privacy and present an algorithm titled FedLinUCB for this problem. In contrast to the prior chapters, here we discuss both the *distributed* and *decentralized* communication protocols separately, as the algorithms for both settings differ. We demonstrate that FedLinUCB obtains a regret of $\widetilde{\mathcal{O}}\left(\left(d + \sqrt{\frac{d^{3/2}}{\varepsilon}}\right)\sqrt{MT}\right)$[3] in the distributed setting, i.e., when agents communicate via a server, and a regret of $\widetilde{\mathcal{O}}\left(\left(d + \sqrt{\frac{d^{3/2}}{\varepsilon}}\right)\sqrt{\bar{\chi}(G_\gamma) \cdot MT}\right)$ in the decentralized peer-to-peer setting, with only $\widetilde{\mathcal{O}}(M \log(MT))$ bits of communication between agents.

The analysis of FedLinUCB relies on several novel arguments. To achieve the required regret rate within the communication budget, the agents rely on a *data-dependent* communication protocol that adapts to the exploration done by any agent. This communication protocol breaks the martingale structure usually present in single-agent contextual bandit algorithms, and hence a more sophisticated reasoning is required to bound the confidence widths of the ridge regressor. Further, in the decentralized setting, the absence of a central server requires the agents to broadcast their messages to the entire network instead, which again requires a delicate clique-wise analysis of the error terms.

In this chapter, we demonstrate the synergy between communication and privacy briefly remarked in Chapter 3 in more detail. We show that if any agent sends $n$ total messages, it

[3] The $\widetilde{\mathcal{O}}(\cdot)$ notation ignores constants and polylogarithmic factors.

must add a noise term of $\mathcal{O}(\log(n))$ variance to its messages in order to preserve privacy, which in turn increases the regret by the same factor. Therefore, we see that by communicating in fewer rounds, agents can, in fact, decrease their overall regret increase due to differential privacy. However, communicating in fewer rounds will lead to underexploration in the primary bandit policy, hence there is a delicate balance between communication and exploration that needs to be maintained.

Now, in many optimization problems such as Bayesian optimization, one would like to define a high-dimensional kernel on the data in case it is not linearly-separable. It is well-known that several widely utilized kernels such as the squared-exponential (RBF) kernel are in fact infinite-dimensional. Using the approach presented in Chapter 6 would be inapplicable in this case, as one would be required to add vast amounts of noise to ensure privacy. This brings us to the topic of Chapter 7, where we study differentially-private Bayesian optimization in the bandit setting via Gaussian processes. We propose an algorithm for differentially-private (single-agent) Gaussian process bandit optimization in infinite-dimensional Hilbert spaces titled *approximate* GP-UCB that achieves no-regret learning while maintaining differential privacy.

The approach utilizes the quadrature Fourier approximation (Mutny & Krause, 2018) of certain stationary kernels that allows us to project infinite-dimensional Hilbert spaces into a finite, approximating Hilbert space. Given this approximate representation, we apply a variation of the tree-based mechanism for differential privacy that perturbs the approximate statistics in order to ensure privacy. We show that our algorithm satisfies $(\varepsilon, \delta)$-DP while obtaining $\widetilde{\mathcal{O}}(\sqrt{T\gamma_T/\varepsilon})$[4] pseudoregret. This bound matches (up to logarithmic factors) the lower bound for isotropic kernels (Scarlett et al., 2017), and admits an identical dependence on $\varepsilon$ as linear bandits (Shariff & Sheffet, 2018). Thirdly, inspired by the recent interest in locally differentially private (LDP) methods (Bebensee, 2019), we present an algorithm that achieves $(\varepsilon, \delta)-$LDP with $\widetilde{\mathcal{O}}(T^{3/4}\sqrt{\gamma_T/\varepsilon})$ pseudoregret. We conjecture that the constraints from LDP necessitate the $\mathcal{O}(T^{1/4})$ departure from typical near-optimal regret.

After the quick detour into single-agent private algorithms, we return to studying the federated bandit problem in Chapter 8. In this chapter we study the federated decision-making problem in the *heterogeneous* setting, i.e., when each agent faces a unique bandit environment, but it is assumed that these environments are related to one another (in order

[4] $\gamma_T$ is the *maximum information gain*, see Definition 7.1.

to make learning possible). We study this problem in the kernelized bandit setting (similar to Chapter 7), i.e., we assume that the mean reward function lies in a reproducing kernel Hilbert space (RKHS) endowed with a kernel $K$. Here, we introduce heterogeneity in a parameteric manner by assuming an "agent similarity kernel" $K_z$ such that the rewards are drawn from functions that live in a composite space of the contextual kernel $K_x$ and the agent similarity kernel, i.e., $K = K_x \odot K_z$.

In this kernelized bandit setting, a relevant single-agent baseline is the single-agent IGP-UCB (Chowdhury & Gopalan, 2017) algorithm, which, for example, obtains a regret of $\widetilde{\mathcal{O}}(\sqrt{MT}(B\sqrt{\gamma^x_{MT}} + \gamma^x_{MT}))$ when run for a total of $MT$ rounds, where $\gamma^x_{MT}$ is the *information gain* after $MT$ rounds, the structural complexity of the RKHS specified by $K_x$, as defined in the previous chapter (Definition 7.1). We propose an algorithm FedUCB-Kernel, that obtains a regret of

$$\widetilde{\mathcal{O}}\left(\sqrt{MT \cdot \bar{\chi}(G_d)}\left(B\sqrt{\gamma^x_{MT}\gamma_z} + \gamma^x_{MT}\gamma_z\right)\right).$$

Here, $\gamma_z$ determines the similarity between functions $f_v$ via the network kernel $K_z$. We can further see that *information gain* via the contexts $\mathcal{X} \subset \mathbb{R}^n$ typically grows as $\mathcal{O}((\log T)^n)$ for popularly employed kernels such as the squared-exponential kernel, and the network similarity $1 \leq \gamma_z \leq M$ grows as the decision problems faced by the agents progressively become dissimilar, matching the isolated case when $\gamma_z = M$. We additionally present lower bounds for the decentralized problem that scale in terms of the network statistics of $G$ similar to the multi-armed case. While the above analysis assumes a known $K_z$, in many cases, $K_z$ is unknown and requires estimation. For this case, we provide an alternative algorithm (without regret guarantees) via kernel mean embeddings (Christmann & Steinwart, 2010). Against state-of-the-art methods on a variety of real-world and synthetic multi-agent networks, our algorithm exhibits superior performance.

### 1.1.3 Reinforcement Learning

In a natural progression, we arrive at reinforcement learning problems, a more general decision-making setting, where an agent must navigate a state space in addition to the action space present in the bandit case.

In Chapter 9, we propose decentralized algorithms for federated reinforcement learning that are provably efficient with limited communication. We consider specifically the *low-*

*rank* MDP, i.e., a Markov decision process that can be described (up to constant factors) by a $d-$dimensional linear representation. We discuss the federated problem of learning *low-rank* MDPs and provide several characterizations of *heterogeneity* or "non i.i.d.-ness" that correspond to real-world federated environments. We then present a federated algorithm for solving low-rank MDPs with $M$ agents that obtains competitive performance with a bounded communication budget. We propose two modes of communication that perform better in different regimes, however, both can be parameterized to obtain the optimal rate.

Existing regret bounds for single-agent episodic RL in this *low-rank* or *linear* MDP setting scale as $\widetilde{\mathcal{O}}(H^2\sqrt{d^3T})$ for $T$ episodes of length $H$ each, leading to a cumulative regret of $\widetilde{\mathcal{O}}(MH^2\sqrt{d^3T})$ if $M$ agents operate in isolation. Similarly, an agent running for $MT$ episodes will consequently obtain $\widetilde{\mathcal{O}}(H^2\sqrt{d^3MT})$ regret. In comparison, we provide an algorithm built on least-squares value iteration (LSVI) titled FedLSVI, which obtains a group regret of $\widetilde{\mathcal{O}}((d+k)H^2\sqrt{(d+\Gamma)MT})$, where $\Gamma$ is a measure of heterogeneity between different MDPs, and $k \ll M$ is the size of the ambient space used to model this heterogeneity. When the MDPs are homogenous, our rate matches the *centralized* single-agent regret. We introduce several new aspects in the analysis of linear MDPs: first, we analyse stochastic communication and function approximation in the federated setting, presenting a novel concentration argument to bound the per-step estimation error. Next, we analyse each communication protocol with varying message sizes and associated regret bounds. For both approaches we provide rigorous analyses of regret and a lower bound on the group regret for learning federated MDPs as well.

Until now, all federated environments considered are *independent*, i.e., the behavior of one agent in its environment does not influence the environment of another agent. This assumption of independence is somewhat characteristic across federated learning including problems outside of decision-making, such as distributed optimization. An interesting problem therefore would be to study federated decision-making in *non-independent* environments, which is the subject of the final Chapter 10. The general setting of multiple agents interacting in the same reinforcement learning environment (also known as a Markov game) is a decades-old problem in game theory, going back to the work of Shapley (1953). We consider a limited formulation of the general framework of Markov games, where agents collectively take actions to maximize the global good, a setting known as *cooperative* Markov games.

For this problem, 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 environment of Chapter 9 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 $\mathcal{O}((|\mathcal{S}||\mathcal{A}|)^M)$ to $o(dM)$, where $d$ is the dimensionality of the approximating function class, and $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces respectively of each agent.

Next, we generalize the cooperative multi-agent reinforcement learning 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 MG-LSVI (Markov Game Least Squares Value Iteration), a decentralized *vector-valued* optimistic value iteration algorithm that even under partial observability conditions, obtains a cumulative *Bayes regret* of $\widetilde{\mathcal{O}}(\bar{\chi}(G)H^2\sqrt{d^3T})$ over $T$ episodes, where $\bar{\chi}(G)$ denotes the *clique covering* number of $G$. MG-LSVI runs in polynomial time and only requires a communication budget of $o(Md^2 \log T)$ rounds per agent in the worst case, which can be much smaller for sparse $G$. This ensures that MG-LSVI 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), MG-LSVI 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.

We now present a brief survey of the relevant background and topics discussed in this thesis.

## 1.2 Background

### 1.2.1 Online Learning and Multi-Armed Bandits

The central algorithmic framework in this thesis is the broad setting of online learning and multi-armed bandits (Thompson, 1933; Bush & Mosteller, 1953). The most basic version of this problem proceeds in rounds $t = 1, 2, ..., T$, where, in each round $t$, an agent must select an action $\mathbf{x}_t$ from a *decision set* $\mathcal{D}_t$ with the objective of maximizing some (stochastic) reward $r_t$. The problem was first introduced in the context of understanding the worst-case outcomes of running blind medical trials by Thompson (1933), and later, the name "bandit" was coined by Bush & Mosteller (1953) while studying learning patterns in mice. The name "bandit" evolved from the gambling slot machine, called the one-armed bandit, as they were desgined to gradually siphon cash from their unwitting participants.

While the bandit problem at first seems astonishingly straightforward, it provides deep insights into the central dilemma of *exploration* and *exploitation* that is inherent in almost every sequential decision-making problem with uncertainty. Applications of bandit problems are numerous: for example, bandit algorithms are typically used in online advertising (Tewari & Murphy, 2017) and personal content recommendation (Li et al., 2010) as employed on online platforms such as Netflix. A bandit algorithm is present in Monte-Carlo Tree Search (MCTS, Kocsis & Szepesvári (2006)), which played an important role in DeepMind's AlphaGo (Silver et al., 2016), widely considered one of the crowning achievements of artificial intelligence research in the 21st century (Granter et al., 2017). Furthermore, the widely prevalent paradigm of reinforcement learning Sutton & Barto (2018) derives its foundations from the humble multi-armed bandit as well. Below we present the multi-armed bandit problem in its most abstract form.

(Abstract) Multi-Armed Bandit Problem

For round $t \in [T]$:

1. Agent selects arm $a_t \in \mathcal{D}_t$.
2. Agent incurs reward $r_t$.

Now, to provide any meaningful algorithm, we place constraints on the decision-making environment, which leads us to a plethora of frameworks including, for instance, the finite-armed stochastic bandit (Auer et al., 2002a), linear bandits (Li et al., 2010), contextual linear bandits (Abbasi-Yadkori et al., 2011), kernelized contextual bandits (Valko et al., 2013) and Gaussian processes (Srinivas et al., 2009). The algorithms mentioned previously typically assume the reward generation process to be *stochastic*, i.e., the reward $r_t$ at any instant is generated from a function $f$ (typically, $f$ is a function of the selected action $\mathbf{x}_t$) followed by a stochastic perturbation $\varepsilon_t$, i.e., $r_t = f(\mathbf{x}_t) + \varepsilon_t$, where $\varepsilon_t$ is determined by randomness from the environment, and is typically assumed to be i.i.d. in each round. When the reward generation process is *nonstochastic*, i.e., the rewards for any action are determined by an adversary in advance, the regime is popularly known as the *adversarial* bandit (Auer et al., 1995).

Before we delve into algorithm design for each of these environments, we must establish a suitable performance metric. While it is evident that the agent selects actions to maximize the cumulative reward $\sum_{t=1}^{T} r_t$, we find it easier to analyse algorithms from the perspective of *regret*, i.e., comparing the performance of the agent relative to the best possible outcome.

**Definition 1.1** (Pseudoregret, Bubeck et al. (2012))**.** *For a bandit problem over T rounds, let* $\mathbf{x}_t^\star = \arg\max_{\mathbf{x}\in\mathcal{D}_t} \mathbb{E}[r_t(\mathbf{x})]$ *denote the optimal action (in expectation) at round t. The pseudoregret for any agent executing a sequence* $\mathbf{x}_1, ..., \mathbf{x}_T$ *of actions is then given as,*

$$\mathfrak{R}_T = \sum_{t=1}^{T} \mathbb{E}\left[r_t(\mathbf{x}_t^\star) - r_t(\mathbf{x}_t)\right].$$

*The expectation is taken over both the randomness of the environment as well as the agents' policy. An algorithm is typically called* ***no-regret*** *if* $\lim_{T\to\infty} \frac{\mathfrak{R}_T}{T} = 0$.

This formulation of regret is prevalent for *stochastic* multi-armed bandits, whereas, it is alternatively described in terms of *losses* incurred in the adversarial setting, where $\ell_t(\mathbf{x}_t)$ is the loss incurred at any round $t$ (usually, the losses and rewards are measured similarly, with one merely being an affine transformation of the other). The primary objective behind algorithm design is to typically recover *no-regret* algorithms, however, one can alternatively consider the problem of *pure exploration* or *best-arm identification* (Audibert et al., 2010), wherein we only wish to identify the best possible action to take within a decision set. In this case, it is typical to seek PAC (Probably Approximately Correct, Valiant (1984)) guarantees on the sample complexity of the algorithm, i.e., a high probability guarantee on the number of rounds required to identify the optimal action, as presented below.

**Definition 1.2** (($\varepsilon, \delta$)-PAC algorithm, Mannor & Tsitsiklis (2004); Even-Dar et al. (2006)). *For a bandit problem let* $\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{D}} \mathbb{E}[r_t(\mathbf{x})]$ *denote the optimal action (in expectation) for a decision set* $\mathcal{D} \in \mathfrak{D}$*, and* $\mu^\star_{\mathcal{D}} = \mathbb{E}[r_t(\mathbf{x}^\star)]$ *denote the optimal reward. Then, an algorithm* $\mathcal{A}$ *is* $(\varepsilon, \delta)-$*PAC over the space* $\mathfrak{D}$ *if for any* $\mathcal{D} \in \mathfrak{D}$*, it outputs an action* $\mathbf{x}_{\mathcal{A}} \in \mathcal{D}$ *such that*

$$\mathbb{P}\left(\mathbb{E}[r(\mathbf{x}_{\mathcal{A}})] > \mu^\star_{\mathcal{D}} - \varepsilon\right) \geq 1 - \delta.$$

*The expectation is taken over both the randomness of the environment as well as the agents' policy. The* ***sample complexity*** *of an algorithm is given by the minimum number of rounds* $T$ *it requires to obtain an* $(\varepsilon, \delta)-$*PAC guarantee.*

In this thesis, we focus primarily on regret minimization, however, we present a brief discussion on the interplay between decentralized multi-agent regret minimization and distributed best-arm idenfitication towards the end as well. For regret minimization, the philosophy behind algorithm design for *adversarial* bandits has largely been different from that employed for *stochastic* bandits. For the latter, the analysis typically relies on developing estimators from noisy feedback, and delicately adjusting confidence intervals to provide high probability bounds on the regret. For the former, the majority of analyses consider convex problems, where the essence of algorithm design is to leverage convexity constraints to restrict the decision space available to the algorithm efficiently, and ensure fast convergence to the optimal policy via carefully constructed regularized objective functions for gradient-based optimization.

For stochastic bandits, a popular class of algorithms, dubbed *Upper Confidence Bound* (UCB) methods, is based on the optimism under uncertainty heuristic that explicitly encourages exploration by penalizing an action in accordance with how many times it has been explored previously. The algorithm was first introduce for multi-armed stochastic bandits in Auer et al. (2002a). The problem setting considered is the $K-$armed stochastic bandit, where the decision set $\mathcal{D}_t$ is a fixed set of $K$ arms, and pulling an arm $k \in [K]$ provides a random reward with mean $\mu_k$, such that the optimal arm $k^\star = \arg\max_{k \in [K]} \mu_k$. The algorithm introduced by Auer et al. (2002a), called `UCB1` obtains a pseudoregret of $\mathcal{O}(\sum_{k \neq k^\star} \frac{\log T}{\Delta_k} + \Delta_k)$, where $\Delta_k = \mu_{k^\star} - \mu_k$ refers to the suboptimality of arm $k$. When the rewards are Bernoulli distributed, this algorithm matches (up to constants) the rate lower bound of $\mathcal{O}(\sum_{k \neq k^\star} \frac{\log T}{\Delta_k})$ for the $K-$armed bandit presented in the early work of Burnetas & Katehakis (1996). For arbitrary reward distributions, however, `UCB1` is suboptimal, and asymptotically-optimal rates have been achieved by the `KL-UCB` algorithm of Garivier & Cappé (2011). Numerous subsequent improvements and variants of this underlying design philosophy have been proposed for different bandit problem settings, including the LinUCB (Li et al., 2010) and OFUL (Abbasi-Yadkori et al., 2011) algorithms for linear contextual bandits, where the decision set $\mathcal{D}_t$ is typically assumed to be a subset of $\mathbb{R}^d$ for some $d > 1$, and the reward for any arm $\mathbf{x}_t$ is given as $\boldsymbol{\theta}^\top \mathbf{x}_t + \varepsilon_t$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is an unknown (but fixed) vector. Our contributions use a similar linear contextual formulation owing to its versatility and ubiquity, but additionally consider the kernel setting, where the function $f$ has a bounded "small" norm in some RKHS $\mathcal{H}$. Algorithms based on similar upper-confidence bound strategies have been proposed for both Bayesian (`GP-UCB`, Srinivas et al. (2009); Chowdhury & Gopalan (2017)) as well as frequentist formulations (`KernelUCB`, Valko et al. (2013)) and are relevant to our own algorithm development.

Another (and perhaps the first) approach to the bandit problem is that of Thompson Sampling, introduced by Thompson (1933), that takes a Bayesian perspective on the problem. Thompson (or posterior) sampling maintains a probability distribution over the parameterized bandit environment, and at each round, we draw a sample parameter from the posterior distribution of the environment (this is computed using the observed rewards from each arm and a prior distribution over the rewards), and then selects the action that with the largest posterior predicted reward. While the Thompson sampling algorithm had its roots as a heuristic policy from the early work of Thompson (1933), it has recently been shown to exhibit powerful performance in practice Chapelle & Li (2011). This led to a recent surge in developing a theoretical understanding of its performance, and a series of seminal work (Agrawal & Goyal, 2012, 2013) established no-regret guarantees for the algorithm. An alternative Bayes regret perspective was provided in the work of Russo & Van Roy (2014) that provided a confidence-bound based reduction of frequentist regret (employed typically in the analysis of UCB algorithms) to the Bayes regret of Thompson sampling. This technique has subsequently been applied to develop Bayes regret bounds for a variety of different settings (Kandasamy et al., 2018; Dubey & Pentland, 2019). Russo & Van Roy (2014) use their prior analysis of Thompson sampling to develop *Information-directed sampling*, an alternative algorithm that selects actions that minimize a ratio between per-round incurred regret and the *information gain*, i.e., the mutual information between the optimal action in any given round and the subsequent observation.

### 1.2.2 Cooperative and Multi-Agent Decision-Making

Cooperative multi-agent decision-making is a central problem in artificial intelligence, typically involving a collection of agents interacting in an environment, and communicating among themselves to improve collective performance. Note that this is in contrast to the typical *parallelized* learning setting, where a single agent (typically termed the *server*) delegates computation to a group of agents (typically term *clients*) to accelerate learning. In the latter problem, client agents typically do not interact with an environment, nor do they have any individual utility function. In contrast, each agent in the cooperative setting executes their own (possibly unique) policy in their (possibly unique) environments.

For instance, cooperative decision-making problems are common in *federated* training of consumer digital products, e.g., to train their mobile keyboard, Google (Hard et al., 2018) uses a cooperative framework, where each device optimizes predictions for their individual user while leveraging shared knowledge via infrequent communications. Increasingly, this paradigm is gaining prominence in medicine (Sheller et al., 2020; Brisimi et al., 2018) to facilitate large-scale collaboration without sharing explicit health records. There is additionally a wealth of literature on cooperative learning of policies in robotics (Landgren et al., 2016a,b) as well as sensor control (Nikfar & Vinck, 2017).

The standard cooperative framework involves a group $\mathcal{V}$ of $M$ agents, each interacting with a multi-agent environment and communicating with other agents either via synchronization orchestrated by a single server (termed *distributed* communication) or via peer-to-peer messages, i.e., the agents are arranged in a network $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E}$ denotes a communication network, i.e., agents $(i, j)$ communicate if the edge $(i, j) \in \mathcal{E}$ (termed *decentralized* communication). See Figure 2-1 for a visual comparison. Each communication protocol presents its own challenges, however, from a technical perspective, the peer-to-peer protocol is notoriously more difficult to analyse compared to the distributed one, as message-passing usually also creates delays in information flow throughout the network. In this thesis, we will consider both protocols and demonstrate the similarities and differences in each. Our primary focus in this setting is on online learning problems (i.e., multi-armed bandits and reinforcement learning) and we now present a brief summary of

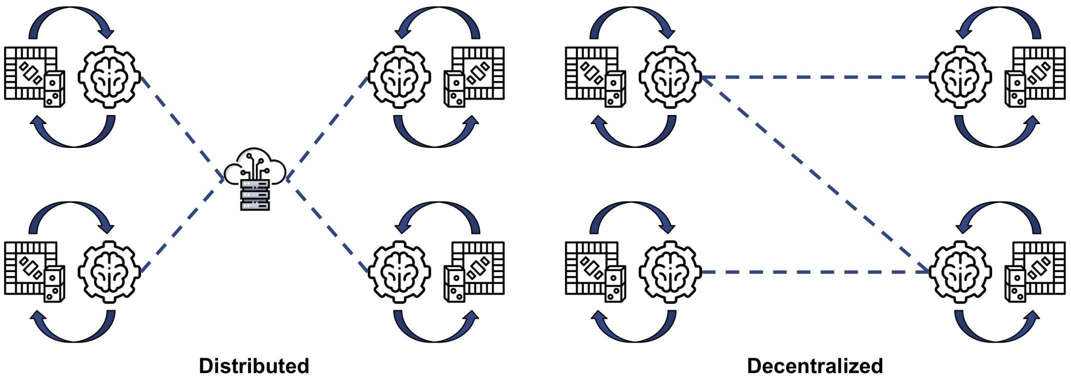


Figure 1-1: A visual representation of the two protocols present in cooperative decision-making for a group of $M$ agents. The dsitributed setting (left) involves periodic communication with a server, which typically requires $\mathcal{O}(M)$ messages be shared each round. In contrast, the decentralized setting (right) involves peer-to-peer communication, and can potentially require $\mathcal{O}(M^2)$ messages be shared per round in the worst case.

existing algorithms for these respective domains. To understand the regret guarantees, we first provide some graph notation.

**Definition 1.3** (Clique covering number)**.** *A clique covering $\mathcal{C}$ of a graph $G$ is a partition of all its vertices into sets $C \in \mathcal{C}$ such that the sub-graph formed by each $C$ is a clique, i.e., all vertices in $C$ are connected to each other in $G$. The smallest number of cliques into which the nodes of $G$ can be partitioned into is called the **clique covering number** $\chi(\bar{G})$.*

**Definition 1.4** (Independence number)**.** *An independent set of a graph $G = (\mathcal{V}, \mathcal{E})$ is a set of vertices $\mathcal{V}' \subseteq \mathcal{V}$ such that no two vertices in $\mathcal{V}'$ are connected. A maximal independent set $\mathcal{V}^\star$ is the largest independent set of $G$, and the corresponding size denotes the **independence number** $\alpha(G) = |\mathcal{V}^\star|$.*

### 1.2.3 Multi-Agent Multi-Armed Bandits

Decentralized cooperative estimation has been explored for sub-Gaussian stochastic bandits using a *running consensus* protocol in (Landgren et al., 2016a,b; Martínez-Rubio et al., 2019) and for adversarial bandits Cesa-Bianchi et al. (2019b); Bar-On & Mansour (2019) using a message-passing protocol. Localized decision-making for sub-Gaussian rewards has also been explored in the work of (Landgren et al., 2018), and a fully-centralized algorithm in (Shahrampour et al., 2017), where all agents select the same action via voting. The stochastic bandit with multiple pulls (Xia et al.; Anantharam et al., 1987) can be seen to equivalent to the cooperative multi-armed bandit on a complete graph $G$ with a centralized actor (since information flows through the network instantaneously). For contextual bandits, similar algorithms have been derived that alternatively utilize message-passing algorithms (Dubey & Pentland, 2020c) or server-synchronization (Wang et al., 2019a). In the competitive multi-agent bandit setting, where agents must avoid collisions, algorithms have been proposed for distributed (Liu & Zhao, 2010b,c; Hillel et al., 2013) and limited-communication (Bistritz & Leshem, 2018) settings. Differentially-private algorithms have also been proposed (Dubey & Pentland, 2020d). Contrasted to cooperative settings, there is extensive research in competitive settings, where multiple agents compete for arms (Bistritz & Leshem, 2018; Bubeck et al., 2019; Liu & Zhao, 2010b,c,a). For strategic experimentation, Brânzei & Peres (2019) provide an interesting perspective on the contrast in exploration strategies between cooperative and competitive agents.

Fundamentally, in the decentralized communication setting, a majority of regret bounds (including those presented in this thesis) for both stochastic and adversarial bandits depend on the independence number $\alpha(G)$ of the communication network (Dubey & Pentland, 2020c,a; Cesa-Bianchi et al., 2019a,b). Interestingly, these bounds can only be obtained when the agents share additional summary statistics, and not just raw observations. In the case when only observations are shared (and not statistics based on individual histories), the regret obtained is a function of the clique number $\chi(\bar{G})$ instead. Since for any graph $G$, we have that $\alpha(G) \leq \chi(\bar{G})$, it is evident that sharing more information is never worse. Indeed, a matching lower bound on the regret has also been shown in the work of Kolla et al. (2018), establishing the optimality of the independence number $\alpha(G)$ for networked communication settings.

For distributed settings, the significant challenge is to provide optimal speed-up in as little communication as possible. However, the theoretical limits of communication vary based on the problem setting considered and the type of guarantee required. For instance, in the problem of distributed *pure exploration*, one can get away with fewer rounds of communication as the objective is to obtain an $(\varepsilon, \delta)-$PAC guarantee, and any agent can individually suffer large regret in the process (e.g., as shown in the work of Hillel et al. (2013)). In contrast, the more relevant setting for cooperative multi-agent setting in modern applications is that of regret minimization, where the utility of any agent cannot be enitrely sacrificed to limit communication. However, in this case as well, we will demonstrate that it is indeed possible to obtain similar guarantees (Dubey & Pentland, 2020b) albeit with slightly higher communication costs.

A closely-related problem is the single-agent *social network* bandit, where a user is picked at random every trial, and the algorithm must infer its contextual mean reward (Cesa-Bianchi et al., 2013; Li et al., 2016; Gentile et al., 2014, 2017), while assuming an underlying *clustering* over the users. This problem setting, while relevant, crucially differs from the cooperative learning, since (a) this is a single-agent setting (only one action is taken every round), and (b) there are no delays or heterogeneity introduced via communication.

### 1.2.4 Cooperative Multi-Agent Reinforcement Learning

Cooperative multi-agent reinforcement learning has a very large body of related work, beginning from classical algorithms in the *fully*-cooperative setting (Boutilier, 1996), i.e., when all agents share identical reward functions. This setting has been explored as multi-agent MDPs in the AI community (Lauer & Riedmiller, 2000; Boutilier, 1996) and as *team Markov games* in the control community (Yoshikawa, 1978; Wang & Sandholm, 2003). In this thesis, we primarily consider the more general *heterogeneous* reward setting, where each agent may have unique reward functions, which corresponds to the *team average* games studied previously (Kar et al., 2013; Zhang et al., 2018b,a). While some of the prior work does indeed provide tractable algorithms that are decentralized and convergent, none consider regret guarantees, owing to the nascent state of research in reinforcement learning with function approximation.

From a theoretical standpoint, we stress that from a fully-observable regret minimization perspective, only environments with sparse communication constraints are interesting as it has been noted abundantly in prior work (e.g., as studied in Szepesvári & Littman (1999)) that a centralized server controlling each agent in realtime can converge to the optimal joint policy. In this thesis, however, we study a more general form of regret in order to discover multiple policies on the *Pareto frontier*, instead of the single policy that maximizes team-average reward. We refer the reader to the illuminating paper by Zhang et al. (2019) for a detailed survey of relevant work.

Closely related to multi-agent RL, parallel reinforcement learning is a very relevant practical setting in large-scale and distributed systems, studied first in Kretchmar (2002). The key dstinction between a typical multi-agent MDP environment and a parallel MDP is that within parallel MDPs, agents have isolated state and action spaces along with isolated reward functions (the reward is only a function of the agent's own state and action), whereas the multi-agent MDP (or cooperative Markov game) involves a joint state and action space and joint reward. The parallel setting can thus be considered as an analog of the *distributed* bandit environment. In this setting, a variant of the SARSA was presented for parallel RL in Grounds & Kudenko (2005), that provides an efficient algorithm but with no regret guarantees. Modern deep-learning based approaches (with no regret guarantees) have been studied recently as well (e.g., Clemente et al. (2017); Espeholt et al. (2018); Horgan et al. (2018); Nair et al. (2015)). In a decentralized variant of parallel reinforcement learning, there has been recent interest from applications (Yu et al., 2020b; Zhuo et al., 2019).

While our focus in this thesis will be limited to approaches motivated by statistical machine learning in order to ensure performance and privacy guarantees, we acknowledge that the broader literature in cooperative multi-agent reinforcement learning includes a variety of perspectives, and we refer the reader to the extensive surveys provided in Busoniu et al. (2008); Hernandez-Leal et al. (2019) for more details.

### 1.2.5 Differential Privacy

While the computer science community has approached the problem of privacy-preserving systems from a variety of computational and statistical approaches, this thesis utilizes *differential privacy*, a cryptographically-secure privacy framework introduced by Dwork (2011); Dwork & Roth (2014) that requires the behavior of an algorithm to fluctuate only slightly (in probability) with any change in its inputs.

**Definition 1.5** (($\varepsilon, \delta$)$-$Differential Privacy, Dwork & Roth (2014))**.** *An algorithm trained over inputs from set $\mathcal{X}$ and producing outputs in the set $\mathcal{Y}$ is $(\varepsilon, \delta)-$differentially private if for any two datasets $X, X' \subset \mathcal{X}$ that differ in only one entry and any $\mathcal{S} \subset \mathcal{Y}$,*

$$\mathbb{P}(\mathcal{A}(X) \in \mathcal{S}) \leq e^{\varepsilon} \cdot \mathbb{P}(\mathcal{A}(X') \in \mathcal{S}) + \delta.$$

*When $\delta \neq 0$, the algorithm is termed $(\varepsilon, \delta)-$**approximately differentially private**, and if the algorithm satisfies the above with $\delta = 0$, it is termed $\varepsilon-$**pure differentially private**.*

While the above model of differential privacy is crucial in ensuring privacy through plausible deniability introduced by randomization, it relies on trust in a centralized authority. Particularly, if a dataset $X$ is comprised of individual rows belonging to distinct users, the users must trust the centralized data aggregating entity that subsequently produces the decision-making algorithm $\mathcal{A}$. In order to eliminate this requirement of trust, an alternate model entitled local differential privacy (LDP) was introduced in the work of Kasiviswanathan et al. (2011), and further popularized by the work of Duchi et al. (2013).

**Definition 1.6** (($\varepsilon, \delta$)$-$Local Differential Privacy, Bebensee (2019))**.** *An algorithm trained over inputs from set $\mathcal{X}$ and producing outputs in the set $\mathcal{Y}$ is $(\varepsilon, \delta)-$differentially private if for any two elements $x, x' \in \mathcal{X}$ and any output $y \in \mathcal{Y}$,*

$$\mathbb{P}(\mathcal{A}(x) = y) \leq e^{\varepsilon} \cdot \mathbb{P}(\mathcal{A}(x') = y) + \delta.$$

*When $\delta \neq 0$, the algorithm is termed $(\varepsilon, \delta)-$**approximate locally differentially private**, and if the algorithm satisfies the above with $\delta = 0$, it is termed $\varepsilon-$**pure locally differentially private**.*

Note that local DP is a much stronger guarantee than DP, as it applies to any pair of input points within a dataset, and not to neighboring datasets as is the case with differential privacy. In this thesis, we will discuss both variants of privacy, and their adaptations to different online learning problems. We now present a brief summary of related work at the intersection of differential privacy and online learning.

### Differential Privacy in Online Learning

A technique to maintain differential privacy for the continual release of statistics was introduced in Chan et al. (2010); Dwork & Smith (2010), known as the *tree-based* algorithm that privatizes the partial sums of $n$ entries by adding at most $\log n$ noisy terms. This method has been used to preserve privacy across several online learning problems, including convex optimization (Jain et al., 2012; Iyengar et al., 2019), online data analysis (Hardt & Rothblum, 2010), collaborative filtering (Calandrino et al., 2011) and data aggregation (Chan et al., 2012). In the single-agent bandit setting, differential privacy using tree-based algorithms have been explored in the multi-armed case (Thakurta & Smith, 2013; Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016) and the contextual case (Shariff & Sheffet, 2018).

For the multi-agent multi-armed stochastic bandit problem, differentially private algorithms have been devised for the centralized (Tossou & Dimitrakakis, 2015b) and decentralized (Dubey & Pentland, 2020d,b) settings. In the competitive multi-agent stochastic bandit case, Tossou and Dimitrakakis (Tossou & Dimitrakakis, 2015b) provide a UCB-based algorithm based on Time-Division Fair Sharing (TDFS). Empirically, the advantages of privacy-preserving contextual bandits has been demonstrated in the work of Malekzadeh et al. (2019), and Hannun et al. (2019) consider a centralized multi-agent contextual bandit algorithm that use secure multi-party computations to provide privacy guarantees (both works do not have any regret guarantees). See Basu et al. (2020) for a summary of regret bounds for private multi-armed bandits. For Gaussian process bandits and Bayesian Optimisation (BO), Kusner et al. (2015) consider the problem of *releasing* GP parameters *after* optimization under differential privacy constraints, by analysing the sensitivity of the final parameters. This thesis, in general, handles a more challenging setting, where parameters must be private *throughout* the optimisation process. An application of DP to the Gaussian process regression was studied in the work of Smith et al. (2016), however, with no regret guarantees. In this thesis, we will extend no-regret private estimation to the Gaussian process problem as well (Dubey, 2021).

In the case of reinforcement learning, the first differentially-private algorithms were presented in the work of Wang & Hegde (2019), that utilize functional perturbations to develop a $Q-$learning algorithm with privacy guarantees. A heuristic approach based on the Laplace mechanism with bit flipping was presented in Ono & Takahashi (2020) for distributed reinforcement learning (i.e., parallel MDPs) with strong empirical performance. The first *joint* differentially private algorithm with regret and PAC guarantees for tabular MDPs was presented in Vietri et al. (2020a), and a local differentially-private variant in the same setting was introduced in Garcelon et al. (2020). This thesis provides differentially-private algorithms for MDPs with general function approximation, which generalizes the existing literature to a much stronger class of models.

### 1.2.6 Robustness

Machine learning at scale is brittly susceptible to a variety of adversarial behavior, including model misspecification (Ghosh et al., 2017; Foster et al., 2020; Lattimore et al., 2020), adversarial perturbations (Moosavi-Dezfooli et al., 2017; Gupta et al., 2019), heavy-tailed data distributions (Dubey & Pentland, 2019), communication failures (Gündüz et al., 2019) and byzantine or contaminated agents (Blanchard et al., 2017; Yin et al., 2018; Dubey & Pentland, 2020d).

In this thesis, we study some of these problems in the context of sequential decision-making and bandit learning. Primarily, we will study the design of algorithms robust to three kinds of adversarial behavior, namely (a) heavy-tailed data, (b) adversarial corruptions contamination and (c) unobserved confounding and heterogeneity. We will now briefly provide some relevant background and related literature.

#### Heavy-Tailed Bandits and Online Learning

Bubeck et al. (2013) first discuss the problem of stochastic bandits with heavy-tailed reward distributions, and propose the Robust-UCB algorithm that uses robust mean estimators to obtain logarithmic regret. Vakili et al. (2013) introduce DSEE, an algorithm that sequences phases of exploration and exploitation to obtain sublinear regret. Thompson Sampling (Thompson, 1933) has been analysed for exponential family bandits (that include Pareto and Weibull heavy-tailed distributions) in the work of Korda et al. (2013), however, these distributions have "lighter" tails owing to the existence of higher order moments. In our own prior work (Dubey & Pentland, 2019), we provide an algorithm for Thompson Sampling for $\alpha$-stable densities (Borak et al., 2005), at family of heavy-tailed densities typically with infinite variance. Yu et al. (2018) provide a purely exploratory algorithm for best-arm identification for $\varepsilon$-heavy tailed rewards. For the linear bandit, Shao et al. (2018); Medina & Yang (2016) provide nearly-optimal algorithms under heavy tails. Best arm selection has also been explored with heavy-tailed reward distributions in the bandit setting Agrawal et al. (2020). No-regret variants of the UCRL2 (Jaksch et al., 2010) and Q-learning (Watkins & Dayan, 1992) for tabular MDPs was introduced recently in the work of Zhuang & Sui (2021).

Most algorithms for heavy-tailed decision-making in the multi-armed bandit and MDP setting rely on the usage of robust mean estimators to construct tight confidence intervals for reward estimation, and such robust estimators typically require excessive communication in the multi-agent setting, as the (near) optimal truncated or median-of-means estimators (Lugosi & Mendelson, 2019) require access to each data sample. This leads us to the question of whether efficient algorithms for robust mean estimation can be developed for streaming applications, in order to allow for tighter controls on communication costs. Among other questions (such as the interaction of heavy-tailed statistics with differential privacy), this demonstrates the technical challenges encountered in robust estimation in the multi-agent setting.

### Sequential Decision-Making with (Adversarial) Corruptions

Robust estimation with adversarial corruptions has a rich history in the bandit literature. We remark that this setting typically lies as an intermediate setting between *stochastic* and *adversarial* observations, e.g., when some fraction of observations are corrupted by an adversary. A large majority of related work in this setting capitalizes on providing *best of both worlds* guarantees, i.e., algorithms that provide optimal rates in both the *stochastic* and *adversarial* observation models, e.g., the work of Bubeck & Slivkins (2012); Seldin & Slivkins (2014); Auer & Chiang (2016); Seldin & Lugosi (2017) provided several algorithms with near-optimal guarantees for both settings in single-agent environments. However, the work of Zimmert & Seldin (2020) demonstrated that the Tsallis-INF algorithm, i.e., mirror descent with Tsallis-entropy regularization provided the optimal best of both worlds guarantees for both stochastic and adversarial bandits, in addition to the *stochastically constrained adversarial* environment, which can be thought of as a stochastic environment corrupted by adversarial perturbations. An alternative arm-elimination perspective was provided for similarly constrained environments in the work of Lykouris et al. (2018). In addition to stochastically-constrained adversarial environments, a very relevant setting for multi-agent algorithm design is robustness to communication failure or byzantine communication. This setting is more closely linked to Huber contamination (Huber, 1965), which has been discussed in single-agent best-arm identification in the work of Altschuler et al. (2019).

# Part I

# Multi-Armed Bandits

# Chapter 2

# Introduction to Federated Multi-Armed Bandits

## 2.1 Introduction

In this chapter we will examine the most fundamental version of stochastic decision-making: the multi-armed bandit, introduced first by Thompson (1933), in a multi-agent federated setting. Building on the problem setting for multi-armed bandits introduced in Chapter 1, we present a federated variant involving $M$ agents below.

(Abstract) Federated Multi-Armed Bandit Problem

For round $t \in [T]$ and agent $m \in [M]$:

1. Environment provides decision set $\mathcal{D}_m(t)$.
2. Agent $m$ selects arm $a_m(t) \in \mathcal{D}_m(t)$.
3. Agent incurs reward $r_{m,t}$.
4. Optionally communicate with other agents.

The typical federated multi-armed setting assumes that the rewards obtained by any agent $m$ from any arm $k$ are drawn from a distribution $\mathcal{D}_k$ with mean $\mu_k$, which is also known as *homogenous* federated learning, as the arm rewards do not change across agents, and depend only on the agent's individual action (and *not* the group action). In Chapter 6, we discuss the *heterogeneous* setting, where reward distributions can vary across agents.

The objective for any agent $m$, is to pull arms $a_m(1), ..., a_m(T)$ over $T$ rounds of the game, such that it obtains the largest cumulative reward $\sum_{t=1}^{T} r_{m,t}$. For arm $k \in [K]$, rewards come from a distribution . The largest expected reward is denoted by $\mu_* = \max_{k\in[K]} \mu_k$, and the corresponding arm(s) is denoted as the *optimal* arm(s) $k^*$. In Chapters 2-5, we will focus exclusively on the i.i.d. setting, that is, for each arm $k$, rewards are independently and identically drawn from a fixed distribution $\mathfrak{D}_k$. We assume (unless stated otherwise), throughout Chapters 2-5 that the rewards are drawn from 1-sub-Gaussian distributions.

### 2.1.1 Single-Agent Regret

We measure performance by *Regret*, which, at any round $T$, compares the obtained (expected) reward against the best mean reward in hindsight.

$$\mathfrak{R}(T) = \mu_* T - \mathbb{E}[\sum_{t=1}^{T} r(t)] = T\mu_* - \sum_{t=1}^{T} \mu_{a_t}$$

This expectation is taken both over the environment and the algorithm's randomness. When the environment obeys simple constraints such as $\sigma$ sub-Gaussianity, one can provide a uniform guarantee on the reward obtained for any specific sequence of actions $a_1, ..., a_T$ for any instantiation of the corresponding random variables. For any fixed sequence of actions $a_1, ..., a_T$ selected by an algorithm, given that we have that $r_t \sim \mu_{a_t}$, we can bound the *instance* regret. Consider the instance regret as the *realized* difference between the rewards from the best arm at each round and the actual reward $r_t$ as $\widehat{\mathfrak{R}}(T)$. Then, we have that $\mathbb{E}[\widehat{\mathfrak{R}}(T)] = \mathfrak{R}(T)$ and that with probability at least $1 - \frac{1}{T}$,

$$\widehat{\mathfrak{R}}(T) \leq \mathfrak{R}(T) + \left|\mathfrak{R}(T) - \widehat{\mathfrak{R}}(T)\right| \leq \mathfrak{R}(T) + \mathcal{O}\left(\sigma\sqrt{T \log(KT)}\right).$$

Here the expectation is taken *only* with respect to the environment. The last inequality follows trivially from a Hoeffding bound, and can also be derived in settings where rewards are not drawn i.i.d. from the arms (e.g., in contextual bandits), using a martingale approach. This suggests that a bound on the expected regret (also known as *pseudoregret*) can provide a high probability bound on the *realized* regret, making the *pseudoregret* a more appealing measure. Chapters 2-5 consider the following class of bandit problems $\mathcal{E}$ for a finite, countable set of actions $\mathcal{A}$, such that $|\mathcal{A}| = K$. $\mathcal{E}$ is considered to be *unstructured*, i.e.

the rewards from each arm are independent of the others.

**Definition 2.1** (Unstructured Bandit Problem)**.** *A class of bandit problems $\mathcal{E}$ is unstructured if its action space $\mathcal{A}$ is finite, and there exists a set of distributions $\mathfrak{P}_a \forall\ a \in \mathcal{A}$ such that*

$$\mathcal{E} = \{\boldsymbol{\nu} = (P_a : a \in \mathcal{A}) : P_a \in \mathfrak{P}_a \forall a \in \mathcal{A}\}.$$

**Lower Bounds**. One can derive the immediate bound of $\mathfrak{R}(T) \geq \sum_{k:\Delta_k>0} \Delta_k$ from the fact that any agent must pull each arm once. A stronger bound on the regret has been provided by Lai & Robbins (1985), and generalized subsequently by Burnetas & Katehakis (1996), that holds for a class of "consistent" policies, i.e., policies that exhibit "uniform" behavior across all stochastic bandit problems.

**Definition 2.2** (Consistent Bandit Policy)**.** *Let $\pi$ be any bandit policy, which is **consistent** if, for any suboptimal arm $k \in [K], k \neq k^*$, horizon $T > 0$, one has $\mathbb{E}[n_k(T)] = o(T^a)$ for any $a > 0$, where $n_k(T)$ denotes the number of times $\pi$ pulls arm $k$ until round $T$.*

For the class of bandit policies defined above, we have the following lower bound.

**Theorem 2.1** (Burnetas & Katehakis (1996))**.** *Let $\mathcal{B}$ be a class of unstructured bandits and $\pi$ be a consistent policy over $\mathcal{B}$. Then, for all $\nu = (P_i)_{i=1}^k \in \mathcal{B}$ such that $\mu_i = \mathbb{E}_{X\sim P_i}[X] < \mu^*$, we have*

$$\liminf_{T\to\infty} \frac{R(T)}{\ln(T)} \geq \sum_{k:\Delta_k>0} \frac{\Delta_k}{d_{\inf}(P_i, \mu^*, \mathcal{M}_i)}, \text{ where,}$$

$$d_{\inf}(P, \mu^*, \mathcal{M}) = \inf_{P' \in \mathcal{M}} \left\{\mathbb{D}_{\mathsf{KL}}(P, P') : \mu(P') > \mu^*\right\}.$$

We refer the reader to Burnetas & Katehakis (1996) for a complete proof of this theorem.

### 2.1.2 Communication Protocols and Group Regret

The standard federated framework involves a group $\mathcal{V}$ of $M$ agents, each interacting with a multi-agent environment and communicating with other agents either via synchronization orchestrated by a single server (termed *distributed* communication) or via peer-to-peer messages, i.e., the agents are arranged in a network $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E}$ denotes a communication network, i.e., agents $(i, j)$ communicate if the edge $(i, j) \in \mathcal{E}$ (termed *decentralized* communication). See Figure 2-1 for a visual comparison. Each communication protocol

Table 2.1: Quantity (with notation) for any graph $G$.

| | | | |
|---|---|---|---|
| Average degree ($\bar{d}$) | Maximum degree ($d_{\max}$) | Degree of $i$ ($d_i$) | Independence number ($\alpha$) |
| Message life ($\gamma$) | Minimum degree ($d_{\min}$) | Neighborhood of $i$ ($\mathcal{N}_i$) | Domination number ($\psi$) |
| $k$-power of $G$ ($G_k$) | Diameter ($d_\star$) | $\mathcal{N}_i \cup \{i\}$ ($\mathcal{N}_i^+$) | Clique covering number ($\bar{\chi}$) |

presents its own challenges, however, from a technical perspective, the peer-to-peer protocol is more difficult to analyse compared to the distributed one, as message-passing usually also creates delays in information flow throughout the network.

**Message-Passing**. Let $G = (\mathcal{V}, \mathcal{E})$ be a connected, undirected graph encoding the communication network, where $\mathcal{E}$ contains an edge $(i, j)$ if agents $i$ and $j$ can communicate directly via messages with each other. After each round $t$, each agent $j$ broadcasts a message $\mathbf{q}_j(t)$ to all their neighbors. Each message is forwarded at most $\gamma$ times through $G$, after which it is discarded. For any value of $\gamma > 1$, the protocol is called *message-passing* (Linial, 1992), but for $\gamma = 1$ it is called *instantaneous reward sharing*, as this setting has no delays in communication. Part I considers the general message-passing setting with $\gamma > 1$, and we remark that algorithms for the peer-to-peer setting are by design, applicable in the distributed setting (one can simply set any individual agent as the server, and consider $G$ to be the star graph with that agent at the center). We now provide some graph notation in Table 2.1 and provide additional relevant terminology.

**Definition 2.3** (Clique covering number)**.** *A clique cover $\mathcal{C}$ of any graph $G = (\mathcal{V}, \mathcal{E})$ is a partition of $\mathcal{V}$ into subgraphs $C \in \mathcal{C}$ such that each subgraph $C$ is fully connected, i.e., a clique. The size of the smallest possible covering $\mathcal{C}^\star$ is known as the* clique covering number *$\bar{\chi}(G)$.*

**Definition 2.4** (Independence number)**.** *The* independence number *$\alpha(G)$ of $G = (\mathcal{V}, \mathcal{E})$ is the size of the largest subset of $\mathcal{V}_\alpha \subseteq \mathcal{V}$ such that no two vertices in $\mathcal{V}_\alpha$ are connected.*

**Definition 2.5** (Domination number)**.** *The* domination number *$\psi(G)$ of $G = (\mathcal{V}, \mathcal{E})$ is the size of the smallest subset $\mathcal{V}_\psi \subseteq \mathcal{V}$ s. t. each vertex not in $\mathcal{V}_\psi$ is adjacent to at least one agent in $\mathcal{V}_\psi$.*

**Group Regret**. The performance measure we consider is a straightforward extension of the single-agent idea of *pseudo regret* called *group* regret, which is the regret (in expectation) incurred by the group $\mathcal{V}$ by pulling suboptimal arms. The group regret is given by

$$\mathfrak{R}(T) = TM\mu_* - \sum_{t=1}^{T} \sum_{m=1}^{M} \mu_{a_m(t)} = \sum_{i=1}^{M} \sum_{k:\Delta_k>0} \Delta_k \cdot \mathbb{E}\left[n_k^i(t)\right].$$

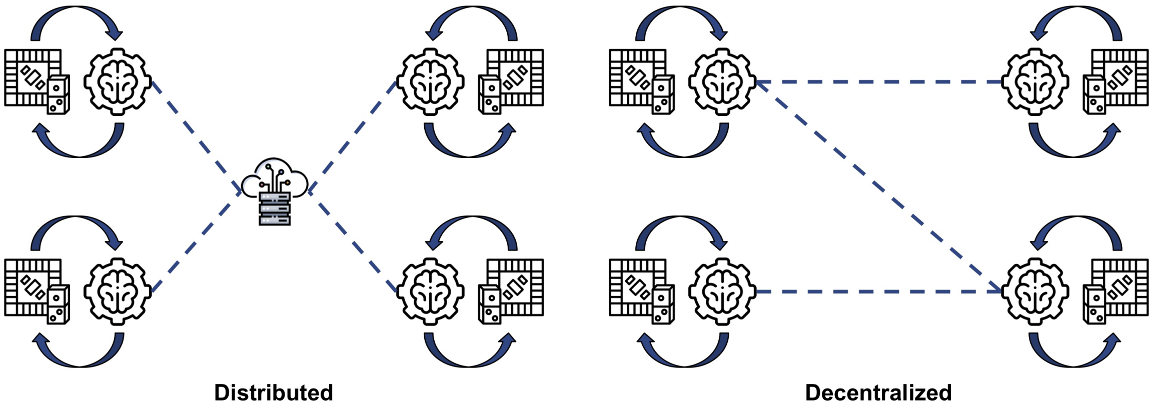


Figure 2-1: A visual representation of the two protocols present in cooperative decision-making for a group of $M$ agents. The dsitributed setting (left) involves periodic communication with a server, which typically requires $\mathcal{O}(M)$ messages be shared each round. In contrast, the decentralized setting (right) involves peer-to-peer communication, and can potentially require $\mathcal{O}(M^2)$ messages be shared per round in the worst case.

Here $n_k^i(t)$ is the number of times agent $i$ pulls the suboptimal arm $k$ up to (and including) round $t$. We now present lower bounds on the group regret in two different settings, based on moderate assumptions about the algorithms and the communication protocol. These results are quite general and are inherently independent of the exact setting, and hence will be used in the later chapters to provide the basic framework to contruct hard bandit instances in different settings.

## 2.2 FedUCB1

In this section we present the most basic algorithm of this part, titled FedUCB1. As the name suggests, this approach extends the fundamental UCB1 algorithm of Auer et al. (2002a) to the federated setting. We will consider the decentralized communication protocol and provide regret bounds as well.

The protocol proceeds as follows. Each agent $1 \leq v \leq M$ maintains a set of observations $\mathcal{S}_{v,k}(t)$ for each arm $k$, where an observation is of the form $(t, m, r_{m,k}(t))$ where $t$ denotes the round in which this arm was pulled by agent $m$, and $r_{m,k}(t)$ denotes the reward obtained by agent $m$ for pulling arm $k$. At the start of each round, the agent selects an arm based on the set of observations $\{\mathcal{S}_{v,k}(t)\}_{k:\Delta_k>0}^K$ and an exploration policy, and then creates a message $\mathbf{q}_v(t)$ to send to its neighbors via the message-passing protocol.

It then sends all *fresh* messages it has to its neighbors, where a message is denoted fresh if it was originally sent at any time $\geq t-\gamma$ to agent $v$, along with its message own $\mathbf{q}_v(t)$. It then collates the messages it has received from other agents, and discards all *stale* messages (i.e., messages that originated more than $\gamma$ rounds prior), updates its observation sets $\mathcal{S}_{v,k}(t)$ and the next round begins henceforth. The agent $v$ creates the following message $\mathbf{q}_m(t)$ to send to its neighbors.

$$\mathbf{q}_m(t) = \langle v, t, \gamma, r_v(t), a_v(t) \rangle$$

The exploration strategy for each agent is straightforward. At the start of the new round, each agent $m$ computes an upper confidence bound for each arm $k \in [K]$ using the latest information from its neighborhood $\mathcal{N}_m(G_\gamma)$, as follows.

$$Q_{m,k}(t) = \frac{\sum_{i \in \mathcal{S}_{v,k}(t)} r_i}{|\mathcal{S}_{v,k}(t)|} + \sqrt{\frac{2\log(t)}{|\mathcal{S}_{v,k}(t)|}}.$$

Here, the first term simply represents the average of all observations in $\mathcal{S}_{v,k}(t)$ and is reminiscent of the typical UCB in single-agent bandit algorithms. The agent then selects the action $k$ with the largest $Q_{m,k}(t)$. For the first $K$ rounds, each agent pulls arms 1 to $K$. The algorithm denoted as FedUCB1, and obtains the following regret.

**Theorem 2.2** (Regret of FedUCB1). *If all agents $m \in [M]$ each run* FedUCB1 *with the messaging protocol described in Algorithm 1, then the group regret incurred after $T$ trials obeys,*

$$\mathfrak{R}(T) \leq \sum_{k:\Delta_k > 0} \bar{\chi}(G_\gamma)\left(\frac{8\log(T)}{\Delta_k}\right) + \left(\sum_{k:\Delta_k>0} \Delta_k\right)(M\gamma + 2).$$

*Here, $\bar{\chi}$ is the clique covering number.*

*Proof.* The key element of this proof is bounding the collective behavior of all agents as a function of the largest clique in $\boldsymbol{C}$ it is part of. At a high level, we will analyse each clique of agents in $G_\gamma$ separately. Now, let a clique covering of $G_\gamma$ be given by $\boldsymbol{C}$. We first bound the regret in each clique $\mathcal{C}$ within the clique covering $\boldsymbol{C}$ of $G_\gamma$. This is done by noticing that the upper confidence bound for any arm at a selected $t$ deviates by a constant amount between agents based on the number of times each agent has pulled an arm. By bounding this deviation, we obtain a relationship between the confidence bound of each arm for each agent within the clique $\mathcal{C}$. Next, we bound the probability of pulling a suboptimal arm within the clique $\mathcal{C}$ using the previous result. Summing over the clique cover $\boldsymbol{C}$ delivers the final form of the result. We begin by decomposing the group regret.

$$\mathfrak{R}(T) = \sum_{m=1}^{M} \mathfrak{R}_m(T) \leq \sum_{\mathcal{C} \in \boldsymbol{C}} \sum_{m \in \mathcal{C}} \sum_{k:\Delta_k > 0}^{K} \Delta_k \mathbb{E}[n_{m,k}(T)] \tag{2.1}$$

$$= \sum_{\mathcal{C} \in \boldsymbol{C}} \sum_{k:\Delta_k > 0}^{K} \Delta_k \left( \sum_{m \in \mathcal{C}} \sum_{t=1}^{T} \mathbb{P}\left(a_m(t) = k\right) \right) \tag{2.2}$$

Consider the cumulative regret $R_{\mathcal{C}}(T)$ within the clique $\mathcal{C}$. For some time $T^k_{\mathcal{C}}$, assume that each agent has pulled arm $k$ for $\eta^k_m$ trials, where $\eta^k_{\mathcal{C}} = \sum_{m \in \mathcal{C}} \eta^k_m$. Then,

$$R_{\mathcal{C}}(T) \leq \sum_{k:\Delta_k > 0}^{K} \Delta_k \left( \eta^k_{\mathcal{C}} + \sum_{m \in \mathcal{C}} \sum_{t=T^k_{\mathcal{C}}}^{T} \mathbb{P}\left( a_m(t) = k, N^{\mathcal{C}}_k(t) \geq \eta^k_{\mathcal{C}} \right) \right). \tag{2.3}$$

Here $N_{\mathcal{C},k}(t)$ denotes the number of times arm $k$ has been pulled by any agent in $\mathcal{C}$. We now examine the probability of agent $m \in \mathcal{C}$ pulling arm $k$. Note that an arm is pulled when one of three events occurs:

$$\text{Event (A): } \hat{\mu}_{m,*}(t-1) \leq \mu_* - \left( \frac{2\log(t)}{|\mathcal{S}_{m,*}(t)|} \right)^{\frac{1}{2}}.$$

$$\text{Event (B): } \hat{\mu}_{m,k}(t-1) \geq \mu_k + \left( \frac{2\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{1}{2}}.$$

$$\text{Event (C): } \mu_* \leq \mu_k + 2\left( \frac{2\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{1}{2}}.$$

Now, let us examine the occurence of event $(C)$:

$$\Delta_k \leq 2\left( \frac{2\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{1}{2}}$$

$$\implies |\mathcal{S}_{m,k}(t)| \leq 2\log(t)\left( \frac{2}{\Delta_k} \right)^2$$

Now, we present a lemma that bounds the effective delays for any agent within the network.

**Lemma 2.1.** *For graph $G$ and agent $m$, consider a subgraph $G_\gamma(m)$ that includes all agents that have a shortest path of length at most $\gamma$ from agent $m$, along with the corresponding paths. Let $\mathcal{S}_{m,k}(t)$ denote the set of all reward samples (across all agents) possessed by agent $m$ for arm $k$ at time $t$, and $\widetilde{N}_{m,k}(t)$ denote the total number of times arm $k$ has been pulled until time $t$ across all* *agents in $G_\gamma(m)$. Then, we have, for all $k \in [K], m \in [M]$,*

$$\widetilde{N}_{m,k}(t) \geq |\mathcal{S}_{m,k}(t)| \geq \max\left\{0, \widetilde{N}_{m,k}(t) + (|G_\gamma(m)| - 1)(1-\gamma)\right\}.$$

*Proof.* Let $\mathcal{S}_{m,k}(t)$ denote the set of all reward samples possessed by agent $m$ for arm $k$ at time $t$. Similarly, let $P_{m,k}(t)$ denote the set of reward samples obtained by agent $m$ for its own pulls of arm $k$ until time $t$. We know, then that $P_{m,k}(t) = P_{m,k}(t-1)$ if arm k was pulled at time t, and $P_{m,k}(t) = P_{m,k}(t-1) \cup \{X_{m,t}\}$ otherwise. Additionally, any message from an agent $m' \in G$ takes $d(m, m') - 1$ iterations to reach agent $m$. Therefore:

$$\mathcal{S}_{m,k}(t) = P_{m,k}(t) \cup \left\{ \bigcup_{m' \in G \setminus \{m\}} P_{m,k}\left(t - d(m', m) + 1\right) \right\}.$$

Note that $P_{m,k}(t)$ and $P^{k'}_{m'}(t')$ are disjoint for all $m \neq m', k, k', t, t'$. Let $n(\mathcal{S})$ denote the cardinality of $\mathcal{S}$. Then,

$$n\left(\mathcal{S}_{m,k}(t)\right) = n\left(P_{m,k}(t)\right) + \left\{ \sum_{m' \in G \setminus \{m\}} n\left(P_{m,k}\left(t - d(m', m) + 1\right)\right) \right\}.$$

Now, in the iterations $t - d(m, m') + 1$ to $t$, agent $m'$ can pull arm $k$ at most $d(m, m') - 1$ times and at least 0 times. Therefore,

$$\begin{aligned}\widetilde{N}_{m,k}(t) \geq n\left(\mathcal{S}_{m,k}(t)\right) &\geq \max\left\{0, \widetilde{N}_{m,k}(t) - \sum_{m' \in G_\gamma(m) \setminus \{m\}} (d(m, m') - 1)\right\} \\ &\geq \max\left\{0, \widetilde{N}_{m,k}(t) + |G_\gamma(m)|(1-\gamma)\right\}\end{aligned}$$

□

We know that for the subgraph $\mathcal{C}$, Lemma 2.1 holds for each $m \in \mathcal{C}$ with delay $\gamma$. Hence, $N_{m,k}(t) \geq N_{\mathcal{C},k}(t) - (|\mathcal{C}| - 1)(1-\gamma)$ for all $t$. Therefore, if we set

$$\eta^k_{\mathcal{C}} = \left\lceil 2\log(t)\left(\frac{2}{\Delta_k}\right)^2 + (|\mathcal{C}| - 1)(\gamma - 1) \right\rceil,$$

we know that event $(C)$ will not occur. Additionally, using the union bound over $N_{m,*}(t)$

and $N_{m,k}(t)$, and Assumption 4.1, we have:

$$\mathbb{P}(\text{Event (A) or (B) occurs}) \leq 2\sum_{s=1}^{t} \frac{1}{s^4} \leq \frac{2}{t^3}.$$

Combining all probabilities, and inserting in Equation (4.4), we have,

$$\begin{aligned}
\mathfrak{R}_{\mathcal{C}}(T) &\leq \sum_{k:\Delta_k>0}^{K} \Delta_k \left( \eta_{\mathcal{C}}^k + \sum_{m\in\mathcal{C}} \sum_{t=T_{\mathcal{C}}^k}^{T} \mathbb{P}\left(a_m(t) = k, N_k^{\mathcal{C}}(t) \geq \eta_{\mathcal{C}}^k\right)\right) \\
&\leq \sum_{k:\Delta_k>0}^{K} \Delta_k \left( \left\lceil 2\log(t) \left(\frac{2}{\Delta_k}\right)^2 + (|\mathcal{C}|-1)(\gamma-1) \right\rceil + \sum_{m\in\mathcal{C}} \sum_{t=1}^{T} \frac{2}{t^3} \right) \\
&\leq \sum_{k:\Delta_k>0}^{K} \Delta_k \left( \left\lceil 2\log(t) \left(\frac{2}{\Delta_k}\right)^2 + (|\mathcal{C}|-1)(\gamma-1) \right\rceil + 4|\mathcal{C}| \right) \\
&\leq \sum_{k:\Delta_k>0}^{K} \Delta_k \left( 2\log(t) \left(\frac{2}{\Delta_k}\right)^2 + (|\mathcal{C}|-1)(\gamma-1) + 1 + 4|\mathcal{C}| \right).
\end{aligned}$$

We can now substitute this result in the total regret.

$$\begin{aligned}
\mathfrak{R}(T) &\leq \sum_{\mathcal{C}\in\boldsymbol{C}} \mathfrak{R}_{\mathcal{C}}(T) \\
&\leq \sum_{\mathcal{C}\in\boldsymbol{C}} \sum_{k:\Delta_k>0}^{K} \Delta_k \left( 2\log(t) \left(\frac{2}{\Delta_k}\right)^2 + (|\mathcal{C}|-1)(\gamma-1) + 1 + 4|\mathcal{C}| \right) \\
&= \sum_{k:\Delta_k>0}^{K} \frac{2\chi\left(\bar{G_\gamma}\right)}{\Delta_k} \log T + \left(3M + \gamma\left(M-1\right)\right) \left( \sum_{k:\Delta_k>0}^{K} \Delta_k \right).
\end{aligned}$$

□

**Remark 2.1.** Observe that the leading term in the bound depends on the clique covering number $\bar{\chi}(G_\gamma)$ of the power graph $G_\gamma$. In comparison to the regret lower bound presented earlier, we see that the algorithm obtains regret within a constant factor of this lower bound, even when the graph $G_\gamma$ is the worst-case connected graph, i.e., a line graph. We have that when $G$ is a line graph, $\alpha(G_{\gamma+1}) = \frac{M}{\gamma+1}$ and the corresponding clique covering number is $\bar{\chi}(G_\gamma) = \frac{M}{\gamma}$. We therefore have that for line graphs,

$$\liminf_{T\to\infty} \frac{\text{UB}(\mathfrak{R}(T))}{\text{LB}(\mathfrak{R}(T))} \leq \frac{\bar{\chi}(G_\gamma)}{\alpha(G_{\gamma+1})} \leq 1 + \frac{1}{\gamma} \leq 2.$$

Similar bounds can be derived for circular graphs and regular graphs.

## 2.3 Asymptotic Lower Bounds

We now present lower bounds on cooperative decision-making. All full proofs are presented in the appendix for brevity. We consider $M$ agents communicating over graph $G$, with $\text{diameter}(G) = \gamma_* \ll M$. We first make some (mild) assumptions on the communication protocol.

**Assumption 2.1** (Communication Protocol)**.** *The communication protocol considered follows:*

1. *Any agent $m$ is capable of sending a message $\boldsymbol{q}_m(t)$ to any other agent $m' \in [M]$, which is earliest received at time $t + \min(0, d(m, m') - 1)$.*
2. *$\boldsymbol{q}_m(t)$ is a function* only *of the history of agent $m$, i.e., for any deterministic and differentiable set of functions $\boldsymbol{F}_t = (f_{i,t})_{i \in [L]}, f_{i,t} : \mathbb{R}^{2t} \to \mathbb{R}$ with Jacobian $\boldsymbol{J}_t$,*

$$\boldsymbol{q}_m(t) = \boldsymbol{F}_t(a_m(1), r_m(1), ..., a_m(t), r_m(t)),$$

*Furthermore $\boldsymbol{F}_t$ satisfies $|det\,(\boldsymbol{J}_t)\,| = \Lambda(m, t)$, where $\Lambda$ is a function only of $m$ and $t$.*

This assumption ensures that (a) information can flow between any two agents, and (b) that given the agent's history, the messages are not stochastic and are independent of any prior knowledge of the bandit problem. Under these assumptions we present a lower bound on the regret in the general, networked communication setting.

**Theorem 2.3** (General Federated Lower Bound)**.** *For any consistent cooperative multi-agent policy $\Pi = (\Pi_t)_{t \in [T]}$ on $M$ agents that satisfies Assumption 2.1 the following is true.*

$$\liminf_{T \to \infty} \frac{R_G(T)}{\log(T)} \geq \sum_{k : \Delta_k > 0} \frac{\Delta_k}{\mathbb{D}_k^{\mathsf{inf}}}.$$

*Here, $\mathbb{D}_k^{\mathsf{inf}} = \inf_{\nu' \in \mathcal{M}_k} \{\mathbb{D}_{\mathsf{KL}}(\nu, \nu') : \mu(\nu') > \mu^*\}$.*

The proof of this result is presented in Section 2.6. This result generalizes that obtained by Anantharam et al. (1987) for a centralized agent with multiple pulls to the case where rewards are obtained after finite delays. Note, however, that Theorem 2.3 does not guarantee an overhead from delayed communication, since it includes protocols that allow information to flow completely through the (connected) network $G$, albeit at a delay (which is independent of $T$). Indeed, in this chapter, we will demonstrate an algorithm that matches this rate exactly while satisfying Assumption 2.1. Making stronger assumptions about the algorithm, however, can lead to a stronger *network-dependent* lower bound, presented next. We first provide some additional shorthand notation.

**Histories**. Let $\mathcal{H}(t)$ denote the interaction history of all agents until round $t$, i.e., $\mathcal{H}(t) = \{(a_m(\tau), r_{m,\tau} : 1 \leq m \leq M, 1 \leq \tau \leq t\}$. Let $\mathcal{H}_m(t)$, for any $1 \leq m \leq M$ denote the history of only agent $m$ and its neighbors in $G_\gamma$ interactions. Further, let $\overline{\mathcal{H}}_m(t)$ denote the history of all agents $n \notin \mathcal{N}_m(G_\gamma)$. Formally,

$$\begin{aligned}\mathcal{H}(t) &= \{(a_m(\tau), r_{m,\tau} : 1 \leq m \leq M, 1 \leq \tau \leq t\},\\ \mathcal{H}_m(t) &= \{(a_n(\tau), r_{n,\tau} : n \in \mathcal{N}_m^+(G_\gamma), 1 \leq \tau \leq t\},\\ \overline{\mathcal{H}}_m(t) &= \{(a_n(\tau), r_{n,\tau} : n \notin \mathcal{N}_m^+(G_\gamma), 1 \leq \tau \leq t\}.\end{aligned}$$

**Definition 2.6** (Non-Altruistic and Individually Consistent (NAIC) Policy, Kolla et al. (2018))**.** *A policy belonging to agent m is called* individually consistent *if, for any sub-optmal arm k, regardless of the policy of the agents in $\mathcal{N}_m(G_\gamma)$, $\mathbb{E}[n_{m,k}(t)|\overline{\mathcal{H}}_m(t)] = o(t^a) \forall a > 0, \forall \overline{\mathcal{H}}_m(t)$.*

*A policy followed by a user m is said to be* non-altruistic *if there exists constants $a_1, a_2$ depending on the horizon T such that the following holds. For any T and suboptimal arm k, then the number of times the policy plays arm k after having obtained $a_1 \log(T)$ samples from it is no more than $a_2$, regardless of all other agents.*

**Remark 2.2.** NAIC policies essentially satisfy the intuitive idea that they will not sacrifice their reward in order to reduce cumulative regret: non-altruism prevents an agent from pulling an arm more than $\mathcal{O}(\log(T))$ times in order to allow other (poorly-connected) agents to explore more. Individual consistency, on the other hand, ensures that the *individual* policies are *consistent* (Definition 2.2), in contrast to the result from Theorem 2.3, which only requires *group consistency*. Kolla et al. (2018) demonstrate that several collaborative policies are in fact NAIC, such as UCB1 and Thompson Sampling. The algorithms we present include algorithms that require certain agents to mimic other agents. This approach, while obtaining better regret, violates tha individual consistency property.

**Theorem 2.4** (NAIC Federated Lower Bound)**.** *For any consistent cooperative multi-agent NAIC policy $\Pi = (\Pi_t)_{t \in [T]}$ on M with $1 \leq \gamma < d_\star(G)$ agents that satisfies Assumption 2.1* *the following is true.*

$$\liminf_{T\to\infty} \frac{R_G(T)}{\log(T)} \geq \alpha(G_{\gamma+1}) \cdot \sum_{k:\Delta_k>0} \frac{\Delta_k}{\mathbb{D}_k^{\mathsf{inf}}}.$$

*Here,* $\mathbb{D}_k^{\mathsf{inf}} = \inf_{\nu' \in \mathcal{M}_k} \{\mathbb{D}_{\mathsf{KL}}(\nu, \nu') : \mu(\nu') > \mu^*\}$.

*Proof.* The complete proof for $\gamma = 1$ can be found in Appendix B of Kolla et al. (2018). To extend this to the case when $\gamma > 1$ we can follow a similar structure. Consider the maximal independent set $S$ of the graph $G_{\gamma+1}$ for any $\gamma < d_\star(G)$. $G_{\gamma+1}$ has an edge $(u, v)$ either if two agents are $\gamma-$neighbors or if $\mathcal{N}_u^+(G_\gamma) \cap \mathcal{N}_v^+(G_\gamma) \neq \phi$. This implies that for any two agents $(u, v)$ in $S$, $\mathcal{N}_u^+(G_\gamma) \cap \mathcal{N}_v^+(G_\gamma) = \phi$. Therefore, we can analyse the regret incurred by each of these agents in $S$ in isolation. We have by Lemma 4 of Kolla et al. (2018) that for any individually consistent agent $v \in S$, arm $1 \leq k \leq K$,

$$\liminf_{T\to\infty} \frac{\mathbb{E}[n_{v,k}(t)|\overline{\mathcal{H}}_v(t)]}{\log(T)} \geq \frac{1}{\mathbb{D}_k^{\mathsf{inf}}}.$$

Let $N_{v,k}(t)$ denote the cumulative arm pulls for arm $k$ within $\mathcal{N}_v^+(G_\gamma)$. We therefore have,

$$\liminf_{T\to\infty} \frac{\mathbb{E}[n_k(t)]}{\log(T)} = \liminf_{T\to\infty} \frac{\mathbb{E}[\sum_{v\in S} N_{v,k}(t)]}{\log(T)} \geq \sum_{v\in S} \liminf_{T\to\infty} \frac{\mathbb{E}[n_{v,k}(t)|\overline{\mathcal{H}}_v(t)]}{\log(T)} \geq \frac{\alpha(G_{\gamma+1})}{\mathbb{D}_k^{\mathsf{inf}}}.$$

The first inequality follows from the fact that $N_{v,k}(t) \geq n_{v,k}(t)$. □

**Remark 2.3.** The above lower bound suggests that for NAIC policies one can expect an asymptotic regret that is much larger than that for a general policy (Theorem 2.3). For instance, when we consider a circular topology with $M$ nodes, one can obtain that $\alpha(G_{\gamma+1}) \geq \lfloor \frac{M}{\gamma+2} \rfloor$, suggesting an $\Omega(M \log(T))$ regret bound. We will see that several introduced policies that are in fact NAIC obtain regret within a constant factor of this lower bound. However, for non-NAIC policies, one can obtain smaller rates as well, albeit at the expense of consistency, i.e., some agents will actively "explore for other agents".

## 2.4 Minimax Lower Bounds

To supplement the previous *problem-dependent* results, we present a lower bound to characterize the minimax optimal rates for the federated multi-armed bandit problem. We present first an assumption on multi-agent policies.

**Assumption 2.2** (Agnostic decentralized policies)**.** *A set of $N$ policies $\pi_1, ..., \pi_N$ are termed agnostic decentralized policies, if for every pair $(i, j)$ of agents that communicate in $G$ and each $t \in [T]$, $\pi_i(t)$ is independent of $\{\pi_j(\tau)\}_{\tau=1}^{t-d(i,j)}$ conditioned on the rewards $\{(A_j(\tau), r_j(\tau))\}_{\tau=1}^{t-d(i,j)}$.*

**Theorem 2.5** (Minimax Rate)**.** *For any multi-agent algorithm $\mathcal{A}$, there exists a $K-$armed environment over $N$ agents with $\Delta_k \leq 1$ such that,*

$$\mathfrak{R}(T; \mathcal{A}) \geqslant 0.027\sqrt{KN(T + \widetilde{d}(G))}.$$

*Furthermore, if $\mathcal{A}$ is an agnostic decentralized policy, there exists a $K-$ armed environment over $N$ agents with $\Delta_k \leq 1$ for any connected graph $G$ and $\gamma \geq 1$ such that*

$$\mathfrak{R}(T; \mathcal{A}) \geqslant 0.019\sqrt{\alpha^\star(G_\gamma)KNT}.$$

*Where $\tilde{d}(G)$ denotes the average delay incurred by any message across the network $G$, and $\alpha^\star(G_\gamma) = \frac{N}{1+\bar{d}_\gamma}$ is Turan's lower bound (Turán, 1941) on $\alpha(G_\gamma)$.*

*Proof.* Let $\mathcal{A}$ be a deterministic (multi-agent) algorithm, and let the empirical distribution of arm pulls across all agents be given by $p^{(i)}(t) = \left(p_1^{(i)}(t), ..., p_K^{(i)}(t)\right)$, where $p_k(t) = \frac{n_i^k(T)}{T}$. Consider the random variable $J_t^{(i)}$ drawn according to $p^{(i)}(t)$ and $\mathbb{P}_i$ denote the law of $J_t$ when drawn from arm $k$ having parameter $\frac{1+\varepsilon}{2}$ (and other arms with parameter $\frac{1-\varepsilon}{2}$). We have,

$$\mathbb{P}_k\left(J_t^{(i)} = j\right) = \mathbb{E}_k\left[\frac{n_i^k(T)}{T}\right].$$

Since on pulling any arm $k' \neq k$, we obtain regret $\varepsilon$, we therefore have for the group regret,

$$\begin{aligned}\mathbb{E}_k\left[\sum_{t=1}^{T}\left(M \cdot r_k(t) - \sum_{i\in\mathcal{V}} r_{A_i}(t)\right)\right] &= \varepsilon \cdot T \cdot \sum_{i\in\mathcal{V}} \mathbb{P}_k\left(J_t^{(i)} = k'\right)\\ &= \varepsilon \cdot T \cdot \sum_{i\in\mathcal{V}}\left(1 - \sum_{k'\neq k}\mathbb{P}_k\left(J_t^{(i)} = k'\right)\right).\end{aligned}$$

By Pinsker's inequality and averaging over all $k \in [K]$, we have for any $i \in \mathcal{V}$,

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{P}_k\left(J_t^{(i)} = k\right) \leqslant \frac{1}{K} + \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{2}\mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0, \mathbb{P}_k)}.$$

We now bound the R.H.S. using the chain rule for KL-divergence. Since we assume that $\mathcal{A}$ is deterministic, we have that the rewards obtained by the agent $i$ until time $t$ from its neighborhood alone determine uniquely the empirical distribution of plays. Here, the analysis diverges from that of the single-agent bandit as a richer set of observations is available to each agent. Denote the set of rewards observed by agent $i$ at instant $\tau$ be given by $\mathcal{O}_i(\tau)$. First, observe that since each reward is i.i.d., we have for any $k$,

$$\mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0(\mathcal{O}_i(\tau)), \mathbb{P}_k(\mathcal{O}_i(\tau))) = |\mathcal{O}_i(\tau)| \cdot \mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}\right)$$

For $k = 0$ the above divergence is 0. When we consider the standard single-agent setting, $|\mathcal{O}_i(\tau)| = 1$, recovering the usual bound. Now, by the chain rule, we have that, at round $t$ for any agent $i$, and arm $k \in [K]$,

$$\begin{aligned}\mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0(t), \mathbb{P}_k(t)) &= \mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0(1), \mathbb{P}_k(1)) + \sum_{\tau=2}^{t} |\mathcal{O}_i(\tau)|\, \mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}\right) \\ &= \mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}\right) \mathbb{E}_0\left[\sum_{j\in\mathcal{V}} n_j^k(t - d(i,j))\right].\end{aligned}$$

Replacing this result in the earlier equation, we have by the concavity of $\mathbb{D}_{\mathsf{KL}}$ divergence:

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K} \mathbb{P}_k\left(J_t^{(i)} = k\right) &\leqslant \frac{1}{K} + \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{2}\mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0, \mathbb{P}_k)} \\ &\leqslant \frac{1}{K} + \frac{1}{K}\sum_{k=1}^{K}\sqrt{\mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}\right) \mathbb{E}_0\left[\sum_{j\in\mathcal{V}} n_j^k(T - d(i,j))\right]} \\ &\leqslant \frac{1}{K} + \sqrt{\left(\frac{TM - \sum_{j=1}^{d^\star(G)} d_{=j}(i)\cdot j}{K}\right)\cdot \mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}\right)}.\end{aligned}$$

Now, observe that the KL divergence between Bernoulli bandits can be bounded as

$$\mathbb{D}_{\mathsf{KL}}(p, q) \leq \frac{(p-q)^2}{q(1-q)}.$$

Substituting we get,

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{P}_k\left(J_t^{(i)} = k\right) \leqslant \frac{1}{K} + \sqrt{\frac{4\varepsilon^2(MT - \sum_{j=1}^{d^\star(G)} d_{=j}(i)\cdot j)}{(1-\varepsilon^2)K}}.$$

Replacing this in the regret and using $\varepsilon \leqslant 1/2$, we get that,

$$\begin{aligned}
&\mathbb{E}_k\left[\sum_{t=1}^{T}\left(M\cdot r_k(t)-\sum_{i\in\mathcal{V}} r_{A_i}(t)\right)\right]\\
&\geqslant \varepsilon\cdot T\cdot\sum_{i\in\mathcal{V}}\left(1-\frac{1}{K}-\sqrt{\frac{4\varepsilon^2(MT-\sum_{j=1}^{d^\star(G)} d_{=j}(i)\cdot j)}{(1-\varepsilon^2)K}}\right)\\
&\geqslant \varepsilon\cdot T\cdot\sum_{i\in\mathcal{V}}\left(\frac{1}{2}-4\varepsilon\sqrt{\frac{(MT-\sum_{j=1}^{d^\star(G)} d_{=j}(i)\cdot j)}{3K}}\right)\\
&=\frac{\varepsilon\cdot MT}{2}-\frac{4\varepsilon^2 MT}{\sqrt{K}}\left(\sum_{i,j\in\mathcal{V}} T-d(i,j)\right)^{1/2}
\end{aligned}$$

Setting $\varepsilon = c\cdot\sqrt{\frac{K}{M(T-\sum_{j=1}^{d^\star(G)}\bar{d}_{=j}\cdot j)}}$ where $c$ is a constant to be tuned later, we have,

$$\begin{aligned}
\mathbb{E}_k\left[\sum_{\tau=1}^{T}\left(M\cdot r_{k,t}-\sum_{i\in\mathcal{V}} r_{A_i(t),t}\right)\right] &\geqslant \left(\frac{c}{2}-\frac{4c^2}{\sqrt{3}}\right)\cdot\sqrt{\frac{KM^2T^2}{M(T-\sum_{j=1}^{d^\star(G)}\bar{d}_{=j}\cdot j)}}\\
&\geqslant 0.027\sqrt{KM(T+\sum_{j=1}^{d^\star(G)}\bar{d}_{=j}\cdot j)}.
\end{aligned}$$

This proves the first part of the theorem. Now, when the policies are decentralized and agnostic, the chain rule step can be factored as follows.

$$\begin{aligned}
\mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0(t),\mathbb{P}_k(t)) &= \mathbb{D}_{\mathsf{KL}}(\mathbb{P}_0(1),\mathbb{P}_k(1))+\sum_{\tau=2}^{t}|\mathcal{O}_i(\tau)|\,\mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2},\frac{1+\varepsilon}{2}\right)\\
&= \mathbb{D}_{\mathsf{KL}}\left(\frac{1-\varepsilon}{2},\frac{1+\varepsilon}{2}\right)\mathbb{E}_0\left[\sum_{j\in\mathcal{N}^+_\gamma(G)} n^k_j(t-d(i,j))\right].
\end{aligned}$$

Note that here instead of taking the cumulative sum over all $\mathcal{V}$ we select only those agents that are within the $\gamma-$neighborhood of $i$ in $G$, since conditioned on these observations the rewards of the agents are independent of all other rewards (by Assumption), and hence the higher-order KL divergence terms are 0. Replacing this in the analysis gives us the following decomposition (after similar steps as the first part):

$$\mathbb{E}_k\left[\sum_{t=1}^{T}\left(Mr_k(t)-\sum_{i\in\mathcal{V}} r_{A_i}(t)\right)\right] \geqslant \frac{MT\varepsilon}{2} - \frac{4\varepsilon^2 T}{\sqrt{3K}}\cdot\sum_{i\in\mathcal{V}}\left(\sum_{j:\mathcal{N}_\gamma^+(i)} T-d(i,j)\right)^{1/2}$$

$$\geqslant \frac{MT\varepsilon}{2} - \frac{4\varepsilon^2 M^{1/2} T}{\sqrt{3K}}\cdot\left(\sum_{i\in\mathcal{V}}\sum_{j:\mathcal{N}_\gamma^+(i)} T-d(i,j)\right)^{1/2}$$

Setting $\varepsilon = c\cdot\sqrt{\frac{MK}{\sum_{i\in\mathcal{V}}\sum_{j:\mathcal{N}_\gamma^+(i)} T-d(i,j)}}$ where $c$ is a constant to be tuned later, we have,

$$\mathbb{E}_k\left[\sum_{t=1}^{T}\left(M\cdot r_k(t)-\sum_{i\in\mathcal{V}} r_{A_i}(t)\right)\right] \geqslant \left(\frac{c}{2}-\frac{4c^2}{\sqrt{3}}\right)\cdot\sqrt{\frac{M^3T^2}{\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i^+(G_\gamma)} T-d(i,j)}}$$

$$\geqslant \left(\frac{c}{2}-\frac{4c^2}{\sqrt{3}}\right)\cdot\sqrt{\frac{M^3T}{\sum_{i\in\mathcal{V}} 1+d_i(G_\gamma)}}$$

$$\geqslant \frac{3}{4}\left(\frac{c}{2}-\frac{4c^2}{\sqrt{3}}\right)\sqrt{\alpha^\star(G_\gamma)MT}$$

$$\geqslant 0.019\sqrt{\alpha^\star(G_\gamma)MT}.$$

The constants in both settings are obtained by optimizing $c$ over $\mathbb{R}$. Extending this to random (instead of deterministic) algorithms is straightforward via Fubini's theorem, see Theorem 2.6 of Bubeck (2010). □

In the forthcoming chapters, we will discuss algorithms that build on this basic model, and study a variety of different constrained environments. We now present a brief summary of relevant work in this problem area.

## 2.5 Related Work and Discussion

**Prior work in Multi-Agent Multi-Armed Bandits**. The federated setup considered here is part of a broader literature on cooperative decision-making for the stochastic multi-armed bandit, which has recently seen a lot of research interest. Decentralized cooperative estimation has been explored for sub-Gaussian stochastic bandits using a *running consensus* protocol in Landgren et al. (2016a,b); Martínez-Rubio et al. (2019) and for adversarial bandits Cesa-Bianchi et al. (2019b); Bar-On & Mansour (2019) using a message-passing protocol.

Localized decision-making for sub-Gaussian rewards has also been explored in the work of Landgren et al. (2018), and a fully-centralized algorithm in Shahrampour et al. (2017), where all agents select the same action via voting.The stochastic bandit with multiple pulls Xia et al.; Anantharam et al. (1987) is equivalent to the cooperative multi-armed bandit on a complete $\mathcal{G}$ with a centralized actor (since there are no delays and all agents have the same information $\forall t \in [T]$).

Contrasted to cooperative settings, there is extensive research in competitive settings, where multiple agents compete for arms Bistritz & Leshem (2018); Bubeck et al. (2019); Liu & Zhao (2010b,c,a). For strategic experimentation, Brânzei & Peres (2019) provide an interesting comparison of exploration in cooperative and competitive agents.

A closely-related problem setting is the single-agent *social network* bandit, where a user is picked at random every trial, and the algorithm must infer its contextual mean reward Cesa-Bianchi et al. (2013); Li et al. (2016); Gentile et al. (2014, 2017), while assuming an underlying *clustering* over the users. This problem setting, while relevant, crucially differs from the one considered herein, since (a) this is a single-agent setting (only one action is taken every round), and (b) there are no delays in the rewards obtained. While a multi-agent variant has been considered Korda et al. (2016), this work also assumes no delays in communication.

## 2.6 Omitted Proofs

### 2.6.1 Proof of Theorem 2.3

The lower bound proceeds in a manner similar to the lower bound achieved in the single-agent case (Lattimore & Szepesvári, 2020). We first state a few intermediary results.

**Theorem 2.6** (Carathéodory's Extension Theorem). *Let $(\Omega_1, \mathcal{F}_1), ..., (\Omega_n, \mathcal{F}_n)$ be measurable spaces and $\bar{\mu} : \mathcal{F}_1 \times ... \times \mathcal{F}_n \to [0, 1]$ be a function such that (a) $\bar{\mu}(\Omega_1 \times ... \times \Omega_n) = 1$, and (b) $\bar{\mu}\left(\cup_{k:\Delta_k>0}^{\infty} A_k\right) = \sum_{k:\Delta_k>0}^{\infty} \bar{\mu}(A_k)$ for all sequences of disjoint sets with $A_k \in \mathcal{F}_1 \times ... \times \mathcal{F}_n$. Let $\Omega = \Omega_1 \times ... \times \Omega_n$ and $\mathcal{F} = \sigma(\mathcal{F}_1 \times ... \times \mathcal{F}_n)$. Then there exists a unique probability measure $\mu$ on $(\Omega, \mathcal{F})$ such that $\mu$ agrees with $\bar{\mu}$ on $\mathcal{F}_1 \times ... \times \mathcal{F}_n$.*

**Theorem 2.7** (Multiagent Divergence Decomposition). *Let $\mathcal{E} = \prod_{k\in[K]} \mathcal{M}_k$ be a structured family of K-armed bandit problems and $\nu = (\nu_k)_{k\in[K]}, \nu' = (\nu'_k)_{k\in[K]} \in \mathcal{E}$ be two bandit problem instances. Then, for any decentralized policy $\Pi_t = (\pi_{m,t})_{m\in[M],t\in[T]}$ that uses a communication protocol satisfying Assumptions 2.1, the following is true.*

$$\mathbb{E}_{\nu\Pi}\left[\log \frac{d\mathbb{P}_{\nu\Pi}}{d\mathbb{P}_{\nu'\Pi}}(A_{1,1}, X_{1,1}, ..., a_m(t), X_{M,T})\right] = \sum_{k:\Delta_k>0}^{K} \mathbb{E}_{\nu\Pi}\left[N_k(T)\right] \mathbb{D}_{\mathsf{KL}}(\nu_{A_k}, \nu'_{A_k}).$$

*Here, $\mathbb{P}_{\nu\Pi}$ and $\mathbb{P}_{\nu'\Pi}$ denote the product measures arising from the interaction of $\nu$ and $\nu'$ with $\Pi$, and $N_k(T)$ denotes the total number of pulls of arm $k$ across all $M$ agents at time $T$.*

*Proof.* Consider an agent $m$, and let the $\gamma$-neighborhood of this agent be given by $\mathcal{N}_\gamma(m)$, such that $N = |\mathcal{N}_\gamma(m)|$. At any instant $t$, let the agents actions be denoted by $a_m(t)$, and the associated outcome variable be $X_{m,t}$. We denote the set of action-reward pairs for the agent at time $t$ as $H_t^m = (A_{m,1}, X_{m,1}, ..., a_m(t), X_{m,t})$.

At any instant $t$, the agent also receives messages from neighboring agents, delayed by their distance in the graph $\mathcal{G}$. By our third assumption we can write each message $Z_{m,m'}(t)$ sent fom agent $m$ to agent $m'$ at time $t$ as the following, for some deterministic, bijective differentiable function(s) $(f_i)_{i\in[L]}$ where $L$ is the length of the message.

$$Z_{m,m'}(t) = f_i\left(A_{m,1}, X_{m,1}, ..., a_m(t), X_{m,t}\right)$$

For each $t \in [T]$, let $\Omega_t = ([K] \times \mathbb{R})^{Mt} \subset \mathbb{R}^{2Mt}$ and $\mathcal{F}_t = \mathfrak{B}(\Omega_T)$. As is the case with single-agent bandits, we can define coordinate projections that govern each of the random variables $A_t^m, X_t^m \; \forall t, m$ by creating an ordering of all elements of $H_t$.

$$a_m(t)\left(a_{1,1}, x_{1,1}, ..., a_m(t), x_{M,T}\right) = a_m(t)$$
$$X_{m,t}\left(a_{1,1}, x_{1,1}, ..., a_m(t), x_{M,T}\right) = x_{m,t}$$

By our assumption on the nature of messages, we can express the density of any message $z_{m,m'}(t)$ as the following. Let $\boldsymbol{F} = (f_i)_{i\in[L]}$.

$$\begin{aligned} p(z_{m,m'}(t)) &= p(\boldsymbol{F}^{-1}\left(z_{m,m'}(t)\right) \left|\det\left(\boldsymbol{J}(z_{m,m'}(t)\right)\right| \\ &= p(\boldsymbol{F}^{-1}\left(\boldsymbol{F}(a_{m,1}, x_{m,1}, ..., a_m(t), x_{m,t})\right)) \left|\det\left(\boldsymbol{J}(z_{m,m'}(t)\right)\right| \\ &= p(a_{m,1}, x_{m,1}, ..., a_m(t), x_{m,t}) \left|\det\left(\boldsymbol{J}(z_{m,m'}(t)\right)\right| \\ &= p(a_{m,1}, x_{m,1}, ..., a_m(t), x_{m,t})\Lambda(m, m', t) \end{aligned}$$

This primarily implies that each message is completely specified by the corresponding inputs. With this probability space $(\Omega_T, \mathcal{F}_T)$ we can then define a **decentralized policy** as a sequence $(\Pi_t)_{t=1}^T$, where $\Pi_t = (\pi_{m,t})_{m\in[M]}$ is a probability kernel from $(\Omega_{t-1}, \mathcal{F}_{t-1})$ to $([K]^M, 2^{[K]^M})$.

We now require a valid measure that connects $\Pi = (\Pi_t)_{t=1}^T$ and $\nu = (\nu_k)_{k\in[K]} \in \mathcal{E}$. The measure we will define will be similar to that of the canonical bandit model, however, we have a few key differences. First, we must note that individual elements $\pi_{m,t}$ of $\Pi_t$ factorize differently based on $\mathcal{G}$. Additionally, conditioned on $H_t^m$, $X_{m,t}$ follows the law $\nu_{a_m(t)}$, i.e. it only depends on the corresponding arm pulled by the agent $m$. Therefore the conditions on the measure can be listed as follows.

1. The conditional distribution of $A_t^m$ given $\cup_{m\in[M]}\left(H_t^m\right)$ is

   $$\pi_{m,t}\left(\cdot | \left(H_{t-1}^m \cup_{m'\in\mathcal{N}_\gamma(m)} \left(H_{t-d(m,m')-1}^{m'}\right)\right)\right)$$

   almost surely. This condition is justified by the fact that each individual policy for an agent $m$ can only be dependent on information in the $\gamma$-neighborhood of the agent, and that information takes $d(m, m')$ steps to reach agent $m$ from any other agent $m'$.

2. The conditional distribution of $X_{m,t}$ given $\bigcup_{m\in[M]}\left(H_t^m\right)$ is $\nu_{A_{m,t}}$ almost surely.

Let $\lambda$ be a $\sigma$-finite measure on $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ for which $\nu_k$ is absolutely continuous with respect to $\lambda$ for all $k \in [K]$. Let $p_k = \frac{d\nu_k}{d\lambda}$ be the Radon-Nikodym derivative of $\nu_k$ with respect to $\lambda$, and $\rho$ be the counting measure over $\mathfrak{B}(\mathbb{R})$, We can define the density $p_{\nu\Pi} : \Omega \to \mathbb{R}$ with respect to the product measure $(\rho \times \mathbb{R})^{MT}$ as the following.

$$
\begin{aligned}
&p_{\nu\Pi}\left(a_{1,1}, x_{1,1}, ..., a_m(t), x_{M,T}\right) = \\
&\prod_{m \in [M]} \pi_{m,T}\left(a_m(t) | a_{1,1}, x_{1,1}, ..., a_{M,T-1}, x_{M,T-1}\right) p\left(a_{1,1}, x_{1,1}, ..., a_{M,T-1}, x_{M,T-1}\right) \nu_{a_m(t)}(x_{m,T}) \\
&= \left( \prod_{m \in [M]} \prod_{t \in [T]} \pi_{m,t} \left( a_m(t) | \bigcup_{m' \in \mathcal{N}_\gamma(m) \cup \{m\}} \left\{ a_{m',1}, x_{m',1}, ... a_{m',t-d(m,m')}, x_{m',t-d(m,m')} \right\} \right) \right) \times \\
&\qquad \left( \prod_{m' \in [M]} \prod_{t \in [T]} \nu_{a_m(t)}(x_{m,t}) \right) \times \left( \prod_{m \in [M]} \prod_{m' \in \mathcal{N}_\gamma(m)} \prod_{t \in [T - d(m,m')]} \Lambda(m, m', t) \right)
\end{aligned}
$$

It can be easily shown that $p_{\nu\Pi}$ is a valid density, and satisfies the two properties listed earlier. By Theorem 2.6, we know that such a distribution exists. Let the corresponding measure be denoted by $\mathbb{P}_{\nu\Pi}$. We now prove a version of the canonical divergence decomposition in the presence of additional observations.

In addition to $\nu$, let $\nu' = (\nu'_k)_{k \in [K]} \in \mathcal{E}$ be the reward distributions associated with a separate $k$-armed bandit problem, and $\mathbb{P}_{\nu'\pi}$ denote the joint measure for $\nu'$ under the same policy $\pi$. Assume that $\mathbb{D}_{\mathsf{KL}}(\nu_k, \nu'_k) < \infty$, for all $k \in [K]$. We then have,

$$
\log \frac{d\mathbb{P}_{\nu\Pi}}{d\mathbb{P}_{\nu'\Pi}} \left(a_{1,1}, x_{1,1}, ..., a_m(t), x_{M,T}\right) = \sum_{m \in [M]} \sum_{t \in [T]} \log \left( \frac{\nu_{a_m(t)}(x_{m,t})}{\nu'_{a_m(t)}(x_{m,t})} \right)
$$

This follows from the chain rule of Radon-Nikodym derivatives and the fact that (a) the policy terms cancel out by the definitions of $p_{\nu\Pi}$ and $p_{\nu'\Pi}$, and (b) communication terms $\Lambda$ cancel out since they are independent of $\nu$. Taking expectations and replacing $H_T = \bigcup_{m \in [M]} (H_T^m)$, we have,

$$
\mathbb{E}_{\nu\Pi} \left[ \log \frac{d\mathbb{P}_{\nu\Pi}}{d\mathbb{P}_{\nu'\Pi}} (H_T) \right] = \sum_{m \in [M]} \sum_{t \in [T]} \mathbb{E}_{\nu\pi} \left[ \log \left( \frac{\nu_{a_m(t)}(X_{m,t})}{\nu'_{A_{m,t}}(X_{m,t})} \right) \right]. \tag{2.4}
$$

Additionally, we also know that, for all $t \in [T], m \in [M]$,

$$
\mathbb{E}_{\nu\Pi} \left[ \log \left( \frac{\nu_{a_m(t)}(X_{m,t})}{\nu'_{A_{m,t}}(X_{m,t})} \right) \right] = \mathbb{E}_{\nu\Pi} \left[ \mathbb{E}_{\nu\Pi} \left[ \log \left( \frac{\nu_{a_m(t)}(X_{m,t})}{\nu'_{A_{m,t}}(X_{m,t})} \right) \right] \middle| a_m(t) \right]
$$

$$= \mathbb{E}_{\nu\Pi}\left[\mathbb{D}_{\mathsf{KL}}(\nu_{a_m(t)}, \nu'_{a_m(t)})\right].$$

Replacing the above identity in Equation (2.4), we have,

$$\begin{aligned}
\mathbb{E}_{\nu\Pi}\left[\log \frac{d\mathbb{P}_{\nu\pi}}{d\mathbb{P}_{\nu'\pi}}(H_T)\right] &= \sum_{m\in[M]}\sum_{t\in[T]} \mathbb{E}_{\nu\Pi}\left[\mathbb{D}_{\mathsf{KL}}(\nu_{a_m(t)}, \nu'_{a_m(t)})\right] \\
&= \sum_{k:\Delta_k>0}^{K}\sum_{m\in[M]}\sum_{t\in[T]} \mathbb{E}_{\nu\Pi}\left[\mathbb{1}\left\{A_{m,t}=k\right\}\right]\mathbb{D}_{\mathsf{KL}}(\nu_{A_k}, \nu'_{A_k}) \\
&= \sum_{k:\Delta_k>0}^{K} \mathbb{E}_{\nu\Pi}\left[N_k(T)\right]\mathbb{D}_{\mathsf{KL}}(\nu_k, \nu'_k).
\end{aligned}$$

□

**Theorem 2.8** (Bretagnolle-Huber Inequality). *Let $P$ and $Q$ be probability measures on the same measurable set $(\Omega, \mathcal{F})$ and let $A$ be an arbitrary event. Then,*

$$\mathbb{P}_P(A) + \mathbb{P}_Q(A^c) \geq \frac{1}{2}\exp\left(-\mathbb{D}_{\mathsf{KL}}(P, Q)\right).$$

*Here $A^c$ denotes the complement event.*

We are now ready to prove Theorem 2.3.

*Proof.* This proof follows the standard approach for single-agent consistent bandit algorithms. Consider any suboptimal arm $i$ and let $\delta > 0$ be arbitrary. Consider $\nu' = (\nu'_k)_{k\in[K]} \in \mathcal{E}$ such that $\mathbb{D}_{\mathsf{KL}}(\nu_k, \nu'_k) \leq \mathbb{D}_i^{\mathsf{inf}} + \delta$, and $\mu(\nu'_i) > \mu^*$, which exists by the definition of $\mathbb{D}_i^{\mathsf{inf}}$. By Theorems 2.8 and 2.7 we have the following for any event $A$.

$$\mathbb{P}_{\nu\Pi}(A) + \mathbb{P}_{\nu'\Pi}(A^c) \geq \frac{1}{2}\exp\left(-\mathbb{E}_{\nu\Pi}[N_i(T)](\mathbb{D}_i^{\mathsf{inf}} + \delta)\right)$$

Let $R_{\mathcal{G}} = R_{\mathcal{G}}(T, \nu, \Pi)$ be the regret obtained by $\Pi$ on $\nu$ and $R'_{\mathcal{G}} = R_{\mathcal{G}}(T, \nu', \Pi)$ be the regret obtained by $\Pi$ on $\nu'$. By choosing $A = \{N_i(T) > T/2\}$, we have,

$$\begin{aligned}
R_{\mathcal{G}} + R'_{\mathcal{G}} &\geq \frac{T}{2}\left(\mathbb{P}_{\nu\Pi}(A)\Delta_i + \mathbb{P}_{\nu'\Pi}(A^c)(\mu'_i - \mu^*)\right) \\
&\geq \frac{T}{2}\min\{\Delta_i, \mu'_i - \mu^*\}\left(\mathbb{P}_{\nu\Pi}(A) + \mathbb{P}_{\nu'\Pi}(A^c)\right) \\
&\geq \frac{T}{4}\min\{\Delta_i, \mu'_i - \mu^*\}\exp\left(-\mathbb{E}_{\nu\Pi}[N_i(T)](\mathbb{D}_i^{\mathsf{inf}} + \delta)\right)
\end{aligned}$$

Rearranging and taking limit inferior, we have,

$$\begin{aligned}\liminf_{T\to\infty}\frac{\mathbb{E}_{\nu\Pi}[N_i(T)]}{\ln(T)} &\geq \frac{1}{\mathbb{D}_i^{\mathsf{inf}}+\delta}\liminf_{T\to\infty}\frac{\ln\left(\frac{T\min\{\Delta_i,\mu_i'-\mu^*\}}{R_{\mathcal{G}}+R_{\mathcal{G}}'}\right)}{\ln(T)}\\ &\geq \frac{1}{\mathbb{D}_i^{\mathsf{inf}}+\delta}\left(1-\limsup_{T\to\infty}\frac{\ln\left(R_{\mathcal{G}}+R_{\mathcal{G}}'\right)}{\ln(T)}\right)\end{aligned}$$

Using the fact that $\Pi$ is consistent, we have for some constant $a > 0$ and constant $C_a$,

$$\geq \frac{1}{\mathbb{D}_i^{\mathsf{inf}}+\delta}\left(1-\limsup_{T\to\infty}\frac{a\log(T)+\ln(C_a)}{\ln(T)}\right).$$

Since $a > 0$ is arbitrary, taking the limit as $\delta$ goes to zero, we have, for any suboptimal arm $i$,

$$\liminf_{T\to\infty}\frac{\mathbb{E}_{\nu\Pi}[N_i(T)]}{\ln(T)} \geq \frac{1}{\mathbb{D}_i^{\mathsf{inf}}}.$$

Plugging this into the definition of regret and rearranging gets us the final result. □

# Chapter 3

# Differentially-Private Federated Multi-Armed Bandits

In this chapter we discuss privacy-preserving approaches to federated learning of multi-armed stochastic bandits. The algorithm presented herein is an upper confidence bound strategy that utilizes *pure* differential privacy achieved by the Laplace mechanism (Dwork & Roth, 2014), however, analogous algorithms can be developed for the *approximate* differential privacy regime by utilizing the Gaussian mechanism instead, with an albeit simpler analysis of sub-Gaussian confidence intervals. We begin by first providing an overview of the precise $\varepsilon$-differential privacy guarantee considered.

## 3.1 Differential Privacy in Federated Multi-Armed Bandits

The standard stting in centralized privacy-preserving decision-making systems involves an agent that interacts with a new user at every round $t$, and must ensure that the policy $\pi(t)$ is differentially private with respect to the rewards obtained from the previous $t-1$ users (Tossou & Dimitrakakis, 2016; Mishra & Thakurta, 2015). In the federated setting, each agent is assumed to interact with a new user each round, and we require that *each* agents policy is differentially-private with respect to *all* previous $M \cdot (t-1)$ rewards. However, observe that for any agent, its policy is a function of its own decision history and only the *messages* it receives from other agents. By making the *messages* $\varepsilon$-DP with respect to the reward sequence an agent observes, we can therefore, ensure that any policy using these messages is also $\varepsilon$-DP with respect to the rewards obtained by other agents (by the post-processing property of DP, Proposition 2.1 of Dwork & Roth (2014)). We use this to define differential privacy in the decentralized bandit context.

**Definition 3.1** ($\varepsilon$-DP message protocol)**.** *For any agent $m$, a message $\mathbf{q}_m(t)$ composed of $L$ independent functions $(f_i)_{i\in[L]}$ at time $t$ is $\varepsilon$-differentially private with respect to its personal reward history if for all histories $H_m(t)$ and $H'_m(t)$ that differ in at most one sample, we have $\forall \mathcal{S}_i \subseteq Range(f_i)$:*

$$\left|\log\left(\prod_{i=1}^{L}\frac{\mathbb{P}\left(f_i \in \mathcal{S}_i | H_m(t)\right)}{\mathbb{P}\left(f_i \in \mathcal{S}_i | H'_m(t)\right)}\right)\right| \leq \varepsilon.$$

This requirement can be satisfied by individually ensuring that each of the $L$ outputs in a message are $(\varepsilon/L)-$DP, but we adopt this definition to allow for tunable thresholds for each parameter. Note that there are several advantages to this definition. First, we see that unlike centralized or single-agent bandit algorithms that inject noise as a part of the central algorithm itself, our definition requires each message to *individually* preserve an irreducible level of privacy regardless of the recipient agents' algorithm. While federated systems are typically designed to have a (trusted) server to route communication between agents (Kairouz et al., 2019), our protocol does not assume any trusted entity in the network.

We additionally can see that this design evidently manifests a privacy-communication frontier: communicating often will require a larger privacy budget (by composition). For instance, if an agent communicates $n = \mathcal{O}(\log T)$ messages, where message each is ensuring $\varepsilon-$DP, the cumulative privacy budget $\varepsilon_T = \mathcal{O}(\varepsilon \cdot \log T)$ by basic composition, but each message can be allowed a different privacy threshold in order to improve performance. In our algorithm, we will observe precisely this phenomenon, whereby each agent maintains $\varepsilon-$DP by allocating different privacy budgets to each outgoing message. We now briefly review the Laplace mechanism, our central technical tool to maintain privacy.

### The Laplace Mechanism

One of the central techniques to introduce differential privacy for functions with range within $\mathbb{R}^d$ is the Laplace mechanism (Section 3.3 of Dwork & Roth (2014)). For any domain $\mathcal{D}$ and function $f : \mathcal{D} \to \mathbb{R}$, we can define the $\ell_1$ *sensitivity* of $f$ for two neighboring datasets $D, D' \in \mathcal{D}$ (i.e., datasets that differ only in one entry) as follows.

$$s_1(f) = \max_{D,D'\in\mathcal{D}} |f(D) - f(D')| \tag{3.1}$$

The Laplace mechanism operates by adding zero-mean Laplace noise to the output of $f$ with a scale that is governed by the level of privacy required and $s_1(f)$, as follows.

**Lemma 3.1** (Theorem 4 of Dwork & Roth (2014))**.** *For any real function $f$ computed on data $D$, releasing $f(D) + X$ where $X$ is drawn from a Laplace distribution with scale parameter $\beta$ is $\frac{\beta}{s_1(f)}$-differentially private with respect to $D$.*

The Laplace mechanism is the underlying approach that we utilize in guaranteeing privacy in the multi-agent setting. We will now describe our message-passing protocol, which uses this mechanism to guarantee differential privacy between any pair of agents.

## 3.2 Differentially-Private Message-Passing

Algorithms for privacy in multi-armed bandits and online learning involve using a binary tree mechanism to compute the running sum of rewards for any arm (Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016). The fundamental intuition behind this strategy is to reduce the effective sensitivity of maintaining a rolling sum. As an example, consider privately maintaining the sum of a series of real numbers $x_1, ..., x_T, x_i \in [0,1]$. Naively computing the sum $s_t = \sum_{\tau=1}^{t} x_\tau$ at any instant $t$ has the drawback that the sensitivity of $s_t$ is $o(1)$, and hence, to ensure that each element of the sequence $s_1, ..., s_T$ obeys $\varepsilon-$DP, one must release $s_t$ by adding Laplace noise with scale $\mathcal{O}(1)$, making the entire sequence $\varepsilon \cdot T-$DP, weakening the privacy budget significantly. Alternatively, one can utilize the tree-based mechanism for maintaining private partial sums, introduced in Chan et al. (2010) and Dwork & Smith (2010). This mechanism involves maintaining a binary tree of individual rewards, where each node in the tree stores a privatized partial sum of its children. Since inserting and accessing elements in a balanced binary tree with $T$ elements requires accessing at most $\mathcal{O}(\log T)$ elements. This effectively reduces the overall privacy budget required to $\mathcal{O}(\varepsilon \cdot \log T)$, greatly reducing the error from the noise. Tossou & Dimitrakakis (2016) utilize this mechanism to provide a bandit algorithm titled DP-UCB that maintains one tree for each of $K$ arms. If an arm has been pulled $n$ times at round $t$, one requires introducing Laplace noise with sensitivity $\frac{\ln n}{\varepsilon}$ to achieve $\varepsilon$-differential privacy. As a consequence, the algorithm obtains regret inversely proportional to privacy level.

**Theorem 3.1** (Theorem 3.2, Tossou & Dimitrakakis (2016)). DP-UCB *obtains regret*

$$\mathfrak{R}(T) = \mathcal{O}\left(\sum_{k>1}^{K} \log(T) \cdot \max\left\{\frac{1}{\varepsilon}\log\left(\frac{\log(T)}{\varepsilon}\right), \frac{1}{\Delta_k}\right\} + \sum_{k>1}^{K} \Delta_k.\right).$$

While the above bound is presented in a rather coarse manner and can be tuned to improve the constants, the primary takeaway is that there is a substantial privacy cost for "easy" arms, i.e., when $\Delta_k$ is large. Furthermore, while this approach is feasible in the single-agent case, in the distributed setting, we will have to maintain $M$ separate binary trees (one for each agent), which would create an overhead of a factor of $M$ in the regret.

### 3.2.1 Communication Mechanism

To mitigate this overhead we propose a *distributed interval* mechanism, which extends the basic tree-based mechanism to the distributed setting, and avoids a multiplicative factor by carefully selecting an update schedule. Under this mechanism, the mean of an arm is updated only (approximately) $T/\varepsilon$ times, which makes it possible to add Laplacian noise that is of a lower scale (due to the communication-privacy frontier discussed earlier). We demonstrate that using a message-passing algorithm that is inspired by the interval mechanism, we obtain a group regret that has a much smaller than the regular $\mathcal{O}(M)$ overhead on the number of agents, and is additive instead. Recall that in the message-passing protocol described in Chapter 2, agent $m$ creates the message $\mathbf{q}_m(t)$ in any communicating round $t \in [T]$. We define $\mathbf{q}_m(t)$ as follows.

$$\mathbf{q}_m(t) = \langle m, t, \gamma, \widetilde{\boldsymbol{\mu}}_m(t), \mathbf{n}_m(t)\rangle \tag{3.2}$$

Here, $\widetilde{\boldsymbol{\mu}}_m(t) = (\widetilde{\mu}_{m,k}(t))_{k\in[K]}$ is a vector of arm-wise reward means, where $\widetilde{\mu}_{m,k}(t) = \widehat{\mu}_{m,k}(t) + \xi_{m,k}(t)$ denotes a noisy version of the mean reward $\mu_{m,k}(t)$ obtained from arm $k$ by agent $m$ only until time $t$, and $\xi_{m,k}(t)$ is a random noise sample drawn from an appropriately chosen Laplace distribution. Similarly, $\mathbf{n}_m(t) = (n_{m,k}(t))_{k\in[K]}$ is a vector of the number of times each arm $k \in [K]$ has been pulled by the agent $m$. Note that since the identity of the arm pulled is not assumed to be private (we assume that only the reward obtained by the environment is considered private, the alternative setting is discussed in Chapter 6). At any time $t$, the agent interacts with the environment, and updates the sum of its own rewards with the reward obtained from the bandit algorithm for the arm pulled. Then, for a set of $n$ rounds $\mathfrak{T} = \{t_0, ..., t_n\}$ selected according to Definition 3.2 (presented below), broadcasts message $\mathbf{q}_m(t)$ following the communication protocol.

**Definition 3.2** (Broadcast Round Schedule)**.** *The set of communicating rounds $\mathfrak{T} = \{t_0, ..., t_n\}$ are given as follows for any fixed value of $\varepsilon \in (0,1)$, parameter $v \in (1, 1.5)$ with $t_0 = 0$, and $t_i$ as*

$$t_i = \inf_{x \in \mathbb{N}} \left\{ x \geq t_{i-1} + 1 : \sum_{t_{i-1}+1}^{x} \frac{1}{\sqrt{i^v}} \geq \frac{1}{\varepsilon \sqrt{x^v}} \right\}.$$

The intuition for such an update protocol can be understood as follows: we select the next round as the smallest interval that ensures an approximate $\mathcal{O}(\frac{1}{\varepsilon})$ gap between successive communication rounds, such that the overall communication is regulated. This can also be verified by the fact that $\max_{i \in [T]} t_{i+1} - i_t \leq \lceil \frac{1}{\varepsilon} \rceil$. The remainder of the form is chosen to ensure a convergent series. After the broadcast, each agent collates messages received from neighboring agents, and updates its copies of the (privatized) mean rewards $\boldsymbol{\mu}_{\nu \to m}(t)$ and arm pull counts $\mathbf{n}_{\nu \to m}(t)$ for each agent $\nu \in \mathcal{N}_m^+(G_\gamma)$. It then discards stale messages (with life parameter $l = 0$), and returns the updated group means for all arms for the bandit algorithm to use. We describe the complete message-passing protocol in Algorithm 1.

### 3.2.2 Privacy Guarantees

To ensure privacy we utilize the Laplace mechanism as described in the earlier sections. The noise $\zeta_{m,k}(t)$ added to the transmitted mean $\widetilde{\mu}_{m,k}(t)$ is a randomly drawn from a Laplace distribution with scale $n_{m,k}(t)^{v/2-1}$ for a fixed parameter $v \in (1, 1.5)$ that can be tuned with prior knowledge. Adding this noise provides us the following privacy guarantee for $\mathbf{q}_m(t)$.

**Lemma 3.2** (Privacy of Outgoing Messages)**.** *For fixed $v \in (1, 1.5)$, each outgoing message $\mathbf{q}_m(t)$ is $n_{m,k}(t)^{-v/2}$-differentially private w.r.t. the reward sequence of arm $k \in [K]$.*

*Proof.* This follows directly from the fact that the only element that is dependent on the reward sequence of arm $k$ is its noisy mean $\widetilde{\mu}_{m,k}(t)$, which is $n_{m,k}(t)^{-v/2}$- differentially private since we add noise of scale $n_{m,k}(t)^{v/2-1}$ and that the sensitivity of $\widehat{\mu}_{m,k}(t)$ is $n_{m,k}(t)^{-1}$. □

Subsequently applying a $k$-fold adaptive composition argument provides us the final privacy guarantee. Note that our final guarantee provides a *pure* and *approximate* DP bound, although our careful selection of message-level privacy constraints provides a much stronger *approximate* DP guarantee.

**Lemma 3.3** (Privacy Guarantee)**.** *After $t$ trials, agent $m \in [M]$ is $(\varepsilon', \delta)$-differentially private with respect to the reward sequence of any arm observed by any other agent $v \in \mathcal{N}_m^+(G_\gamma)$ communicating via Algorithm 1 with parameter $v \in (1, 1.5)$, where, for $\varepsilon \in (0,1], \delta \in (0,1]$, $\varepsilon'$ satisfies*

$$\varepsilon' \leq \min\left(\varepsilon\frac{(t-d(m,m'))^{1-v/2}}{1-v/2}, 2\varepsilon\zeta(v) + \sqrt{2\varepsilon\zeta(v)\ln(1/\delta)}\right).$$

*Proof.* We can see that for any arm $k$, the estimate $\hat{\mu}_k^m(t)$ is composed of the sum of rewards from all other agents in $\mathcal{N}_\gamma(m)$. However, with respect to the reward sequence of any single agent $m' \in \mathcal{N}_\gamma(m)$, this term only depends on the differential privacy of the outgoing messages $q_{m'}$ obtained by agent $m$ until time $t - d(m,m')$ (since it takes at least $d(m,m')$ trials for a message from $m'$ to reach $m$). Now, we present a lemma that will assist us in bounding the final value of $\varepsilon'$.

**Lemma 3.4.** *Each output message by any agent at time $t'$ is $(\varepsilon', \delta') - DP$, where*

$$\varepsilon' \leq \min\left(\varepsilon\sum_{\tau=1}^{t'}\frac{1}{\sqrt{\tau^v}}, \varepsilon\sum_{\tau=1}^{t'}\frac{\exp\left(\frac{1}{\sqrt{\tau^v}}\right)-1}{\sqrt{\tau^v}} + \sqrt{\varepsilon\sum_{\tau=1}^{t'}\frac{2\log(1/\delta')}{\tau^v}}\right).$$

We can replace the summations in the first term with an integral to get the term $\varepsilon\frac{(t')^{1-v/2}}{1-v/2}$. The second term can be bound similarly by first using the fact that $e^x \leq 1 + 2x$ for $x \in [0,1]$ and then integrating. Finally, applying a $k$-fold composition theorem (Dwork et al., 2010) with $t' = t - d(m,m')$ gives us the result. □

**Remark 3.1** (Privacy Guarantee)**.** It is important to note that if all agents follow the protocol in Algorithm 1, then the privacy guarantee is sufficient, regardless of the algorithm any agent individually uses to make decisions. This is true since for any agent, the the complete sequence of messages it receives from any other agent is differentially private with respect to the origin's reward sequence for any arm and at any instant of the problem. In this setting, one can additionally assume (as is common in federated learning scenarios) that a trusted server exists. In the trusted server setting, it is easy to see that the problem reduces to that of single-agent private decision-making, as the server need only insert one Laplace noise sample to ensure privacy for all agents (after agents communicate $\boldsymbol{\mu}, \mathbf{n}$ to the server).

## 3.3 UCB Exploration and Regret Guarantees

The exploration strategy for each agent is straightforward. At the start of the new round, each agent $m$ computes an upper confidence bound for each arm $k \in [K]$ using the latest information from its neighborhood $\mathcal{N}_m(G_\gamma)$, as follows.

$$Q_{m,k}(t) = \frac{\sum_v n_{v\to m,k}(t) \cdot \mu_{v\to m,k}(t) + n_{m,k}(t) \cdot \widehat{\mu}_{m,k}(t)}{\sum_v n_{v\to m,k}(t) + n_{m,k}(t)} + \sqrt{\frac{2\log(t)}{\sum_v n_{v\to m,k}(t) + n_{m,k}(t)}} \tag{3.3}$$

The summation $\sum_v$ is taken over the neighborhood $\mathcal{N}_m(G_\gamma)$. The agent then selects the action $k$ with the largest $Q_{m,k}(t)$. For the first $K$ rounds, each agent pulls arms 1 to $K$. The algorithm denoted as FedUCB1 is described in Algorithm 2, with the following regret.

**Theorem 3.2** (Regret of FedUCB1)**.** *If all agents $m \in [M]$ each run* FedUCB1 *with the messaging protocol described in Algorithm 1, then the group regret incurred after $T$ trials obeys,*

$$\mathfrak{R}(T) \leq \sum_{k:\Delta_k>0} \bar{\chi}(G_\gamma)\left(\frac{8\log(T)}{\Delta_k}\right) + \left(\sum_{k:\Delta_k>0} \Delta_k\right)\left((M\gamma+2) + M\left(\frac{1}{\varepsilon} + \zeta(1.5)\right)\right).$$

*Here, $\bar{\chi}$ is the clique covering number, and $\zeta$ is the Riemann zeta function.*

*Proof.* The proof approach is to partition the entire graph $G$ into subgraphs that are individually analysed, and the analysis for each subgraph largely follows the analysis structure of UCB1 (Auer et al., 2002a), with several new arguments that handle the noise introduced by private messages and delays in communication. Let a maximal clique covering of $G_\gamma$ be given by $\mathfrak{C}$. We begin by decomposing the group regret along the clique covering $\mathfrak{C}$.

$$\mathfrak{R}(T) = \sum_{m=1}^{M} \mathfrak{R}_m(T) = \sum_{\mathcal{C}\in\mathfrak{C}} \sum_{m\in\mathcal{C}} \sum_{k:\Delta_k>0} \Delta_k \cdot \mathbb{E}[n_{m,k}(T)] \tag{3.4}$$

$$= \sum_{\mathcal{C}\in\mathfrak{C}} \sum_{k=1}^{K} \Delta_k \cdot \left(\sum_{m\in\mathcal{C}} \sum_{t=1}^{T} \mathbb{P}\left(a_m(t) = k\right)\right). \tag{3.5}$$

Consider now the cumulative regret $\mathfrak{R}_\mathcal{C}(T)$ within the clique $\mathcal{C} \in \mathfrak{C}$. For some round $T^k_\mathcal{C}$, assume that each agent has pulled arm $k$ for $\eta^k_m$ trials, where $\eta^k_\mathcal{C} = \sum_{m\in\mathcal{C}} \eta^k_m$. Then,

$$\mathfrak{R}_\mathcal{C}(T) \leq \sum_{k=1}^{K} \Delta_k \left(\eta^k_\mathcal{C} + \sum_{m\in\mathcal{C}} \sum_{t=T^k_\mathcal{C}}^{T} \mathbb{P}\left(A_{m,t} = k, N^\mathcal{C}_k(t) \geq \eta^k_\mathcal{C}\right)\right). \tag{3.6}$$

Here $N_k^{\mathcal{C}}(t)$ denotes the number of times arm $k$ has been pulled by any agent in $\mathcal{C}$. We now examine the probability of agent $m \in \mathcal{C}$ pulling arm $k$. First note that the empirical mean of any arm can be given by the latest messages accumulated by agent $m$ until that time. This can be given by the following, for any arm $k \in [K]$.

$$\hat{\mu}_{m,k}(t-1) = \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \left( \frac{\sum_{v \in \mathcal{N}_m^+(G_\gamma)} \sum_{\tau=1}^{t-1} r_v(\tau) \cdot \mathbb{1}\{a_v(\tau) = k\}}{n_{m'}^k(t - d(m,m'))} + Y_{m'}^k \right)$$

Here, $Y_{m'}^k \sim \mathcal{L}\left(n_{m'}^k(t-d(m,m'))^{v/2-1}\right)$. For convenience, let's denote the noise-free mean $\hat{\mu}_k^m(t-1) - \sum_{m' \in \mathcal{N}(m)} Y_{m'}^k$ as $Z_k^m(t-1)$, and $N_k^m(t) = n_m^k(t) + \sum_{m' \in \mathcal{N}(m)} n_{m'}^k(t - d(m,m'))$. Note that an arm is pulled when one of three events occurs:

$$\text{Event (A): } Z_k^m(t-1) \leq \mu_* - \sigma\sqrt{\frac{2\ln t}{N_*^m(t)}} - \sum_{m' \in \mathcal{N}(m)} Y_{m'}^*$$

$$\text{Event (B): } Z_k^m(t-1) \geq \mu_k + \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} + \sum_{m' \in \mathcal{N}(m)} Y_{m'}^k$$

$$\text{Event (C): } \mu_* \leq \mu_k + 2\sigma\sqrt{\frac{2\ln t}{N_k^m(t)}}$$

We will first analyse events $(A)$ and $(B)$. We know from Dwork & Roth (2014) that for any $N$ random variables $Y_i \sim \mathcal{L}(b_i)$,

$$\Pr\left(\left|\sum_i Y_i\right| \geq \ln\left(\frac{1}{\omega}\right)\sqrt{8\sum_i b_i^2}\right) \leq \omega \tag{3.7}$$

For some $\omega \in (0,1)$, let $h_{k,m}(t) = \ln\left(\frac{1}{\omega}\right)\sqrt{8\sum_{m'}(n_{m'}^k(t-d(m,m'))^{v-2}}$. Let us examine the probability of event (A) occuring.

$$\Pr(B) = \Pr\left(Z_k^m(t-1) \geq \mu_k + \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} + \sum_{m' \in \mathcal{N}(m)} Y_{m'}^k\right) \tag{3.8}$$

$$= \Pr(\hat{\mu}_k^m(t-1) - Z_k^m(t-1) \geq h_{k,m}(t) \ \ \& \ \ Z_k^m(t-1) \geq \mu_k + \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - h_{k,m}(t)) \tag{3.9}$$

$$\leq \Pr\left(\hat{\mu}_k^m(t-1) - Z_k^m(t-1) \geq h_{k,m}(t)\right) + \Pr\left(Z_k^m(t-1) \geq \mu_k + \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - h_{k,m}(t)\right) \tag{3.10}$$

$$\leq \omega + \Pr\left( Z_k^m(t-1) \geq \mu_k + \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - h_{k,m}(t) \right) \tag{3.11}$$

$$\leq \omega + \exp\left( -2N_k^m(t) \left( \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - h_{k,m}(t) \right)^2 \right) \tag{3.12}$$

$$= t^{-3.5} + \exp\left( -2N_k^m(t) \left( \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - 3.5\ln t\sqrt{8\sum_{m'}(n_{m'}^k(t-d(m,m'))^{v-2}} \right)^2 \right) \tag{3.13}$$

$$\leq t^{-3.5} + \exp\left( -2N_k^m(t) \left( \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - 3.5\ln t\sqrt{8M}\left(\varepsilon^{-1}-\gamma\right)^{v/2-1} \right)^2 \right) \tag{3.14}$$

$$\leq 2t^{-3.5} \tag{3.15}$$

Here, we use Hoeffding's Inequality in Equation (3.12), and the fact that $v \in (1, 1.5)$ in Equation (3.14) and that $n_m^k(t) \geq \varepsilon^{-1}$, and choose $\sum_{m'} n_{m'}^k(t - d(m,m'))$ such that

$$\exp\left( -2N_k^m(t) \left( \sigma\sqrt{\frac{2\ln t}{N_k^m(t)}} - 3.5\ln t\sqrt{8M}\left(\varepsilon^{-1}-\gamma\right)^{v/2-1} \right)^2 \right) \leq t^{-3.5}.$$

We see that as long as $N_k^{\mathcal{C}}(t) \geq \frac{(\varepsilon^{-1}-\gamma)^{v/2-1}(2\sqrt{2}\sigma-\sqrt{3.5})}{\sqrt{24M\ln t}} + \frac{|\mathcal{C}|}{\varepsilon}$, this is true, following the fact that $\gamma < 1/\varepsilon$. We can repeat the same process for the event (A). Finally, let us analyse event (C). For (C) to be true, we must have the following to be true.

$$N_k^m(t) < \frac{8\sigma^2\ln t}{\Delta_k^2}. \tag{3.16}$$

Since $N_k^m(t) \geq N_k^{\mathcal{C}} - M\gamma$, we see that this event does not happen for any agent $m \in \mathcal{C}$ if we set

$$\eta_k^{\mathcal{C}} = \left\lceil \max\left\{ \frac{8\sigma^2\ln T}{\Delta_k^2} + M\gamma, \frac{(\varepsilon^{-1}-\gamma)^{v/2-1}(2\sqrt{2}\sigma-\sqrt{3.5})}{\sqrt{24M\ln t}} + \frac{|\mathcal{C}|}{\varepsilon} \right\} \right\rceil.$$

Next, we should notice that the second term decreases as $t$ increases. We therefore, can decompose the regret of the entire clique as follows.

$$R_{\mathcal{C}}(T) \leq \sum_{k=1}^{K} \Delta_k \left( \eta_{\mathcal{C}}^k + \frac{|\mathcal{C}|}{\varepsilon} + \sum_{m\in\mathcal{C}} \sum_{t=T_{\mathcal{C}}^k}^{T} \mathbb{P}\left( A_{m,t} = k, N_k^{\mathcal{C}}(t) \geq \eta_{\mathcal{C}}^k \right) \right) \tag{3.17}$$

$$\leq \sum_{k:\Delta_k>0} \Delta_k \left( \frac{8\sigma^2 \ln T}{\Delta_k^2} + M\gamma + 2 + \frac{|\mathcal{C}|}{\varepsilon} + \sum_{m\in\mathcal{C}} \sum_{t=1}^{T} 4t^{-1.5} \right) \tag{3.18}$$

$$\leq \sum_{k:\Delta_k>0} \Delta_k \left( \frac{8\sigma^2 \ln T}{\Delta_k^2} + M\gamma + 2 + \frac{|\mathcal{C}|}{\varepsilon} + \sum_{m\in\mathcal{C}} \sum_{t=1}^{T} 4t^{-1.5} \right) \tag{3.19}$$

$$\leq \sum_{k:\Delta_k>0} \Delta_k \left( \frac{8\sigma^2 \ln T}{\Delta_k^2} + M\gamma + 2 + |\mathcal{C}| \left( \frac{1}{\varepsilon} + \zeta(1.5) \right) \right) \tag{3.20}$$

Summing over all cliques $\mathcal{C} \in \mathfrak{C}$, we get

$$R_G(T) \leq \sum_{C\in\mathcal{C}_\gamma} \sum_{k:\Delta_k>0} \Delta_k \left( \frac{8\sigma^2 \ln T}{\Delta_k^2} + M\gamma + 2 + |\mathcal{C}| \left( \frac{1}{\varepsilon} + \zeta(1.5) \right) \right). \tag{3.21}$$

Choosing $\mathfrak{C}$ to be the minimal clique partition of $G_\gamma$,we obtain the final form of the bound.

□

We now present a few remarks regarding the presented algorithm and analysis.

## 3.4 Discussion

The obtained regret has two components - the dependence on the communication graph $G$ and the privacy budget $\varepsilon$. Since the additional regret due to privacy is additive (and independent of the network $G$), we discuss the two dependencies separately, starting with the network dependence first.

As discussed earlier, the constant in the leading $\log(T)$ term scales as $\bar{\chi}(G_\gamma)$, similar to the non-private case. In Chapter 4 we will see that this leading constant can be improved to $\alpha(G_\gamma)$, i.e., the independence number of the power graph, using more sophisticated techniques that either involve more communication or mimicking other agents. We remark that similar algorithms can also be derived in the differentially-private setting without many changes to the privacy analysis.

We now come to the dependence on the privacy budget $\varepsilon$. Observe that the additional regret due to private message-passing is a fixed term of $\frac{M}{\varepsilon} + M \cdot \zeta(1.5)$. We remark that the second term is an artifact of our analysis: the varying privacy budget per message is combined by an advanced composition theorem (Dwork et al., 2010) to provide the final privacy guarantee, which leads to an additional suboptimality. This term can potentially be improved at some cost to the leading term in the regret, by reducing the number of

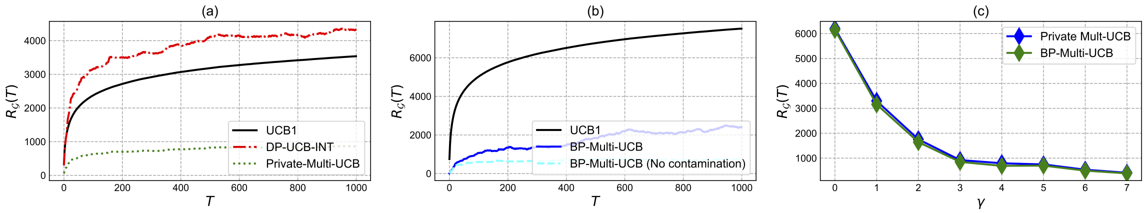


Figure 3-1: Experimental comparisons on random graphs. Each figure is constructed by averaging over 100 trials.

outgoing messages further.

Next, we conjecture that the first term of $\frac{M}{\varepsilon}$ can be reduced only to $\frac{\max_{v \in \mathcal{V}} d_\gamma(v)}{\varepsilon}$, where $\max_{v \in \mathcal{V}} d_\gamma(v)$ denotes the maximum degree of any agent in $G_\gamma$. While we cannot prove this rigorously, we provide some intuition. Consider any arbitrary agent $v$ with degree $d$. If this agent utilizes observations from all of their neighbors, the confidence bound will inevitably be stretched by an amount $\frac{d}{\varepsilon}$ by the fact that additional observations are private. One way that this can potentially be avoided is to allow a majority of (perhaps poorly connected agents) mimic the most well-connected agents. Even if all agents mimic only one "best-connected" agent $v^\star$, we will incur an additional regret of $\frac{d_\gamma(v^\star)}{\varepsilon}$, suggesting that the conjectured rate cannot be improved on its dependence on $G$. We conjecture that this is the optimal constant for any algorithm that utilizes all of its communication, since diregarding messages will lead to an increase in the leading $\log(T)$ term.

Coming to the dependence on $\varepsilon$, the following claim from Shariff & Sheffet (2018) regarding single-agent lower bounds in the joint differentially-private setting can provide some insight.

**Claim 3.1** (Claim 14 of Shariff & Sheffet (2018), Private Regret Lower Bound)**.** *The expected regret of any $\varepsilon-$DP algorithm for the MAB is $\Omega(K \log(T)/\varepsilon)$. Combined with the standard (non-private) bound of $\Omega(\sum_{k:\Delta_k > 0} \log(T)/\Delta_k)$, we have that the minimum single-agent regret obeys $\mathfrak{R}(T) = \Omega(\max\{K \log(T)/\varepsilon, \sum_{k:\Delta_k>0} \log(T)/\Delta_k\})$.*

The above claim clearly suggests that even with a time-varying privacy budget, one must incur a regret that scales as $\frac{1}{\varepsilon}$ for $T$ observations. Given that the federated setting involves multiple agents, the conjectured bound follows.

We now describe experimental comparisons.

## 3.5 Experiments

For our experimental setup, we consider rewards drawn from randomly initialized Bernoulli distributions, with $K = 5$ as our default operating environment. We initialize the connectivity graph $G$ as a sample from a random Erdos-Renyi (Bollobás & Riordan, 2003) family with edge probability $p = 0.1$, and set the communication parameter $\gamma = d_\star(G)/2$. We describe the experiments and associated benchmark algorithms individually.

We compare the group regret $\mathfrak{R}(T)$ of $M = 200$ agents over 100 randomly initialized trials of the above setup. The benchmark algorithms we compare with are (a) single-agent UCB1 (Auer et al., 2002a) running individually on each agent, and (b) DP-UCB-INT of (Tossou & Dimitrakakis, 2016) running individually on each agent, under the exact setup for $\varepsilon$ and $\delta$ as described in (Tossou & Dimitrakakis, 2016). We choose this benchmark as it is the state-of-the-art in the single-agent stochastic bandit case. The results of this experiment are plotted in Figure 4-1(a). We observe that the performance of our algorithm is significantly better than both private benchmarks, however, it incurs higher regret (as expected) than the non-private version.

**Testing the effect of $\gamma$.** To understand the effect of $\gamma$ on the obtained regret, we repeated the same experiment ($M = 200$, 100 trials of randomly generated Erdos-Renyi graphs with $p = 0.1$) with our two algorithms and compared their obtained group regret at $T = 1000$ trials. We observe a sharp decline as $\gamma$ increases from 1 to $d_\star(G)$, and matches the optimal group regret at $\gamma = d_\star(G)$, as hypothesized by our regret bound. The results of this experiment are summarized in Figure 4-1(c).

## 3.6 Omitted Proofs

### 3.6.1 Proof of Lemma 3.4

*Proof.* Consider the set of communicating rounds in Definition 3.2 be given by $\mathfrak{T}(\varepsilon)$ for any value of $\varepsilon$, i.e., $\mathfrak{T}(\varepsilon)[: t] = \{\frac{1}{\varepsilon}, \frac{2}{\varepsilon}, \frac{3}{\varepsilon}, ..., t\}$ and consider for any agent $m$ and arm $k$ the shorthand $p(t) = n_{m,k}(t)$. Now, observe that by a $k-$fold adaptive composition (Dwork & Roth, 2014), we have that the algorithm is $\varepsilon', \delta'$-DP for any $\delta' \in (0, 1]$ such that for any agent

$$\begin{aligned}\varepsilon' \le \min\{B_1, B_2\}, \text{ where, } B_1 &= \sum_{\tau \in \mathfrak{T}(\varepsilon)[:p(t)]} \frac{1}{\sqrt{\tau^v}} \le \sum_{\tau \in \mathfrak{T}(\varepsilon)[:t]} \frac{1}{\sqrt{\tau^v}} \le \varepsilon \sum_{\tau=1}^{t} \frac{1}{\sqrt{\tau^v}}, \text{ and} \\ B_2 &= \sum_{\tau \in \mathfrak{T}(\varepsilon)[:p(t)]} \frac{\exp\left(\frac{1}{\sqrt{\tau^v}}\right) - 1}{\sqrt{\tau^v}} + \sqrt{\sum_{\tau \in \mathfrak{T}(\varepsilon)[:p(t)]} \frac{2\log(1/\delta')}{\tau^v}} \\ &\le \varepsilon \sum_{\tau=1}^{t} \frac{\exp\left(\frac{1}{\sqrt{\tau^v}}\right) - 1}{\sqrt{\tau^v}} + \sqrt{\varepsilon \sum_{\tau=1}^{t} \frac{2\log(1/\delta')}{\tau^v}}.\end{aligned}$$

This concludes the proof. □

## 3.7 Algorithm Pseudocode

**Algorithm 1** Private FedUCB Message-Passing Protocol

```
Input: Agent $m \in [M]$, Iteration $t \in [T]$, series $W = (w_0, w_1, ...w_T)$, such that $W(i) = w_i$, series counter $i_k^m$ for each $k \in [K]$; $i_k^m = w_0$ if $t = 0$ $\forall k$, set of existing messages $Q_m(t-1)$, $Q_m(t) = \phi$ if $t = 0$, privacy constant $v \in (1, 1.5)$, existing reward sums $s_{m,k}(t)$ $\forall k \in [K]$ such that $s_{m,k}(0) = 0 \forall k$.
Obtain $X_{m,t}, A_{m,t}$ from bandit algorithm.
Set $s_{m,k}(t) = s_{m,k}(t-1) + X_{m,t}$ for $k = A_{m,t}$.
Set $n_k^m(t) = n_k^m(t-1) + 1$ for $k = A_{m,t}$.
for Arm $k$ in $[K]$ do
    if $n_k^m(t) = W(i_k^m)$ then
        Set $\hat{v}_m^k(t) = s_{m,k}(t)/n_k^m(t) + \mathcal{L}(n_k^m(t)^{v/2-1})$.
        Set $i_k^m = i_k^m + 1$.
    else
        Set $\hat{v}_m^k(t) = v_m^k(t-1)$.
    end if
end for
Set $q_m(t) = \langle m, t, \hat{\mathbf{v}}_m(t), \boldsymbol{n}_m(t) \rangle$.
Set $Q_m(t) = Q_m(t-1) \cup \{q_m(t)\}$.
for Each neighbor $m'$ in $\mathcal{N}_1(m)$ do
    SendMessages$(m, m', Q_m(t))$.
end for
for Each neighbor $m'$ in $\mathcal{N}_1(m)$ do
    $Q'$ =ReceiveMessages$(m', m)$
    $Q_m(t) = Q_m(t) \cup Q'$.
end for
Set $N_k^m(t) = n_k^m(t), \hat{\mu}_k^m(t) = s_{m,k}(t)$ $\forall k \in [K]$.
for $q' = \langle m', t', x'_1, ..., x'_K, a'_1, ..., a'_K \rangle \in Q_m(t)$ do
    if IsLatestMessage$(q')$ then
        for Arm $k \in [K]$ do
            Set $N_k^m(t) = N_k^m(t) + a'_k$.
            Set $\hat{\mu}_k^m(t) = \hat{\mu}_k^m(t) + a'_k \cdot x'_k$.
        end for
    end if
end for
for Arm $k \in [K]$ do
    $\hat{\mu}_k^m(t) = \hat{\mu}_k^m(t)/N_k^m(t)$.
end for
```

**Algorithm 2** Private FedUCB

```
Input: Agent $m \in [M]$, trial $t \in [T]$, arms $k \in [K]$, mean $\hat{\mu}_k^m(t)$ and count $n_k^m(t)$ estimates
    for each arm $k \in [K]$, from Algorithm 1.
if $t \leq K\lceil 1/\varepsilon \rceil$ then
    $A_{m,t} = t \mod K$.
else
    for Arm $k \in [K]$ do
        $\mathrm{UCB}_k^m(t) = \sqrt{\frac{2\ln t}{N_k^m(t)}}$.
    end for
    $A_{m,t} = \arg\max_{k \in [K]} \left\{ \hat{\mu}_k^m(t) + \mathrm{UCB}_k^{(m)}(t) \right\}$.
end if
$X_{m,t} = \textsc{Pull}(A_{m,t})$.
return $A_{m,t}, X_{m,t}$.
```

# Chapter 4

# Federated Multi-Armed Bandits with Heavy-Tailed Rewards

Prior work on the cooperative or federated bandit learning has largely been on reward distributions that are sub-Gaussian (Landgren et al., 2016a,b). While this is certainly applicable in several domains, increasing evidence suggests that assumptions of sub-Gaussianity may not hold for numerous applications specific to *federated* decision-making, in problems such as distributed load estimation of internet traffic (Hernandez-Campos et al., 2004; Crovella et al., 1998), multi-agent modeling of supply chain networks (Thadakamaila et al., 2004), modeling information cascades in economic multi-agent models (De Vany et al., 1999; Konovalov, 2010) and, among others, numerous problems in distributed modeling for social science (Barabasi, 2005; Eom & Jo, 2015). In this chapter, we therefore consider reward distributions that admit heavier tails, made precise as follows.

**Definition 4.1** (Heavy-Tailed Random Variables). *A random variable $X$ is light-tailed if it admits a finite moment generating function, i.e. there exists $u_0 > 0$ such that $\forall |u| \leq u_0$,*

$$M_X(u) \triangleq \mathbb{E}[\exp(uX)] < \infty.$$

*Otherwise $X$ is heavy-tailed. We define a random variable $X$ to be $\varepsilon$-heavy tailed if all moments of $X$ of order $> 1+\varepsilon$ are infinite, i.e., $\mathbb{E}[|X|^{1+\alpha}] = \infty$ for all $\alpha > \varepsilon$.*

Extending robust estimation to the cooperative case is not straightforward. Robust estimators do not naturally lend themselves to consensus-based algorithms (Landgren et al., 2016a,b) that have been widely utilized in cooperative decision-making, since require sending messages that are at least $\mathcal{O}(\log T)$ bits per round, which is infeasible in practice. We demonstrate that using a message-passing protocol with robust estimators allow for near-optimal rates for the problem. Specifically, we first present an algorithm that extends the message-passing upper-confidence bound approach from Chapter 2 to handle heavy-tailed reward distributions with the help of robust mean estimators. This algorithm handles the combined issue of delayed observations and heavy-tailed noise simultaneously, and additionally we present a new online algorithm to update the robust (trimmed) mean with varying confidence levels that reduces the runtime complexity in time $\mathcal{O}(\log T)$ per episode (improving from $\mathcal{O}(T)$ time complexity).

## 4.1 Problem Setup and Univariate Robust Estimation

Recall that in the decentralized federated protocol, agents $m \in \mathcal{M}$ communicate via messages $\mathbf{q}_v(t) = \langle v, t, A_{v,t}, X_{v,t} \rangle$. This message is first sent to its neighbors in $G$, and it is subsequently forwarded by any agent that receives it until time $t + \gamma$, after which it is discarded. $0 \leq \gamma \leq \text{diameter}(G)$ is therefore the communication *density*, where lower values of $\gamma$ imply less communication in the network. Let $\mathcal{Q}_v(t)$ denote the set of incoming messages received by agent $v$ at instant $t$. During any trial, the agent first pulls an arm, and creates the message $\mathbf{q}_v(t)$. It then processes all messages in $\mathcal{Q}_v(t)$, and updates its beliefs as per any bandit algorithm. Finally, it discards all messages older than $t - \gamma$ and forwards all remaining messages in $\mathcal{Q}_v(t) \cup \{\mathbf{q}_v(t)\}$ to all its neighbors in $G$. Now, before presenting the algorithms, we briefly discuss our setup for univariate robust mean estimation.

### 4.1.1 Univariate Robust Estimation

. It has long been known that traditional "sub-Gaussian" error rates are unachievable for heavy-tailed distributions where $\varepsilon < 1$, i.e., distributions with infinite variance (Lugosi & Mendelson, 2019). Furthermore, Devroye et al. (2016) demonstrate that even when the variance is finite, it is impossible to construct a single estimator that is sub-Gaussian for any non-trivial range of confidence parameters $\delta$, i.e., we require a different estimator for each confidence level $\delta$. This result especially is an issue for bandit algorithms, as we require adjusting our confidence level after each new trial in order to effectively balance exploration and exploitation. Later in this chapter we will demonstrate that we can avoid $\mathcal{O}(T)$ updates by carefully selecting a confidence level schedule, reducing the rate of updating to $\mathcal{O}(\log T)$. We now present an overview of the estimators we use and the error rates they achieve.

The simplest univariate robust estimator is the trimmed mean, that rejects outlying samples based on an upper bound on the largest finite moment.

**Definition 4.2** (Trimmed Mean)**.** *Consider $n$ copies $X_1, ..., X_n$ of a heavy-tailed random variable $X$ such that $\mathbb{E}[X] = \mu, \mathbb{E}[|X|^{1+\varepsilon}] \leq u$ for some $\varepsilon \in (0,1]$. The online trimmed mean, for some $\delta \in (0,1)$ is defined as*

$$\hat{\mu}_O = \frac{1}{n}\sum_{\tau=1}^{n} X_\tau \cdot \mathbf{1}\left\{|X_\tau| \leq \left(\frac{u \cdot \tau}{\log(1/\delta)}\right)^{\frac{1}{1+\varepsilon}}\right\}.$$

This estimator is known to achieve near-sub-Gaussian rates when the variance exists.

**Lemma 4.1** (Trimmed Mean Error, Bubeck et al. (2013))**.** *Consider $n$ copies $X_1, ..., X_n$ of a heavy-tailed random variable $X$ such that $\mathbb{E}[X] = \mu, \mathbb{E}[|X|^{1+\varepsilon}] \leq u$ for some $\varepsilon \in (0,1]$. The online trimmed mean satisfies, with probability at least $1-\delta$,*

$$|\hat{\mu}_O - \mu| \leq 4(u)^{\frac{1}{1+\varepsilon}}\left(\frac{\log(1/\delta)}{n}\right)^{\frac{\varepsilon}{1+\varepsilon}}.$$

Another estimator we will utilize is the median-of-means estimator.

**Lemma 4.2** (Median-of-Means Estimator)**.** *Let $X_1, ..., X_n$ be copies of a heavy-tailed random variable $X$ such that $\mathbb{E}[X] = \mu$ and $\mathbb{E}[|X-\mu|^{1+\varepsilon}] \leq v$ for some $\varepsilon \in (0,1]$. Let $k = \lfloor 8\log(e^{1/8}/\delta) \wedge n/2 \rfloor$ and $N = \lfloor n/k \rfloor$. Let the group-wise means be given by the following.*

$$\hat{\mu}_1 = \frac{1}{N}\sum_{i=1}^{N} X_i, ..., \hat{\mu}_k = \frac{1}{N}\sum_{i=(k-1)N+1}^{kN} X_i.$$

*Let $\hat{\mu}_M$ denote the median of $\hat{\mu}_1, ..., \hat{\mu}_k$. Then with probability at least $1-\delta$,*

$$|\hat{\mu}_M - \mu| \leq (12v)^{\frac{1}{1+\varepsilon}}\left(\frac{16\log(e^{1/8}/\delta)}{n}\right)^{\frac{\varepsilon}{1+\varepsilon}}.$$

Alternative robust mean estimators exist, such as Catoni's estimator (Catoni, 2012). In the analysis, we assume that a mean estimator exists that achieves the following rate.

**Assumption 4.1** (Estimator Rate Assumption)**.** *Let $\mathcal{S} = \{X_1, ..., X_n\}$ be $n$ samples of an $\varepsilon$-heavy tailed random variable, where $\varepsilon \in (0, 1]$, and $\mathbb{E}[X] = \mu$. For positive constants $c, \rho$ suppose that there exists a robust estimator $\hat{\mu}(\mathcal{S}, \delta)$ such that, with probability at least $1 - \delta$,*

$$|\hat{\mu}(\mathcal{S}, \delta) - \mu| \leq 2\rho^{\frac{1}{1+\varepsilon}} \left( \frac{c \log(1/\delta)}{n} \right)^{\frac{\varepsilon}{1+\varepsilon}} .$$

### 4.1.2 Online Estimation of the Trimmed Mean

The next sections can potentially work with any robust mean estimator, however, we select the trimmed mean due to its simplicity and fast time complexity of $\mathcal{O}(T)$ per round, and demonstrate now that it can be improved to $\mathcal{O}(\log T)$ per round.

The trimmed mean estimator requires selecting a sample $X_i$ at time $t$ only if $|X_i| \leq (2ui \log(t))^{1/(1+\varepsilon)}$ (Definition 4.2). This implies that the $i^{th}$ reward sample an agent has will be selected at the smallest time $t$ such that $(|X_i|^{1+\varepsilon}/(i)) \leq 2u \log(t)$. When $T$ is known, we can utilize a binary search tree to make an update to the robust mean $\mathcal{O}(\log(t))$ instead of $\mathcal{O}(t)$ at time $t$. We outline this procedure in Algorithm 6. We assume that for any $t$, a new set of observations $O_t$ is available, which it incorporates into the robust mean with $\mathcal{O}(\log(t))$ per sample (instead of typically recomputing the mean for each $t$). The complexity stems from the binary search, assuming the dictionary lookup is $\mathcal{O}(1)$.

When $T$ is unknown, one can simply run a "doubling" routine with $T = 1, 2, 4, 8, ...$, reconstructing the binary tree every time $T$ is doubled. Observe that the doubling at any time $t$ will take $\mathcal{O}(t)$ steps to reconstruct the tree, which leads to an amortized per-round cost of $\mathcal{O}(\log^2(T))$ instead.

## 4.2 Fully-Decentralized Algorithm

In this section we present our first algorithm, a basic extension of the algorithm discussed in Chapter 2 to heavy tailed rewards. Recall that in the fully-decentralized setting, each agent acts independently, i.e., there is no centralized controller that dictates actions. In this setting, each agent $v$ maintains a set $\mathcal{S}_{v,k}(t)$ of rewards obtained from arm $k$, which it updates at each trial from its own pulls and incoming messages. Any agent $v$ only selects observations that originate within its local $\gamma$-neighborhood $\mathcal{N}_v^+(G_\gamma)$, and does not utilize observations from agents outside this neighborhood. It then computes the robust mean of

$\mathcal{S}_{v,k}(t)$ via the estimator $\hat{\mu}(\mathcal{S}_{v,k}(t), \delta)$. Using Assumption 4.1, it then estimates a UCB for each arm mean, and selects the arm with the largest UCB (Algorithm 3). At any instant, the $Q-$values (UCB) computed by the agent can be written, for any arm $k$ as (following Assumption 4.1):

$$Q_{v,k}(t) = \hat{\mu}\left(\mathcal{S}_{v,k}(t), \frac{1}{t^2}\right) + 2\rho^{\frac{1}{1+\varepsilon}}\left(\frac{2c\log(t)}{|\mathcal{S}_{v,k}(t)|}\right)^{\frac{\varepsilon}{1+\varepsilon}}. \tag{4.1}$$

The challenges in bounding the regret of this algorithm are to account for the varying information present across the network (due to delays in communication), along with the robust estimation of the mean reward from each arm. We present the regret bound below.

**Theorem 4.1.** *There exists an absolute constant $C > 0$ independent of $T, K, M, \gamma$ and $G$ such that the group regret for Algorithm 3 when run with parameter $\gamma$ and mean estimator $\hat{\mu}(n, \delta)$ satisfies:*

$$\mathfrak{R}(T) \leq C \cdot \bar{\chi}\left(G_\gamma\right)\left(\sum_{k:\Delta_k>0} \frac{1}{2\Delta_k^{1/\varepsilon}}\right)\log T + (3+\gamma)\cdot M \cdot \left(\sum_{k:\Delta_k>0} \Delta_k\right).$$

*Here $\bar{\chi}(\cdot)$ refers to the clique covering number.*

*Proof.* Let a clique covering of $G_\gamma$ be given by $\boldsymbol{C}_\gamma$. We first bound the regret in each clique $\mathcal{C}$ within the clique covering $\boldsymbol{C}_\gamma$ of $G_\gamma$. This is done by noticing that the upper confidence bound for any arm at a selected $t$ deviates by a constant amount between agents based on the number of times each agent has pulled an arm. By bounding this deviation, we obtain a relationship between the confidence bound of each arm for each agent within the clique $\mathcal{C}$. Next, we bound the probability of pulling a suboptimal arm within the clique $\mathcal{C}$ using the previous result. Summing over the clique cover $\boldsymbol{C}_\gamma$ delivers the final form of the result. We begin by decomposing the group regret.

$$\mathfrak{R}(T) = \sum_{m=1}^{M} \mathfrak{R}_m(T) \leq \sum_{\mathcal{C}\in\boldsymbol{C}_\gamma}\sum_{m\in\mathcal{C}}\sum_{k=1}^{K} \Delta_k \mathbb{E}[n_{m,k}(T)] \tag{4.2}$$

$$= \sum_{\mathcal{C}\in\boldsymbol{C}_\gamma}\sum_{k=1}^{K} \Delta_k \left(\sum_{m\in\mathcal{C}}\sum_{t=1}^{T} \mathbb{P}\left(a_m(t) = k\right)\right) \tag{4.3}$$

Consider the cumulative regret $R_{\mathcal{C}}(T)$ within the clique $\mathcal{C}$. For some time $T^k_{\mathcal{C}}$, assume that each agent has pulled arm $k$ for $\eta^k_m$ trials, where $\eta^k_{\mathcal{C}} = \sum_{m\in\mathcal{C}} \eta^k_m$. Then,

$$R_{\mathcal{C}}(T) \leq \sum_{k=1}^{K} \Delta_k \left( \eta^k_{\mathcal{C}} + \sum_{m\in\mathcal{C}} \sum_{t=T^k_{\mathcal{C}}}^{T} \mathbb{P}\left( a_m(t) = k, N^{\mathcal{C}}_k(t) \geq \eta^k_{\mathcal{C}} \right) \right). \tag{4.4}$$

Here $N_{\mathcal{C},k}(t)$ denotes the number of times arm $k$ has been pulled by any agent in $\mathcal{C}$. We now examine the probability of agent $m \in \mathcal{C}$ pulling arm $k$. Note that an arm is pulled when one of three events occurs:

$$\text{Event (A): } \hat{\mu}_{m,*}(t-1) \leq \mu_* - v^{\frac{1}{1+\varepsilon}} \left( \frac{2c\log(t)}{|\mathcal{S}_{m,*}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.5}$$

$$\text{Event (B): } \hat{\mu}_{m,k}(t-1) \geq \mu_k + v^{\frac{1}{1+\varepsilon}} \left( \frac{2c\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.6}$$

$$\text{Event (C): } \mu_* \leq \mu_k + 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.7}$$

Now, let us examine the occurence of event $(C)$:

$$\Delta_k \leq 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c\log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.8}$$

$$\implies |\mathcal{S}_{m,k}(t)| \leq 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} \tag{4.9}$$

We know that for the subgraph $\mathcal{C}$, Lemma 2.1 holds for each $m \in \mathcal{C}$ with delay $\gamma$. Hence, $N_{m,k}(t) \geq N_{\mathcal{C},k}(t) - (|\mathcal{C}|-1)(1-\gamma)$ for all $t$. Therefore, if we set

$$\eta^k_{\mathcal{C}} = \left\lceil 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + (|\mathcal{C}|-1)(\gamma-1) \right\rceil,$$

we know that event $(C)$ will not occur. Additionally, using the union bound over $N_{m,*}(t)$ and $N_{m,k}(t)$, and Assumption 4.1, we have:

$$\mathbb{P}(\text{Event (A) or (B) occurs}) \leq 2\sum_{s=1}^{t} \frac{1}{s^4} \leq \frac{2}{t^3}. \tag{4.10}$$

Combining all probabilities, and inserting in Equation (4.4), we have,

$$\mathfrak{R}_{\mathcal{C}}(T) \leq \sum_{k=1}^{K} \Delta_k \left( \eta_{\mathcal{C}}^k + \sum_{m \in \mathcal{C}} \sum_{t=T_{\mathcal{C}}^k}^{T} \mathbb{P}\left( a_m(t) = k, N_k^{\mathcal{C}}(t) \geq \eta_{\mathcal{C}}^k \right) \right) \tag{4.11}$$

$$\leq \sum_{k=1}^{K} \Delta_k \left( \left\lceil 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + (|\mathcal{C}|-1)(\gamma-1) \right\rceil + \sum_{m \in \mathcal{C}} \sum_{t=1}^{T} \frac{2}{t^3} \right) \tag{4.12}$$

$$\leq \sum_{k=1}^{K} \Delta_k \left( \left\lceil 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + (|\mathcal{C}|-1)(\gamma-1) \right\rceil + 4|\mathcal{C}| \right) \tag{4.13}$$

$$\leq \sum_{k=1}^{K} \Delta_k \left( 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + (|\mathcal{C}|-1)(\gamma-1) + 1 + 4|\mathcal{C}| \right). \tag{4.14}$$

We can now substitute this result in the total regret.

$$\mathfrak{R}(T) \leq \sum_{\mathcal{C} \in \boldsymbol{\mathcal{C}}} \mathfrak{R}_{\mathcal{C}}(T) \tag{4.15}$$

$$\leq \sum_{\mathcal{C} \in \boldsymbol{\mathcal{C}}} \sum_{k=1}^{K} \Delta_k \left( 2cv^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + (|\mathcal{C}|-1)(\gamma-1) + 1 + 4|\mathcal{C}| \right) \tag{4.16}$$

$$= \sum_{k=1}^{K} \frac{4cv^{\frac{1}{\varepsilon}} \chi\left(\bar{G_\gamma}\right)}{(2\Delta_k)^{1/\varepsilon}} \log T + \left(3M + \gamma\left(M-1\right)\right) \left( \sum_{k=1}^{K} \Delta_k \right). \tag{4.17}$$

□

**Remark 4.1** (Regret Bound)**.** The leading term in the previous bound depends on $\bar{\chi}(G_\gamma)$, the clique covering number of the power graph, and arises from the partitioning of the graph that omits "beyond-clique" edges, i.e., edges that are not present in the clique cover. This term can be loose when the graph is sparse, e.g., for a linear graph $G$, $\bar{\chi}(G_\gamma) = \mathcal{O}(M/\gamma)$. Note that the bound is still tight for the extreme cases of communication, e.g., when $\gamma = 0$, i.e., no communication, $\bar{\chi}(G_\gamma) = M$, recovering the independent learning bound. When $\gamma = \text{diam}(G), \bar{\chi}(G_\gamma) = 1$, recovering the optimal rate (see Section 2.6).

Observe that this approach requires sending messages $\mathbf{q}_m(t)$ which are of length $\mathcal{O}(1)$. We can improve the $\bar{\chi}(G_\gamma)$ leading term to a smaller quantity known as the *domination number* $\psi(G_\gamma)$ of the power graph by allowing for larger messages to be sent, i.e., messages that are $\mathcal{O}(K)$ in size. We first augment the earlier message $\mathbf{q}_m(t)$ as follows.

$$\mathbf{q}_m(t) = \langle m, t, a_m(t), x_m(t), \widehat{\boldsymbol{\mu}}_m(t), \mathbf{n}_m(t) \rangle. \tag{4.18}$$

Here, $\widehat{\boldsymbol{\mu}}_m(t) = (\hat{\mu}_{m,k})_{k\in[K]}$ are the robust mean estimates used by agent $m$ to make decisions at time $t$, and $\mathbf{n}_m(t) = (|\mathcal{S}_{m,k}(t)|)_{k\in[K]}$ is the vector containing the number of reward samples possessed by agent $m$ until time $t$. The algorithm FedUCB1 is then modified as follows. Each agent $m$ also maintains a set $\mathcal{W}_m$ of the most recent $(t_v, \widehat{\boldsymbol{\mu}}_v(t_v), \mathbf{n}_v(t_v))$ for each $v \in \mathcal{N}_m^+(G_\gamma)$, which they update with the latest message received from agent $v$. At any instant, the agent chooses, for each arm $k$, the corresponding $\hat{\mu}_{v^*,k}(t_{v^*})$ and $|\mathcal{S}_{v^*,k}(t_{v^*})|$ in $\mathcal{W}_m$ with the largest $|\mathcal{S}_{v^*,k}(t_{v^*})|$ to construct its upper confidence bound, as follows.

$$Q_{m,k}(t) = \hat{\mu}\left(\mathcal{S}_{v^*,k}(t_{v^*}), \frac{1}{t_{v^*}^2}\right) + 2\rho^{\frac{1}{1+\varepsilon}}\left(\frac{2c\log(t_{v^*})}{|\mathcal{S}_{v^*,k}(t_{v^*})|}\right)^{\frac{\varepsilon}{1+\varepsilon}}, \text{ where } v^* = \arg\min_{v\in\mathcal{W}_m} |\mathcal{S}_{v,k}(t_v)|. \tag{4.19}$$

Since weakly connected agents in the network will not have many additional observations to produce tight confidence intervals compared to the better connected agents in $G$, by allowing agents to communicate *all* of their estimates in each message, it is possible for weakly-connected agents to leverage the estimators from strongly-connected ones. We call this algorithm FedUCB1-Best (Algorithm 4), and present the regret bound below.

**Theorem 4.2.** *There exists an absolute constant $C > 0$ independent of $T, K, M, \gamma$ and $G$ such that the group regret for* FedUCB1-Best *when run with parameter $\gamma$ and mean estimator $\hat{\mu}(n, \delta)$ satisfies:*

$$\mathfrak{R}(T) \le C \cdot \psi(G_\gamma) \cdot \left(\sum_{k:\Delta_k>0} \Delta_k^{-1/\varepsilon}\right)\log(T) + (\psi(G_\gamma)(\gamma+2) + M)\left(\sum_{k:\Delta_k>0} \Delta_k\right).$$

*Here $\psi(\cdot)$ denotes the domination number.*

*Proof.* Consider the maximal dominating set of $G_\gamma$ given by $V'$. We can decompose the group regret based on $V'$ as follows.

$$\mathfrak{R}(T) = \sum_{m\in V'} \mathfrak{R}_m(T) + \sum_{m\in V\setminus V'} \mathfrak{R}_m(T) \tag{4.20}$$

$$\le \sum_{m\in V'}\left(\mathfrak{R}_m(T) + \sum_{m'\in\mathcal{N}_m^+(G_\gamma)} \mathfrak{R}_{m'}(T)\right) \tag{4.21}$$

Now, for any agent $m$, consider the total regret for all agents in $\mathcal{N}_m^+(G_\gamma) \cup \{m\}$.

$$\mathfrak{R}_m(T) + \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \mathfrak{R}_{m'}(T) = \sum_{k:\Delta_k > 0} \Delta_k \left( \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \sum_{t=1}^{T} \mathbb{P}\left(a_{m'}(t) = k\right) \right) \tag{4.22}$$

For any set of constants $\eta_k^{m'} > 0, k \in [K], m' \in \mathcal{N}_m^+(G_\gamma)$, and $\tilde{n}_k(t) = \sum_{v \in \mathcal{N}_m^+(G_\gamma)} n_{v,k}(t), \beta_k = \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \eta_k^m$,

$$\leq \sum_{k:\Delta_k > 0} \Delta_k \left( \beta_k + \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \sum_{t=1}^{T} \mathbb{P}\left(a_{m'}(t) = k, \tilde{n}_k(t) > \beta_k\right) \right) \tag{4.23}$$

For any arm $k \in [K]$, at any trial $t$, any agent $v \in \mathcal{N}_m^+(G_\gamma)$ chooses the confidence bound based on the pair $\hat{\mu}_{m^*,k}(t), n_{m^*,k}(t)$, where $n_{m^*,k}(t) = \max_{u \in \mathcal{N}_v^+(G_\gamma)} |\mathcal{S}_{u,k}(t - d(m,u))|$, and $\hat{\mu}_{m^*,k}(t)$ is the corresponding robust mean. Now, we know that $m'$ is in the $\gamma$-neighborhood of $m$, therefore,

$$\begin{aligned} n_{m^*,k}(t_{m^*}) &= \max_{u \in \mathcal{N}_m^+(G_\gamma)} |\mathcal{S}_{u,k}(t - d(m,u))| \geq |\mathcal{S}_{u,m}(t - d(m,u))| \\ &= \tilde{n}_k(t - d(m,u)) \geq \tilde{n}_k(t) - d(m,u) \geq \tilde{n}_k(t) - \gamma. \end{aligned}$$

Now, any agent $v \in \mathcal{N}_m^+(G_\gamma)$ pulls a suboptimal arm $k$ when one of three events occurs.

$$\text{Event (A): } \hat{\mu}_{m^*,*}(t-1) \leq \mu_* - v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t_{m^*})}{|\mathcal{S}_{m^*,*}(t_{m^*})|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.24}$$

$$\text{Event (B): } \hat{\mu}_{m^*,k}(t-1) \geq \mu_k + v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t_{m^*})}{|\mathcal{S}_{m^*,k}(t_{m^*})|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.25}$$

$$\text{Event (C): } \mu_* \leq \mu_k + 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t_{m^*})}{|\mathcal{S}_{m^*,k}(t_{m^*})|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.26}$$

Now, let us examine the occurence of event $(C)$:

$$\Delta_k \leq 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t_{m^*})}{|\mathcal{S}_{m^*,k}(t_{m^*})|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \tag{4.27}$$

$$\implies n_{m^*,k}(t_{m^*}) \leq c v^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} \tag{4.28}$$

Therefore, by setting $\beta_k = \left\lceil cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma \right\rceil$, Event (C) will not occur for any agent $m' \in \mathcal{N}_m^+(G_\gamma)$. Additionally, using the union bound over $n_{m^*,*}(t)$ and $n_{m^*,k}(t)$, and Assumption 4.1, we have:

$$\mathbb{P}(\text{Event (A) or (B) occurs}) \leq 2 \sum_{s=\gamma+1}^{t} \frac{1}{(s-\gamma)^4} \leq \frac{2}{(t-\gamma)^3}.$$

Replacing this in the total regret for $m' \in \mathcal{N}_m^+(G_\gamma) \cup \{m\}$, we have,

$$\begin{aligned}
&\mathfrak{R}_m(T) + \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \mathfrak{R}_{m'}(T) \\
&\leq \sum_{k:\Delta_k>0} \Delta_k \left( \beta_k + \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \sum_{t=1}^{T} \mathbb{P}\left(a_{m'}(t) = k, \tilde{n}_k(t) > \beta_k\right) \right) \\
&\leq \sum_{k:\Delta_k>0} \Delta_k \left( \left\lceil cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma \right\rceil + \sum_{m' \in \mathcal{N}_m^+(G_\gamma)} \sum_{t=\gamma+1}^{T+\gamma} \frac{2}{(t-\gamma)^3} \right) \\
&\leq \sum_{k:\Delta_k>0} \Delta_k \left( cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma + 2 + |\mathcal{N}_m^+(G_\gamma)| \right)
\end{aligned}$$

Summing over all agents $m \in V'$, we have,

$$\mathfrak{R}(T) \leq \psi(G_\gamma) \left( \sum_{k:\Delta_k>0} \Delta_k^{-1/\varepsilon} \right) \left( 2cv^{\frac{1}{\varepsilon}} \right) \log(T) + \left( \psi(G_\gamma)(\gamma+2) + M \right) \left( \sum_{k:\Delta_k>0} \Delta_k \right).$$

□

Contrasted to the regret bound of $O(\sqrt{|V|\alpha(G)TK \ln K})$ obtained by Cesa-Bianchi et al. (2019b) for the nonstochastic case (where communication is also $O(K)$ per agent), our algorithm obtains lower group regret in the stochastic case. Additionally, this implies a $\mathcal{O}(\log(M))$ improvement over the previous state-of-the-art bound for the stochastic case (Martínez-Rubio et al., 2019).

## 4.3 Partially-Decentralized Algorithm

We now demonstrate that the improved $\mathcal{O}\left(\psi(G_\gamma) \cdot \sum_{k:\Delta_k>0} \frac{\log(T)}{\Delta_k^{1/\varepsilon}}\right)$ can be achieved in fact with messages of length $\mathcal{O}(1)$ instead of $\mathcal{O}(K)$ by modifying the exploration strategy for poorly-connected agents in the network. In settings where the number of arms is prohibitively large, i.e., $K = \Omega(M)$, this alternate algorithm can provide competitive performance. The algorithm, dubbed CentUCB1, is a version of the "follow-the-leader" strategy. Here, the agents are partitioned into "leaders" and "followers". The leader agents follow the same procedure identically to Algorithm 3, and the follower agents simply copy the most recent action they have observed of their associated leader. We select the leaders as the members of the smallest dominating set $V' \subseteq V$ of $G_\gamma$. For each follower agent $v \in V \setminus V'$ we assign a leader $l(v)$ to it such that $(a)$ there is an edge between $v$ and $l(v)$ in $G_\gamma$, and $(b)$ $l(v)$ has maximum degree in $V' \cap \mathcal{N}_v^+(G_\gamma)$, i.e. $l(v) \in V'$ such that $l(v) = \arg\max_{v' \in V' \cap \mathcal{N}_v^+(G_\gamma)} \deg(v)$. Algorithm 5 describes this algorithm particularly from its differences with FedUCB1. We present the associated regret bound.

**Theorem 4.3.** *There exists an absolute constant $C > 0$ independent of $T, K, M, \gamma$ and $G$ such that the group regret for* CentUCB1 *when run with parameter $\gamma$ and mean estimator $\hat{\mu}(n, \delta)$ satisfies:*

$$\mathfrak{R}(T) \leq C \cdot \psi(G_\gamma) \left( \sum_{k:\Delta_k > 0} \Delta_k^{-1/\varepsilon} \right) \log(T) + \left(\psi(G_\gamma) + 1\right)(M+1) \left( \sum_{k:\Delta_k > 0} \Delta_k \right)$$

*Here $\psi(\cdot)$ denotes the domination number.*

*Proof.* We begin by decomposing the group regret into leader and follower nodes, grouped according to their position in $G$ and the minimal dominating set.

$$\begin{aligned} \mathfrak{R}(T) &= \sum_{m \in G} \mathfrak{R}_m(T) && (4.29) \\ &= \sum_{m \in V'} \left( \mathfrak{R}_m(T) + \sum_{f \in \mathcal{N}_m(G_\gamma)} \mathfrak{R}_f(T) \right) && (4.30) \\ &= \sum_{t=1}^{T} \sum_{k:\Delta_k > 0} \sum_{m \in V'} \Delta_k \left( \mathbb{P}\left\{a_m(t) = k\right\} + \sum_{f \in \mathcal{N}_m(G_\gamma)} \mathbb{P}\left\{a_f(t) = k\right\} \right). && (4.31) \end{aligned}$$

For constants $\eta_k^m, \eta_k^f > 0, m \in V', f \in \mathcal{N}_m(G_\gamma)$, let $U_m^k$ be the event when $\eta_k^m + \sum_{f \in \mathcal{N}_m(G_\gamma)} \eta_k^f \leq N_k^m(t) + \sum_{f \in \mathcal{N}_m(G_\gamma)} N_k^f(t)$. Then we have,

$$\mathfrak{R}(T) = \sum_{k:\Delta_k > 0} \sum_{m \in V'} \Delta_k \Big( \eta_k^m + \sum_{f \in \mathcal{N}_m(G_\gamma)} \eta_k^f + \sum_{t=1}^{T} \mathbb{P}\left\{a_m(t) = k; U_k^m\right\}$$

$$+ \sum_{f \in \mathcal{N}_m(G_\gamma)} \sum_{t=1}^{T} \mathbb{P}\left\{a_f(t) = k; U_k^m\right\} \Big). \quad (4.32)$$

We know that $a_f(t) = a_m(t - d(f,m))$. Let $\beta_k^m = \eta_k^m + \sum_{f \in \mathcal{N}_m(G_\gamma)} \eta_k^f$ for brevity. Therefore,

$$\mathfrak{R}(T) = \sum_{k:\Delta_k>0} \sum_{m \in V'} \Delta_k \left( \beta_k^m + \sum_{t=1}^{T} \mathbb{P}\left\{a_m(t) = k; U_k^m\right\} + \sum_{t=1}^{T-d(m,f)} \mathbb{P}\left\{a_m(t) = k; U_k^m\right\} \right) + \sum_{m \in V'} \sum_{f \in \mathcal{N}_m(G_\gamma)} d(m,f) \quad (4.33)$$

$$\implies \mathfrak{R}(T) \leq \sum_{k:\Delta_k>0} \sum_{m \in V'} \Delta_k \left( \beta_k^m + (|\mathcal{N}_m(G_\gamma)| + 1) \left( \sum_{t=1}^{T} \mathbb{P}\left\{a_m(t) = k; U_k^m\right\} \right) \right). \quad (4.34)$$

We see that a suboptimal arm is pulled when one of three events occurs.

$$\text{Event (A): } \hat{\mu}_{m,*}(t-1) \leq \mu_* - v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t)}{|\mathcal{S}_{m,*}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \quad (4.35)$$

$$\text{Event (B): } \hat{\mu}_{m,k}(t-1) \geq \mu_k + v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \quad (4.36)$$

$$\text{Event (C): } \mu_* \leq \mu_k + 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \quad (4.37)$$

Now, let us examine the occurence of event $(C)$:

$$\Delta_k \leq 2v^{\frac{1}{1+\varepsilon}} \left( \frac{2c \log(t)}{|\mathcal{S}_{m,k}(t)|} \right)^{\frac{\varepsilon}{1+\varepsilon}} \quad (4.38)$$

$$\implies |\mathcal{S}_{m,k}(t)| \leq c v^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} \quad (4.39)$$

Since agent $m$ can communicate only with its neighborhood $\mathcal{N}_\gamma(m)$, and $\mathcal{N}_m(G_\gamma) \subseteq \mathcal{N}_\gamma(m)$. Therefore $\sum_{f \in \mathcal{N}_m(G_\gamma) \cup \{m\}} N_f^k(t) \leq \sum_{f \in N_\gamma(m) \cup \{m\}} N_f^k(t)$, and since each message from a neighbor $f$ takes time $d(m,f) - 1$ time to reach agent $m$, we have that $\sum_{f \in \mathcal{N}_m(G_\gamma) \cup \{m\}} N_f^k(t) - \sum_{f \in N_\gamma(m) \cup \{m\}} (d(f,m) - 1) \leq |\mathcal{S}_{m,k}(t)|$. Using this in the previous equation and the fact that $d(m,f) \leq \gamma$, we have

$$\implies \sum_{f \in \mathcal{N}_m(G_\gamma) \cup \{m\}} N_f^k(t) \leq c v^{\frac{1}{\varepsilon}} \log(t) \left( \frac{2}{\Delta_k} \right)^{1+\frac{1}{\varepsilon}} + \gamma(|\mathcal{N}_\gamma(m)| + 1) \quad (4.40)$$

Therefore, we know that if $\beta_k^m \geq \left\lceil cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma(|\mathcal{N}_\gamma(m)|+1)\right\rceil$ then Event (C) does not occur. Additionally, using the union bound over $n_{m,*}(t)$ and $n_{m,k}(t)$, and Assumption 4.1, we have:

$$\mathbb{P}(\text{Event (A) or (B) occurs}) \leq 2\sum_{s=1}^{t} \frac{1}{s^4} \leq \frac{2}{t^3}. \tag{4.41}$$

Combining all probabilities, and inserting in the individual regret, we have,

$$\mathfrak{R}(T) \leq \sum_{k:\Delta_k>0} \sum_{m\in V'} \Delta_k \left(\beta_k^m + (|\mathcal{N}_m(G_\gamma)|+1)\left(\sum_{t=1}^{T} \mathbb{P}\left\{a_m(t)=k; U_k^m\right\}\right)\right) \tag{4.42}$$

$$\leq \sum_{k:\Delta_k>0} \sum_{m\in V'} \Delta_k \left(\left\lceil cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma(|\mathcal{N}_\gamma(m)|+1)\right\rceil + (|\mathcal{N}_m(G_\gamma)|+1)\left(\sum_{t=1}^{T} \frac{2}{t^3}\right)\right) \tag{4.43}$$

$$\leq \sum_{k:\Delta_k>0} \sum_{m\in V'} \Delta_k \left(cv^{\frac{1}{\varepsilon}} \log(T) \left(\frac{2}{\Delta_k}\right)^{1+\frac{1}{\varepsilon}} + \gamma(|\mathcal{N}_\gamma(m)|+1) + (|\mathcal{N}_m(G_\gamma)|+2)\right) \tag{4.44}$$

Since $|V'| \leq \psi(G_\gamma)$, where $\alpha(\cdot)$ denotes the domination number, we have,

$$\leq \sum_{k:\Delta_k>0} \left(2^{1+1/\varepsilon}\Delta_k^{-1/\varepsilon} cv^{1/\varepsilon}\psi(G_\gamma)\log(T) + \sum_{m\in V'} \Delta_k(\gamma(|\mathcal{N}_\gamma(m)|+1) + (|\mathcal{N}_m(G_\gamma)|+2))\right) \tag{4.45}$$

$$\leq \left(\sum_{k:\Delta_k>0} \Delta_k^{-1/\varepsilon}\right)\left(2^{1+1/\varepsilon} cv^{1/\varepsilon}\psi(G_\gamma)\right)\log(T) + (\psi(G_\gamma)M\gamma + M + \psi(G_\gamma))\left(\sum_{k:\Delta_k>0} \Delta_k\right) \tag{4.46}$$

□

Since $\psi(G) \leq \bar{\chi}(G)$ for any graph $G$, the centralized version of the FedUCB1 algorithm obtains regret strictly no worse compared to the decentralized version. A few additional remarks can be made, inspired by Bar-On & Mansour (2019).

**Remark 4.2.** The average regret from Algorithm 5 is $\mathcal{O}((\psi(G_\gamma)/M)K\log(T))$, i.e. optimal when $\gamma = \text{diam}(G)$. When $\gamma = \sqrt{K}$, Algorithm 5 can obtain a per-agent regret of $\mathcal{O}(\Delta_*^{-1/\varepsilon}\sqrt{K}\log(T))$. This can be shown by noticing that when $G$ is connected, $\psi(G_\gamma) \leq G\psi(G_\gamma) \leq \lceil 2M/(\gamma+2)\rceil$. Also note that we need only $\sqrt{K}$ leaders at most to obtain this regret. When $\gamma = \mathrm{diam}(G)$, then, any arbitrarily chosen leader can deliver optimal regret, regardless of its position in $G$.

## 4.4 Experiments

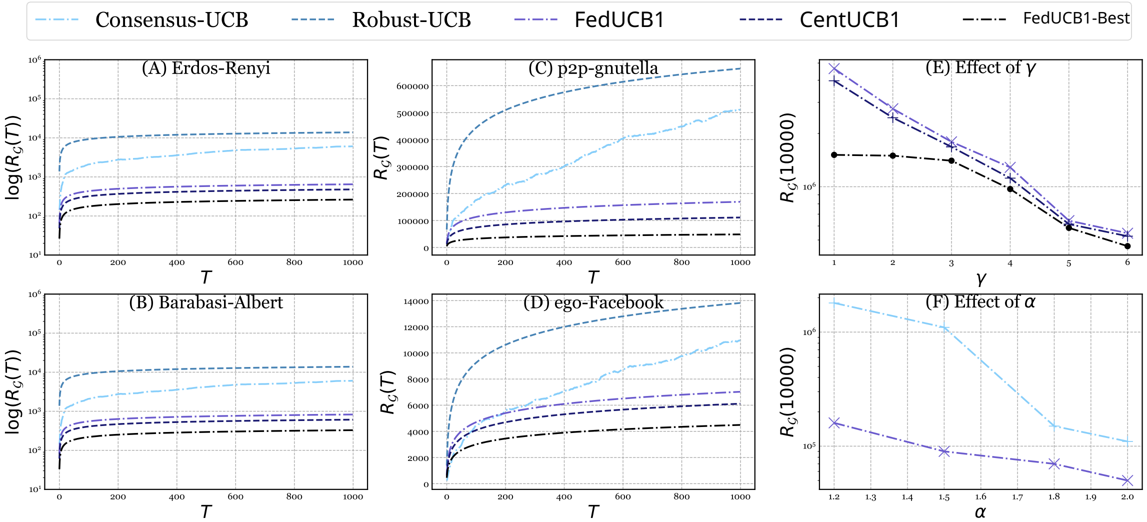


Figure 4-1: Experimental benchmarks, where each experiment is averaged over 100 trials. Figures (A) and (B) compare performance on samples of random graphs; (C) and (D) compare performance on two classes of real-world networks, and (E) and (F) are ablations.

Our primary contributions are in leveraging cooperation to accelerate overall decision-making, and the most interesting aspects of this study pertain to how graph structures, scalability, heavy tails and decentralized vs. centralized estimation affect the group regret.

*Reward Distributions*. We conduct experiments using $\alpha$-stable densities (Lévy, 1925), that admit finite moments only of order $< \alpha \leq 2$, and we consider $\alpha$-stable densities where $\alpha \geq 1$. The $\alpha$-stable family includes several widely used distributions, such as Gaussian ($\alpha = 2$, only light-tailed density), Lévy ($\alpha = 0.5$) and Cauchy ($\alpha$=1).

*Graph Partitioning*. For Algorithm 5, we require computing the minimal dominating set of $G_\gamma$. We use the approximate algorithm presented in (Lucas, 2014) that uses the QUBO (Glover & Kochenberger, 2018) solver.

**Experiment 1: Random Graphs**. We set $K = 5$, $\alpha = 1.9$ for the standard $\alpha$-stable density, and sample arm means randomly from the interval $[0, 1]$ for each arm every experiment. We then construct random graphs on 200 agents from the Erdos-Renyi (ER) ($p = 0.7$) and Barabasi-Albert (BA) ($m = 5$) random graph families, and compare all three of our algorithms (using the trimmed mean estimator, with $\gamma = \text{diam}(G)/2$) with the Consensus-UCB and single-agent Robust-UCB(Bubeck et al., 2013) algorithms. We compare the group regret $\mathfrak{R}(T)$ vs. $T$, averaged over 100 random graphs and bandit instances. The results for Erdos-Renyi graphs (Figure (4-1A)) and Barabasi-Albert graphs (Figure (4-1B)) demonstrate that while our algorithms outperform the baselines (in the order dictated by regret bounds), the gain is larger for the former. We attribute this to connectivity, i.e., since Barabasi-Albert graphs have "hubs", the clique number $\bar{\chi}(G)$ for these graphs is larger.

**Experiment 2: Real-World Networks**. We select the p2p-Gnutella04 (Figure 4-1C) and ego-Facebook (Figure 4-1D) network structures from the SNAP repository (Leskovec & Sosič, 2016) to experiment with in the real-world setting. For both experiments, we sample subgraphs of 500 nodes, and use these subgraphs. A common misconception is to compare our distributed *multi-agent* problem with the *social network clustering* problem (Gentile et al., 2014; Li et al., 2016), which is more scalable since it is *single-agent* (i.e., one action chosen per trial). These networks are chosen because they represent two diverse situations cooperative decision-making can be applicable in – social networks and peer-to-peer communication networks. In both cases, we observe a similar trend. The gains are larger in the p2p-Gnutella case since ego-Facebook is dense (with fewer nodes), hence Consensus-UCB performs better as well.

**Experiment 3: Effect of $\gamma$ and $\alpha$**. As ablation experiments, we investigate the effect of communication density $\gamma$ (Figure 4-1E) and tail parameter $\alpha$ (Figure 4-1F) on the group regret. For both experiments, we construct random graphs on 200 agents from the Erdos-Renyi ($p = 0.7$) family. We compare the group regret at $T = 10000$ trials as a function of $\gamma$, and $\alpha$, respectively. First, we observe that communication density has a significant effect on all but the FedUCB1-Best algorithm. Next, we see that Consensus-UCB progressively gets worse as the tail gets heavier (i.e, $\alpha \to 1^+$).

## 4.5 Lower Bound and Discussion

For the specific case of $(1+\varepsilon)$-heavy tailed rewards, we present an extension of the lower bounds from Chapter 2 to the modified environment.

**Corollary 4.1** (Lower Bound on Heavy-Tailed Cooperative Regret)**.** *For any* $\Delta \in (0, 1/4)$, *there exist $K \geq 2$ distributions $\nu_1, ..., \nu_K$ satisfying $\mathbb{E}_{X\sim\nu_k}[|X|^{1+\varepsilon}] \leq u$, and $\mathbb{E}_{X\sim\nu_*}[X] - \mathbb{E}_{X\sim\nu_k}[X] = \Delta \forall k \in K$, such that any consistent decentralized policy $\Pi_t = (\pi_{m,t})_{m\in[M],t\in[T]}$ that satisfies Assumption 2.1 obtains group regret of $\Omega(K\Delta^{-1/\varepsilon}\log(T))$. Furthermore, if the decentralized policy is NAIC (Definition 2.6) then it must obtain group regret of $\Omega(K\alpha(G_{\gamma+1}) \cdot \Delta^{-1/\varepsilon}\log(T))$*

*Proof.* Consider $\nu_1(x) = \left(1 - \alpha^{1+\varepsilon}\right)\delta(x) + \alpha^{1+\varepsilon}\delta(x - 1/\alpha)$, where $\alpha = (2\Delta)^{\frac{1}{\varepsilon}}$, $\delta(x-\gamma)$ is the Dirac distribution at $\gamma$, and let $\forall\, k \in [2, K]$, $\nu_k(x) = \left(1 - \alpha^{1+\varepsilon} + \Delta\alpha\right)\delta(x) + \left(\alpha^{1+\varepsilon} - \Delta\alpha\right)\delta(x - 1/\alpha)$. We can see that

$$\mathbb{E}_{X\sim\nu_i}[|X|^{1+\varepsilon}] = 1 \;\forall i \in [K] \text{ and } \mathbb{E}_{X\sim\nu_1}[X] - \mathbb{E}_{X\sim\nu_i}[X] = \Delta \;\forall i \in [2, K].$$

Hence, $\nu_i$ satisfy the constraints stated in the Theorem. Now, we can see that $\nu_1$ corresponds to a scaled Bernoulli distribution with parameter $\alpha^{1+\varepsilon}$, and similarly, $\nu_i, i \in [2, K]$ correspond to a scaled Bernoulli distribution with parameter $\alpha^{1+\varepsilon} - \Delta\alpha$. Since an algorithm operating on $\nu_1, ..., \nu_K$ will exhibit identical behavior on reward distributions $\mathcal{B}(\alpha^{1+\varepsilon}), ..., \mathcal{B}(\alpha^{1+\varepsilon} - \Delta\alpha)$, we can note the following for Bernoulli distributions for two Bernoulli distributions with arm parameter $\mu_k$ optimal parameter $\mu^*$.

$$\mathbb{D}_k^{\mathsf{inf}} = \mu_k \log\left(\frac{\mu_k}{\mu_*}\right) + (1-\mu)\log\left(\frac{1-\mu_k}{1-\mu_*}\right) = \mathbb{D}_{\mathsf{KL}}(\mu_k, \mu^*) \tag{4.47}$$

Therefore we can apply Theorem 2.3 directly to obtain the following lower bound on the number of pulls of any suboptimal arm.

$$\begin{aligned}
\mathbb{E}[N_k(T)] &\geq \left(\frac{1}{\mathbb{D}_{\mathsf{KL}}(\nu_k, \nu_*)}\right)\ln T && (4.48)\\
&\overset{(a)}{=} \left(\frac{1}{\mathbb{D}_{\mathsf{KL}}(\mathcal{B}(\alpha^{1+\varepsilon} - \Delta\alpha), \mathcal{B}(\alpha^{1+\varepsilon}))}\right)\ln T && (4.49)\\
&\overset{(b)}{\geq} \left(\frac{\alpha^{\varepsilon-1} - \alpha^{2\varepsilon}}{\Delta^2}\right)\ln T && (4.50)\\
&= \left(\frac{2^{1-\frac{1}{\varepsilon}}}{\Delta^{1+\frac{1}{\varepsilon}}}\right)\ln T. && (4.51)
\end{aligned}$$

Here, $(a)$ is obtained by the equivalence of $\nu_1, ..., \nu_K$ to Bernoulli distributions, and $(b)$ is obtained by the inequality $\mathbb{D}_{\mathsf{KL}}(\mathcal{B}(\theta_2), \mathcal{B}(\theta_1)) \leq \frac{(\theta_1-\theta_2)^2}{\theta_1(1-\theta_1)}$. We now decompose the regret.

$$R(T) = \sum_{k=2}^{K} \Delta_k \mathbb{E}[N_k(T)] \tag{4.52}$$

$$\geq \sum_{k=2}^{K} \left( \frac{2^{1-\frac{1}{\varepsilon}}}{\Delta^{\frac{1}{\varepsilon}}} \right) \ln T. \tag{4.53}$$

The second part follows from an identical analysis applied directly to Theorem 2.4 for NAIC policies. □

**Discussion**. In this chapter, we presented a treatment of federated bandit estimation under heavy tails. We provided the first asymptotic lower bound on cooperative estimation that holds for arbitrary graphs $G$ and a wide variety of communication protocols. We present the first robust cooperative estimation algorithms that can all provide optimal regret, even without knowledge of $G$. However, our work leaves several open questions in *robust* multi-agent decision-making.

First, we note that our best algorithm provides a group regret of $\mathcal{O}(G\psi(G_\gamma)K \ln T)$, which is similar to the results obtained in the non-stochastic case Cesa-Bianchi et al. (2019b); Martínez-Rubio et al. (2019). The $G\psi(G_\gamma)$ overhead can be attributed to the fact that information does not flow completely through the network (cf. Assumption 2.1a). This leads us to believe that tighter lower bounds can be obtained by taking this aspect of the communication protocol into account. Moreover, in realistic settings, messages incur stochasticity, i.e. they can be dropped at random, or propagate with varying delay $\gamma$. This line of work has been studied in the single-agent setting Pike-Burke et al. (2017); Vernade et al. (2018), however the problem becomes more challenging when multiple agents interact simultaneously. The extension of our setting to the contextual case is not trivial. Robust single-agent estimation for linear bandits is a difficult problem from both the algorithmic and computational point of view, since statistically optimal multivariate estimators require exponential time to compute Lugosi & Mendelson (2019). Furthermore, delay creates a $\sqrt{\gamma}$ scaling of the regret Neu et al. (2010), which is amplified in the multi-agent setting. Addressing such scenarios is an interesting next step in this line of research.

## 4.6 Algorithm Pseudocode

**Algorithm 3** FedUCB1

```
Input: Arms $k \in [K]$, parameters $\varepsilon, c, \rho$, estimator $\hat{\mu}(n, \delta)$
$S_k^v \leftarrow \phi \; \forall k \in [K]$, $\mathcal{Q}_v(t) \leftarrow \phi, \forall v \in V$.
for each iteration $t \in [T]$ do
    for each agent $v \in V$ do
        if $t \leq K$ then
            $a_m(t) \leftarrow t$.
        else
            for Arm $k \in [K]$ do
                $\hat{\mu}_k^{(v)} \leftarrow \hat{\mu}(S_k^v, 1/t^2)$.
                $\text{UCB}_k^{(v)}(t) \leftarrow \rho^{\frac{1}{1+\varepsilon}} \left(\frac{2c\log(t)}{|S_k^v|}\right)^{\frac{\varepsilon}{1+\varepsilon}}$.
            end for
            $A_{v,t} \leftarrow \arg\max_{k \in [K]} \left\{\hat{\mu}_k^{(v)}(t) + \text{UCB}_k^{(v)}(t)\right\}$.
        end if
        $X_{v,t} \leftarrow \textsc{Pull}(A_{v,t})$.
        $S_{A_{v,t}}^v \leftarrow S_{A_{v,t}}^v \cup \{X_{v,t}\}$
        $\mathcal{Q}_v(t) \leftarrow \mathcal{Q}_v(t) \cup \{\langle v, t, A_{v,t}, X_{v,t}\rangle\}$.
        for each neighbor $v'$ in $\mathcal{N}_1(v)$ do
            $\textsc{SendMessages}(v, v', \mathcal{Q}_v(t))$.
        end for
    end for
    for each agent $v \in V$ do
        $Q_v(t+1) \leftarrow \phi$.
        for each neighbor $v'$ in $\mathcal{N}_1(v)$ do
            $Q' \leftarrow \textsc{ReceiveMessages}(v', v)$
            $Q_v(t+1) \leftarrow Q_v(t+1) \cup Q'$.
        end for
        for $\langle v', t', a', x'\rangle \in Q_v(t+1)$ do
            if $v' \in \textsc{Clique}(v, G_\gamma)$ then
                $S_{a'}^v \leftarrow S_{a'}^v \cup \{x'\}$.
            end if
        end for
    end for
end for
```

**Algorithm 4** FedUCB1-Best

```
**Input**: $K, \varepsilon, \hat{\mu}_R(n,\delta), c, v$.
Set $\mathcal{S}_{m,k} = \phi\ \forall k \in [K]$, $Q_m(t) = \phi\ \forall m \in [M]$
Set $W_m = \mathbf{1}^{|\mathcal{N}_\gamma(m)| \times 2K}\ \forall m \in [M]$.
**for** For each iteration $t \in [T]$ **do**
    **for** For each agent $m \in [M]$ **do**
        **if** $t \leq K$ **then**
            $a_m(t) = t$.
        **else**
            **for** Arm $k \in [K]$ **do**
                $\hat{\mu}_k^{(m)}(t) = \hat{\mu}_R(\mathcal{S}_{m,k}, 2\log(t))$.
            **end for**
            $W_m = \left(\hat{\mu}_k^{(m)}(t), |\mathcal{S}_{m,k}|\right)_{k\in[K]}$.
            **for** Arm $k \in [K]$ **do**
                $m^* = \arg\max_m \{s_k^m : (\mu_{m,k}, s_k^m) \in W\}$.
                $\hat{\mu}_k^{m,*}, |S_k^{m,*}| = W_{m^*,k}$
                $\mathrm{UCB}_k^{(m)}(t) = v^{\frac{1}{1+\varepsilon}} \left(\frac{2c\log(t)}{|S_k^{m,*}|}\right)^{\frac{\varepsilon}{1+\varepsilon}}$.
            **end for**
            $a_m(t) = \arg\max_{k\in[K]} \left\{\hat{\mu}_k^{m,*}(t) + \mathrm{UCB}_k^{(m)}(t)\right\}$.
        **end if**
        $a_m(t) r_{m,t} = \textsc{Pull}(a_m(t))$.
        $S_{a_m(t)}^m = S_{a_m(t)}^m \cup \{a_m(t) r_{m,t}\}$
        $Q_m(t) = \textsc{PruneDeadMessages}(Q_m(t))$.
        $q_m(t) = \left\langle m, t, \gamma, a_m(t), a_m(t) r_{m,t}, \hat{}^{(m)}(t), \mathbf{N^m(t)} \right\rangle$
        $Q_m(t) = Q_m(t) \cup \{q_m(t)\}$.
        Set $l = l - 1\ \forall \langle m', t', l, a', x', \mathbf{d}', \mathbf{n}' \rangle$ in $Q_m(t)$.
        **for** Each neighbor $m'$ in $\mathcal{N}_1(m)$ **do**
            $\textsc{SendMessages}(m, m', Q_m(t))$.
        **end for**
    **end for**
    **for** For each agent $m \in [M]$ **do**
        $Q_m(t+1) = \phi$.
        **for** Each neighbor $m'$ in $\mathcal{N}_1(m)$ **do**
            $Q' = \textsc{ReceiveMessages}(m', m)$
            $Q_m(t+1) = Q_m(t+1) \cup Q'$.
        **end for**
        **for** $\langle m', t', l', a', x', \mathbf{d}', \mathbf{n}' \rangle\ \in Q_m(t+1)$ **do**
            $S_{a'}^m = S_{a'}^m \cup \{x'\}$.
            $W_{m'} = (\mathbf{d}', \mathbf{n}')$
        **end for**
    **end for**
**end for**
```

**Algorithm 5** CentUCB1

```
Input: Same as Algorithm 3.
Set $S_k^v \leftarrow \phi\ \forall k \in [K]$, $\mathcal{Q}_v(t) \leftarrow \phi$, $A_v^* \leftarrow 1$, for all $v \in V$.
for each iteration $t \in [T]$ do
    for each agent $v \in V$ do
        if $t \leq K$ then
            $A_{v,t} \leftarrow t$.
        else if $v \in V'$ or $t \leq d(v, l(v))$ then
            Run lines 8-12 of Algorithm 3.
        else
            $A_{v,t} \leftarrow A_v^*$.
        end if
        Run lines 14-19 of Algorithm 3.
    end for
    for each agent $v \in V$ do
        Run lines 22-26 of Algorithm 3.
        for $\langle v', t', a', x' \rangle \in \mathcal{Q}_v(t+1)$ do
            $S_{a'}^v \leftarrow S_{a'}^v \cup \{x'\}$.
        end for
        $A_v^* =$ChooseLastAction$(\cup_k S_k^v(t+1))$.
    end for
end for
```

**Algorithm 6** Online Trimmed Mean Estimator

```
Input: $u, T$.
Create dictionary $D$ of size $T$, where $D(t) = \phi\ \forall t \in [T]$.
Create BST $B$ with entries $((2u \log(t))^{1/(1+\varepsilon)})_{t \in [T]}$.
$\hat{S}_O \leftarrow 0, n \leftarrow 0$
for $t \in [T]$ do
    $O_t \leftarrow$ Observations$(t)$.
    for $x_t \in O_t$ do
        $n \leftarrow n + 1$
        $i_t \leftarrow \max\left(t, \text{Search}(B, (|x_t|^{1+\varepsilon}/n))\right)$.
        $D(i_t) \leftarrow D(i_t) \cup \{x_t\}$.
    end for
    for $x \in D(t)$ do
        $\hat{S}_O \leftarrow \hat{S}_O + x$.
    end for
    $\hat{\mu}_O(t) \leftarrow \hat{S}_O / n$.
end for
```

# Chapter 5

# Handling Byzantine and Adversarial Corruptions

In prior treatment of the cooperative multi-armed bandit, adversarial loss functions have also been considered (Cesa-Bianchi et al., 2019b). However, in this line of work, the motivation is very different from that of federated learning systems: it is assumed that each of the multiple agents interact with *identical* adversarial bandit instances, and the objective is to effectively *distribute* arms between agents to achieve the optimal rate. For example, consider the adversarial (nonstochastic) multi-armed bandit, e.g., the setup discussed in Auer et al. (2002b). In the single-agent setting, the environment (or adversary) selects a sequence of $T$ vector losses $\ell_1, ..., \ell_T, \ell_\tau \in [0,1]^K, \tau \in [T]$. The agent, at each round $\tau$, selects an action $a(\tau) \in [K]$ and observes the corresponding loss element in $\ell_\tau$.

When considering the multi-agent variant of this problem, prior research assumes that the sequence of losses $\ell_1, ..., \ell_T$ are identical for all players, and the central advantage of the multi-agent setup is to explore $M \leq K$ arms simultaneously, albeit with delays in communication (Cesa-Bianchi et al., 2019b,a). This setting appears to be inapplicable in modern federated learning systems: it is unlikely that the assumption of *identical* adversarial instances in large-scale decentralized systems. Furthermore, the strong robustness guarantees that arrive with nonstochastic algorithms come at a price: guarantees for nonstochastic bandit algorithms are significantly weaker, whereas one can obtain logarithmic regret for stochastic algorithms. Nevertheless, it is unlikely that application domains obey nice stochastic properties typically assumed in Chapters 2-4. Non-stochastic corruptions in federated bandit systems can arise from a variety of circumstances, including byzantine agents, imperfect communication, and inherent non-stochasticity of the environment itself.

Therefore, in this chapter we discuss federated multi-armed bandits with corruption. We focus on two types of corruption - Huber contamination, where a constant, small fraction of rewards are drawn from arbitrary corruption distributions, and another, more challenging setting, where rewards can arbitrarily be corrupted from the default stochastic assumption. For both settings, we provide novel algorithms that provide competitive performance, without knowledge of the corruption levels in either setting.

## 5.1 Huber Contamination

In this section, we examine the federated bandit problem when there exist *byzantine* agents, that, at any trial, instead of reporting the true rewards, provide a random sample from an alternate (but fixed) distribution, with some probability $\epsilon$. Once a message is created, however, we assume that it is received correctly by all subsequent agents, and any corruption occurs only at the source (see Remark 5.1). Specifically, for any agent $m$ and arm $k$, the random reward $x_{m,k}(t)$ is drawn from the mixture distribution $(1-\epsilon)\cdot P_k + \epsilon \cdot Q$, where $P_k$ is the reward distribution for arm $k$ and $Q$ is an arbitrary unknown contamination distribution. The general problem of estimating statistics of a distribution $P_1$ from samples of a mixture distribution of the form $(1-\epsilon)\cdot P_1 + \epsilon \cdot P_2$ for $\epsilon < 1/2$ is a classic contamination model in robust statistics, known as Huber's $\epsilon$-contamination model (Huber, 2011).

**Remark 5.1** (Modeling Assumption)**.** One might consider the setting where in addition to messages being corrupted at the source, it is possible for a message to be corrupted by any intermediary byzantine agent with the same probability $\epsilon$. From a technical perspective, this setting is not very different than the first setting, since the probability of any incoming message being corrupted becomes at most $\gamma\epsilon$ (by the union bound and that $d(m, m') \leq \gamma$ for any pair of agents $m, m'$ that can communicate), and the remainder of the analysis is identical henceforth. Therefore, we simply consider the first setting.

**Problem Setting and Messaging Protocol**. We consider for any agent $m$, the message $\mathbf{q}_m(t) = \langle m, t, a_m(t), \hat{x}_m(t)\rangle$ created at time $t \in [T]$. For byzantine agents, $\hat{x}_m(t) = x_m(t)$ i.i.d. with probability $1-\epsilon$, and a random sample from an unknown (but fixed) distribution $Q$ with probability $\epsilon$. For non-byzantine agents, $\hat{x}_m(t) = x_m(t)$ with probability 1. In the univariate setting, a popular approach to robustly estimate the mean in the presence of outliers is to utilize a *trimmed* estimator of the mean that is robust to outlying samples, that works as long as $\epsilon$ is small. We utilize the trimmed estimator to design our algorithm as well, defined as follows.

**Definition 5.1** (Robust Trimmed Estimator)**.** *Consider a set of* $2N$ *samples* $X_1, Y_2, ..., X_N, Y_N$ *of a mixture distribution* $(1-\epsilon)\cdot P + \epsilon \cdot Q$ *for some* $\epsilon \in [0, 1/2)$. *Let* $X_1^*, \leq X_2^* \leq ... \leq X_N^*$ *denote a non-decreasing arrangement of* $X_1, ..., X_N$. *For some confidence level* $\delta \in (0,1)$, *let* $\alpha = \max(\epsilon, \frac{\ln(1/\delta)}{N})$, *and let* $\mathcal{Z}$ *be the shortest interval containing* $N\left(1 - 2\alpha - \sqrt{2\alpha\frac{\ln(4/\delta)}{N}} - \frac{\ln(4/\delta)}{N}\right)$ *points of* $\{Y_1^*, ..., Y_N^*\}$. *The trimmed estimator* $\hat{\mu}_R(\{X_1, Y_2, ..., X_N, Y_N\}, \delta)$ *is defined as*

$$\hat{\mu}_R = \frac{1}{\sum_i^N \mathbb{1}\left\{X_i \in \mathcal{Z}\right\}} \sum_{i=1}^N X_i \mathbb{1}\left\{X_i \in \mathcal{Z}\right\}. \tag{5.1}$$

We now present a confidence interval for this quartile-based trimmed mean estimator.

**Theorem 5.1** (Confidence Interval, Prasad et al. (2019))**.** *Let* $\delta \in (0,1)$. *Then for any distribution* $P$ *with mean* $\mu$ *and finite variance* $\sigma^2$, *we have, with probability at least* $1-\delta$,

$$|\hat{\mu}_R - \mu| \leq \sigma\epsilon + \sqrt{\frac{\sigma \ln(1/\delta)}{N}}.$$

We first describe an algorithm for the federated bandit when the likelihood of contamination $\epsilon$ is known in advance. While this indeed appears to be an impractical scenario, it will be the basis of the final algorithm that operates without knowledge of $\epsilon$, only assuming that $\epsilon \in [0, 1/2)$. Notice that $\epsilon \geq 1/2$ is a highly unusual pathological case, as in that case it is impossible to distinguish the "contaminating" distribution $Q$ from the "true" distribution $P$ (Prasad et al., 2019).

We consider a harder problem: we assume that *all* agents are byzantine, i.e., each reward sample is drawn from the aforementioned mixture distribution. While this primarily aids in the simplicity of the algorithm and analysis, it provides a worst-case algorithm, instead of assuming a certain fraction of agents are byzantine. From the perspective of any agent, if all $M$ agents are byzantine, then it can expect to obtain approximately $O(TM\epsilon)$ corrupted messages. If however, only a fraction $f < 1$ of agents are byzantine, then it can expect to obtain approximately $O(TMf\epsilon)$ corrupted messages, which would imply that (in expectation), all $M$ agents are byzantine with $\epsilon' = f\epsilon$. This information can be incorporated at runtime as well, and hence we proceed with the conservative assumption.

The algorithm itself, dubbed RobustFedUCB($\epsilon$) (Algorithm 7), is an extension of FedUCB, where instead of the sub-Gaussian confidence intervals utilized for the basic setting, we utilize the confidence intervals obtained from the robust estimator described earlier. It requires the value of $\epsilon$ to be known in advance, and the regret analysis is identical to that of Theorem 4.1, except we utilize the confidence intervals from Theorem 5.1.

**Theorem 5.2.** *There exists an absolute constant $C > 0$ independent of $T, K, M, \gamma$ and $G$ such that the group regret for* RobustFedUCB($\epsilon$) *when run with parameter $\gamma$ satisfies, for all $\epsilon < \Delta_k^2/4$,*

$$\mathfrak{R}(T) \leq C \cdot \bar{\chi}\left(G_\gamma\right) \cdot \left(\sum_{k:\Delta_k>0} \frac{\Delta_k}{(\Delta_k - 2\epsilon)^2}\right) \log(T) + M \cdot (3 + \gamma\chi\left(G_\gamma\right)) \left(\sum_{k:\Delta_k>0} \Delta_k\right).$$

*Here, $\bar{\chi}(\cdot)$ refers to the minimum clique number.*

*Proof.* The proof is identical to that of Theorem 4.1 with the modification that the confidence intervals utilized are derived from Theorem 5.1. □

**Remark 5.2** (Deviation from Optimal Rates)**.** When comparing the group regret bound to the optimal rate achievable in the single-agent case, there is an additive constant that arises from the delay in the network. Identical to the FedUCB1, this additive constant reduces to the constant corresponding to $MT$ individual trials of the UCB algorithm when information flows instantly throughout the network, i.e. $G$ is connected. Additionally, we observe identical dependencies on other graph parameters in the leading term as well, except for the modified denominator $(\Delta_k - 2\epsilon)^2$. This term arises from the inescapable bias that the mixture distribution $Q$ introduces into estimation, and (Prasad et al., 2019) show that the optimal asymptotic bias of $\mathcal{O}(\epsilon)$ and unimprovable in general. Corresponding to this result, we present a multi-agent bandit lower bound that achieves a similar dependence.

**Theorem 5.3** (Lower Bound for Uniform Huber Contamination)**.** *For any $\Delta \in (0, 1/4)$, there exist $K \geq 2$ distributions $\nu_1, ..., \nu_K$ satisfying $\mathbb{E}_{X\sim\nu_*}[X] - \mathbb{E}_{X\sim\nu_k}[X] = \Delta \forall k \in K$ such that the following holds: Consider any consistent decentralized policy $\Pi_t = (\pi_{m,t})_{m\in[M],t\in[T]}$ that uses a communication protocol satisfying Assumptions 2.1, on a bandit problem with reward distributions $\nu_1, ..., \nu_K$ such that an adversary can corrupt any message with at most $\epsilon-$Huber contamination.*

*Then, the policy satisfies,*

$$\liminf_{T\to\infty}\frac{\mathfrak{R}(T)}{\log(T)} \geq \sum_{k=2}^{K}\frac{3}{\Delta(1-2\epsilon)^2} = \sum_{k=2}^{K}\frac{3\Delta}{(\Delta-2\epsilon)^2+4\epsilon\cdot\Delta(1-\Delta)}.$$

*Furthermore, if the policy satisfies NAIC (Definition 2.6), then,*

$$\liminf_{T\to\infty}\frac{\mathfrak{R}(T)}{\log(T)} \geq \alpha(G_{\gamma+1})\cdot\sum_{k=2}^{K}\frac{3}{\Delta(1-2\epsilon)^2} = \alpha(G_{\gamma+1})\cdot\sum_{k=2}^{K}\frac{3\Delta}{(\Delta-2\epsilon)^2+4\epsilon\cdot\Delta(1-\Delta)}.$$

*Proof.* The proof follows in the standard construction outlined in Chapter 2, but we create a different hard instance that makes it difficult to identify the optimal arm by corrupting its reward. By Theorem 2.3, we obtain the following lower bound on the number of pulls of any suboptimal arm over the entire group of agents.

$$\liminf_{T\to\infty}\frac{\mathbb{E}[n_k(T)]}{\log(T)} \geq \frac{1}{\mathbb{D}_{\mathsf{KL}}(\nu_k,\nu_*)}. \tag{5.2}$$

Consider the problem where $P_\star = \mathcal{B}(\frac{1+\Delta}{2})$ for some $\Delta \in (0,1)$ and $P_k = \mathcal{B}(\frac{1}{2})\forall k \neq \star$. Now, given the true reward distributions $P_1, ..., P_K$, the adversary sets the arm reward distributions as follows. All suboptimal arms are left unperturbed. The reward distribution for the optimal arm is corrupted as $\nu_\star(1-\epsilon)P_\star + \epsilon\cdot Q_2$, where $Q_2 = \mathcal{B}(\frac{1-\Delta}{2})$. We therefore have that $\nu_k \sim \mathcal{B}(\frac{1}{2})$, and $\nu_\star \sim \mathcal{B}((1-\epsilon)\cdot\frac{1+\Delta}{2}+\epsilon\cdot\frac{1-\Delta}{2})$. Therefore, we have,

$$\mathbb{D}_{\mathsf{KL}}(\nu_k,\nu_\star) = \mathbb{D}_{\mathsf{KL}}\left(\mathcal{B}\left(\frac{1}{2}\right),\mathcal{B}\left((1-\epsilon)\cdot\frac{1+\Delta}{2}+(\epsilon)\cdot\frac{1-\Delta}{2}\right)\right) \leq \frac{\Delta^2(1-2\epsilon)^2}{3}. \tag{5.3}$$

The inequality follows from the fact that $\mathbb{D}_{\mathsf{KL}}(\theta_1,\theta_2) \leq \frac{(\theta_1-\theta_2)^2}{\theta_1(1-\theta_1)}$. Replacing this result in the regret, we obtain that

$$\liminf_{T\to\infty}\frac{\mathfrak{R}(T)}{\log(T)} = \sum_{k:\Delta_k>0}^{K}\Delta_k\mathbb{E}[n_k(T)] \geq \sum_{k:\Delta_k>0}^{K}\frac{3}{\Delta(1-2\epsilon)^2} = \sum_{k:\Delta_k>0}^{K}\frac{3\Delta}{(\Delta-2\epsilon)^2+4\epsilon\cdot\Delta(1-\Delta)}. \tag{5.4}$$

The second bound follows from the same analysis applied to Theorem 2.4. □

**Remark 5.3.** We see that regret becomes progressively worse as $\epsilon \to \frac{1}{2}$, consistent with intuition. A rewrite of the bound allows the comparison with the regret obtained by RobustFedUCB, where we see an additional $\mathcal{O}(\epsilon\Delta(1-\Delta))$ factor in the denominator. We believe that the lower bound is tight, and using alternative exploration strategies such as arm elimination can potentially eliminate this from the upper bound, as UCB relies on the concentration of the robust estimator, which can lead to excessive exploration.

**Remark 5.4** (Improving Network Dependence)**.** It can be seen that by using a similar leader-follower approach as described in Section 4.3 or utilizing $\mathcal{O}(K)-$sized messages with a FedUCB1-Best strategy with the alternate trimmed estimator will provide regret $\mathcal{O}(\psi(G_\gamma)\cdot\sum_{k:\Delta_k>0}\frac{\Delta_k\cdot\log(T)}{(\Delta_k-2\epsilon)^2}+M\gamma\cdot\sum_{k:\Delta_k>0}\Delta_k)$. We omit these proofs as they are straightforward combinations of the above approach with the analysis from Chapter 4.

## 5.2 Finite-Budgeted Adversarial Corruption in Communication

In this section, we assume that any reward when transmitted can be corrupted a maximum value of $\epsilon$, i.e., $\max_{t,n}|r_n(t)-\tilde{r}_n(t)|\leq\epsilon$ where $\tilde{r}_n(t)$ denotes the transmitted reward. Furthermore, we assume that the corruptions can be *adaptive*, i.e., can depend on the prior actions and rewards of each agent. This model includes natural settings, where messages can be corrupted during transmission, as well as *byzantine* communication (Dubey & Pentland, 2020d). If $\epsilon$ were known, we could then extend algorithms for misspecified bandits (Ghosh et al., 2017) to create a robust estimator and a subsequent UCB1-like algorithm that obtains a regret of $\mathcal{O}(\bar{\chi}(G_\gamma)K(\frac{\log T}{\Delta})+TNK\epsilon)$. However, this approach has two issues. First, $\epsilon$ is typically not known, and the dependence on $G_\gamma$ can be improved as well. We present an arm-elimination algorithm called CHARM (Adversarial Corruptions) that provides better guarantees on regret, without knowledge of $\epsilon$ in Algorithm 8.

The central motif in CHARM's design is to eliminate bad arms by an epoch-based exploration, an idea that has been successful in the past for adversarially-corrupted stochastic bandits (Lykouris et al., 2018; Gupta et al., 2019). The challenge, however, in a message-passing decentralized setting is two-fold. First, agents have different amounts of information based on their position in the network, and hence badly positioned agents in $G$ may be exploring for much larger periods. Secondly, communication between agents is delayed, and hence any agent naively incorporating stale observations may incur a heavy bias from delays. To ameliorate the first issue, we partition the group of agents into two sets - *exploring agents* ($\mathcal{I}$) and *imitating agents* ($\mathcal{V}\setminus\mathcal{I}$). The idea is to only allow well-positioned agents in $\mathcal{I}$ to direct the exploration strategy for their neighboring agents, and the rest simply imitate their exploration strategy. We select $\mathcal{I}$ as a minimal dominating set of $G_\gamma$, hence $|\mathcal{I}| = \psi(G_\gamma)$. Furthermore, since $\mathcal{V} \setminus \mathcal{I}$ is a vertex cover, this ensures that each *imitating* agent is connected (at distance at most $\gamma$) to at least one agent in $\mathcal{I}$. Next, observe that there are two sources of delay: first, any *imitating* agent must wait at most $\gamma$ trials to observe the latest action from its corresponding *exploring* agent, and second, each *exploring* agent must wait an additional $\gamma$ trials for the feedback from all of its imitating agents. We propose that each *exploring* agent run `UCB1` for $2\gamma$ rounds after each epoch of arm elimination, using *only* local pulls. This prevents a large bias due to these delays, at a small cost of $\mathcal{O}(\log\log T)$ suboptimal pulls.

**Theorem 5.4** (`CHARM` Regret)**.** *Algorithm 8 obtains, with probability at least $1-\delta$, cumulative group regret of*

$$\mathfrak{R}(T) = \mathcal{O}\left(KTN\gamma\epsilon + \psi(G_\gamma)\sum_{k:\Delta_k>0}\frac{\log T}{\Delta_k}\log\left(\frac{K\psi(G_\gamma)\log T}{\delta}\right) + N\Delta_k + \frac{N\log(N\gamma\log T)}{\Delta_k}\right).$$

*Proof.* We decompose the regret based on the independent set cover and epoch. Let $\mathcal{I} \subseteq \mathcal{V}$ be an independent set of $G_\gamma$ and $M_i$ be the number of epochs run for the subgraph covered by agent $i$. Observe that the total regret can be written as,

$$\mathfrak{R}(T) = \sum_{i\in\mathcal{I}}\left(\sum_{k=1}^{K}\sum_{t=1}^{T}\Delta_k\cdot\left(\mathbb{P}(A_i(t)=k) + \sum_{j\in\mathcal{N}(i)}\mathbb{P}(A_j(t)=k)\right)\right). \tag{5.5}$$

First, observe that $A_j(t) = A_i(t-d(i,j))$ for all $j\in\mathcal{N}(i)$ and all $t\in[d(i,j),T]$. Rearranging the above, we have,

$$\mathfrak{R}(T) \leqslant \sum_{i\in\mathcal{I}}\left(\sum_{k=1}^{K}\Delta_k\cdot\left(\sum_{t=1}^{T}\mathbb{P}(A_i(t)=k) + \sum_{j\in\mathcal{N}_\gamma(i)}\left(\sum_{t=1}^{T-d(i,j)}\mathbb{P}(A_i(t)=k)+d(i,j)\right)\right)\right) \tag{5.6}$$

$$\leqslant \sum_{i\in\mathcal{I}}\left(\sum_{k=1}^{K}\Delta_k\cdot|\mathcal{N}_i^+(G_\gamma)|\cdot\left(\sum_{t=1}^{T-\gamma}\mathbb{P}(A_i(t)=k)+\gamma\right)\right) \tag{5.7}$$

$$= \sum_{i\in\mathcal{I}}\left(|\mathcal{N}_i^+(G_\gamma)|\sum_{k=1}^{K}\Delta_k\left(\sum_{t=1}^{T-\gamma}\mathbb{P}(A_i(t)=k)\right)\right) + N\gamma\sum_{k=1}^{K}\Delta_k. \tag{5.8}$$

(5.9)

Now, observe that we run two algorithms in tandem for each independent set in $G_\gamma$. Let us split the total number of rounds of the game into epochs that run arm elimination and the intermittent periods of running `UCB1`. We denote the cumulative regret in the $i^{th}$ independent set from rounds $\gamma$ to $T$ as $\mathfrak{R}_i(T)$, and analyse it separately.

$$\mathfrak{R}_i(T) \leqslant |\mathcal{N}_i^+(G_\gamma)| \sum_{k=1}^{K} \left( \Delta_k \left( \sum_{t \leq T-\gamma : t \in \mathcal{M}_i} \mathbb{P}(A_i(t) = k) + \sum_{t \leq T-\gamma : t \notin \mathcal{M}_i} \mathbb{P}(A_i(t) = k) \right) \right). \tag{5.10}$$

Here $\mathcal{M}_i$ denotes the rounds in which arm elimination is played in the independent set $i$. Since each `UCB1` period after each epoch is of length $2\gamma$, we have at most $2\gamma M_i$ rounds of isolated `UCB1`. We analyse the second term in the bound first. By the standard analysis of the `UCB1` algorithm (Auer & Ortner, 2007), we have that the leader agent incurs $\mathcal{O}(K \log T/\Delta)$ regret. We therefore have,

$$|\mathcal{N}_i^+(G_\gamma)| \sum_{k=1}^{K} \left( \Delta_k \left( \sum_{t \notin \mathcal{M}_i} \mathbb{P}(A_i(t) = k) \right) \right) \leqslant |\mathcal{N}_i^+(G_\gamma)| \cdot \sum_{k=1}^{K} \left( \left(1 + \frac{\pi^2}{3}\right) \Delta_k + \frac{4\log(2\gamma M_i)}{\Delta_k} \right).$$

Now, we analyse the first term in the regret bound. By Theorem 5.5, we have that with probability at least $1-\delta$ simultaneously for each independent group corresponding to agent $i \in \mathcal{I}$,

$$\sum_{k=1}^{K} \left( \Delta_k \left( \sum_{m \in \mathcal{M}_i} \mathbb{E}\left[ n_k^i(m) \right] \right) \right) \leqslant \mathcal{O}\left( \gamma\epsilon \cdot KT |\mathcal{N}_i^+(G_\gamma)| + \sum_{k:\Delta_k > 0} \frac{\log T}{\Delta_k} \log\left( \frac{K\alpha(G_\gamma)}{\delta} \log T \right) \right).$$

Summing over each leader agent, we have that with probability at least $1-\delta$,

$$\sum_{i \in \mathcal{I}} \sum_{k=1}^{K} \left( \Delta_k \left( \sum_{m \in \mathcal{M}_i} \mathbb{E}\left[ n_k^i(m) \right] \right) \right) \leqslant \mathcal{O}\left( \gamma\epsilon \cdot KTN + \sum_{k:\Delta_k > 0} \frac{\log T}{\Delta_k} \log\left( \frac{K\alpha(G_\gamma)}{\delta} \log T \right) \right).$$

Next, observe that for all $i$, $|\mathcal{M}_i| \leq \log(MT)$ by Lemma 5.1. Replacing this result in the `UCB1` regret for each leader, and summing over all $i \in \mathcal{I}$, we have,

$$\mathfrak{R}(T) = \mathcal{O}\left( \gamma\epsilon \cdot KTN + \sum_{k:\Delta_k > 0} \alpha(G_\gamma) \frac{\log T}{\Delta_k} \log\left( \frac{K\alpha(G_\gamma) \log T}{\delta} \right) + N\Delta_k + \frac{N \log(N\gamma \log T)}{\Delta_k} \right).$$

□

**Lemma 5.1.** *For any leader $i$, let $L^i(m)$ denote the length of the $m^{th}$ epoch of arm elimination. Then, we have that $L^i_m$ satisfies,*

$$2^{2m-2}\lambda \leq L^i(m) \leq K2^{2m-2}\lambda.$$

*Furthermore, the number of arm elimination epochs for agent $i$ satisfies $M_i \leq \log_2(T - 2\gamma)$.*

*Proof.* For any leader $i$, let $\hat{k}$ be the optimal arm under $r^i(m)$, therefore $r^i_\star(m) - r^i_{\hat{k}}(m) \leq 0$ and therefore $\Delta^m_{\hat{k}} = 2^{-m}$, and therefore $L^i_{m+1} \geq n^{m+1}_{\hat{k}} = \lambda(\Delta^m_{\hat{k}})^{-2} \geq 2^{2m}\lambda$. Next, observe that $\Delta^m_k \geq 2^{-m}$ for each arm $k$, and therefore $n^{m+1}_k \leq 2^{2m}\lambda$, giving the upper bound.

For the second part, observe that $\sum_{m=1}^{M_i} L^i_m \leq T - 2\gamma M_i \leq T - 2\gamma$, and that $L^i_m \geq \frac{2^{2m-2}\lambda}{|\mathcal{N}^+_i(G_\gamma)|}$. Summing over $m \in [M_i]$ and taking the logarithm provides us with the result. □

**Lemma 5.2.** *Denote $\mathcal{E}$ to be the event for which,*

$$\left\{ \forall m, i, k, \left| r^i_k(m) - \mu_k \right| \leq 2\gamma\epsilon + \frac{\Delta^i_k(m-1)}{16} \bigwedge \sum_{\substack{t \in \mathcal{M}_i(m) \\ j \in \mathcal{N}^+_i(G_\gamma)}} X^j_k(t + d(i,j)) \leq 2n^i_k(m) \right\}$$

*Then, we have that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$.*

*Proof.* Recall that at each step in the epoch, the leader agent picks an arm $k$ with probability $p^i_k(m) = \frac{n^i_k(m)}{L^i(m)}$, and let $X^j_k(t)$ denote whether agent $j$ picks arm $k$ at time $t$. Let $C_{j \to i}(t) = \tilde{r}_{j \to i}(t) - r_j(t)$ denote the corruption in the transmitted reward from agent $j$ when it reaches agent $i$, and $\mathcal{M}_i(m) = [T^i_{m-1} + 1, ..., T^i_m]$ denote the $L^i(m)$ steps in the $m^{th}$ epoch for the arm elimination algorithm run by the leader $i$. We then have,

$$r^i_k(m) = \frac{1}{n^i_k(m)} \left( \sum_{\substack{t \in \mathcal{M}_i(m) \\ j \in \mathcal{N}^+_i(G_\gamma)}} X^j_k(t + d(i,j)) \cdot \left( r_j(t + d(i,j)) + C_{j \to i}(t + d(i,j)) \right) \right)$$

For simplicity, let

$$A^i_k(m) = \sum_{\substack{t \in \mathcal{M}_i(m) \\ j \in \mathcal{N}^+_i(G_\gamma)}} X^j_k(t + d(i,j)) \cdot r_j(t + d(i,j)), B^i_k(m) = \sum_{\substack{t \in \mathcal{M}_i(m) \\ j \in \mathcal{N}^+_i(G_\gamma)}} X^j_k(t + d(i,j)) \cdot C_{j \to i}(t + d(i,j)).$$

We can bound the first summation by a multiplicative version of the Chernoff-Hoeffding bound (**?**) as each $r_j$ is bounded within $[0,1]$ and $X_k^i$ is a random variable in $\{0,1\}$ with mean $p_k^i(m)L^i(m)\mu_k \leq n_k^i(m)$. We obtain that with probability at least $1-\beta/2$,

$$\left|\frac{A_k^i(m)}{n_k^i(m)} - \mu_i\right| \leq \sqrt{\frac{3\log(\frac{4}{\beta})}{n_k^i(m)}}.$$

To bound the second term, we must construct a filtration that ensures that the corruption is measurable. For the set $\mathcal{N}_i^+(G_\gamma)$, consider an order $\sigma$ of the $N$ agents, such that $\sigma[1]=i$, followed by the agents at distance 1 from $i$, then the agents at distance 2, and so on until distance $\gamma$, and next consider the ordering $\{\tilde{r}_\tau\}_{\tau=1}^{|\mathcal{N}_i^+(G_\gamma)|t}$ of the rewards generated by all agents within $\mathcal{M}_i(m)$ where $\tilde{r}_\tau$ is the reward obtained by agent $j = (\sigma(\tau) \mod |\mathcal{N}_i^+(G_\gamma)|)$ during the round $\lfloor \frac{\tau}{|\mathcal{N}_i^+(G_\gamma)|} \rfloor + d(i,j)$, and similarly consider an identical ordering of the pulled arms $\{\widetilde{X}_\tau\}_{\tau=1}^{|\mathcal{N}_i^+(G_\gamma)|t}$. Now consider the filtration $\{\mathcal{F}_t\}_{t=1}^{T|\mathcal{N}_i^+(G_\gamma)|}$ generated by the two stochastic processes of $\tilde{r}$ and $\widetilde{X}$. Clearly, the corruption $C_{\sigma(j)\to i}(t)$ is deterministic conditioned on $\mathcal{F}_{t-1}$. Moreover, we have that the pulled arm satisfies, for all $\tau \in [|\mathcal{N}_i^+(G_\gamma)|t]$ that $\mathbb{E}[\tilde{X}_\tau|\mathcal{F}_{\tau-1}] = p_k^i(m)$. Furthermore, since the corruption in each round is bounded and deterministic, we have that the sequence $Z_\tau = (\widetilde{X}_\tau - p_k^i(m)) \cdot \widetilde{C}_\tau$ (where $\widetilde{C}_\tau$ is the corresponding ordering of corruptions) is a martingale difference sequence with respect to $\{\mathcal{F}_\tau\}_{\tau=1}^T$. Now, consider the slice of $[|\mathcal{N}_i^+(G_\gamma)|t]$ that is present within $B_k^i(m)$, and let the corresponding indices be given by the set $\widetilde{\mathcal{M}}_i(m)$. Using the fact that the observed rewards are bounded, we have that,

$$\sum_{\tau\in\widetilde{\mathcal{M}}_i(m)} \mathbb{E}[Z_\tau^2|\mathcal{F}_{\tau-1}] \leq \sum_{\tau\in\widetilde{\mathcal{M}}_i(m)} |\widetilde{C}_\tau| \cdot \mathbb{V}(Z_\tau) \leq p_k^i(m) \cdot \sum_{\tau\in\widetilde{\mathcal{M}}_i(m)} \widetilde{C}_\tau \leq \gamma C L^i(m).$$

We then have by Freedman's inequality that with probability at least $1-\frac{\beta}{4}$,

$$\frac{B_k^i(m)}{n_k^i(m)} \leq \frac{p_k^i(m)}{n_k^i(m)}\left(\sum_{\tau\in\widetilde{\mathcal{M}}_i(m)} \widetilde{C}_\tau + \frac{\gamma C L^i(m) + \log(4/\beta)}{n_k^i(m)}\right) \leq 2\gamma\epsilon + \sqrt{\frac{\log(4/\beta)}{16 n_k^i(m)}}.$$

The last inequality follows from the fact that $n_k^i(m) \geq \lambda \geq 16\ln(4/\beta)$. With the same probability, we can derive a bound for the other tail. Now, observe that since each $X_k^i$ is a random variable with mean $p_k^i$, we have by the multiplicative Chernoff-Hoeffding bound (Lemma **??**) that the probability that the sum of $L^i(m)$ i.i.d. bernoulli trials with mean $p_k^i(m)$

is greater than $2p_k^i(m) \cdot L^i(m) = 2n_k^i(m)$ is at most $2\exp(-n_k^i(m)/3) \leq 2\exp(-\lambda/3) \leq \beta$.

To conclude the proof, we apply each of the above bounds with $\beta = \frac{\delta}{2K\alpha(G_\gamma)\log T}$ to each epoch and arm. Observe that $\beta \geq 4\exp\left(-\frac{\lambda}{16}\right)$. Now, since $\log(4/\beta) = \lambda/(32)^2$ we have that,

$$\mathbb{P}\left(\left|r_k^i(m) - \mu_k\right| \geq 2\gamma\epsilon + \frac{\Delta_k^i(m-1)}{16} \bigwedge \sum_{\substack{t\in\mathcal{M}_i(m)\\ j\in\mathcal{N}_i^+(G_\gamma)}} X_k^j(t + d(i,j)) \geq 2n_k^i(m)\right) \leq \frac{\delta}{2K\alpha(G_\gamma)\log T}.$$

The proof concludes by a union bound over all epochs, arms and agents in $\mathcal{I}$. □

**Lemma 5.3.** *If the event $\mathcal{E}$ (Lemma 5.2) occurs then for each $i \in \mathcal{I}, m \in \mathcal{M}_i$,*

$$-2\gamma\epsilon - \frac{\Delta_\star^i(m-1)}{8} \leq r_\star^i(m) - \mu_\star \leq 2\gamma\epsilon.$$

*Proof.* Observe that $r_\star^i(m) \geq r_{k^\star}^i(m) - \frac{1}{16}\Delta_{k^\star}^i(m-1)$. This fact coupled with the fact that $\mathcal{E}$ holds provides the lower bound. The upper bound is obtained by observing that,

$$r_\star^i(m) \leq \max_i \left\{\mu_i + 2\gamma\epsilon + \frac{\Delta_k^i(m-1)}{16} - \frac{\Delta_k^i(m-1)}{16}\right\} \leq \mu_\star + 2\gamma\epsilon.$$

□

**Lemma 5.4.** *If the event $\mathcal{E}$ (Lemma 5.2) occurs then for each $i \in \mathcal{I}, m \in \mathcal{M}_i$,*

$$\Delta_k^i(m) \geq \frac{\Delta_k}{2} - 6\gamma\epsilon\sum_{n=1}^{m} 8^{n-m} - \frac{3}{4}2^{-m}.$$

*Proof.* We first bound $\Delta_k^i(m) \leq 2(\Delta_k + 2^{-m} + 2\gamma\epsilon \cdot \sum_{n=1}^m 8^{n-m})$ under $\mathcal{E}$ by induction. Observe that when $m = 1$ we have that trivially $\Delta_k^i(1) \leq 1 \leq 2 \cdot 2^{-1}$. Now, if the bound holds for epoch $m-1$ for any agent, we have by Lemma 5.3,

$$r_\star^i(m) - r_k^i(m) = r_\star^i(m) - \mu_\star + \mu_\star - \mu_k + \mu_k - r_k^i(m) \leq 4\gamma\epsilon + \Delta_k + \frac{\Delta_k^i(m-1)}{16}.$$

Replacing the induction hypothesis in the upper bound, we have,

$$\begin{aligned} r^i_\star(m) - r^i_k(m) &\leq 4\gamma\epsilon + \Delta_k + \frac{1}{8}\left(\Delta_k + 2^{-(m-1)} + 2\gamma\epsilon \cdot \sum_{n=1}^{m-1} 8^{n-m+1}\right) \\ &\leq 2(\Delta_k + 2^{-m} + 2\gamma\epsilon \cdot \sum_{n=1}^{m} 8^{n-m}). \end{aligned}$$

Now, we bound the gaps as,

$$\Delta^i_k(m) \geq r^i_\star(m) - r^i_k(m) \geq \Delta_k - 4\gamma\epsilon - \left(\frac{\Delta^i_{k^\star}(m-1)}{8} - \frac{\Delta^i_k(m-1)}{16}\right).$$

The last inequality follows from Lemma 5.3 and the event $\mathcal{E}$. Replacing the bound from induction we obtain,

$$\begin{aligned} \Delta^i_k(m) &\geq \Delta_k - 4\gamma\epsilon - \left(\frac{6\gamma\epsilon}{8}\sum_{n=1}^{m} 2^{n-m} + \frac{3}{8}2^{-(m-1)} + \frac{\Delta_k}{8}\right) \\ &\geq \frac{\Delta_k}{2} - 6\gamma\epsilon\sum_{n=1}^{m} 8^{n-m} - \frac{3}{4}2^{-m}. \end{aligned}$$

□

**Theorem 5.5.** *The cumulative regret for all agents within each independent set corresponding to leader $i \in \mathcal{I}$ satisfy simultaneously, with probability at least $1-\delta$,*

$$\sum_{m=1}^{\mathcal{M}_i}\sum_{k=1}^{K}\Delta_k\mathbb{E}[n^i_k(m)] = \mathcal{O}\left(\log\left(\frac{K\alpha(G_\gamma)}{\delta}\log(T)\right)\log(T)\left(\sum_{k=1}^{K}\frac{1}{\Delta_k}\right) + \gamma\epsilon\cdot KT\cdot|\mathcal{N}^+_i(G_\gamma)|\right).$$

*Proof.* We bound the regret in each epoch $m \in \mathcal{M}_i$ for each arm $k \neq k^\star$ based on three cases.

**Case 1**. $0 \leq \Delta_k \leq 4/2^m$: We have that $n^i_k(m) \leq \lambda 2^{2(m-1)}$ since $\Delta^i_k(m-1) \geq 2^{m-1}$, and hence,

$$\Delta_k\mathbb{E}[n^i_k(m)] \leq \frac{4\lambda}{\Delta_k^2}\cdot\Delta_k = 4\lambda\cdot\frac{1}{\Delta_k}.$$

**Case 2**. $\Delta_k > 4/2^m$ and $\gamma\epsilon\sum_{n=1}^{m} 8^{n-m} \leq \Delta_k/64$: We have by Lemma 5.4,

$$\Delta^i_k(m) \geq \frac{\Delta_k}{2} - 6\gamma\epsilon\sum_{n=1}^{m} 8^{n-m} - \frac{3}{4}2^{-m} \geq \Delta_k\left(\frac{1}{2} - \frac{3}{32} - \frac{3}{8}\right) = \frac{\Delta_k}{32}.$$

Therefore, we have that $n_k^i(m) \leq \frac{1024\lambda}{\Delta_k^2}$, and hence the regret is,

$$\Delta_k \mathbb{E}[n_k^i(m)] \leq \frac{1024\lambda}{\Delta_k^2} \cdot \Delta_k = 1024\lambda \cdot \frac{1}{\Delta_k}.$$

**Case 3**. $\Delta_k > 4/2^m$ and $\gamma\epsilon \sum_{n=1}^m 8^{n-m} > \Delta_k/64$: This implies that $\Delta_k \leq 64\gamma\epsilon \cdot \sum_{n=1}^m 8^{n-m}$. Therefore,

$$\begin{aligned}\Delta_k \mathbb{E}[n_k^i(m)] &\leq 64\lambda\gamma\epsilon \left(\sum_{n=1}^m 8^{n-m}\right) \cdot 2^{2(m-1)} \\ &\leq 64\lambda\gamma\epsilon \left(\frac{8^{m+1}}{7}\right) \cdot \frac{2^{2(m-1)}}{2^{3m}} \\ &\leq \frac{512}{7}\gamma\epsilon \cdot L^i(m).\end{aligned}$$

Here the last inequality follows from Lemma 5.1. Putting it together and summing over all epochs and arms, we have with probability at least $1-\delta$ simultaneously for each $i \in \mathcal{I}$,

$$\begin{aligned}\sum_{m=1}^{\mathcal{M}_i}\sum_{k=1}^{K} \Delta_k \mathbb{E}[n_k^i(m)] \leq 1024^2 \log\left(\frac{8K\alpha(G_\gamma)}{\delta}\log(T)\right)\log(T)\left(\sum_{k=1}^K \frac{1}{\Delta_k}\right) \\ + 74\gamma\epsilon \cdot KT \cdot |\mathcal{N}_i^+(G_\gamma)|.\end{aligned}$$

□

**Remark 5.5** (Regret Optimality)**.** Theorem 5.4 demonstrates a trade-off between communication density and the adversarial error, as seen by the first two terms in the regret bound. The first term ($KTN\gamma\epsilon$) is a bound on the cumulative error introduced due to message-passing, which is increasing in $\gamma$, whereas the second term denotes the logarithmic regret due to exploration, where $\psi(G_\gamma)$ decreases as $\gamma$ increases: for $\gamma = d_\star(G), \psi(G_\gamma) = 1$, matching the lower bound in Dubey & Pentland (2020a). This too, is expected, as fewer exploring agents are needed with a higher communication budget. Furthermore, we conjecture that the first term is optimal (in terms of $T$, up to graphical constants): a linear lower bound has been demonstrated for the single-agent setting in Lykouris et al. (2018).

**Remark 5.6** (Computational complexity)**.** While the dominating set problem is known to be NP-complete (Karp, 1972), the problem admits a polynomial-time approximation scheme (PTAS) (Crescenzi et al., 1995) for certain graphs, for which our bounds hold exactly. However, CHARM can work on any dominating set of size $n$, and suffer regret of $\widetilde{\mathcal{O}}(KTN\gamma\epsilon + n\sum_{k>1}\frac{\log T}{\Delta_k})$[1].

[1] The $\widetilde{\mathcal{O}}$ notation ignores absolute constants and $\log\log(\cdot)$ factors in $T$.

## 5.3 Algorithm Pseudocode

**Algorithm 7** Robust-FedUCB($\epsilon$)

```
Input: Agent $m$, arms $k \in [K]$, mean estimator $\hat{\mu}_R(n, \delta)$
Set $S_k^m = \phi \; \forall k \in [K]$, $Q_m(t) = \phi$.
for For $t \in [T]$ do
    if $t \leq K$ then
        $a_m(t) = t$.
    else
        for Arm $k \in [K]$ do
            $\hat{\mu}_k^{(m)} = \hat{\mu}_R(S_k^m, 1/t^2)$.
            $\text{UCB}_k^{(m)}(t) = \sigma\sqrt{\epsilon} + \sqrt{\frac{\sigma \ln(1/\delta)}{|S_k^m(t)|}}$.
        end for
        $a_m(t) = \arg\max_{k \in [K]} \left\{ \hat{\mu}_k^{(m)}(t) + \text{UCB}_k^{(m)}(t) \right\}$.
    end if
    $x_m(t) = \textsc{Pull}(a_m(t))$.
    $S_{A_{m,t}}^m = S_{A_{m,t}}^m \cup \{x_m(t)\}$
    $Q_m(t)$ = $\textsc{PruneDeadMessages}$($Q_m(t)$).
    $Q_m(t) = Q_m(t) \cup \{\langle m, t, \gamma, a_m(t), x_m(t) \rangle\}$.
    Set $l = l - 1 \; \forall \langle m', t', a', x' \rangle$ in $Q_m(t)$.
    for Each neighbor $m'$ in $\mathcal{N}_1(m)$ do
        $\textsc{SendMessages}(m, m', Q_m(t))$.
    end for
    $Q_m(t+1) = \phi$.
    for Each neighbor $m'$ in $\mathcal{N}_1(m)$ do
        $Q'$ =$\textsc{ReceiveMessages}(m', m)$
        $Q_m(t+1)$ = $Q_m(t+1) \cup Q'$.
    end for
    for $\langle m', t', a', x' \rangle \; \in Q_m(t+1)$ do
        $S_{a'}^m = S_{a'}^m \cup \{x'\}$.
    end for
end for
```

**Algorithm 8** `CHARM`: Cooperative Hybrid Arm Elimination

```
**Parameters**. Confidence $\delta \in (0,1)$, horizon $T$, graph $G$ with exploration set $\mathcal{I} \subseteq \mathcal{V}$.
Initialize $T_i(0) = K \forall i \in \mathcal{I} \lambda = 1024 \log\left(\frac{8K\alpha(G_\gamma)}{\delta} \log_2 T\right)$ and $\Delta_k^i(0) = 1, \forall\, k \in [K]$ and $i \in \mathcal{I}$.
for each subgraph $\mathcal{N}_i^+(G_\gamma)$ where $i \in \mathcal{I}$ do
  for $t = 1, ..., K$, each agent $j \in \mathcal{N}_i^+(G_\gamma)$ do
    Play arm $K$ and get reward $r_j(t)$.
  end for
  for epoch $m_i = 1, 2, ...,$ do
    Set $n_k^i(m) = \lambda(\Delta_k^i(m-1))^{-2} \forall k \in [K]$.
    $N_i(m) = \sum_k n_k^i(m)$ and $T_i(m) = T_i(m-1) + N_i(m) + 2\gamma$.
    for agent $j \in \mathcal{N}_i^+(G_\gamma)$ do
      for $t = T_i(m_i - 1)$ to $s = T_i(m_i - 1) + 2\gamma$ do
        if $j \neq i$ then
          if $t \leq K + d(i,j)$ then
            Pull random arm.
          else
            Pull $A_j(t) = A_i(t - d(i,j))$ and get reward $r_j(t)$.
          end if
        else
          Pull $A_j(t) = \texttt{UCB1}(t)$
        end if
      end for
      for $t = T_i(m_i - 1) + 2\gamma$ to $T_i(m_i)$ do
        if $j \neq i$ then
          Pull $A_j(t) = A_i(t - d(i,j))$ and get reward $r_j(t)$.
        else
          Pull an arm $A_i(t) = k \in [K]$ with probability $n_k^i(m)/n_k(m)$.
        end if
      end for
    end for
  end for
end for
```

# Part II

# Contextual Bandits

# Chapter 6

# Differentially-Private Federated Contextual Bandits

## 6.1 Introduction

The previous part was concerned primarily with stochastic multi-armed bandits and various communication and environmental constraints. In this part we examine bandit problems with *contexts*, e.g., linear contextual bandits and Gaussian process bandits. The fundamental difference between the *contextual* and *non-contextual* bandit settings is that contextual bandit problems have *time-varying* decision sets (which can potentially depend on the history of the agent as well), in contrast to the previous part, where the set of arms is fixed. This poses additional challenges in the federated setting, where more information about the decision sets and sophisticated estimators are required to obtain good performance.

Specifically, the contextual bandit problem is a very interesting candidate for private methods, since in most application areas such as online recommender systems or medicine, the involved contexts and rewards *both* typically contain sensitive user information (see, e.g., Malekzadeh et al. (2019) for more details on applications). There is an increasing body of work on online learning and multi-armed bandits in cooperative settings (Dubey & Pentland, 2020c; Landgren et al., 2016a; Martínez-Rubio et al., 2019), and private single-agent learning (Shariff & Sheffet, 2018; Malekzadeh et al., 2019), but methods for private federated bandit learning are still elusive, despite their immediate applicability.

In this chapter, we study the federated contextual bandit problem under constraints

Table 6.1: Comparison of communication complexity and regret speed-up for FedLinUCB.

| Algorithm | Threshold $S$ | Communication | Regret Speed-up |
|---|---|---|---|
| FedLinUCB<br>(Distributed) | $\infty$ | $0$ | $1$ |
| | $\mathcal{O}(1)$ | $\mathcal{O}(M\sqrt{dT\log(MT)})$ | $\mathcal{O}(\sqrt{M})$ |
| | $\mathcal{O}\left(\frac{T}{dM^2\log(MT)}\right)$ | $\mathcal{O}(dM\log(MT))$ | $\mathcal{O}\left(\sqrt{\frac{M}{\log(MT)}}\right)$ |
| | $\mathcal{O}\left(\frac{T\log(MT)}{dM^2}\right)$ | $\mathcal{O}(dM^3)$ | $\mathcal{O}\left(\frac{\sqrt{M}}{\log(MT)}\right)$ |
| FedLinUCB<br>(Decentralized) | $c > 1$ | $\mathcal{O}\left(d^2M(\frac{\log(MT)}{\log(c)})\right)$ | $\mathcal{O}\left(\sqrt{\frac{M}{c}}\right)$ |
| | $\mathcal{O}\left(\log(MT)\right)$ | $\mathcal{O}(d^3M^3)$ | $o(1)$ |

of differential privacy. We consider both multi-agent paradigms of distributed and decentralized learning separately, in contrast to the earlier chapters. We provide a rigorous formulation of $(\varepsilon, \delta)$-differential privacy in the federated contextual bandit, and present two variants of FedLinUCB, a no-regret algorithm that ensures that each agent is private with respect to the data from all other agents, and provides a tunable parametric control over the communication budget.

We demonstrate that FedLinUCB obtains a group regret of $\widetilde{\mathcal{O}}\left(\left(\sqrt{\frac{d^{3/2}}{\varepsilon}}+d\right)\sqrt{MT}\right)$[1] when run on a distributed setting over $M$ agents and $T$ rounds, where $\varepsilon$ denotes the privacy budget. Our approach relies on a new self-normalized inequality that holds simultaneously for all agents even when the communication protocol is *data-dependent*, as it is in our case. Furthermore, this rate is achieved with only $\mathcal{O}(dM\log^2(MT)\log(K))$ bits of communication, and can be tuned based on the threshold parameter to even a constant value with some degradation in regret.

For the decentralized peer-to-peer communication setting, we demonstrate that FedLinUCB obtains a group pseudoregret of $\widetilde{\mathcal{O}}\left(\left(\sqrt{\frac{d^{3/2}}{\varepsilon}}+d\right)\sqrt{\bar{\chi}(G_\gamma)\cdot MT}\right)$ where, as in the earlier chapters $\bar{\chi}(G_\gamma)$ denotes the minimal clique covering number of the $\gamma$ power graph of $G$, where $\gamma$ specifies how long messages persist in the network. In contrast to the previous analyses, the analysis for FedLinUCB follows a different structure, where we use a decentralized "broadcast" mechanism to bound the regret.

[1] The $\widetilde{\mathcal{O}}(\cdot)$ ignores polylogarithmic factors and constants.

We assume: *(a)* bounded action set: $\forall i, t,\ \|\mathbf{x}_i(t)\| \leq L$, *(b)* bounded mean reward: $\forall \mathbf{x}, \langle \boldsymbol{\theta}_\star, \mathbf{x} \rangle \leq 1$, *(c)* bounded target parameter: $\|\boldsymbol{\theta}_\star\| \leq S$, *(d)* sub-Gaussian rewards: $\forall i, t,\ \eta_i(t)$ is $\sigma$-sub-Gaussian, *(e)* $\forall i, t\ \mathcal{D}_i(t)$ is compact and finite, *(f)* bounded reward: $\forall i, t,\ |y_i(t)| \leq B$.[3]

Figure 6-1: The assumptions on the environment in this chapter.

## 6.2 Problem Setup

**Federated Contextual Bandit**. This is an extension of the linear contextual bandit (Li et al., 2010; Abbasi-Yadkori et al., 2011) involving a set of $M$ agents. At every trial $t \in [T]$, each agent $i \in [M]$ is presented with a *decision set* $\mathcal{D}_i(t) \subset \mathbb{R}^d$ from which it selects an action $\mathbf{x}_i(t) \in \mathbb{R}^d$. It then obtains a reward $y_i(t) = \mathbf{x}_i(t)^\top \boldsymbol{\theta}_\star + \eta_i(t)$ where $\boldsymbol{\theta}_\star \in \mathbb{R}^d$ is an unknown (but fixed) parameter and $\eta_i(t)$ is a noise parameter sampled i.i.d. every trial for every agent. The objective of the agents is to minimize the cumulative *group pseudoregret*:[2]

$$\mathfrak{R}(T) = \sum_{i=1}^{M} \sum_{t=1}^{T} \langle \mathbf{x}_i^\star(t) - \mathbf{x}_i(t), \boldsymbol{\theta}_\star \rangle,$$

Where $\mathbf{x}_i^\star(t) = \arg\max_{\mathbf{x} \in \mathcal{D}_i(t)} \langle \mathbf{x}, \boldsymbol{\theta}_\star \rangle$ is the optimal action. In the single-agent setting, the best regret bound obtained scales as $\widetilde{\mathcal{O}}(d\sqrt{T})$ obtained by the upper confidence bound (UCB) linear algorithm LinUCB (Abbasi-Yadkori et al., 2011; Li et al., 2010; Auer et al., 2002a). In the *non-private* distributed contextual bandit setting, a group pseudoregret of $\widetilde{\mathcal{O}}(d\sqrt{MT})$ has been achieved (Wang et al., 2019a, 2020c), which matches the regret obtained for a single LinUCB agent pulling $MT$ arms. We will demonstrate that the baseline of a single-agent pulling $MT$ arms does provide a lower bound on the regret for the distributed setting.

**Differential Privacy**. The contextual bandit problem involves two sets of variables that any agent must private to the other participating agents – the available decision sets $(\mathcal{D}_i(t))_{t \in [T]}$ and observed rewards $(y_i(t))_{t \in [T]}$. The adversary model assumed here is to prevent any two colluding agents $j$ and $k$ to obtain non-private information about any specific element in agent $i$'s history. That is, we assume that each agent is a trusted entity that interacts with a new user at each instant $t$. Therefore, the context set $(\mathcal{D}_i(t))$ and outcome $(y_{i,t})$ are sensitive variables that the user trusts only with the agent $i$. Hence, we wish to

[2] The *pseudoregret* is an expectation (over the randomness of $\eta_i(t)$) of the stochastic quantity *regret*, and is more amenable to high-probability bounds. However, a bound over the pseudoregret can also bound the regret with high probability, e.g., by a Hoeffding concentration (see, e.g., (Valko et al., 2013)).

keep $(\mathcal{D}_i(t))_{t\in[T]}$ private. However, the agent only stores the chosen actions $(\mathbf{x}_i(t))_{t\in[T]}$ (and not all of $\mathcal{D}_i(t)$), and hence making our technique differentially private with respect to $((\mathbf{x}_i(t), y_i(t)))_{t\in[T]}$ will suffice. We first denote two sequences $S_i = ((\mathbf{x}_i(t), y_i(t)))_{t\in[T]}$ and $S'_i = \left((\mathbf{x}'_{i,t}, y'_{i,t})\right)_{t\in[T]}$ as $t-$neighbors if for each $t' \neq t$, $(\mathbf{x}_i(t), y_i(t)) = (\mathbf{x}'_{i,t'}, y'_{i,t})$. We can now provide the formal definition for federated differential privacy:

**Definition 6.1** (Federated Differential Privacy)**.** *In a federated learning setting with $M \geq 2$ agents, a randomized multiagent contextual bandit algorithm $A = (A_i)_{i=1}^M$ is $(\varepsilon, \delta, M)$-federated differentially private under continual multi-agent observation if for any $i, j$ s.t. $i \neq j$, any $t$ and set of sequences $\mathbf{S}_i = (S_k)_{k=1}^M$ and $\mathbf{S}'_i = (S_k)_{k=1, k\neq i}^M \cup S'_i$ such that $S_i$ and $S'_i$ are $t$-neighboring, and any subset of actions $\mathcal{S}_j \subset \mathcal{D}_{j,1} \times \mathcal{D}_{j,2} \times ... \times \mathcal{D}_{j,T}$ of actions, it holds that:*

$$\mathbb{P}\left(A_j\left(\mathbf{S}_i\right) \in \mathcal{S}_j\right) \leq e^{\varepsilon} \cdot \mathbb{P}\left(A_j\left(\mathbf{S}'_i\right) \in \mathcal{S}_j\right) + \delta.$$

Our notion of federated differential privacy is formalizing the standard intuition that "the action chosen by any agent must be sufficiently impervious (in probability) to any single $(\mathbf{x}, y)$ pair from any other agent". Here, we essentially lift the definition of joint differential privacy (Shariff & Sheffet, 2018) from the individual $(\mathbf{x}, y)$ level to the entire history $(\mathbf{x}_t, y_t)_t$ for each participating agent. Note that our definition in its current form does not require each algorithm to be private with respect to its own history, but only the histories belonging to other agents, i.e., each agent can be trusted with its own data. This setting can also be understood as requiring all *outgoing communication* from any agent to be *locally differentially private* (Yang et al., 2020a) to the personal history $(\mathbf{x}_t, y_t)_t$. We can alternatively relax this assumption and assume that the agent cannot be trusted with its own history, in which case the notion of joint or local DP at the individual level (i.e., $(\mathbf{x}_t, y_t)$) must be considered, as done in Yang et al. (2020a).

The same guarantee can be obtained if each agent $A_i$ is $(\varepsilon, \delta)$-differentially private with respect to any other agent $j$'s observations, for all $j$. A composition argument (Dwork, 2011)[4] over all $M$ agents would provide $(\sqrt{2M\log(1/\delta')}\varepsilon + M\varepsilon(e^{\varepsilon}-1), M\delta + \delta')$-differential privacy with respect to the overall sequence. To keep the notation simple we adopt the

[4] Under the stronger assumption that each agent interacts with a completely different set of individuals, we do not need to invoke the composition theorem (as $\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, ..., \mathbf{x}_{M,t}$ are independent for each $t$). However, in the case that one individual could potentially interact simultaneously with all agents, this is not true (e.g., when for some $t$, $\mathcal{D}_i(t) = \mathcal{D}_{j,t}\ \forall i, j$) and we must invoke the $k$-fold composition Theorem (Dwork & Smith, 2010) to ensure privacy.

$(\varepsilon, \delta, M)$ format.

## 6.3 Federated LinUCB with Differential Privacy

In this section, we introduce our algorithm for federated learning with differential privacy. For the remainder of this section, for exposition, we consider the single-agent setting and drop an additional index subscript we use in the actual algorithm (e.g., we refer to the action at time $t$ as $\mathbf{x}_t$ and not $\mathbf{x}_i(t)$ for agent $i$). We build on the celebrated LinUCB algorithm, an application of the optimism heuristic to the linear bandit case (Li et al., 2010; Abbasi-Yadkori et al., 2011), designed for the single-agent problem. The central idea of the algorithm is, at every round $t$, to construct a *confidence set* $\mathcal{E}_t$ that contains $\boldsymbol{\theta}_\star$ with high probability, followed by computing an upper confidence bound on the reward of each action within the decision set $\mathcal{D}_t$, and finally selecting the action with the largest UCB, i.e., $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}_t} (\max_{\boldsymbol{\theta}\in\mathcal{E}_t} \langle \mathbf{x}, \boldsymbol{\theta} \rangle)$. The confidence set is an ellipsoid centered on the regularized linear regression estimate (for $\mathbf{X}_{<t} = \left[\mathbf{x}_1^\top \; \mathbf{x}_2^\top \; ... \; \mathbf{x}_{t-1}^\top\right]^\top$ and $\mathbf{y}_{<t} = \left[y_1 \; y_2 \; ... \; y_{t-1}\right]^\top$):

$$\mathcal{E}_t := \left\{ \boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\|_{\mathbf{V}_t} \leq \beta_t \right\},$$

$$\text{where } \hat{\boldsymbol{\theta}}_t := \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \left[ \|\mathbf{X}_{<t}\boldsymbol{\theta} - \mathbf{y}_{<t}\|_2^2 + \|\boldsymbol{\theta}\|_{\mathbf{H}_t}^2 \right].$$

The regression solution can be given by $\hat{\boldsymbol{\theta}}_t := (\mathbf{G}_t + \mathbf{H}_t)^{-1}\mathbf{x}_{<t}^\top \mathbf{y}_{<t}$, where $\mathbf{G}_t = \mathbf{x}_{<t}^\top \mathbf{x}_{<t}$ is the Gram matrix of actions, $\mathbf{H}_t$ is a (time-varying) regularizer, and $\beta_t$ is an appropriately chosen exploration parameter. Typically in non-private settings, the regularizer is constant, i.e., $\mathbf{H}_t = \lambda \mathbf{I} \; \forall t, \lambda > 0$ (Abbasi-Yadkori et al., 2011; Li et al., 2010), however, in our case, we will carefully select $\mathbf{H}_t$ to introduce privacy, using a strategy similar to Shariff & Sheffet (2018). Given $\mathbf{V}_t = \mathbf{G}_t + \mathbf{H}_t$, let $\text{UCB}_t(\mathbf{x}; \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \mathbf{x} \rangle + \beta_t \|\mathbf{x}\|_{\mathbf{V}_t^{-1}}$.

In the federated setting, since there are $M$ learners that have distinct actions, the communication protocol is a key component of algorithm design: communication often creates *heterogeneity* between agents, e.g., for any two agents, their estimators $(\hat{\boldsymbol{\theta}}_t)$ at any instant are certainly distinct, and the algorithm must provide a control over this heterogeneity, to bound the group regret. We additionally require that communication between agents is $(\varepsilon, \delta)$-private, making the problem more challenging.

### 6.3.1 Distributed Environment with a Server

We first consider the distributed communciation environment where there exists a controller that coordinates communication between different agents, as is typical in large-scale distributed learning. We consider a set of $M$ agents that each are interacting with the contextual bandit, and periodically communicate with the controller, that synchronizes them with other agents. We present the algorithm FedLinUCB-Dist in Algorithm 9.

#### Overview

Algorithm 9 works as follows. Consider an agent $i$, and assume that synchronization had last taken place at instant $t'$. At any instant $t > t'$, the agent has two sets of parameters - **(A)** the first being all observations up to instant $t'$ for all $M$ agents and **(B)** the second being its own observations from instant $t'$ to $t$. Since **(A)** includes samples from other agents, these are privatized, and represented as the Gram matrix $\mathbf{S}(t'+1) = \sum_{i\in[M]} \widehat{\mathbf{U}}_i(t'+1)$ and reward vector $\mathbf{s}_{t'+1} = \sum_{i\in[M]} \widehat{\mathbf{u}}_{i,t'+1}$. Algorithm 9 privatizes its own observations as well (for simplicity in the analysis) and hence $\mathbf{S}, \mathbf{s}$ are identical for all agents at all times.

Moreover, since the group parameters are noisy variants of the original parameters, i.e., $\widehat{\mathbf{U}}_i(t) = \mathbf{G}_i(t) + \mathbf{H}_i(t)$ and $\widehat{\mathbf{u}}_i(t) = \mathbf{u}_i(t) + \mathbf{h}_i(t)$ (where $\mathbf{H}_i(t)$ and $\mathbf{h}_i(t)$ are perturbations), we can rewrite $\mathbf{S}(t), \mathbf{s}(t)$ as (for any instant $t > t'$),

$$\mathbf{S}(t) = \sum_{i\in[M]} \left( \sum_{\tau=1}^{t'} \mathbf{x}_i(\tau)\mathbf{x}_i(\tau)^\top + \mathbf{H}_i(t') \right), \mathbf{s}(t) = \sum_{i\in[M]} \left( \sum_{\tau=1}^{t'} y_i(\tau)\mathbf{x}_i(\tau) + \mathbf{h}_i(t') \right). \tag{6.1}$$

When we combine the group parameters with the local (unsynchronized) parameters, we obtain the final form of the parameters for any agent $i$ as follows (for any instant $t > t'$):

$$\mathbf{V}_i(t) = \sum_{\tau=t'}^{t-1} \mathbf{x}_i(\tau)\mathbf{x}_i(\tau)^\top + \mathbf{S}(t), \tilde{\mathbf{u}}_i(t) = \sum_{\tau=t'}^{t-1} y_i(\tau)\mathbf{x}_i(\tau) + \mathbf{s}(t) \tag{6.2}$$

Then, with a suitable sequence $(\beta_i(t))_{t=1}^T$, the agent selects the action following the linear UCB objective:

$$\mathbf{x}_i(t) = \arg\max_{\mathbf{x}\in\mathcal{D}_i(t)} \left( \langle \widehat{\boldsymbol{\theta}}_{i,t}, \mathbf{x} \rangle + \beta_i(t) \|\mathbf{x}\|_{\mathbf{V}_i(t)^{-1}} \right) \text{ where } \widehat{\boldsymbol{\theta}}_{i,t} = \mathbf{V}_i(t)^{-1}\tilde{\mathbf{u}}_i(t). \tag{6.3}$$

The central idea of the algorithm is to therefore carefully perturb the Gram matrices $\mathbf{V}_i(t)$

and the reward vector $\mathbf{u}_i(t)$ with random noise $(\mathbf{H}_i(t), \mathbf{h}_i(t))$ based on the sensitivity of these elements. First, each agent updates its local (unsynchronized) estimates. These are used to construct the UCB in a manner identical to the standard OFUL algorithm (Abbasi-Yadkori et al., 2011). If, for any agent $i$ a threshold condition is met, then the agents synchronize their observations via the server. We describe this synchronization condition in Section 6.3.1 after providing an overview of the perturbation mechanism next.

### Changing Regularizers and Exploration Sequence

The perturbations $\mathbf{H}_i(t), \mathbf{h}_i(t)$ are designed keeping the privacy setting in mind. In our paper, we defer this to the subsequent section in a subroutine known as PRIVATIZER, and concentrate on the performance guarantees first. The PRIVATIZER subroutine provides suitable perturbations based on the privacy budget ($\varepsilon$ and $\delta$). In this paper, we assume these budgets to be identical for all agents (however, the algorithm and analysis hold for unique privacy budgets as well, as long as a lower bound on the budgets is known). In turn, the quantities $\varepsilon$ and $\delta$ affect the algorithm (and regret) via the quantities $\rho_{\min}, \rho_{\max}$, and $\kappa$ which can be understood as spectral bounds on $\mathbf{H}_i(t), \mathbf{h}_i(t)$.

**Definition 6.2** (Sparsely-accurate $\rho_{\min}, \rho_{\max}$ and $\kappa$)**.** *Consider a subsequence $\bar{\boldsymbol{\sigma}}$ of $[T] = 1, ..., T$ of size $n$. The bounds $0 \leq \rho_{\min} \leq \rho_{\max}$ and $\kappa$ are $(\alpha/2nM, \bar{\boldsymbol{\sigma}})$-accurate for $(\mathbf{H}_i(t))_{i\in[M], t\in\bar{\boldsymbol{\sigma}}}$ and $(\mathbf{h}_i(t))_{i\in[M], t\in\bar{\boldsymbol{\sigma}}}$, if, for each round $t \in \bar{\boldsymbol{\sigma}}$, with probability at least $1 - \alpha/2M$,*

$$\left\| \sum_{i=1}^{M} \mathbf{H}_i(t) \right\| \leq \rho_{\max}, \quad \left\| \left( \sum_{i=1}^{M} \mathbf{H}_i(t) \right)^{-1} \right\| \leq \frac{1}{\rho_{\min}}, \quad \left\| \sum_{i=1}^{M} \mathbf{h}_i(t) \right\|_{(\sum_i \mathbf{H}_i(t))^{-1}} \leq \kappa.$$

The motivation for obtaining accurate bounds $\rho_{\min}, \rho_{\max}$ and $\kappa$ stems from the fact that in the non-private case, the quantities that determine regret are not stochastic conditioned on the obtained sequence $(\mathbf{x}_t, y_t)_{t\in[T]}$, whereas the addition of stochastic regularizers in the private case requires us to have control over their spectra to achieve any meaningful regret. To form the UCB, recall that we additionally require a suitable exploration sequence $\beta_i(t)$ for each agent, which is defined as follows.

**Definition 6.3** (Accurate $(\beta_i(t))_{i\in[M], t\in[T]}$)**.** *A sequence $(\beta_i(t))_{i\in[M], t\in[T]}$ is $(\alpha, M, T)$-accurate for $(\mathbf{H}_i(t))_{i\in[M], t\in[T]}$ and $(\mathbf{h}_i(t))_{i\in[M], t\in[T]}$, if it satisfies $\|\tilde{\boldsymbol{\theta}}_i(t) - \boldsymbol{\theta}_\star\|_{\mathbf{V}_i(t)} \leq \beta_i(t)$ with probability at least $1 - \alpha$ for all rounds $t = 1, ..., T$ and agents $i = 1, ..., M$ simultaneously.*

**Theorem 6.1.** *Consider an instance of the problem where synchronization occurs exactly $n$ times on instances $\bar{\boldsymbol{\sigma}}$, up to and including $T$ trials, and $\rho_{\min}, \rho_{\max}$ and $\kappa$ are $(\alpha/2nM)$-accurate. Then, for Algorithm 9, the sequence $(\beta_i(t))_{i\in[M],t\in[T]}$ is $(\alpha, M, T)$-accurate where:*

$$\beta_i(t) := \sigma\sqrt{2\log\left(\frac{2t}{\alpha}\right) + d\log\left(\frac{\det(\mathbf{V}_i(t))}{M\rho_{\min}}\right)} + S\sqrt{M\rho_{\max}} + \kappa.$$

*Proof.* Let $\mathbf{X}_{i,<t}$ denote the set of all observations available to the agent (including private communication from other agents). Furthermore, let the noise-free Gram matrix of all observations as $\mathbf{G}_i(t) = \sum_{\tau=1}^{t}\mathbf{x}_i(\tau)\mathbf{x}_i(\tau)^\top + \sum_{\tau=1}^{t_s}\sum_{j=1,j\neq i}^{M}\mathbf{x}_j(\tau)\mathbf{x}_j(\tau)^\top$ (where $t_s$ is the last synchronization iteration). We also have that $\mathbf{V}_i(t) = \mathbf{G}_i(t) + \sum_{j=1}^{M}\mathbf{H}_j(t)$, and for any $t$ between synchronization rounds $t_s$ and $t_{s+1}$, $\mathbf{H}_j(t) = \mathbf{H}_{j,t_s}$ for all $j$. By definition, $\tilde{\boldsymbol{\theta}}_i(t) = \mathbf{V}_i(t)^{-1}\tilde{\mathbf{u}}_i(t)$, $\tilde{\mathbf{u}}_i(t) = \mathbf{u}_i(t) + \sum_{j\in[M]}\mathbf{h}_j(t)$ and $\mathbf{u}_i(t) = \mathbf{X}_{i,<t}^\top\mathbf{y}_{<t}$. Therefore, we have,

$$\begin{aligned}
\boldsymbol{\theta}_\star - \tilde{\boldsymbol{\theta}}_i(t) &= \boldsymbol{\theta}_\star - \mathbf{V}_i(t)^{-1}\left(\mathbf{X}_{i,<t}^\top\mathbf{y}_{<t} + \sum_{j\in[M]}\mathbf{h}_j(t)\right)\\
&= \boldsymbol{\theta}_\star - \mathbf{V}_i(t)^{-1}\left(\mathbf{X}_{i,<t}^\top\mathbf{X}_{i,<t}\boldsymbol{\theta}_\star + \mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t} + \sum_{j\in[M]}\mathbf{h}_j(t)\right)\\
&= \boldsymbol{\theta}_\star - \mathbf{V}_i(t)^{-1}\left(\mathbf{V}_i(t)\boldsymbol{\theta}_\star - \sum_{j\in[M]}\mathbf{H}_j(t)\boldsymbol{\theta}_\star + \mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t} + \sum_{j\in[M]}\mathbf{h}_j(t)\right)\\
&= \mathbf{V}_i(t)^{-1}\left(\sum_{j\in[M]}\mathbf{H}_j(t)\boldsymbol{\theta}_\star - \mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t} - \sum_{j\in[M]}\mathbf{h}_j(t)\right).
\end{aligned}$$

Multiplying both sides by $\mathbf{V}_i(t)^{1/2}$ gives

$$\begin{aligned}
\mathbf{V}_i(t)^{1/2}\left(\boldsymbol{\theta}_\star - \tilde{\boldsymbol{\theta}}_i(t)\right) &= \mathbf{V}_i(t)^{-1/2}\left(\sum_{j\in[M]}\mathbf{H}_j(t)\boldsymbol{\theta}_\star - \mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t} - \sum_{j\in[M]}\mathbf{h}_j(t)\right)\\
\implies \left\|\boldsymbol{\theta}_\star - \tilde{\boldsymbol{\theta}}_i(t)\right\|_{\mathbf{V}_i(t)} &= \left\|\sum_{j\in[M]}\mathbf{H}_j(t)\boldsymbol{\theta}_\star - \mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t} - \sum_{j\in[M]}\mathbf{h}_j(t)\right\|_{\mathbf{V}_i(t)^{-1}} \qquad \text{(Applying } \|\cdot\|\text{)}\\
&\leq \left\|\sum_{j\in[M]}\mathbf{H}_j(t)\boldsymbol{\theta}_\star\right\|_{\mathbf{V}_i(t)^{-1}} + \left\|\mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t}\right\|_{\mathbf{V}_i(t)^{-1}} + \left\|\sum_{j\in[M]}\mathbf{h}_j(t)\right\|_{\mathbf{V}_i(t)^{-1}}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Triangle inequality)}
\end{aligned}$$

Making the substitution $\mathbf{H}_t = \sum_{j\in[M]} \mathbf{H}_j(t)$,

$$\leq \|\mathbf{H}_t\boldsymbol{\theta}_\star\|_{\mathbf{H}_t^{-1}} + \left\|\mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t}\right\|_{\mathbf{V}_i(t)^{-1}} + \left\|\mathbf{h}_j(t)\right\|_{\mathbf{H}_t^{-1}}$$

$$\text{(Since } \mathbf{V}_i(t) \succcurlyeq \textstyle\sum_{j\in[M]} \mathbf{H}_j(t))$$

$$\leq \|\boldsymbol{\theta}_\star\|_{\mathbf{H}_t} + \left\|\mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t}\right\|_{(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})^{-1}} + \sum_{j\in[M]} \left\|\mathbf{h}_j(t)\right\|_{\mathbf{H}_t^{-1}}.$$

$$\text{(Since } \forall i \in [M], \mathbf{V}_i(t) \succcurlyeq \mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})$$

Now, note that since we only require at most $Mn$ different noise matrices, we only need the noise sequences $\mathbf{H}$ by a union bound over all $TM$ rounds ($T$ rounds per agent), we can say that simultaneously for all $i \in [M], t \in [T]$, with probability at least $1-\alpha/2$, $\|\boldsymbol{\theta}_\star\|_{\mathbf{H}_t} \leq \sqrt{\|\mathbf{H}_t\|}\|\boldsymbol{\theta}_\star\| \leq S\sqrt{\rho_{\max}}$ and $\|\sum_{j\in[M]} \mathbf{h}_j(t)\|_{\mathbf{H}_t^{-1}} \leq \kappa$. At this point, to control the remaining term, one might consider directly applying the self-normalized martingale bound (Theorem 1 of Abbasi-Yadkori et al. (2011)), however, this cannot be done as the data-dependent communication breaks the martingale structure of the sequence $\mathbf{x}_1, ..., \mathbf{x}_t$ (the observations depend on when the last synchronization has taken place). To remedy this issue, we control the worst-case deviation over all possible synchronization points $1 \leq t_s \leq t$, and then apply a union bound. Conditioned on a fixed synchronization round $t_s$, the martingale structure is preserved, and therefore we can apply Theorem 1 of Abbasi-Yadkori et al. (2011). Therefore, we can say that conditioned on synchronization being done on round $1 \leq t_s \leq t$, with probability at least $1-\alpha/2t$ simultaneously for all $1 \leq t \leq T$ and agents $i$,

$$\left\|\mathbf{X}_{i,<t}^\top\boldsymbol{\eta}_{<t}\right\|_{(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})^{-1}} \leq \sigma\sqrt{2\log\frac{2t}{\alpha} + \log\frac{\det(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})}{\det(M\rho_{\min}\mathbf{I})}}$$

$$\leq \sigma\sqrt{2\log\frac{2t}{\alpha} + d\log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{tL^2}{d\rho_{\min}}\right)}.$$

The last step follows from (a) noting that $\forall i \in [M], \mathbf{G}_i(t) + M\rho_{\max}\mathbf{I} \succcurlyeq \mathbf{V}_i(t) \succcurlyeq \mathbf{G}_i(t) + M\rho_{\min}\mathbf{I} \implies \det(\mathbf{G}_i(t)+M\rho_{\max}\mathbf{I}) \geq \det(\mathbf{V}_i(t)) \geq \det(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})$, and (b) the trace-determinant inequality. Next, we have by a union bound over all $1 \leq t_s \leq t$ that with probability at least $1 - \alpha/2$,

$$\max_{1 \leq t_s \leq t} \left\| \mathbf{X}_{i,<t}^\top \boldsymbol{\eta}_{<t} \right\|_{(\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})^{-1}} \leq \sigma \sqrt{2 \log \frac{2t}{\alpha} + d \log \left( \frac{\rho_{\max}}{\rho_{\min}} + \frac{tL^2}{d\rho_{\min}} \right)}.$$

Putting it all together, we have that all $\beta_i(t)$ are bounded by $\bar{\beta}_t$, given by,

$$\bar{\beta}_t := \sigma \sqrt{2 \log \frac{2t}{\alpha} + d \log \left( \frac{\rho_{\max}}{\rho_{\min}} + \frac{tL^2}{d\rho_{\min}} \right)} + S\sqrt{\rho_{\max}} + \kappa.$$

□

The key point here is that for the desired levels of privacy $(\varepsilon, \delta)$ and synchronization rounds $(n)$, we can calculate appropriate $\rho_{\min}, \rho_{\max}$ and $\kappa$, which in turn provide us a UCB algorithm with guarantees. We now present the synchronization condition and communication complexity as a function of $\rho_{\min}, \rho_{\max}$ and $\kappa$.

### Synchronization and Communication Complexity

The synchronization event is triggered by the server if, for any agent $i$, the log-determinant of the local Gram matrix exceeds the synchronized Gram matrix ($\mathbf{S}_i(t)$) by an amount $D/\Delta t_i$ (where $\Delta t_i$ is the time since the last synchronization), then it sends a signal to the controller, that synchronizes *all* agents with their latest action/reward pairs. Specifically, synchronization occurs whenever the following condition is met.

$$\log \left( \frac{\det \left( \mathbf{V}_i(t) + \mathbf{x}_i(t)\mathbf{x}_i(t)^\top + M(\rho_{\max} - \rho_{\min})\mathbf{I} \right)}{\det \left( \mathbf{S}_i(t) \right)} \right) \geq \frac{D}{\Delta t_i}.$$

The synchronization is done using a *privatized* version of the Gram matrix and rewards, carried out by the subroutine PRIVATIZER (Section 6.5, Alg. 11). This synchronization ensures that the *heterogeneity* between the agents is controlled, allowing us to control the overall regret and limit communication as well.

**Proposition 6.1** (Communication Complexity). *If Algorithm 9 is run with threshold $D$, then total rounds of communication $n$ obeys,*

$$n \leq 2\sqrt{\left( \frac{dT}{D} \right) \cdot \log \left( \frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}} \right)} + 4.$$

This statement is proved in Section 6.9.1.

### 6.3.2 Decentralized Peer-to-Peer Environment

In this environment, we assume that the collection of agents communicate by directly sending messages to each other, identical to the multi-armed bandit setting. The communication network is denoted by an undirected (connected) graph $G = (\mathcal{V}, \mathcal{E})$, where, edge $e_{ij} \in \mathcal{E}$ if agents $i$ and $j$ can communicate directly. The protocol operates as follows: every trial, each agent interacts with their respective bandit, and obtains a reward. At any trial $t$, after receiving the reward, each agent $v$ sends the message $\mathbf{m}_v(t)$ to all its neighbors in $G$. This message is forwarded from agent to agent $\gamma$ times (taking one trial of the bandit problem each between forwards), after which it is dropped. This communication protocol, based on the *time-to-live* (delay) parameter $\gamma \leq d_\star(G)$ is a common technique to control communication complexity, known as the LOCAL protocol (Fraigniaud, 2016; Linial, 1992; Suomela, 2013). Each agent $v \in \mathcal{V}$ therefore also receives messages $\mathbf{m}_{v'}(t - d(v, v'))$ from all the agents $v'$ such that $d(v, v') \leq \gamma$, i.e., from all agents in $\mathcal{N}_v^+(G_\gamma)$.

There are several differences from the distributed setting: first, since agents receive messages from different other agents based on their position in $G$, they generally have heterogenous information throughout, as there is no global synchronization via a server. Second, information does not flow instantaneously through $G$, and messages can take up to $\gamma$ rounds to be communicated, due to delays incurred, similar to the multi-armed bandit case. This requires (mainly technical) changes to the distributed algorithm in order to control the regret. To account for these changes in the environment, we describe a different algorithm FedLinUCB-Network as follows, and the pseudocode is presented in Algorithm 10.

**Subsampling**. The first key algorithmic change from the earlier variant is the idea of subsampling, inspired by Weinberger & Ordentlich (2002). We consider $T$ rounds of the bandit problem as $\gamma$ interleaved bandit problems, each of length $\frac{T}{\gamma}$. We call each of these interleaved problems as "phases". Each agent $i$ maintains $\gamma$ total estimators $\bar{\boldsymbol{\theta}}_{i,g}, g = 1, ..., \gamma$ (one for each phase), and uses each estimator in a round-robin fashion, i.e., at $t = 1$ each agent uses $\bar{\boldsymbol{\theta}}_{i,1}$, and at $t = 2$, each agent uses $\bar{\boldsymbol{\theta}}_{i,2}$, and so on.

**Broadcast-based Synchronization**. The synchronization process is different from the distributed case as there is no server to assist. The estimators (and their associated $\mathbf{V}_{i,g}$, $\tilde{\mathbf{u}}_{i,g}$) are updated in a manner similar to Algorithm 9: if the log det of the $g^{th}$ Gram matrix exceeds a threshold $D/(\Delta_{i,g}+1)$, then the agent broadcasts a request to synchronize. Each agent $j$ within the $\gamma$-clique of $i$ that receives this request broadcasts its personal observations within $\mathbf{V}_{j,g}$, $\tilde{\mathbf{u}}_{j,g}$ to all agents. The message broadcast by any agent $v$ during synchronization for phase $g$ is therefore,

$$\mathbf{m}_{v,g}(t) = \left\langle v, t, g, \widehat{\mathbf{U}}_i^g(t), \widehat{\mathbf{u}}_i^g(t) \right\rangle .$$

Where, similar to the distributed case, $\widehat{\mathbf{U}}_i^g(t), \widehat{\mathbf{u}}_i^g(t)$ denote the (privatized) personal Gram and bias for phase $g$ until round $t$. In contrast to the multi-armed setting, the *contextual* setting requires us to broadcast the covariance of the actions as well ($\widehat{\mathbf{U}}_i^g(t)$), and hence each message is $\log(MT\gamma) + (d^2+d)\log(BL) = \mathcal{O}(d^2\log(MT\gamma))$ bits long. Therefore, each $\mathbf{V}_{i,g}$, $\tilde{\mathbf{u}}_{i,g}$ is updated with action/reward pairs from *only* the trials they were employed in (across all agents). This ensures that if a signal to synchronize the $g^{th}$ set of parameters has been broadcast by an agent $i$, all agents within the $\gamma$-clique of $i$ will have synchronized their $g^{th}$ parameters by the next 2 rounds they will be used again (i.e., $2\gamma$ trials later).

## 6.4 Regret Guarantees

### 6.4.1 Distributed Group Regret

**Theorem 6.2** (Distributed Group Regret)**.** *Assuming Theorem 6.1 holds, and synchronization occurs in at least $n = \Omega\left(d\log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right)\right)$ rounds, Algorithm 9 obtains the following group pseudoregret with probability at least $1-\alpha$:*

$$\mathfrak{R}(T) = \mathcal{O}\left(\sigma\sqrt{MT}\left(d\log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right) + \sqrt{\log\frac{2}{\alpha}} + \sqrt{d}\cdot\kappa\right)\right).$$

*Proof.* Consider a hypothetical agent that takes the following actions sequentially,

$$\mathbf{x}_1(1), \mathbf{x}_2(1), \ldots, \mathbf{x}_1(2), \mathbf{x}_2(2), \ldots, \mathbf{x}_{M-1}(T), \mathbf{x}_M(T).$$

Let $\boldsymbol{W}_{i,t} = M\rho_{\min}\mathbf{I} + \sum_{j=1}^{M}\sum_{u=1}^{t-1}\mathbf{x}_j(u)\mathbf{x}_j(u)^\top + \sum_{j=1}^{i-1}\mathbf{x}_j(t)\mathbf{x}_j(t)^\top$ be the Gram matrix formed until the hypothetical agent reaches $\mathbf{x}_i(t)$. We state a result to bound the above potential.

**Lemma 6.1** (Elliptical Potential, Lemma 3 of Abbasi-Yadkori et al. (2011))**.** *Let $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathbb{R}^d$ be vectors such that $\|\mathbf{x}\|_2 \leq L$. Then, for any positive definite matrix $\mathbf{U}_0 \in \mathbb{R}^{d \times d}$, define $\mathbf{U}_t := \mathbf{U}_{t-1} + \mathbf{x}_t \mathbf{x}_t^\top$ for all t. Then, for any $\nu > 1$,*

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

By Lemma 6.1, we have that

$$\sum_{t=1}^{T} \sum_{i=1}^{M} \|\mathbf{x}_i(t)\|^2_{\boldsymbol{W}_{i,t}^{-1}} \leq 2d \log \left( 1 + \frac{TL^2}{d\rho_{\min}} \right).$$

Now, in the original setting, let $T_1, T_2, ..., T_{p-1}$ be the trials at which synchronization occurs. After any round $T_k$ of synchronization, consider the cumulative Gram matrices of all observations obtained until that round as $\mathbf{V}_k, k = 1, ..., p-1$, regularized by $M\rho_{\min}\mathbf{I}$, i.e., $\mathbf{V}_k = \sum_{i \in [M]} \sum_{t=1}^{T_k} \mathbf{x}_i(t)\mathbf{x}_i(t)^\top + M\rho_{\min}\mathbf{I}$. Finally, let $\mathbf{V}_p$ denote the (regularized) Gram matrix with all trials at time $T$, and $\mathbf{V}_0 = M\rho_{\min}\mathbf{I}$. Therefore, we have that $\det(\mathbf{V}_0) = (M\rho_{\min})^d$, and that $\det(\mathbf{V}_p) \leq \left(\frac{\text{tr}(\mathbf{V}_p)}{d}\right)^d \leq \left(M\rho_{\max} + \frac{MTL^2}{d}\right)^d$. Therefore, for any $\nu > 1$,

$$\log_\nu \left( \frac{\det(\mathbf{V}_p)}{\det(\mathbf{V}_0)} \right) \leq d \log_\nu \left( \frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}} \right).$$

Let $R = \left\lceil d \log_\nu \left( \frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}} \right) \right\rceil$. It follows that in all but $R$ periods between synchronization $1 \leq \frac{\det(\mathbf{V}_k)}{\det(\mathbf{V}_{k-1})} \leq \nu$. We consider the event $E$ to be the period $k$ when the above holds. Now, for any $T_{k-1} \leq t \leq T_k$, consider the immediate pseudoregret for any agent $i$. We now present a standard lemma to bound the per-round regret by the upper confidence bound.

**Lemma 6.2.** *The instantaneous pseudoregret $r_{i,t}$ obtained by any agent i at any instant t obeys the following:*

$$r_{i,t} \leq 2\bar{\beta}_T \left\|\mathbf{x}_i(t)\right\|_{\mathbf{V}_i(t)^{-1}}.$$

By the above lemma, we have

$$\begin{aligned} r_{i,t} &\leq 2\bar{\beta}_T \|x_{i,t}\|_{\mathbf{V}_i(t)^{-1}} \\ &\leq 2\bar{\beta}_T \|x_{i,t}\|_{(\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})^{-1}} && (\mathbf{V}_i(t) \succcurlyeq \mathbf{G}_i(t) + M\rho_{\min}\mathbf{I}) \end{aligned}$$

$$
\begin{aligned}
&\leq 2\bar{\beta}_T \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}} \cdot \sqrt{\frac{\det(\boldsymbol{W}_{i,t})}{\det(\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})}} \\
&\leq 2\bar{\beta}_T \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}} \cdot \sqrt{\frac{\det(\mathbf{V}_k)}{\det(\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})}} && (\mathbf{V}_k \succcurlyeq \boldsymbol{W}_{i,t}) \\
&\leq 2\bar{\beta}_T \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}} \cdot \sqrt{\frac{\det(\mathbf{V}_k)}{\det(\mathbf{V}_{k-1})}} && (\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I} \succcurlyeq \mathbf{V}_{k-1}) \\
&\leq 2\nu\bar{\beta}_T \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}}. && (\text{Event } E \text{ holds})
\end{aligned}
$$

Now, we can sum up the immediate pseudoregret over all such periods where $E$ holds to obtain the total regret for these periods. With probability at least $1 - \alpha$,

$$
\begin{aligned}
\text{Regret}(T, E) = \sum_{i=1}^{M} \sum_{t \in [T]:\text{E is true}} r_{i,t} &\leq \sqrt{MT \left( \sum_{i=1}^{M} \sum_{t \in [T]:\text{E is true}} r_{i,t}^2 \right)} \\
&\leq 2\nu\bar{\beta}_T \sqrt{MT \left( \sum_{i=1}^{M} \sum_{t \in [T]:\text{E is true}} \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}} \right)} \leq 2\nu\bar{\beta}_T \sqrt{MT \left( \sum_{i=1}^{M} \sum_{t \in [T]} \|x_{i,t}\|_{\boldsymbol{W}_{i,t}^{-1}} \right)} \\
&\leq 2\nu\bar{\beta}_T \sqrt{2MTd \log_\nu \left( 1 + \frac{TL^2}{d\rho_{\min}} \right)}.
\end{aligned}
$$

Now let us consider the periods in which $E$ does not hold. In any such period between synchronization of length $t_k = T_k - T_{k-1}$, we have, for any agent $i$, the regret accumulated given by:

$$
\begin{aligned}
\text{Regret}([T_{k-1}, T_k]) = \sum_{t=T_{k-1}}^{T_k} \sum_{i=1}^{M} r_{i,t} &\leq 2\nu\bar{\beta}_T \left( \sum_{i=1}^{M} \sqrt{t_k \sum_{t=T_{k-1}}^{T_k} \|\mathbf{x}_i(t)\|^2_{\mathbf{V}_i(t)^{-1}}} \right) \\
&\leq 2\nu\bar{\beta}_T \left( \sum_{i=1}^{M} \sqrt{t_k \log_\nu \left( \frac{\det(\mathbf{V}_{i,t+t_k})}{\det(\mathbf{V}_i(t))} \right)} \right) \\
&\leq 2\nu\bar{\beta}_T \left( \sum_{i=1}^{M} \sqrt{t_k \log_\nu \left( \frac{\det(\mathbf{G}_{i,t+t_k} + M\rho_{\max}\mathbf{I})}{\det(\mathbf{G}_i(t) + M\rho_{\min}\mathbf{I})} \right)} \right)
\end{aligned}
$$

By Algorithm 9, we know that for all agents, $t_k \log_\nu\left(\frac{\det(\mathbf{G}_{i,t+t_k}+M\rho_{\max}\mathbf{I})}{\det(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})}\right) \leq D$ (since there would be a synchronization round otherwise), therefore

$$\text{Regret}([T_{k-1}, T_k]) \leq 2\nu\bar{\beta}_T M\sqrt{D}.$$

Now, note that of the total $p$ periods between synchronizations, only at most $R$ periods will not have event $E$ be true. Therefore, the total regret over all these periods can be bound as,

$$\text{Regret}(T, \bar{E}) \leq R \cdot 2\nu\bar{\beta}_T M\sqrt{D} \leq 2\nu\bar{\beta}_T M\sqrt{D}\left(d\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right) + 1\right).$$

Adding it all up together gives us,

$$\begin{aligned}\mathfrak{R}(T) &= \text{Regret}(T, E) + \text{Regret}(T, \bar{E}) \\ &\leq 2\nu\bar{\beta}_T\left[\sqrt{2MTd\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right)} + M\sqrt{D}\left(d\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right) + 1\right)\right]\end{aligned}$$

Setting $D = 2Td\left(\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right) + 1\right)^{-1}$ and using shorthand $\Phi = \log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right)$, we have:

$$\mathfrak{R}(T) \leq 4\nu\bar{\beta}_T\sqrt{2MTd\Phi + 1} \leq 4\nu\left(\sigma\sqrt{2\log\frac{2}{\alpha} + d\Phi} + S\sqrt{\rho_{\max}} + \kappa\right)\sqrt{2MTd\Phi}.$$

Optimizing over $\nu$ gives us the final result. Asymptotically, we have (setting $\nu = e$), with probability at least $1 - \alpha$,

$$\mathfrak{R}(T) = \mathcal{O}\left(\sigma\sqrt{MT}\left(d\log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right) + \sqrt{\log\frac{2}{\alpha} + \sqrt{d}\cdot\kappa}\right)\right).$$

□

This regret bound is obtained by setting $D = 2Td\left(\log\left(\rho_{\max}/\rho_{\min} + TL^2/d\rho_{\min}\right) + 1\right)^{-1}$, which therefore ensures $\mathcal{O}(M\log T)$ rounds of total communication (by Proposition 9.3). However, the remarks next clarify a more sophisticated relationship between communication, privacy and regret.

**Remark 6.1** (Communication Complexity and Regret)**.** Theorem 6.2 assumes that the number of rounds of communication is essentially $\mathcal{O}(M\log T)$. This rate (and corresponding

$D$) is chosen to provide a balance between privacy and utility, and in fact, can be altered depending on the application. By Theorem 6.2, a simple substitution suggests that if we allow $\mathcal{O}(MT)$ rounds of communication (i.e., synchronize every round), the regret can be improved by a factor of $\mathcal{O}(\sqrt{\log(T)})$, to match the single-agent $\widetilde{\mathcal{O}}(d\sqrt{MT})$ up to the term $\kappa\sqrt{d}$. Similarly, we can select $D$ such that with $\mathcal{O}(M^{1.5}d^3)$ rounds of communication (i.e., independent of $T$), we incur cumulative regret of $\mathcal{O}((d+\kappa\sqrt{d})\log^2(MT)\sqrt{MT})$. We will demonstrate subsequently in Section 6.5 that the term $\kappa$ arises from the noise added to preserve privacy, and is 0 when the algorithm is not private, hence all our bounds will match the $\Omega(\sqrt{dMT})$ lower bound for a single-agent pulling $M$ arms serially (up to a factor $\sqrt{d}$).

Theorem 6.2 demonstrates the relationship between communication complexity (i.e., number of synchronization rounds) and the regret bound for a fixed privacy budget, via the dependence on the bounds $\rho_{\min}, \rho_{\max}$ and $\kappa$. We now present similar results (for a fixed privacy budget) on group regret for the decentralized setting, which is more involved, as the delays and lack of a centralized controller make it difficult to control the heterogeneity of information between agents. Subsequently, in the next section, we will present results on the privacy guarantees of our algorithms.

### 6.4.2 Decentralized Group Regret

**Theorem 6.3** (Decentralized Group Regret)**.** *Assuming Theorem 6.1 holds, and the number of synchronization rounds are at least*

$$n = \Omega\left(\frac{d(\bar{\chi}(G_\gamma)\cdot\gamma)}{1+L^2}\log\left(\frac{\rho_{\max}}{\rho_{\min}}+\frac{TL^2}{d\rho_{\min}}\right)\right),$$

FedLinUCB-Network *obtains the following group pseudoregret with probability at least* $1-\alpha$*:*

$$\mathfrak{R}(T) = O\left(\sigma\sqrt{M(\bar{\chi}(G_\gamma)\cdot\gamma)T}\left(d\log\left(\frac{\rho_{\max}}{\rho_{\min}}+\frac{TL^2}{d\rho_{\min}}\right)+\sqrt{\log\left(\frac{2\bar{\chi}(G_\gamma)}{\alpha}\right)}+\kappa\sqrt{d}\right)\right).$$

The proof can be found in Section 6.9.3.

**Remark 6.2** (Decentralized Group Regret)**.** Decentralized FedLinUCB obtains an identical dependence on the privacy bounds $\rho_{\min}, \rho_{\max}$ and $\kappa$ and horizon $T$ as Algorithm 9, since the underlying bandit subroutines are identical for both. The key difference is in additional the leading factor of $\sqrt{\bar{\chi}(G_\gamma)\cdot\gamma}$, which arises from the delayed spread of information: if

$G$ is dense, e.g., complete, then $\gamma = 1$ and $\bar{\chi}(G_\gamma) = 1$, since there is only one clique of $G$. In the worst case, if $G$ is a line graph, then $\bar{\chi}(G_\gamma) = M/\gamma$, giving an additional factor of $M$ (i.e., it is as good as each agent acting individually). In more practical scenarios, we expect $G$ to be hierarchical, and expect a delay overhead of $\mathcal{O}(1)$ and not as a function of $M$.

**Remark 6.3** (Comparison with Multi-Armed Bound)**.** The regret of FedLinUCB-Network in comparison with the multi-armed case introduces an additional factor of $\gamma$ in the leading term. Observe that while we believe that this is in fact an artefact of our proof technique, where we use only $\mathcal{O}(M\log(T))$ messaging rounds (instead of communicating every round, as in the bandit case), however, we remark that the bandit result, e.g., in Theorem 2.2, also incurs a $\gamma$ factor in the problem independent version of the bound.

### 6.4.3 Lower Bounds

In this section, we discuss lower bounds for the federated contextual bandit problem. We first provide a minimax result from Basu et al. (2020) for the single-agent multi-armed bandit under joint differential privacy.

**Theorem 6.4** (Theorem 2 of Basu et al. (2020))**.** *For any $\varepsilon \in (0, 1/2]$ and horizon $T \geq K$, any single-agent policy that is $\varepsilon-$ joint DP at all instances for a $K-$armed bandit must incur regret*

$$\mathfrak{R}(T) = \Omega\left(\sqrt{\frac{(K-1)T}{2\varepsilon(e^{2\varepsilon}-1)}}\right).$$

Using this result we present our lower bound for the distributed setting.

**Theorem 6.5.** *Let $\pi$ be a multi-agent bandit policy over $M$ agents that satisfies $(\varepsilon, \delta, M)-$federated differential privacy. We have that for any number of trials $T$, there exists a $d-$dimensional bandit problem with $d \geq 2$ such that,*

$$\mathfrak{R}(T) = \Omega\left(\sqrt{\frac{dMT}{\varepsilon(e^{2\varepsilon}-1)}}\right).$$

*Proof.* We consider a reduction from linear to $K$-armed stochastic bandits. Consider the set of all possible policies for a $T$ round bandit for any individual agent be contained in the set $\mathbf{\Pi}_s$, and let the set of all possible policies for any single-agent $MT$ round bandit be contained in the set $\mathbf{\Pi}_M$. For any set of $M$ policies $\pi_1, ..., \pi_M$ such that $\pi_m \in \mathbf{\Pi}_s \forall m \in [M]$, we have that the group pseudoregret:

$$\mathfrak{R}(T;\pi_1,...,\pi_M)=\sum_{m=1}^{M}\mathfrak{R}_m(T;\pi_m)$$

Where any agent $m$ is faced with the sequence of decision sets $\mathfrak{D}_m=\mathcal{D}_m(1),...,\mathcal{D}_m(T)$. Now, consider the following environment where any agent $m$ is provided the following sequence of decision sets $\widetilde{\mathfrak{D}}_m=\mathcal{D}'_m(1),...,\mathcal{D}'_m(MT)$ where $\mathcal{D}'_m(i)=\mathcal{D}_m(\lceil\frac{i}{M}\rceil)$, and for each policy $\pi\in\mathbf{\Pi}_s$, consider the policy $\tilde{\pi}$ such that $\tilde{\pi}(t)=\pi(\lceil\frac{i}{M}\rceil)$ for each $1\le t\le MT$, and denote the compound policy space $\widetilde{\mathbf{\Pi}}=\{\tilde{\pi}|\pi\in\mathbf{\Pi}\}$. We can therefore see that the following holds for each $\pi\in\mathbf{\Pi}$ (with corresponding $\tilde{\pi}\in\widetilde{\mathbf{\Pi}}$):

$$\begin{aligned}(\text{Regret of }\pi\text{ in }\mathfrak{D}_m)(T)&\ge\frac{1}{M}(\text{Regret of }\tilde{\pi}\text{ in }\widetilde{\mathfrak{D}}_m)(MT)\\&\ge\inf_{\tilde{\pi}\in\widetilde{\mathbf{\Pi}}}\frac{1}{M}(\text{Regret of }\tilde{\pi}\text{ in }\widetilde{\mathfrak{D}}_m)(MT)\\&\ge\inf_{\tilde{\pi}\in\mathbf{\Pi}_M}\frac{1}{M}(\text{Regret of }\tilde{\pi}\text{ in }\widetilde{\mathfrak{D}}_m)(MT)\\&=\Omega\left(\sqrt{\frac{(K-1)T}{2M\varepsilon(e^{2\varepsilon}-1)}}\right).\end{aligned}$$

The last line follows from Theorem 6.4 and holds since each agent is $(\varepsilon,\delta)$-JDP if they obey $(\varepsilon,\delta,M)-$federated DP. Summing over all agents, setting $K=d$ and using the fact that $d-1\ge d/2$ for all $d\ge 2$ gives us the result. □

**Discussion**. The above bound approaches the upper bound for small $\varepsilon$, but is not tight for large $\varepsilon$. However, our upper bound (Corollary 6.1) is only valid for $\varepsilon\le 1$ which demonstrates that the performance of our algorithm is near-optimal (up to $\sqrt{d}$ factors). Note that, for non-private settings, we can directly apply the network-dependent bound of Theorem 2.4 into this setting via a similar linear bandits to multi-armed reduction by setting $\Delta_k=\sqrt{\frac{K\alpha(G_{\gamma+1})\log(T)}{MT}}$ to obtain the rate of $\sqrt{\alpha(G_{\gamma+1})MT}$, which once again, for sparse $G$ is within constant factors of our bound. Determining optimal network-dependent rates for the private setting, however, is an open problem for future work.

## 6.5 Privacy Guarantees

We now discuss the privacy guarantees for both algorithms. Here we present results for the distributed algorithm, but our results hold (almost identically) for the decentralized case as well (see appendix). Note that each agent interacts with data from other agents only via the cumulative parameters $\mathbf{S}(t)$ and $\mathbf{s}(t)$. These, in turn, depend on $\boldsymbol{Z}_{i,t} = \mathbf{U}_i(t) + \mathbf{H}_i(t)$ and $\boldsymbol{z}_{i,t} = \bar{\mathbf{u}}_i(t) + \mathbf{h}_i(t)$ for each agent $i$, on the instances $t$ that synchronization occurs.

**Proposition 6.2** (see Dwork (2011); Shariff & Sheffet (2018))**.** *Consider $n \leq T$ synchronization rounds occuring on trials $\bar{\boldsymbol{\sigma}} \subseteq [T]$. If the sequence $(\boldsymbol{Z}_{i,t}, \boldsymbol{z}_{i,t})_{t \in \bar{\boldsymbol{\sigma}}}$ is $(\varepsilon, \delta)$-differentially private with respect to $(\mathbf{x}_i(t), y_i(t))_{t \in [T]}$, for each agent $i \in [M]$, then all agents are $(\varepsilon, \delta, M)$-federated differentially private.*

**Tree-Based Mechanism**. Let $x_1, x_2, ...x_T$ be a (matrix-valued) sequence of length $T$, and $s_i = \sum_{t=1}^{i} x_t$ be the incremental sum of the sequence that must be released privately. The tree-based mechanism (Dwork & Smith, 2010) for differential privacy involves a trusted entity maintaining a binary tree $\mathcal{T}$ (of depth $m = 1 + \lceil \log_2 T \rceil$), where the leaf nodes contain the sequence items $x_i$, and each parent node maintains the (matrix) sum of its children. Let $n_i$ be the value stored at any node in the tree. The mechanism achieves privacy by adding a noise $h_i$ to each node, and releasing $n_i + h_i$ whenever a node is queried. Now, to calculate $s_t$ for some $t \in [T]$, the procedure is to traverse the tree $\mathcal{T}$ up to the leaf node corresponding to $x_t$, and summing up the values at each node on the traversal path. The advantage is that we only access at most $m$ nodes, and add $m = \mathcal{O}(\log T)$ noise (instead of $\mathcal{O}(T)$).

The implementation of the private release of $(\mathbf{U}_i(t), \bar{\mathbf{u}}_i(t))$ is done by the ubiquitous tree-based mechanism for partial sums. We aggregate both into a single matrix $\boldsymbol{M}_{i,t} \in \mathbb{R}^{(d+1)\times(d+1)}$, by first concatenating: $\boldsymbol{A}_{i,t} := [\mathbf{x}_{i,1:t}, \mathbf{y}_{i,1:t}] \in \mathbb{R}^{t \times (d+1)}$, and then computing $\boldsymbol{M}_{i,t} = \boldsymbol{A}_{i,t}^\top \boldsymbol{A}_{i,t}$. Furthermore, the update is straightforward:

$$\boldsymbol{M}_{i,t+1} = \boldsymbol{M}_{i,t} + \left[\mathbf{x}_i(t)^\top \; y_i(t)\right]^\top \left[\mathbf{x}_i(t)^\top \; y_i(t)\right].$$

Recall that in our implementation, we only communicate in synchronization rounds (and not every round). Assume that two successive rounds of synchronization occur at time $t'$ and $t$. Then, at instant $t$, each agent $i$ inserts $\sum_{\tau=t'}^{t} [\mathbf{x}_i(\tau)^\top \; y_i(\tau)]^\top [\mathbf{x}_i(\tau)^\top \; y_i(\tau)]$ into $\mathcal{T}$, and computes $(\mathbf{U}_i(t), \bar{\mathbf{u}}_i(t))$ by summing up the entire path up to instant $t$ via the tree mechanism. Therefore, the tree mechanism accesses at most $m = 1 + \lceil \log_2 n \rceil$ nodes (where $n$ total rounds of communication occur until instant $T$), and hence noise that ensures each node guarantees $(\varepsilon/\sqrt{8m \ln(2/\delta)}, \delta/2m)$-privacy is sufficient to make the outgoing sequence $(\varepsilon, \delta)$-private. This is different from the setting in the joint DP single-agent bandit (Shariff & Sheffet, 2018), where observations are inserted *every* round, and not *only* for synchronization rounds. To make the partial sums private, a noise matrix is also added to each node in $\mathcal{T}$. We utilize additive Gaussian noise: at each node, we sample $\widehat{\boldsymbol{N}} \in \mathbb{R}^{(d+1)\times(d+1)}$, where each $\widehat{\boldsymbol{N}}_{i,j} \sim \mathcal{N}(0, \sigma_N^2)$, and $\sigma_N^2 = 16m(L^2+1)^2 \log(2/\delta)^2/\varepsilon^2$, and symmetrize it (see step 6 of Algorithm 11). The total noise $\mathbf{H}_i(t)$ is the sum of at most $m$ such terms, hence the variance of each element in $\mathbf{H}_i(t)$ is $\leq m\sigma_N^2$. We can bound the operator norm of the top-left $(d \times d)$-submatrix of each noise term. Therefore, to guarantee $(\varepsilon, \delta, M)$-federated DP, we require that with probability at least $1 - \alpha/nM$:

$$\|\mathbf{H}_i(t)\|_{\text{op}} \leq \Lambda = \frac{\sqrt{32}m(L^2+1)}{\varepsilon} \cdot \log\left(\frac{4}{\delta}\right)\left(4\sqrt{d} + 2\log\left(\frac{2nM}{\alpha}\right)\right).$$

**Remark 6.4** (Privacy Guarantee)**.** The procedure outlined above guarantees that each of the $n$ outgoing messages $(\mathbf{U}_i(t), \bar{\mathbf{u}}_i(t))$ (where $t$ is a synchronization round) for any agent $i$ is $(\varepsilon, \delta)$-differentially private. This analysis considers the $L_2$-sensitivity with respect to a single differing observation, i.e., $(\mathbf{x}, y)$ and not the entire message itself, i.e., the complete sequence $(\mathbf{x}_i(\tau), y_i(\tau))_{\tau=t'}^{t}$ (where $t'$ and $t$ are successive synchronization rounds), which may potentially have $O(t)$ sensitivity and warrants a detailed analysis. While our analysis is sufficient for the user-level adversary model, there may be settings where privacy is required at the message-level as well, which we leave as future work.

However, as noted by (Shariff & Sheffet, 2018), this $\mathbf{H}_i(t)$ would not always be PSD. To ensure that it is always PSD, we can shift each $\mathbf{H}_i(t)$ by $2\Lambda\mathbf{I}$, giving a bound on $\rho_{\max}$. Similarly, we can obtain bounds on $\rho_{\min}$ and $\kappa$:

**Proposition 6.3.** *Fix $\alpha > 0$. If each agent $i$ samples noise parameters $\mathbf{H}_i(t)$ and $\mathbf{h}_i(t)$ using the tree-based Gaussian mechanism mentioned above for all $n$ trials of $\bar{\boldsymbol{\sigma}}$ in which communication occurs, then the following $\rho_{\min}, \rho_{\max}$ and $\kappa$ are $(\alpha/2nM, \bar{\boldsymbol{\sigma}})$-accurate bounds:*

$$\rho_{\min} = \Lambda,\ \rho_{\max} = 3\Lambda,\ \kappa \leq \sqrt{\frac{m(L^2+1)}{\varepsilon\sqrt{2}}\left(\sqrt{d} + 2\log\left(\frac{2nM}{\alpha}\right)\right)}.$$

*Proof.* The proof follows directly from Shariff and Sheffet (Shariff & Sheffet, 2018), Proposition 11 but we take the cumulative sum across all agents. □

**Remark 6.5** (Strategyproof Analysis)**.** The mechanism presented above assumes the worst-case communication, i.e., synchronization occurs every round, therefore at most $T$ partial sums will be released, $m = 1 + \lceil \log T \rceil$. This is not true for general settings, where infrequent communication would typically require $m = \mathcal{O}(\log \log T)$, for instance, when we only synchronize $\mathcal{O}(M \log T)$ times. However, if any agent is byzantine and requests a synchronization every trial, $m$ must be $1 + \lceil \log T \rceil$ to ensure privacy. In case the protocol is fixed in advance (i.e., synchronization occurs on a pre-determined set $\bar{\boldsymbol{\sigma}}$ of $n$ rounds), then we can set $m = 1 + \lceil \log n \rceil$ to achieve the best utility at the desired privacy budget.

**Remark 6.6** (Decentralized Protocol)**.** FedLinUCB-Network requires the following bounds for $\rho_{\min}, \rho_{\max}$ and $\kappa$, with $m = 1 + \lceil \log(T/\gamma) \rceil$.

$$\rho_{\min} = \Lambda_d,\ \rho_{\max} = 3\Lambda_d,\ \kappa \leq \sqrt{\frac{m(L^2+1)}{\varepsilon\sqrt{2}}\left(\sqrt{d} + 2\log\left(\frac{2nM\gamma}{\alpha}\right)\right)},\ \text{where}$$

$$\Lambda_d = \frac{\sqrt{32}m(L^2+1)}{\varepsilon} \cdot \log\left(\frac{4}{\delta}\right)\left(4\sqrt{d} + 2\log\left(\frac{2nM\gamma}{\alpha}\right)\right).$$

An additional term of $\log(\gamma)$ appears in $\Lambda$ and $\kappa$, since we need to now maintain $\gamma$ partial sums with at most $T/\gamma$ elements. Unsurprisingly, there is no dependence on the network $G$, as privatization is done at the source itself.

**Discussion**. In the decentralized version of the algorithm, each agent maintains $\gamma$ sets of parameters that are used in a round-robin manner. Note that this implies that each parameter set is used at most $T/\gamma$ times per agent. This implies that in the worst case, each agent will communicate at most $T/\gamma$ partial sums related to this parameter set, hence needing at most $1 + \lceil \log(T/\gamma) \rceil$ separate nodes of the tree-based mechanism for the particular set of parameters, which leads to the additional $\log \gamma$ in the bounds as well. Next, when determining $\kappa$, each agent will not require $M$-accurate bounds, since it communicates with only $|C|$ other agents. However, in the worst case, a centrally-positioned node belongs to a small clique but can still communicate with all other $M - 1$ nodes (i.e., they can still obtain the partial sums broadcasted), and hence we maintain the factor $M$ to ensure privacy.

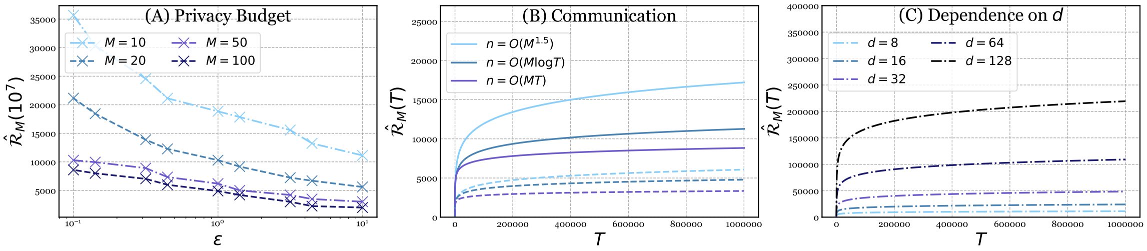


Figure 6-2: A comparison of distributed FedLinUCB on 3 different axes. Fig. (A) describes the variation in asymptotic per-agent regret for varying privacy budget $\varepsilon$ (where $\delta = 0.1$); (B) describes the effect of $n$ in private (solid) vs. non-private (dashed) settings; (C) describes the effect of $d$ in per-agent regret in the private setting ($n = O(M \log T), \varepsilon = 1, \delta = 0.1$). Experiments averaged over 100 runs.

**Corollary 6.1** (($\varepsilon, \delta$)-dependent Regret)**.** FedLinUCB *with the* Privatizer *subroutine in Alg. 11 run with privacy parameters* $(\varepsilon, \delta)$, *obtains a group regret of* $\widetilde{O}\left(\left(\sqrt{\frac{d^{3/2}}{\varepsilon}} + d\right)\sqrt{MT}\right)$ *in the distributed setting and* $\widetilde{O}\left(\left(\sqrt{\frac{d^{3/2}}{\varepsilon}} + d\right)\sqrt{(\bar{\chi}(G_\gamma)\cdot\gamma)MT}\right)$ *regret in the decentralized setting.*

*Proof.* Follows directly by substituting the values of $\rho_{\min}, \rho_{\max}$ and $\kappa$ into the respective regret bounds. □

## 6.6 Experiments

In the experiments we focus on the distributed environment for simplicity, and on the variation of the regret with communication complexity and privacy budget. For all experiments, we assume $L = S = 1$. For any $d$, we randomly fix $\boldsymbol{\theta}_\star \in \mathcal{B}_d(1)$. Each $\mathcal{D}_i(t)$ is generated as follows: we randomly sample $K \leq d^2$ actions $\mathbf{x}$, such that for $K - 1$ actions $0.5 \leq \langle \mathbf{x}, \boldsymbol{\theta}_\star \rangle \leq 0.6$ and for the optimal $\mathbf{x}^*, 0.7 \leq \langle \mathbf{x}^*, \boldsymbol{\theta}_\star \rangle \leq 0.8$ such that $\Delta \geq 0.1$ always. $y_i(t)$ is sampled from $\text{Ber}(\langle \mathbf{x}_i(t), \boldsymbol{\theta}_\star \rangle)$ such that $\mathbb{E}[y_i(t)] = \langle \mathbf{x}_i(t), \boldsymbol{\theta}_\star \rangle$ and $|y_i(t)| \leq 1$. Results are in Fig. 6-2, and experiments are averaged on 100 trials.

**Experiment 1: Privacy Budget**. In this setting, we set $n = \mathcal{O}(M \log T), d = 10$ (to balance communication and performance), and plot the average per-agent regret after $T = 10^7$ trials for varying $M$ and $\varepsilon$, while keeping $\delta = 0.1$. Figure 6-2A describes the results, competitive even at large privacy budget.

**Experiment 2: Communication**. We examine the regret curves for $M = 100$ agents on $n = \mathcal{O}(M^{1.5}), \mathcal{O}(M \log T), \mathcal{O}(MT)$ communication for both private ($\varepsilon = 1$) and non-private settings. We observe a tradeoff as highlighted in Remark 6.1.

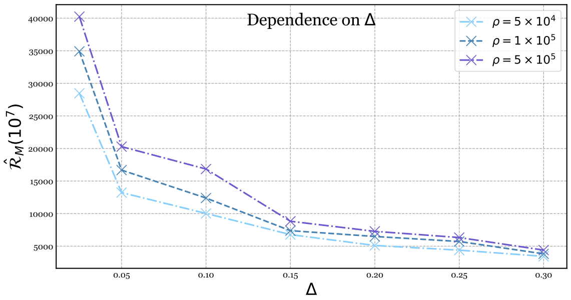


Figure 6-3: An experimental comparison of distributed FedLinUCB-Dist on varying the minimum gap between arms Δ, for various values of the privacy budget $\rho_{\text{min}}$.

**Experiment 3: Dependence on** $d$. As an ablation, we provide the average per-agent regret curves for $M = 100$ by varying the dimensionality $d$. We observe essentially a quadratic dependence.

**Experiment 4: Varying** Δ. Finally, we conduct ablations with variations in Δ. Figure 6-3 summarizes the results when run on $M = 10$ agents for different privacy budgets and arm gaps. As expected, the overall regret decreases as the gap increases, and the algorithm becomes less sensitive to privacy budget altogether.

## 6.7 Related Work

**Multi-Agent and Distributed Bandits**. Bandit learning in multi-agent distributed settings has received attention from several academic communities. Channel selection in distributed radio networks consider the (context-free) multi-armed bandit with collisions (Liu & Zhao, 2010a,b,c) and cooperative estimation over a network with delays (Landgren et al., 2016a,b, 2018). For the contextual case, recent work has considered non-private estimation in networks with delays (Dubey & Pentland, 2020a,c; Wang et al., 2019a; Korda et al., 2016). A closely-related problem is that of bandits with side information (Cesa-Bianchi et al., 2013; Buccapatnam et al., 2014), where the single learner obtains multiple observations every round, similar to the multi-agent communicative setting. Our work builds on the remarkable work of Abbasi-Yadkori et al. (2011), which in turn improves the LinUCB algorithm introduced in Li et al. (2010).

**Differential Privacy**. Our work utilizes *differential privacy*, a cryptographically-secure privacy framework introduced by Dwork (2011); Dwork & Roth (2014) that requires the behavior of an algorithm to fluctuate only slightly (in probability) with any change in its inputs. A technique to maintain differential privacy for the continual release of statistics was introduced in Chan et al. (2010); Dwork & Smith (2010), known as the *tree-based* algorithm that privatizes the partial sums of $n$ entries by adding at most $\log n$ noisy terms. This method has been used to preserve privacy across several online learning problems, including convex optimization (Jain et al., 2012; Iyengar et al., 2019), online data analysis (Hardt & Rothblum, 2010), collaborative filtering (Calandrino et al., 2011) and data aggregation (Chan et al., 2012). In the single-agent bandit setting, differential privacy using tree-based algorithms have been explored in the multi-armed case (Thakurta & Smith, 2013; Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016) and the contextual case (Shariff & Sheffet, 2018). In particular, our work builds on the setting from Shariff & Sheffet (2018), extending their single-agent results to the federated multi-agent setting. For the multi-agent multi-armed (i.e., context-free) bandit problem, differentially private algorithms have been devised for the distributed (Tossou & Dimitrakakis, 2015b) and decentralized (Dubey & Pentland, 2020d) settings. Empirically, the advantages of privacy-preserving contextual bandits has been demonstrated in the work of Malekzadeh *et al.*(Malekzadeh et al., 2019), and Hannun *et al.*(Hannun et al., 2019) consider a centralized multi-agent contextual bandit algorithm that use secure multi-party computations to provide privacy guarantees (both works do not have any regret guarantees). To the best of our knowledge, this paper is the first to investigate differential privacy for contextual bandits in the federated learning setting, in both distributed and decentralized environments.

## 6.8 Discussion

From a technical perspective, we make improvements along several fronts – while there has been prior work on multi-agent private linear bandits (Hannun et al., 2019; Malekzadeh et al., 2019) our work is the first to provide rigorous guarantees on private linear bandits in the multi-agent setting. Our work additionally provides the first algorithm with regret guarantees for contextual bandits in decentralized networks, extending the work of many on multi-armed bandits (Martínez-Rubio et al., 2019; Landgren et al., 2016a,b; Dubey & Pentland, 2020d,a). Specifically, we introduce an analysis of multi-agent contextual bandit algorithms when communication is *data-dependent*, which can be applicable in many scenarios beyond our bandit setting.

There are several unresolved questions in this line of work, as highlighted by our algorithm itself. In the decentralized case, our algorithm obtains a communication overhead of $\mathcal{O}(\sqrt{\bar{\chi}(G_\gamma) \cdot \gamma})$, which we comment is an artefact of our proof technique and can potentially be improved to smaller quantities such as the independence number of the power graph $\alpha(G_\gamma)$ by more communication budgets, as suggested by our results in the multi-armed setting. Establishing the optimal rates for this problem and examining asynchronous methods are valuable lines of future inquiry.

## 6.9 Full Proofs

### 6.9.1 Proof of Proposition 9.3

We denote the number of (common) bandit trials between two rounds of communication as an epoch. Let $n' = \sqrt{\frac{DT}{d\log(\rho_{\max}/\rho_{\min} + TL^2/(d\rho_{\min}))}} + 1$. There can be at most $\lceil T/n' \rceil$ rounds of communication such that they occur after an epoch of length $n'$. On the other hand, if there is any round of communication succeeding an epoch (that begins, say at time $t$) of length $< n'$, then for that epoch, $\log \frac{\det(\mathbf{S}_i(t+n'))}{\det(\mathbf{S}_i(t))} > \frac{D}{n'}$. Let the communication occur at a set of rounds $t'_1, ..., t'_n$. Now, since:

$$\sum_{i=1}^{n-1} \log \frac{\det\left(\mathbf{S}_i(t'_{i+1})\right)}{\det\left(\mathbf{S}_i(t_i)\right)} = \log \frac{\det\left(\mathbf{S}_i(T)\right)}{\det\left(\mathbf{S}_i(0)\right)} \leq d \cdot \log\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{d\rho_{\min}}\right),$$

We have that the total number of communication rounds succeeding epochs of length less than $n'$ is upper bounded by $\log \frac{\det(\mathbf{S}_{i,T})}{\det(\mathbf{S}_{i,0})} \leq d\log(\rho_{\max}/\rho_{\min} + TL^2/(d\rho_{\min})) \cdot (n'/D)$. Combining both the results together, we have the total rounds of communication as:

$$\begin{aligned} n &\leq \lceil T/n' \rceil + \lceil d\log(\rho_{\max}/\rho_{\min} + TL^2/(d\rho_{\min})) \cdot (n'/D) \rceil \\ &\leq T/n' + d\log(\rho_{\max}/\rho_{\min} + TL^2/(d\rho_{\min})) \cdot (n'/D) + 2. \end{aligned}$$

Replacing $n'$ from earlier gives us the final result.

### 6.9.2 Proof of Lemma 6.2

At every round, each agent $i$ selects an "optimistic" action $\mathbf{x}_i(t)$ such that,

$$(\mathbf{x}_i(t), \bar{\boldsymbol{\theta}}_{i,t}) = \underset{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{D}_i(t) \times \mathcal{E}_{i,t}}{\arg\max} \langle \mathbf{x}, \boldsymbol{\theta} \rangle .$$

Let $\mathbf{x}_i^\star(t)$ be the optimal action at time $t$ for agent $i$, i.e., $\mathbf{x}_i^\star(t) = \arg\max_{\mathbf{x} \in \mathcal{D}_i(t)} \langle \mathbf{x}, \boldsymbol{\theta}_\star \rangle$. We can then decompose the immediate pseudoregret $r_{i,t}$ for agent $i$ as the following.

$$\begin{aligned} r_{i,t} &= \langle \mathbf{x}_i^\star(t), \boldsymbol{\theta}_\star \rangle - \langle \mathbf{x}_i(t), \boldsymbol{\theta}_\star \rangle \\ &\leq \langle \mathbf{x}_i(t), \bar{\boldsymbol{\theta}}_{i,t} \rangle - \langle \mathbf{x}_i(t), \boldsymbol{\theta}_\star \rangle && \text{(Since } (\mathbf{x}_i(t), \bar{\boldsymbol{\theta}}_{i,t}) \text{ is optimistic)} \\ &= \langle \mathbf{x}_i(t), \bar{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_\star \rangle \end{aligned}$$

$$
\begin{aligned}
&= \left\langle \mathbf{V}_i(t)^{-1/2}\mathbf{x}_i(t), \mathbf{V}_i(t)^{1/2}\left(\bar{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_\star\right)\right\rangle && (\mathbf{V}_i(t) \succcurlyeq \mathbf{0}) \\
&\leq \|\mathbf{x}_i(t)\|_{\mathbf{V}_i(t)^{-1}} \|\bar{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_i(t)} && \text{(Cauchy-Schwarz)} \\
&\leq \|\mathbf{x}_i(t)\|_{\mathbf{V}_i(t)^{-1}} \left(\|\bar{\boldsymbol{\theta}}_{i,t} - \tilde{\boldsymbol{\theta}}_{i,t}\|_{\mathbf{V}_i(t)} + \|\tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_\star\|_{\mathbf{V}_i(t)}\right) && \text{(Triangle inequality)} \\
&\leq 2\beta_{t,i} \|\mathbf{x}_i(t)\|_{\mathbf{V}_i(t)^{-1}} && (\text{Since } \bar{\boldsymbol{\theta}}_{i,t}, \boldsymbol{\theta}_\star \in \mathcal{E}_{i,t}) \\
&\leq 2\bar{\beta}_T \|\mathbf{x}_i(t)\|_{\mathbf{V}_i(t)^{-1}}. && \text{(By Proposition 6.1)}
\end{aligned}
$$

### 6.9.3 Proof of Theorem 6.3

The proof for this setting is similar to the distributed variant. We can first partition the power graph $G_\gamma$ of the communication network $G$ into a clique cover $\mathcal{C} = \cup_i C_i$. The overall regret can then be decomposed as the following.

$$
\begin{aligned}
\mathfrak{R}(T) &= \sum_{i=1}^{M}\sum_{t=1}^{T} r_{i,t} \\
&= \sum_{C\in\mathcal{C}}\sum_{i\in C}\sum_{t=1}^{T} r_{i,t} \\
&= \sum_{C\in\mathcal{C}} \text{Regret}_C(T).
\end{aligned}
$$

Here, $\text{Regret}_C(T)$ denotes the cumulative pseudoregret of all agents within the clique $C$. It is clear that since there is no communication between agents in different cliques, their behavior is independent and we can analyse each clique separately. Now, consider $\tau$ sequences given by $s_1, ..., s_\tau$, where $s_i = (i, i+\tau, i+2\tau, ..., i + (\lceil T/\tau \rceil - 1)\tau)$. For any clique $C$ we can furthermore decompose the regret as follows.

$$
\begin{aligned}
\text{Regret}_C(T) &= \sum_{i\in C}\sum_{t=1}^{T} r_{i,t} \\
&= \sum_{i\in C}\sum_{j=1}^{\tau}\sum_{t\in s_j} r_{i,t} \\
&= \sum_{j=1}^{\gamma} \text{Regret}_{C,j}(T).
\end{aligned}
$$

Here $\text{Regret}_{C,j}(T)$ denotes the cumulative pseudoregret of the $j^{th}$ subsequence. We will now bound each of these regret terms individually, with an identical argument as the distributed case. This can be done since the behavior of the algorithm in each of these subsequences is independent: each sequence $s_j$ corresponds to a different Gram matrix and least-squares estimate, and is equivalent to each agent running $\gamma$ parallel bandit algorithms. We now bound each of the regret terms $\text{Regret}_{C,j}(T)$ via an identical argument as the distributed case. Let $\bar{\boldsymbol{\sigma}}_j$ be the subsequence of $[T]$ containing every $j^{th}$ index, i.e., $\bar{\boldsymbol{\sigma}}_j = j, j+\gamma, j+2\gamma, ..., j+\lceil T/\gamma - 1\rceil\gamma$. For any clique $C$, and index $j$ we compare the pulls of each agent within $C$ to the pulls taken by an agent pulling arms in a round-robin manner $(\mathbf{x}_{i,j})_{i\in C, j\in\bar{\boldsymbol{\sigma}}_j}$. This corresponds to a total of $|C|T/\gamma$ pulls.

Now, it is crucial to see that according to Algorithm 10, if a signal to synchronize has been sent by any agent $i \in C$ at any time $t$ (belonging to subsequence $j$), then the $j^{th}$ parameter set $\mathbf{V}_i(t)^{(j)}$ is used first at time $t$ (by each agent in $C$), then at time $t+\gamma$ (at which time each agent in $C$ broadcasts their parameters, since by this time each other agent will have received the signal to synchronize, as they are at most distance $\gamma$ apart), and next at time $t+2\gamma$, upon which they will be fully synchronized. Now, if we denote the rounds $n_{C,j} \subset \bar{\boldsymbol{\sigma}}_j$ as the rounds in which each agent broadcasted their $j^{th}$ parameter sets, then for any $\tau \in n_{C,j}$, all agents in $C$ have identical $j^{th}$ parameter sets at instant $\tau+1$, and for each instant $\tau - 1$, all agents in $C$ will obey the synchronization threshold for the $j^{th}$ set of parameters, (log-det condition).

Now, we denote by $\boldsymbol{W}_{i,t}^{(j,C)}$ the Gram matrix obtained by the hypothetical round-robin agent for subsequence $j$ in clique $C$. By Lemma 6.1, we have that

$$\sum_{t\in\bar{\boldsymbol{\sigma}}_j}\sum_{i\in C}\|\mathbf{x}_i(t)\|^2_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}} \leq 2d\log\left(1+\frac{TL^2}{\gamma d\rho_{\min}}\right).$$

After any round $T_k$ of synchronization, consider the cumulative Gram matrices of all observations obtained until that round as $\mathbf{V}_k^{(j,C)}, k = 1, ..., n_{j,C}-1$, regularized by $|C|\rho_{\min}\mathbf{I}$, i.e., $\mathbf{V}_k^{(j,C)} = \sum_{i\in C}\sum_{t\in\bar{\boldsymbol{\sigma}}_j : t<T_k}\mathbf{x}_i(t)\mathbf{x}_i(t)^\top + |C|\rho_{\min}\mathbf{I}$. Finally, let $\mathbf{V}_p$ denote the (regularized) $j^{th}$ Gram matrix with all trials from agents within $C$ at time $T$, and $\mathbf{V}_0^{(j,C)} = |C|\rho_{\min}\mathbf{I}$. Therefore, we have that $\det(\mathbf{V}_0^{(j,C)}) = (|C|\rho_{\min})^d$, and that $\det(\mathbf{V}_{n_{j,C}}) \leq \left(\frac{\text{tr}(\mathbf{V}_{n_{j,C}}^{(j,C)})}{d}\right)^d \leq (|C|\rho_{\max} + |C|TL^2/(\gamma d))^d$. Therefore, for any $\nu > 1$,

$$\log_\nu\left(\frac{\det(\mathbf{V}_{n_{j,C}}^{(j,C)})}{\det(\mathbf{V}_0^{(j,C)})}\right) \leq d\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{\gamma d\rho_{\min}}\right).$$

Let $R = \left\lceil d \log_{\nu}\left(\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{\gamma d \rho_{\min}}\right)\right\rceil$. It follows that in all but $R$ periods between synchronization,

$$1 \le \frac{\det(\mathbf{V}_k^{(j,C)})}{\det(\mathbf{V}_{k-1}^{(j,C)})} \le \nu.$$

We consider the event $E$ to be the period $k$ when Equation $\mathbf{V}_i(t)^{(j,C)} \succcurlyeq \mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I}$ holds. Now, for any $T_{k-1} \le t \le T_k$, consider the immediate pseudoregret for any agent $i$. By Lemma 6.2, we have for any agent $i \in C$ and $t \in \bar{\boldsymbol{\sigma}}_j$:

$$\begin{aligned}
r_{i,t} &\le 2\bar{\beta}_T \|x_{i,t}\|_{(\mathbf{V}_i(t)^{(j,C)})^{-1}} \\
&\le 2\bar{\beta}_T \|x_{i,t}\|_{(\mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I})^{-1}} && (\mathbf{V}_i(t)^{(j,C)} \succcurlyeq \mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I}) \\
&\le 2\bar{\beta}_T \|x_{i,t}\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}} \cdot \sqrt{\frac{\det(\boldsymbol{W}_{i,t}^{(j,C)})}{\det(\mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I})}} \\
&\le 2\bar{\beta}_T \|x_{i,t}\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}} \cdot \sqrt{\frac{\det(\mathbf{V}_k^{(j,C)})}{\det(\mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I})}} && (\mathbf{V}_k^{(j,C)} \succcurlyeq \boldsymbol{W}_{i,t}^{(j,C)}) \\
&\le 2\bar{\beta}_T \|x_{i,t}\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}} \cdot \sqrt{\frac{\det(\mathbf{V}_k^{(j,C)})}{\det(\mathbf{V}_{k-1}^{(j,C)})}} && (\mathbf{G}_i(t)^{(j,C)} + |C|\rho_{\min}\mathbf{I} \succcurlyeq \mathbf{V}_{k-1}^{(j,C)}) \\
&\le 2\nu\bar{\beta}_T \|x_{i,t}\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}}. && (\text{Event } E \text{ holds})
\end{aligned}$$

Now, we can sum up the immediate pseudoregret over all such periods where $E$ holds to obtain the total regret for these periods. With probability at least $1-\alpha$,

$$\begin{aligned}
\text{Regret}_{C,j}(T,E) &= \sum_{i\in C}\sum_{t\in\bar{\boldsymbol{\sigma}}_j:\text{E is true}} r_{i,t} \\
&\le \sqrt{\frac{|C|T}{\gamma}\left(\sum_{i\in C}\sum_{t\in\bar{\boldsymbol{\sigma}}_j:\text{E is true}} r_{i,t}^2\right)} \\
&\le 2\nu\bar{\beta}_T\sqrt{\frac{|C|T}{\gamma}\left(\sum_{i\in C}\sum_{t\in\bar{\boldsymbol{\sigma}}_j:\text{E is true}} \|\mathbf{x}_i(t)\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}}\right)} \\
&\le 2\nu\bar{\beta}_T\sqrt{\frac{|C|T}{\gamma}\left(\sum_{i\in C}\sum_{t\in\bar{\boldsymbol{\sigma}}_j} \|\mathbf{x}_i(t)\|_{(\boldsymbol{W}_{i,t}^{(j,C)})^{-1}}\right)}
\end{aligned}$$

$$\leq 2\nu\bar{\beta}_T\sqrt{2\frac{|C|T}{\gamma}d\log_\nu\left(1+\frac{TL^2}{\gamma d\rho_{\min}}\right)}.$$

Now let us consider the periods in which $E$ does not hold for any subsample $j$. In any such period between synchronization of length $t_k = T_k - T_{k-1}$, we have, for any agent $i$, the regret accumulated given by:

$$\begin{aligned}
\text{Regret}_{C,j}([T_{k-1}, T_k]) &= \sum_{t=T_{k-1}}^{T_k}\sum_{i\in C} r_{i,t} \\
&\leq 2\nu\bar{\beta}_T\left(\sum_{i\in C}\sqrt{t_k\sum_{t=T_{k-1}}^{T_k}\|\mathbf{x}_i(t)\|^2_{(\mathbf{V}_i(t)^{(j)})^{-1}}}\right) \\
&\leq 2\nu\bar{\beta}_T\left(\sum_{i\in C}\sqrt{t_k\log_\nu\left(\frac{\det(\mathbf{V}^j_{i,t+t_k})}{\det(\mathbf{V}^j_{i,t})}\right)}\right) \\
&\leq 2\nu\bar{\beta}_T\left(\sum_{i\in C}\sqrt{t_k\log_\nu\left(\frac{\det(\mathbf{G}^j_{i,t+t_k}+|C|\rho_{\max}\mathbf{I})}{\det(\mathbf{G}^j_{i,t}+|C|\rho_{\min}\mathbf{I})}\right)}\right)
\end{aligned}$$

By Algorithm 9, we know that for all agents, $t_k\log_\nu\left(\frac{\det(\mathbf{G}_{i,t+t_k}+M\rho_{\max}\mathbf{I})}{\det(\mathbf{G}_i(t)+M\rho_{\min}\mathbf{I})}\right) \leq D$ (since there would be a synchronization round otherwise), therefore

$$\leq 2\nu\bar{\beta}_T|C|\sqrt{D}.$$

Now, note that of the total $p$ periods between synchronizations, only at most $R$ periods will not have event $E$ be true. Therefore, the total regret over all these periods can be bound as,

$$\begin{aligned}
\text{Regret}_{C,j}(T,\bar{E}) &\leq R\cdot 2\nu\bar{\beta}_T|C|\sqrt{D} \\
&\leq 2\nu\bar{\beta}_T|C|\sqrt{D}\left(d\log_\nu\left(\frac{\rho_{\max}}{\rho_{\min}}+\frac{TL^2}{d\rho_{\min}}\right)+1\right).
\end{aligned}$$

In a manner identical to the distributed case, we can obtain the total pseudoregret within a clique $C$ for subsampling index $j$ by choosing an appropriate value of $D$. Note that since the broadcast between agents happens at an additional delay of 1 trial, the value $D$ must be scaled by a factor of $1 + \sup_{\mathbf{x} \in \mathcal{D}_i(t)} \|\mathbf{x}\|_2^2$ to bound the additional term in the determinant (by the matrix-determinant lemma), giving us the extra $1 + L^2$ term from the proof. Finally, summing up the above over all $C \in \mathcal{C}$ and $j \in [\gamma]$ and noting that $|C| \leq M \forall C \in \mathcal{C}$, and that $|\mathcal{C}| = \bar{\chi}(G_\gamma)$ gives us the final result.

## 6.10 Algorithm Pseudocode

**Algorithm 9** FedLinUCB-Dist$(D, M, T, \rho_{\min}, \rho_{\max})$

```
**Initialization**: $\forall i$, set $\mathbf{S}_i(1) \leftarrow M\rho_{\min}\mathbf{I}, \mathbf{s}_i(1) \leftarrow \mathbf{0}, \widehat{\mathbf{Q}}_i(0) \leftarrow \mathbf{0}, \mathbf{U}_i(1) \leftarrow \mathbf{0}, \bar{\mathbf{u}}_i(1) \leftarrow \mathbf{0}$.
**for** For each iteration $t \in [T]$ **do**
    **for** For each agent $i \in [M]$ **do**
        Set $\mathbf{V}_i(t) \leftarrow \mathbf{S}_i(t) + \mathbf{U}_i(t), \tilde{\mathbf{u}}_i(t) \leftarrow \mathbf{s}_i(t) + \bar{\mathbf{u}}_i(t)$.
        Receive $\mathcal{D}_i(t)$ from environment.
        Compute regressor $\bar{\boldsymbol{\theta}}_i(t) \leftarrow \mathbf{V}_i(t)^{-1}\tilde{\mathbf{u}}_i(t)$.
        Compute $\beta_i(t)$ following Proposition 6.1.
        Select $\mathbf{x}_i(t) \leftarrow \arg\max_{\mathbf{x} \in \mathcal{D}_i(t)} \langle \mathbf{x}, \bar{\boldsymbol{\theta}}_i(t) \rangle + \beta_i(t) \|\mathbf{x}\|_{\mathbf{V}_i(t)^{-1}}$.
        Obtain $y_i(t)$ from environment.
        Update $\mathbf{U}_i(t+1) \leftarrow \mathbf{U}_i(t) + \mathbf{x}_i(t)\mathbf{x}_i(t)^\top, \mathbf{u}_i(t+1) \leftarrow \mathbf{u}_i(t) + \mathbf{x}_i(t)y_i(t)$.
        Update $\widehat{\mathbf{Q}}_i(t) \leftarrow \widehat{\mathbf{Q}}_i(t-1) + [\mathbf{x}_i(t)^\top \; y_i(t)]^\top [\mathbf{x}_i(t)^\top \; y_i(t)]$
        **if** $\log\det\left(\mathbf{V}_i(t) + \mathbf{x}_i(t)\mathbf{x}_i(t)^\top + M(\rho_{\max} - \rho_{\min})\mathbf{I}\right) - \log\det\left(\mathbf{S}_i(t)\right) \geq D/\Delta t_i$ **then**
            SYNCHRONIZE $\leftarrow$ TRUE.
        **end if**
        **if** SYNCHRONIZE **then**
            [$\forall$ AGENTS] Agent sends $\widehat{\mathbf{Q}}_i(t) \rightarrow$ PRIVATIZER and gets $\widehat{\mathbf{U}}_i(t+1), \widehat{\mathbf{u}}_i(t+1) \leftarrow$ PRIVATIZER.
            [$\forall$ AGENTS] Agent communicates $\widehat{\mathbf{U}}_i(t+1), \widehat{\mathbf{u}}_i(t+1)$ to controller.
            [CONTROLLER] Compute $\mathbf{S}(t+1) \leftarrow \sum_{i=1}^M \widehat{\mathbf{U}}_i(t+1), \mathbf{s}(t+1) \leftarrow \sum_{i=1}^M \widehat{\mathbf{u}}_i(t+1)$.
            [CONTROLLER] Communicate $\mathbf{S}(t+1), \mathbf{s}_i(t+1)$ back to agent.
            [$\forall$ AGENTS] $\mathbf{S}_i(t+1) \leftarrow \mathbf{S}(t+1), \mathbf{s}_i(t+1) \leftarrow \mathbf{s}(t+1)$.
            [$\forall$ AGENTS] $\widehat{\mathbf{Q}}_i(t+1) \leftarrow \mathbf{0}$.
        **else**
            $\mathbf{S}_i(t+1) \leftarrow \mathbf{S}_i(t), \mathbf{s}_i(t+1) \leftarrow \mathbf{s}_i(t), \Delta t_i \leftarrow \Delta t_i + 1$.
            $\Delta t_i \leftarrow 0, \mathbf{U}_i(t+1) \leftarrow \mathbf{0}, \bar{\mathbf{u}}_{i,t+1} \leftarrow \mathbf{0}$.
        **end if**
    **end for**
**end for**
```

**Algorithm 10** FedLinUCB-Network$(D, M, T, \rho_{\min}, \rho_{\max}, G, \gamma)$

```
Initialization: $\forall i, \forall g \in [\gamma]$, set $\mathbf{S}^g_{i,1} \leftarrow \mathbf{0}, \mathbf{s}^g_{i,1} \leftarrow \mathbf{0}, \mathbf{H}^g_{i,0} \leftarrow \mathbf{0}, \mathbf{h}^g_{i,0} \leftarrow \mathbf{0}, \mathbf{U}^g_{i,1} \leftarrow \mathbf{0}, \bar{\mathbf{u}}^g_{i,1} \leftarrow \mathbf{0}$.
for each iteration $t \in [T]$ do
    for each agent $i \in [M]$ do
        Set subsampling index $g \leftarrow t \mod \gamma$.
        Run lines 4-13 of Algorithm 1 with $\mathbf{S}^g_{i,t}, \mathbf{s}^g_{i,t}, \mathbf{U}^g_{i,t}, \bar{\mathbf{u}}^g_{i,t}, \mathbf{V}^g_{i,t}, \tilde{\mathbf{u}}^g_{i,t}, \mathbf{H}^g_{i,t}, \mathbf{h}^g_{i,t}$
        if $\log\left(\frac{\det\left(\mathbf{V}_i(t)^g + \mathbf{x}^g_{i,t}(\mathbf{x}^g_{i,t})^\top + M(\rho_{\max}-\rho_{\min})\mathbf{I}\right)}{\det\left(\mathbf{S}^g_{i,t}\right)}\right) \geq \frac{D}{(\Delta t_{i,g}+1)(1+L^2)}$ then
            REQUEST TO SYNCHRONIZE$(i, g) \leftarrow$ TRUE.
        end if
        if REQUEST TO SYNCHRONIZE$(i, g)$ then
            BROADCAST Message SYNCHRONIZE$(i, g, t)$ from agent $i$ at time $t$ to all neighbors.
        end if
        for message $m$ received at time $t$ by agent $i$ do
            if $m$ = SYNCHRONIZE$(i', g', t')$ then
                if $i'$ belongs to the same clique as $i$ and $t' \geq t - \gamma$ then
                    Agent sends $\widehat{\mathbf{Q}}^{(g')}_{i,t} \rightarrow$ PRIVATIZER and gets $\widehat{\mathbf{U}}^{(g')}_{i,t+1}, \widehat{\mathbf{u}}^{(g')}_{i,t+1} \leftarrow$ PRIVATIZER.
                    Agent broadcasts $\widehat{\mathbf{U}}^{(g')}_{i,t+1}, \widehat{\mathbf{u}}^{(g')}_{i,t+1}$ to all neighbors.
                end if
                if $m = \widehat{\mathbf{U}}^{(g')}_{i',t'+1}, \widehat{\mathbf{u}}^{(g')}_{i',t'+1}$ then
                    if $i'$ belongs to the same clique as $i$ and $t' \geq t - \gamma$ then
                        Agent updates $\mathbf{S}^{(g')}_{i,t+1}, \mathbf{S}^{(g')}_{i,t+1}$ with $\widehat{\mathbf{U}}^{(g')}_{i',t'+1}, \widehat{\mathbf{u}}^{(g')}_{i',t'+1}$.
                    end if
                end if
            end if
        end for
    end for
end for
```

**Algorithm 11** PRIVATIZER$(\varepsilon, \delta, M, T)$ for any agent $i$

```
Initialization:
If communication rounds $n$ are fixed a priori, set $m \leftarrow 1 + \lceil \log n \rceil$, else $m \leftarrow 1 + \lceil \log T \rceil$.
Create binary tree $\mathcal{T}$ of depth $m$.
for node $n$ in $\mathcal{T}$ do
    Sample noise $\widehat{\boldsymbol{N}} \in \mathbb{R}^{(d+1)\times(d+1)}$, where $\widehat{\boldsymbol{N}}_{ij} \sim \mathcal{N}\left(0, 16m(L^2+1)^2 \log(2/\delta)^2/\varepsilon^2\right)$.
    Store $\boldsymbol{N} = (\widehat{\boldsymbol{N}} + \widehat{\boldsymbol{N}}^\top)/\sqrt{2}$ at node $n$.
end for
Runtime:
for each communication round $t \leq n$ do
    Receive $\widehat{\mathbf{Q}}_i(t)$ from agent, and insert it into $\mathcal{T}$ (see (Jain et al., 2012), Alg. 5).
    Compute $\boldsymbol{M}_{i,t+1}$ using the least nodes of $\mathcal{T}$ (see (Jain et al., 2012), Alg. 5).
    Set $\widehat{\mathbf{U}}_i(t+1) = \mathbf{U}_i(t+1) + \mathbf{H}_i(t)$ as top-left $d \times d$ submatrix of $\boldsymbol{M}_{i,t+1}$.
    Set $\widehat{\mathbf{u}}_i(t+1) = \mathbf{u}_i(t+1) + \mathbf{h}_i(t)$ as first $d$ entries of last column of $\boldsymbol{M}_{i,t+1}$.
    Return $\widehat{\mathbf{U}}_i(t+1), \widehat{\mathbf{u}}_i(t+1)$ to agent.
end for
```

## Chapter 7

# Private Gaussian Process Bandit Optimization

### 7.1 Introduction

In this chapter, we discuss differentially private algorithms for Gaussian Process (GP) bandit optimization (Srinivas et al., 2009). GP bandit optimization is a sequential decision problem that has a variety of human-centered applications, e.g., clinical drug trials (Costabal et al., 2019; Park et al., 2013; Peterson et al., 2017), personalized shopping recommendations (Rohde et al., 2018; Zhou et al., 2019), news feed ranking (Agarwal et al., 2018; Letham & Bakshy, 2019; Vanchinathan et al., 2014). It is increasingly becoming desirable that algorithms interacting with such data maintain the privacy of the individuals whose information is used (Cummings & Desai, 2018).

Similar to the contextual bandit, GP bandit optimization involves learning a function $f$ via repeated interaction in rounds. At any round $t = 1, 2, ...$, the learner is presented with a *decision set* $\mathcal{D}_t \subset \mathbb{R}^d$ from which it must select an action $\mathbf{x}_t$ and obtain a random reward $y_t = f(\mathbf{x}_t) + \xi_t$, where $\xi_t$ is sampled i.i.d. from the environment. The algorithm selects actions in order to minimize regret $\mathfrak{R}(T) = \sum_t [f(\mathbf{x}^\star_t) - f(\mathbf{x}_t)]$, where $\mathbf{x}^\star_t = \max_{\mathbf{x} \in \mathcal{D}_t} f(\mathbf{x})$. Unlike prior chapters, we will only be considering the single-agent setting in this chapter, and we remark that the results obtained herein can be extended to the federated setting following similar techniques as Chapter 6.

This problem he problem is more challenging for general Gaussian Process optimization. Most application settings assume that the target function $f$ lies in a (potentially) infinite-dimensional reproducing kernel Hilbert space (RKHS), and the standard techniques for introducing privacy are inapplicable due to the *curse of dimensionality* (Liu & Guillas, 2017; Meeds & Welling, 2014): the posterior mean and variance for these methods require storing the sample path $(\mathbf{x}_t, y_t)_t$, and are $\Omega(t)$ to evaluate. Moreover, as the learnt function itself is dependent on the sample path (containing sensitive data), privatized release of the function is also a challenge (Smith et al., 2016). In this chapter, we propose algorithms that guarantee differential privacy with respect to continual observation during optimization, and also the private release of learnt parameters.

The contributions in this chapter can be listed as follows. First, we propose a generic framework (and regret bound) for GP bandits that utilizes a finite-dimensional $\epsilon-$uniform approximation of infinte-dimensional kernels and integrates random perturbations to the GP posterior, allowing for various no-regret private GP algorithms based on the kernel approximation method and privacy guarantee required.

Next, In the joint differentially private (JDP) setting (Definition 7.6), we propose a novel GP-UCB algorithm (Algorithm 12) for stationary kernels admitting a decomposable Fourier transform (Assumption 7.1) that satisfies $(\varepsilon, \delta)$-JDP while obtaining $\widetilde{\mathcal{O}}(\sqrt{T\gamma_T/\varepsilon})$[1] pseudoregret. This bound matches (up to logarithmic factors) the lower bound for isotropic kernels (Scarlett et al., 2017), and admits an identical dependence on $\varepsilon$ as linear bandits (Shariff & Sheffet, 2018). Thirdly, inspired by the recent interest in locally differentially private (LDP) methods (Bebensee, 2019), we present an algorithm that achieves $(\varepsilon, \delta)-$LDP with $\widetilde{\mathcal{O}}(T^{3/4}\sqrt{\gamma_T/\varepsilon})$ pseudoregret. We conjecture that the constraints from LDP necessitate the $\mathcal{O}(T^{1/4})$ departure from typical near-optimal regret (Remark 7.7).

Our approach can be coarsely summarized with two steps - we first project $f$ from its (infinite-dimesional) RKHS into a finite-dimensional approximating RKHS, following which, we directly perturb the posterior mean and variance of the resulting GP (in the approximating space) to ensure privacy without the curse of dimensionality, which allows us to provide a no-regret solution. Our approach additionally avoids the parameter release problem (Smith et al., 2016; Kusner et al., 2015) since we do not explicitly store the sample path for prediction, and rely instead on cumulative sums (Remark 7.1).

**Organization**. We first discuss crucial related work and introduce necessary notation

[1] $\gamma_T$ is the *maximum information gain*, see Definition 7.1.

and preliminaries, subsequent to which we introduce our general framework for GP-UCB using noisy approximate features. We discuss quadrature Fourier features and present our algorithm and its associated regret bounds. Next, we discuss the two models of privacy studied, and present privacy mechanisms followed by experimental comparisons.

## 7.2 Related Work

**Gaussian Process Bandits**. Gaussian Processes (Williams & Rasmussen, 2006) have been widely used for the bandit optimization of unknown functions in an RKHS. The seminal work of Srinivas et al. (2010) introduced the nonparameteric GP-UCB algorithm, that introduced contextual-bandit style confidence bounds for optimisation in infinite-dimensional RKHSes. A variant of the *expected improvement* decision rule (Močkus, 1975) was proposed via the GP-EI algorithm (Snoek et al., 2012). By a stronger martingale analysis, Chowdhury & Gopalan (2017) achieve the IGP-UCB algorithm, that improves GP-UCB regret by a factor of $\mathcal{O}(\ln^{3/2} T)$. For a family of isotropic squared-exponential $d$-dimensional kernels, Scarlett et al. (2017) establish lower bounds on the achievable regret of $\Omega(\sqrt{T(\log T)^{d+2}})$, which matches (ignoring polylogarithmic factors) the $\widetilde{\mathcal{O}}(\sqrt{T})$ rate achieved by IGP-UCB and GP-UCB. Our work relies on the research in approximate methods for kernel approximation, which has seen a lot of recent interest. The seminal work of Rahimi & Recht (2008) proposed random Fourier features (RFF) by a Monte-Carlo approximation of the Fourier basis, with additional work establishing finite-sample convergence rates (Avron et al., 2017). We propose a noisy variant of the more efficient quadrature Fourier features (QFF) (Munkhoeva et al., 2018) that have been previously employed in GP optimization with success (Mutny & Krause, 2018). An alternative approach based on sampling fewer points from the algorithm's history based on matrix sketching has been proposed in Calandriello et al. (2019).

**Differentially-Private Bandit Learning**. Differentially private (DP) methods for bandit optimisation have received significant attention recently. For the multi-armed bandit case, UCB and Thompson sampling algorithms have been proposed for pure-DP (Mishra & Thakurta, 2015), with subsequent improvements (Tossou & Dimitrakakis, 2015a). For the contextual linear bandit, Shariff & Sheffet (2018) introduce an algorithm that utilizes matrix perturbations that our work effectively generalizes to infinite-dimensional stationary GPs. Note that this algorithm is inapplicable for general GPs as it assumes that the features are finite-dimensional. See Basu et al. (2020) for a summary of regret bounds for private multi-armed bandits. For Gaussian process bandits and Bayesian Optimisation (BO), Kusner et al. (2015) consider the problem of *releasing* GP parameters *after* optimization under differential privacy constraints, by analysing the sensitivity of the final parameters. Our work handles a more challenging setting, where parameters must be private *throughout* the optimisation process. An application of DP to the Gaussian process regression problem was studied in the work of Smith et al. (2016), however with no regret guarantees.

## 7.3 Preliminaries

**Gaussian Process Bandit Optimization**. We consider the problem of sequential reward maximization under a fixed but unknown reward function $f : \mathcal{D} \to \mathbb{R}$ over a (potentially infinite) set of actions (arms) $\mathcal{D} \subset \mathbb{R}^d$. The problem proceeds in rounds $t = 1, 2, ..., T$ where, in each round, the objective is to select an action $\mathbf{x}_t \in \mathcal{D}_t$ and obtain a reward $y_t = f(\mathbf{x}_t) + \xi_t$ such that the cumulative reward $\sum_{t \in [T]} y_t$ is maximized depending on the history $(\mathbf{x}_\tau, y_\tau)_{\tau<t}$, and $\xi_t$ is sampled from a zero-mean sub-Gaussian distribution with parameter $\lambda$. Gaussian Process (GP) modeling proposes to use a Gaussian likelihood model for observations and a GP prior for the uncertainty over $f$. A Gaussian Process (GP) over $\mathcal{D}$, denoted by $\mathrm{GP}(\mu(\cdot), k(\cdot, \cdot))$ is a collection of random variables $(f(\mathbf{x}))_{\mathbf{x}\in\mathcal{D}}$ such that every finite subset of variables $(f(\mathbf{x}_\tau))_{\tau=1}^{t}$ is jointly Gaussian with mean $\mathbb{E}[f(\mathbf{x}_\tau)] = \mu(\mathbf{x}_\tau)$ and covariance $\mathbb{E}[(f(\mathbf{x}_\tau) - \mu(\mathbf{x}_\tau))(f(\mathbf{x}_{\tau'}) - \mu(\mathbf{x}_{\tau'}))] = k(\mathbf{x}_\tau, \mathbf{x}_{\tau'})$, $\tau, \tau' \in [t]$ where $k(\cdot, \cdot)$ is the kernel function associated with the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k(\mathcal{D})$ in which we assume $f$ has norm at most $B$, i.e., $\|f\|_k \leq B$. We use an initial prior distribution $\mathrm{GP}(0, \rho^2 k(\cdot, \cdot))$ for some $\rho > 0$. Consequently it is also assumed that the noise samples $\xi_t$ are drawn from $\mathcal{N}(0, \lambda\rho^2)$[2]. We then obtain that the observed samples $\mathbf{y}_t = (y_\tau)_{\tau<t}$ and $f(\mathbf{x})$ are jointly Gaussian given $\mathbf{X}_t = (\mathbf{x}_\tau)_{\tau<t}$,

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{y}_t \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \rho^2 k(\mathbf{x}, \mathbf{x}) & \rho^2 \mathbf{k}_t(\mathbf{x})^\top \\ \rho^2 \mathbf{k}_t(\mathbf{x}) & \rho^2(\mathbf{K}_t + \lambda \mathbf{I}) \end{bmatrix}\right).$$

Where $\mathbf{K}_t = (k(\mathbf{x}_\tau, \mathbf{x}_{\tau'}))_{\tau,\tau'}^{t,t}$ is the matrix of kernel evaluations at time $t$, and $\mathbf{k}_t(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), ..., k(\mathbf{x}_t, \mathbf{x})]^\top$ is the vector of kernel evaluations of any input $\mathbf{x}$. Conditioned on

[2] The algorithm only requires $\xi_t$ to be $\lambda$-sub-Gaussian, i.e., the *agnostic* setting (Srinivas et al., 2010).

$(\mathbf{x}_\tau, y_\tau)_{\tau<t}$, the posterior mean and variance of $f$ is given as,

$$\mu_t(\mathbf{x}) = k_t(\mathbf{x})^\top(\mathbf{K}_t + \lambda\mathbf{I})^{-1}\mathbf{y}_t, \tag{7.1}$$

$$\sigma_t^2(\mathbf{x}) = \Big(k(\mathbf{x},\mathbf{x}) - \mathbf{k}_t(\mathbf{x})^\top(\mathbf{K}_t + \lambda\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x})\Big). \tag{7.2}$$

The kernel $k(\cdot,\cdot)$ additionally admits a representation in terms of its feature space $\Phi$ such that $k(\mathbf{x},\mathbf{x}') = \Phi(\mathbf{x})^\top\Phi(\mathbf{x})$, where $\Phi : \mathbb{R}^d \to \mathbb{R}^m$ is the feature embedding. This provides an alternative representation of the posterior mean and variance,

$$\mu_t(\mathbf{x}) = (\mathbf{S}_t + \lambda\mathbf{I})^{-1}\Phi(\mathbf{X})^\top\mathbf{y}_t, \tag{7.3}$$

$$\sigma_t^2(\mathbf{x}) = \rho^2\Phi(\mathbf{x})^\top(\mathbf{S}_t + \lambda\mathbf{I})^{-1}\Phi(\mathbf{x}), \text{ for} \tag{7.4}$$

$$\mathbf{S}_t = \Phi(\mathbf{X}_t)^\top\Phi(\mathbf{X}_t), \Phi(\mathbf{X}_t) = [\Phi(\mathbf{x}_1)^\top, ..., \Phi(\mathbf{x}_{t-1})^\top]^\top. \tag{7.5}$$

$\Phi$ can potentially be infinite-dimensional (e.g., for squared-exponential $k$), and hence this representation is not applicable to many popular kernel families. The regret achieved by existing algorithms depends on the *maximum information gain*, a quantity that depends on the covariance structure of the feature space.

**Definition 7.1** (Information Gain Srinivas et al. (2010))**.** *For $y_t = f(\mathbf{x}_t) + \xi_t$, let $A \subset \mathcal{X}$ be a finite subset such that $|A| = T$. Let $\mathbf{y}_A = \mathbf{f}_A + \boldsymbol{\varepsilon}_A$ where $\mathbf{f}_A = (f(\boldsymbol{x}_i))_{\boldsymbol{x}_i \in A}$ and $\boldsymbol{\varepsilon}_A \sim \mathcal{N}(0, \rho^2)$. The information gain is $\gamma_T \triangleq \max_{A\subset\mathcal{X}:|A|=T} H(\mathbf{y}_A) - H(\mathbf{y}_A|f)$, where $H(\cdot)$ is the entropy of a random variable. For linear $k$, $\gamma_T = \mathcal{O}(d\log T)$. For RBF $k$, $\gamma_T = \mathcal{O}((\log T)^{d+1})$. For Matérn $k$ with $\nu > 1$, $\gamma_T = \mathcal{O}(T^{\frac{d(d+1)}{2\nu+d(d+1)}}(\log T))$.*

**Differential Privacy** (DP). Following the standard definition of DP, within the continual observation setting of sequential decision-making, this would imply that the algorithm be private with respect to all values $(\mathbf{x}_\tau, y_\tau)_{\tau=1}^T$ at each $t \in [T]$. However, as demonstrated in Shariff & Sheffet (2018), any algorithm DP with respect to $(\mathbf{x}_t, y_t)$ at the instance $t$ provably incurs $\Omega(T)$ regret. Therefore we adopt the notion of *joint* differential privacy, which does not require privacy with respect to the inputs $(\mathbf{x}_t, y_t)$ at each instant $t \in [T]$ (Section 7.5.1). We additionally consider the stronger notion of *local* DP, which additionally requires that the algorithm cannot access $(\mathbf{x}_\tau, y_\tau)_{\tau<t}$ directly (Section 7.5.2).

In comparison to the federated privacy definition of Chapter 6, this definition of differential privacy is different: the previous setting required *outgoing messages* to be private, whereas in this setting, we require that the algorithm obey either *joint* or *local* differential privacy with respect to the *personal* observations at all times. We make these definitions more precise in Sections 7.5.1 and 7.5.2.

## 7.4 Noisy Proximal Features & GP-UCB

The primary challenge in creating differentially-private algorithms for *bandit estimation* in arbitrary RKHSes is the curse of dimensionality - the two central quantities $\mu_t$ and $\sigma_t^2$ both require the point-wise kernel evaluations $(\mathbf{k}_t(\mathbf{x}))$ and the kernel Gram matrix $(\mathbf{K}_t)$ at all times, potentially requiring $\mathcal{O}(\sqrt{t})$ noise in order to preserve privacy. In this paper, we tackle this hurdle by optimizing $f$ under a surrogate RKHS $\mathcal{F}_m$ that of finite dimension $m$ instead of the original (potentially infinite-dimensional) RKHS $\mathcal{H}_k$. To ensure a reasonable bound on the regret, we require that $\mathcal{F}_m$ approximates $\mathcal{H}_k$ closely, as formalized below.

**Definition 7.2** (Uniform Approximation)**.** *Let $k : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ be a stationary kernel with associated RKHS $\mathcal{H}_k$, and $\Phi : \mathcal{D} \to \mathbb{R}^m$. Then $\Phi$ $\epsilon$-uniformly approximates $k$ iff $\sup_{\mathbf{x},\mathbf{x}' \in \mathcal{D}} |k(\mathbf{x},\mathbf{x}') - \Phi(\mathbf{x})^\top \Phi(\mathbf{x})| \leq \epsilon$. The corresponding **approximating space** defined by $\Phi$ is given by*

$$\mathcal{F}_m(\Phi) \triangleq \left\{ f(\cdot) = \boldsymbol{\theta}^\top \Phi(\cdot) | \boldsymbol{\theta} \in \mathbb{R}^m \right\}.$$

Therefore, if $\mathcal{F}_m$ (resp. $\Phi$) can approximate $\mathcal{H}_k$ without many features, one can devise an *approximate* Gaussian process algorithm directly using $\Phi$.

$$\mathbf{G}_t = \Phi(\mathbf{X}_t)^\top \Phi(\mathbf{X}_t) + \lambda \mathbf{I}, \mathbf{u}_t = \mathbf{G}_t^{-1} \Phi(\mathbf{X}_t)^\top \mathbf{y}_t.$$

These parameters allow us to obtain the posterior mean $\mu_t(\mathbf{x}) = \mathbf{u}_t^\top \Phi(\mathbf{x})$ and variance $\sigma_t^2(\mathbf{x}) = \rho^2 \|\Phi(\mathbf{x})\|^2_{\mathbf{G}_t^{-1}}$. However, these parameters are obviously not differentially private with respect to the sequences $(\mathbf{X}_t, \mathbf{y}_t)$. Similar to the previous chapter, an efficient way to achieve privacy is to ensure that at each instant $t$, $(\mathbf{G}_t, \mathbf{u}_t)$ are differentially-private with respect to the sequence $(\mathbf{x}_\tau, y_\tau)_{\tau<t}$, which can be achieved by carefully perturbing $(\mathbf{G}_t, \mathbf{u}_t)$ with random noise $(\mathbf{H}_t, \mathbf{h}_t)$ to create differentially-private parameters. While the exact form of $\mathbf{H}_t, \mathbf{h}_t$ will be specified by the nature of privacy (see Section 7.5), we can represent a variety of noise models by spectral bounds, summarized by the following abstraction.

**Definition 7.3** (Spectral Bounds on Noise)**.** *For a sequence of perturbations* $(\mathbf{H}_t)_{t=1}^T$ *and* $(\mathbf{h}_t)_{t=1}^T$*, the bounds* $0 < \lambda_{\min} \leq \lambda_{\max}$ *are* $(\zeta/2T)-$*accurate if with probability at least* $1 - \zeta/2T$*, for each* $t$ *in* $[T]$*:*

$$\|\mathbf{H}_t\| \leq \lambda_{\max}, \|\mathbf{H}_t^{-1}\| \leq 1/\lambda_{\min}, \|\mathbf{h}_t\|_{\mathbf{H}_t^{-1}} \leq \kappa.$$

Let us use the shorthand $\mathbf{G}_t = \mathbf{S}_t + \lambda\mathbf{I}$, where $\mathbf{S}_t = \Phi(\mathbf{X}_t)^\top\Phi(\mathbf{X}_t)$. The perturbed $\mathbf{S}_t$ and $\mathbf{u}_t$ are given as $\widetilde{\mathbf{S}}_t = \mathbf{S}_t + \mathbf{H}_t, \widetilde{\mathbf{u}}_t = \mathbf{u}_t + \mathbf{h}_t$ for any sequence $(\mathbf{H}_t, \mathbf{h}_t)$.

### 7.4.1 GP-UCB **with Noisy Proximal Features**

Our algorithm is built on the GP-UCB algorithm (Srinivas et al., 2010) that constructs a confidence ellipsoid around the posterior $\widetilde{\mu}_t$ such that the function $f$ lies within the confidence ellipsoid with high probability. The key observation, is that we do not need to optimize for $f$ directly. Given an $\epsilon$-uniformly approximating feature $\Phi$ (resp. $\mathcal{F}_m$), then the following result guarantees the existence of a function close to $f$ in $\mathcal{F}_m$.

**Lemma 7.1** (Existence of Proximal Space (Lemma 4 of Mutny & Krause (2018)))**.** *Let* $k$ *be a kernel defining the RKHS* $\mathcal{H}_k$ *and* $f \in \mathcal{H}_k$*, such that the spectral characteristic function is bounded by* $B$*. Assuming that the defining points of* $f$ *come from the set* $\mathcal{D}$*, let* $\mathcal{F}_m$ *be an approximating space with a mapping* $\Phi$ *such that this mapping is an* $\epsilon$*-approximation to the kernel* $k$*. Then there exists* $\widehat{\mu} \in \mathcal{F}_m$ *(with corresponding feature* $\widehat{\boldsymbol{\theta}}$ *such that* $\widehat{\mu}(\mathbf{x}) = \langle\widehat{\boldsymbol{\theta}}, \Phi(\mathbf{x})\rangle$*), such that* $\sup_{\mathbf{x}\in\mathcal{D}} |\widehat{\mu}(\mathbf{x}) - f(\mathbf{x})| \leq B\epsilon$*.*

Hence we know that there exists a fixed point $\widehat{\mu} \in \mathcal{F}_m$ such that $\sup_{\mathbf{x}\in\mathcal{D}} |\widehat{\mu}(\mathbf{x}) - f(\mathbf{x})| \leq B\epsilon$. This implies that the regret incurred at any instant $t$ when optimizing for $f$ is at most $B\epsilon$ larger than the regret obtained when optimizing for $\widehat{\mu}$. We therefore optimize directly in the surrogate space $\mathcal{F}_m$ to learn $\widehat{\mu}$. GP-UCB with noisy approximate features selects, for a sequence $(v_t)_{t=1}^T$, the action $\mathbf{x}_t \in \mathcal{D}_t$ determined as:

$$\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}_t} \widetilde{\mu}_t(\mathbf{x}) + v_t^{1/2} \cdot \widetilde{\sigma}_t(\mathbf{x}).$$

The sequence $(v_t)_{t=1}^T$ is chosen such that $\widetilde{\mu}_t(\mathbf{x})$ is close to $\widehat{\mu}(\mathbf{x})$ with high probability. To accomplish this, we present the central result as follows.

**Theorem 7.1** ($v_t$ concentration)**.** *Let $\lambda_{\min}, \lambda_{\max}$ and $\kappa$ be $(\zeta/2T)$ -accurate and regularizers $\mathbf{H}_t \succcurlyeq 0\ \forall t \in [T]$ are PSD. Let $\widehat{\mu}$ be a function in the RKHS $\mathcal{F}_m$ that $\epsilon$-approximates $f \in \mathcal{H}_k$ (Lemma 7.1). Then, with probability at least $1-\zeta/2$, for any $\mathbf{x} \in \mathcal{D}$ we have for each $t \in [T]$ simultaneously,*

$$|\widehat{\mu}(\mathbf{x}) - \widetilde{\mu}_t(\mathbf{x})| \leq \widetilde{\sigma}_t(\mathbf{x}) \left( B\sqrt{\frac{\lambda_{\max}}{\rho^2}+1} + \frac{tB\epsilon}{\rho\sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho} + \sqrt{\log\det\left(\frac{\widetilde{\mathbf{S}}_t + \lambda\mathbf{I}}{\lambda+\lambda_{\min}}\right) + 2\ln\frac{2}{\zeta}} \right).$$

*The sequence $(v_t^{1/2})_{t=1}^T$ is chosen as the multiplicative factor of $\widetilde{\sigma}_t(\mathbf{x})$, i.e., $|\widehat{\mu}(\mathbf{x}) - \widetilde{\mu}_t(\mathbf{x})| \leq v_t^{1/2}\widetilde{\sigma}_t(\mathbf{x})$.*

*Proof.* We wish to bound $\widehat{\mu}(\mathbf{x}) - \widetilde{\mu}_t(\mathbf{x}) = \left\langle \widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_t, \Phi(\mathbf{x}) \right\rangle$. First, we bound this inner product by a suitable matrix norm:

$$\begin{aligned}
\left\langle \widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_t, \Phi(\mathbf{x}) \right\rangle &\leq \left\| \widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_t \right\|_{\mathbf{V}_t} \|\Phi(\mathbf{x})\|_{\mathbf{V}_t^{-1}} \\
&= \frac{\widetilde{\sigma}_t(\mathbf{x})}{\rho^2} \left\| \widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_t \right\|_{\mathbf{V}_t} \\
&= \frac{\widetilde{\sigma}_t(\mathbf{x})}{\rho^2} \left\| \widehat{\boldsymbol{\theta}} - \mathbf{V}_t^{-1}\Phi(\mathbf{X}_t)^\top \mathbf{y}_t - \mathbf{V}_t^{-1}\mathbf{h}_t \right\|_{\mathbf{V}_t} \\
&\leq \frac{\widetilde{\sigma}_t(\mathbf{x})}{\rho^2} \left( \left\| \widehat{\boldsymbol{\theta}} - \mathbf{V}_t^{-1}\Phi(\mathbf{X}_t)^\top \mathbf{y}_t \right\|_{\mathbf{V}_t} + \|\mathbf{h}_t\|_{\mathbf{V}_t^{-1}} \right).
\end{aligned}$$

Now, let $\mathbf{z}_t = (\langle \widehat{\boldsymbol{\theta}}, \phi(\mathbf{x}_\tau)\rangle + \xi_\tau)_{\tau<=t}$. By Lemma 7.1, we know that for each $|z_t^i - y_t^i| \leq B\epsilon$, and therefore $\|\mathbf{z}_t - \mathbf{y}_t\|_2 \leq B\epsilon\sqrt{t}$. Using this fact:

$$\left\langle \widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_t, \Phi(\mathbf{x}) \right\rangle \leq \frac{\widetilde{\sigma}_t(\mathbf{x})}{\rho^2} \left( \underbrace{\left\| \widehat{\boldsymbol{\theta}} - \mathbf{V}_t^{-1}\Phi(\mathbf{X}_t)^\top \mathbf{z}_t \right\|_{\mathbf{V}_t}}_{\textcircled{A}} + \underbrace{\left\| \Phi(\mathbf{X}_t)^\top (\mathbf{z}_t - \mathbf{y}_t) \right\|_{\mathbf{V}_t^{-1}}}_{\textcircled{B}} + \underbrace{\|\mathbf{h}_t\|_{\mathbf{V}_t^{-1}}}_{\textcircled{C}} \right).$$

Controlling $\textcircled{A}$: Note that $\mathbf{z}_t = \Phi(\mathbf{X}_t)\widehat{\boldsymbol{\theta}} + \boldsymbol{\xi}_t$, and therefore $\Phi(\mathbf{X}_t)^\top \mathbf{z}_t = \Phi(\mathbf{X}_t)^\top\Phi(\mathbf{X}_t)\widehat{\boldsymbol{\theta}} + \Phi(\mathbf{X}_t)^\top\boldsymbol{\xi}_t = \mathbf{V}_t\widehat{\boldsymbol{\theta}} - (\mathbf{H}_t + \lambda\mathbf{I})\,\widehat{\boldsymbol{\theta}} + \Phi(\mathbf{X}_t)^\top\boldsymbol{\xi}_t$. Replacing this in $\textcircled{A}$,

$$\begin{aligned}
\left\| \widehat{\boldsymbol{\theta}} - \mathbf{V}_t^{-1}\Phi(\mathbf{X}_t)^\top \mathbf{z}_t \right\|_{\mathbf{V}_t} &= \left\| \widehat{\boldsymbol{\theta}} - \mathbf{V}_t^{-1} \left( \mathbf{V}_t\widehat{\boldsymbol{\theta}} - (\mathbf{H}_t + \lambda\mathbf{I})\,\widehat{\boldsymbol{\theta}} + \Phi(\mathbf{X}_t)^\top\boldsymbol{\xi}_t \right) \right\|_{\mathbf{V}_t} \\
&= \left\| (\mathbf{H}_t + \lambda\mathbf{I})\,\widehat{\boldsymbol{\theta}} + \Phi(\mathbf{X}_t)^\top\boldsymbol{\xi}_t \right\|_{\mathbf{V}_t^{-1}} \\
&\leq \left\| (\mathbf{H}_t + \lambda\mathbf{I})\,\widehat{\boldsymbol{\theta}} \right\|_{\mathbf{V}_t^{-1}} + \left\| \Phi(\mathbf{X}_t)^\top\boldsymbol{\xi}_t \right\|_{\mathbf{V}_t^{-1}}
\end{aligned}$$

$$
\begin{aligned}
&\leq \left\|(\mathbf{H}_t+\lambda\mathbf{I})\,\widehat{\boldsymbol{\theta}}\right\|_{(\mathbf{H}_t+\lambda\mathbf{I})^{-1}} + \left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{\mathbf{V}_t^{-1}} && (\mathbf{V}_t \succcurlyeq \mathbf{H}_t+\lambda\mathbf{I})\\
&\leq \left\|\widehat{\boldsymbol{\theta}}\right\|_{\mathbf{H}_t+\lambda\mathbf{I}} + \left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{\mathbf{V}_t^{-1}} && (\mathbf{V}_t \succcurlyeq \mathbf{H}_t+\lambda\mathbf{I})\\
&\leq \|\widehat{\boldsymbol{\theta}}\|_2\sqrt{\lambda_{\max}+\rho^2} + \left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{\mathbf{V}_t^{-1}}\\
& && (\mathbf{H}_t \preccurlyeq \lambda_{\max}\mathbf{I}, \text{ union bound } \forall t\in[T] \text{ w. p. } 1-\zeta/2)\\
&\leq B\sqrt{(\lambda_{\max}+\rho^2)} + \left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{\mathbf{V}_t^{-1}} && (\text{Lemma 7.10})\\
&\leq B\sqrt{(\lambda_{\max}+\rho^2)} + \left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{(\mathbf{S}_t+(\lambda+\lambda_{\min})\mathbf{I})^{-1}} && (\mathbf{H}_t \succcurlyeq \lambda_{\min}\mathbf{I})
\end{aligned}
$$

To bound the second term on the RHS, we use the "self-normalized bound for vector-valued martingales" of Abbasi-Yadkori *et al.*(Abbasi-Yadkori et al., 2011) (Theorem 1), which gives us that with probability $1-\varepsilon/2$ for all $t \in [T]$ simultaneously,

$$
\left\|\Phi(\mathbf{X}_t)^\top \boldsymbol{\xi}_t\right\|_{(\mathbf{S}_t+(\lambda+\lambda_{\min})\mathbf{I})^{-1}} \leq \rho\sqrt{2\log\frac{2}{\zeta} + \log\frac{\det\left(\mathbf{S}_t+\lambda+\lambda_{\min}\mathbf{I}\right)}{\det\left(\lambda+\lambda_{\min}\mathbf{I}\right)}}.
$$

Controlling Ⓑ:

$$
\begin{aligned}
\|\Phi(\mathbf{X}_t)^\top(\mathbf{z}_t-\mathbf{y}_t)\|_{\mathbf{V}_t^{-1}} &\leq \left\|\Phi(\mathbf{X}_t)^\top\left(\mathbf{z}_t-\mathbf{y}_t\right)\right\|_{\mathbf{H}_t^{-1}} && (\mathbf{V}_t \succcurlyeq \mathbf{H}_t)\\
&\leq \frac{\|\Phi(\mathbf{X}_t)^\top(\mathbf{z}_t-\mathbf{y}_t)\|_2}{\sqrt{\lambda_{\min}}} && (\mathbf{H}_t \succcurlyeq \lambda_{\min}\mathbf{I})\\
&\leq \frac{\|\Phi(\mathbf{X}_t)\|_2\,\|\mathbf{z}_t-\mathbf{y}_t\|_2}{\sqrt{\lambda_{\min}}} && (\text{Cauchy-Schwarz})\\
&\leq \frac{tB\epsilon}{\sqrt{\lambda_{\min}}}. && (\|\Phi(\mathbf{x})\|_2 \leq 1 \text{ and definition of } \mathbf{z}_t)
\end{aligned}
$$

Controlling Ⓒ: We can see that $\|\mathbf{h}_t\|_{\mathbf{V}_t^{-1}} \leq \|\mathbf{h}_t\|_{\mathbf{H}_t^{-1}}$ and with probability $1-\zeta/2$ (from the control of Ⓐ), for all rounds this is bounded by $\kappa$.

Combining all three, we obtain that with probability at least $1-\zeta/2$, for any $\mathbf{x}\in\mathcal{D}$, and simultaneously for all $t\in[T]$:

$$
\widehat{\mu}(\mathbf{x}) - \widetilde{\mu}_t(\mathbf{x}) \leq \widetilde{\sigma}_t(\mathbf{x}) \underbrace{\left(\frac{B}{\rho}\sqrt{(\lambda_{\max}+\rho^2)} + \sqrt{2\log\frac{2}{\zeta} + \log\frac{\det\left(\widetilde{\mathbf{S}}_t+\lambda_{\min}\mathbf{I}\right)}{\det\left((\lambda+\lambda_{\min})\mathbf{I}\right)}} + \frac{tB\epsilon}{\rho\sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho}\right)}_{v_t^{1/2}}.
$$

□

The complete algorithm is summarized in Algorithm 12, and prooof is presented in the appendix. Note that we describe the algorithm abstractly for any $\epsilon$-uniformly approximating feature $\Phi$ with dimensionality $m$, and Theorem 7.1 (and the regret bound) hold for any such feature approximation that also satisfies $\sup_{\mathbf{x}\in\mathcal{D}}\|\Phi(\mathbf{x})\| \leq 1$. The algorithm is described in two separate entities, the Server and the Privatizer, where the privatizer entity has access to the raw rewards and contexts, and the server only obtains privatized versions of the statistics. We now present specific $\Phi$ such that we obtain an efficient algorithm.

**Remark 7.1** (Parameter Release)**.** $\widetilde{\mu}$ can be determined entirely only with the parameters $\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t$ (Equation 7.3). If the noise variables $\mathbf{H}_t, \mathbf{h}_t$ are constructed such that the resulting parameters satisfy privacy constraints (see next section), these parameters are by design differentially private and hence $\widetilde{\mu}$ can be released without using the sample path $(\mathbf{x}_t, y_t)_{t\leq T}$.

### 7.4.2 Noisy Quadrature Fourier Features

Bochners' theorem (Bochner, 1933) states that there exists an integral form for stationary $k$, where the integrand is a product of identical features of the inputs:

$$k(\mathbf{x}-\mathbf{y}) = \int_\Omega \begin{pmatrix}\sin(\boldsymbol{\omega}^\top\mathbf{x})\\ \cos(\boldsymbol{\omega}^\top\mathbf{x})\end{pmatrix}^\top \begin{pmatrix}\sin(\boldsymbol{\omega}^\top\mathbf{y})\\ \cos(\boldsymbol{\omega}^\top\mathbf{y})\end{pmatrix} p(\boldsymbol{\omega}).$$

When the above integral is approximated by a Monte-Carlo average, we obtain the powerful Random Fourier Features (RFF, (Rahimi & Recht, 2008)) approximation. Random Fourier features, while approximating a variety of kernels, are not efficient since $\epsilon_{\text{RFF}} = \mathcal{O}(m^{-1/2})$, requiring prohibitively many features $m$ for our purpose. We consider Quadrature Fourier Features (QFF, Dao et al. (2017)), a stronger approximation that is motivated by numerical integration, and allows $\epsilon$ to decay *exponentially* in $m$. To define QFF, we require that the kernel be Fourier decomposable.

**Assumption 7.1** (Decomposability of $k$)**.** *Let $k$ be a stationary kernel defined on $\mathbb{R}^d \times \mathbb{R}^d$ and $k(\mathbf{x},\mathbf{y}) \leq 1 \; \forall \; \mathbf{x},\mathbf{y} \in \mathbb{R}^d$ with a Fourier transform that decomposes product-wise*[3]*, i.e.,*

$$p(\boldsymbol{\omega}) = \pi_{j=1}^d p_j(\boldsymbol{\omega}_j).$$

[3] This is satisfied for commonly-used kernels, e.g., squared exponential. Matérn kernels are decomposable when $d=1$. For $d>1$, Mutny & Krause (2018) present a modified Matérn kernel that can be used a surrogate.

**Definition 7.4** (Quadrature Fourier Features)**.** *Let $\mathcal{D} = [0,1]^d$, and $\mathbf{x}, \mathbf{y} \in \mathcal{D}$. Fix $m = (\bar{m})^d$ for some $\bar{m} > 1$, and let $p(\boldsymbol{\omega}) = \exp\left(-\sum_{i=1}^{d} \frac{\omega_i^2 \nu_i^2}{2}\right)$ be the Fourier transform of $k$. The QFF features $\Phi(\mathbf{x})$ is defined as:*

$$\Phi(\mathbf{x})_i = \begin{cases} \sqrt{\Pi_{j=1}^{d} 1/\nu_i Q(\omega_{i,j}) \cos(\boldsymbol{\omega}_i^\top \mathbf{x})} & \textit{if } i \leq \frac{m}{2} \\ \sqrt{\Pi_{j=1}^{d} 1/\nu_i Q(\omega_{m-i,j}) \sin(\boldsymbol{\omega}_{m-i}^\top \mathbf{x})} & \textit{o.w.} \end{cases}$$

*Where $Q(\omega_{i,j}) = \frac{2^{m-1/2} m! \sqrt{\pi}}{\nu_j m^2 H_{m-1}(\omega_{i,j})}$ and $H_t$ is the $t^{th}$ Hermite polynomial, and hence $\Phi$ is of dimensionality $2m$. The set $(\boldsymbol{\omega}_i)_{i=1}^{m}$ is the Cartesian product of $\{\bar{\omega}_j\}_{j=1}^{\bar{m}}$, where each element $\bar{\omega}_i \in \mathbb{R}$ and is a zero of the $i^{th}$ Hermite polynomial. See Hildebrand (1987) for details.*

**Theorem 7.2** (QFF Error (Mutny & Krause, 2018))**.** *Let $\Phi(\cdot), m$ and $\bar{m}$ be as defined above, $\mathcal{D} = [0,1]^d$ and $\nu = \min_i \nu_i$. Then,*

$$\sup_{\mathbf{x},\mathbf{y} \in \mathcal{D}} |k(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x})^\top \Phi(\mathbf{y})| \leq d 2^{d-1} \sqrt{\frac{\pi}{2}} \frac{1}{\bar{m}^{\bar{m}}} \left(\frac{e}{4\nu^2}\right)^{\bar{m}}.$$

**Remark 7.2.** Theorem 7.2 implies that the error $\epsilon$ decays exponentially in $m$ when $m > \nu^{-2}$. Mutny & Krause (2018) evaluate this phase transition in detail, where a break is observed in simulations. For any known kernel $k$ however, we can simply select $m > \nu^{-2}$ to ensure decay. Moreover, for additive kernels, it can be demonstrated that the dependence is exponential in the effective dimension, which can be much less than $d$.

By adding appropriate $(\mathbf{H}_t, \mathbf{h}_t)$ to maintain privacy, we obtain *noisy* quadrature Fourier features (NQFF).

**Definition 7.5** (Noisy Quadrature Fourier Features (NQFF))**.** *Let $\Phi : \mathbb{R}^d \to \mathbb{R}^m$ be an $\epsilon-$approximation QFF to the stationary kernel $k$, and $(\mathbf{H}_t, \mathbf{h}_t)_{t=1}^{T}$ be a sequence of perturbations. Then, at any instant $t$, we can define the noisy QFF as $\widetilde{\Phi}(\mathbf{X}_t) = \begin{bmatrix} \Phi(\mathbf{X}_t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_t \end{bmatrix}$, where $\boldsymbol{\Gamma}_t^\top \boldsymbol{\Gamma}_t = \mathbf{H}_t$ can eb obtained by the eigendecomposition of $\mathbf{H}_t$.*

### 7.4.3 Regret Analysis

We first present the regret bound for GP-UCB with generic $\epsilon$-uniformly approximating features $\Phi$ with dimensionality $m$. Note that this bound is applicable to any approximation technique that satisfies $\sup_{\mathbf{x} \in \mathcal{D}} \|\Phi(\mathbf{x})\| \leq 1$, and suitable $\lambda_{\min}, \lambda_{\max}$ and $\kappa$.

**Theorem 7.3** (Regret Bound)**.** *Let $k$ be a stationary kernel with the associated RKHS $\mathcal{H}_k$, and $\mathcal{F}_m$ be an RKHS with feature $\Phi(\cdot)$ of dimensionality $m$, that $\epsilon$-uniformly approximates every $f \in \mathcal{H}_k$ when $\|f\| \leq B$. Furthermore, assume $\lambda_{\min}, \lambda_{\max}$ and $\kappa$ such that they are $(\zeta/2T)$ -accurate and all regularizers $\mathbf{H}_t \succcurlyeq 0\ \forall t \in [T]$ are PSD. Then for $(v_t)_{t=1}^{T}$ chosen by Theorem 7.1,* GP-UCB *with noisy proximal features obtains the following cumulative regret with probability at least $1-\zeta$:*

$$\mathfrak{R}(T) \leq 2\sqrt{Tv_T\gamma_T} + \frac{2T^3\sqrt{v_T\epsilon}}{3\rho} + 2TB\epsilon.$$

*Where $\gamma_T$ is the maximum information gain (Definition 7.1).*

*Sketch.* The first key observation is to bound the per-round regret from $f$ with the per-round regret from optimizing $\widehat{\mu}$. Next, we utilize standard techniques from the analysis of GP-UCB to bound the regret in terms of $v_t$ and $\widetilde{\sigma}_t$ (using Theorem 7.1 twice), and finally provide a bound on $\widetilde{\sigma}_t$ in terms of the true information gain $\gamma_T$. Summing over all rounds and manipulating proves the result.

*Proof.* We first bound the instantaneous regret $r_t$ at any instant $t$.

$$\begin{aligned}
r_t &= f(\mathbf{x}_*) - f(\mathbf{x}_t) \\
&\leq \widehat{\mu}(\mathbf{x}_*) - \widehat{\mu}(\mathbf{x}_t) + 2B\epsilon && \text{(Lemma 7.1)} \\
&\leq v_t\widetilde{\sigma}_t(\mathbf{x}_*) + \widetilde{\mu}_t(\mathbf{x}_*) - \widehat{\mu}(\mathbf{x}_t) + 2B\epsilon && \text{(Theorem 7.1 } (\forall t \in [T] \text{ w.p. } \geq 1-\varepsilon/2)) \\
&\leq v_t\widetilde{\sigma}_t(\mathbf{x}_t) + \widetilde{\mu}_t(\mathbf{x}_t) - \widehat{\mu}(\mathbf{x}_t) + 2B\epsilon && \text{(Algorithm)} \\
&\leq 2v_t\widetilde{\sigma}_t(\mathbf{x}_t) + 2B\epsilon && \text{(Theorem 7.1 } (\forall t \in [T] \text{ w.p. } \geq 1-\varepsilon/2)) \\
&\leq 2v_t\sigma_t(\mathbf{x}_t) + 2B\epsilon + \frac{2t^2v_t\sqrt{\epsilon}}{\rho}. && \text{(Lemma 7.11)}
\end{aligned}$$

Now, we can sum over all rounds $t \in [T]$ to obtain the overall regret:

$$\begin{aligned}
\mathfrak{R}(T) = \sum_{t=1}^{T} r_t &\leq 2\sum_{t=1}^{T}\left(v_t\sigma_t(\mathbf{x}_t) + B\epsilon + \frac{t^2v_t\sqrt{\epsilon}}{\rho}\right) \\
&\leq 2v_T\left(\sum_{t=1}^{T}\sigma_t(\mathbf{x}_t) + \sqrt{\epsilon}\sum_{t=1}^{T}\frac{t^2}{\rho}\right) + 2TB\epsilon \\
&\leq 2v_T\left(\sum_{t=1}^{T}\sigma_t(\mathbf{x}_t)\right) + v_T\frac{T^3\sqrt{\epsilon}}{3\rho} + 2TB\epsilon
\end{aligned}$$

$$\leq 2v_T \left( \sqrt{T \left( \sum_{t=1}^{T} \sigma_t^2(\mathbf{x}_t) \right)} + \frac{T^3 \sqrt{\epsilon}}{3\rho} \right) + 2TB\epsilon$$

$$\leq 2v_T \left( \sqrt{T \log \left( \mathbf{I} + (\lambda + \lambda_{\min})^{-1} \mathbf{K}_T \right)} + \frac{T^3 \sqrt{\epsilon}}{3\rho} \right) + 2TB\epsilon$$

(Lemma 5.4 of Srinivas *et al.*(Srinivas et al., 2010))

$$\leq 2v_T \left( \sqrt{T \gamma_T} + \frac{T^3 \sqrt{\epsilon}}{3\rho} \right) + 2TB\epsilon$$

(Lemma 5.4 of Srinivas *et al.*(Srinivas et al., 2010))

Consider the substitutions $\overline{\lambda} = \lambda_{\max} + \rho^2$ and $\underline{\lambda} = \lambda + \lambda_{\min}$. Then we have,

$$= 2 \left( \frac{B}{\rho} \sqrt{\overline{\lambda}} + \sqrt{2 \log \frac{2}{\zeta} + \log \frac{\det \left( \mathbf{S}_T + \underline{\lambda} \mathbf{I} \right)}{\det \left( \underline{\lambda} \mathbf{I} \right)}} + \frac{TB\epsilon}{\rho \sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho} \right) \left( \sqrt{T \gamma_T} + \frac{T^3 \sqrt{\epsilon}}{3\rho} \right) + 2TB\epsilon.$$

Further simplifying and substituting $\widetilde{T} = \left( \sqrt{T\gamma_T} + \frac{T^3\sqrt{\epsilon}}{3\rho} \right)$ for brevity,

$$= 2 \left( \frac{B}{\rho} \sqrt{\overline{\lambda}} + \sqrt{2 \log \frac{2}{\zeta} + \log \frac{\det \left( \Phi(\mathbf{X}_T)^\top \Phi(\mathbf{X}_T) + \underline{\lambda} \mathbf{I} \right)}{\det \left( \underline{\lambda} \mathbf{I} \right)}} + \frac{TB\epsilon}{\rho \sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho} \right) \cdot \widetilde{T} + 2TB\epsilon.$$

By the Hadamard inequality,

$$\leq 2 \left( \frac{B}{\rho} \sqrt{\overline{\lambda}} + \sqrt{2 \log \frac{2}{\zeta} + \log \det \left( \frac{\operatorname{diag}(\Phi(\mathbf{X}_T)^\top \Phi(\mathbf{X}_T))}{\lambda + \lambda_{\min}} + \mathbf{I} \right)} + \frac{TB\epsilon}{\rho \sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho} \right) \cdot \widetilde{T} + 2TB\epsilon$$

$$\leq 2 \left( B \sqrt{\left( \frac{\lambda_{\max}}{\rho^2} + 1 \right)} + \sqrt{2 \log \frac{2}{\zeta} + m \log\left(1 + \frac{T}{\rho + \lambda_{\min}}\right)} + \frac{TB\epsilon}{\rho \sqrt{\lambda_{\min}}} + \frac{\kappa}{\rho} \right) \cdot \widetilde{T} + 2TB\epsilon. \quad (7.6)$$

□

By replacing $v_T$ in the result, and manipulating terms, we can conclude that if we have

$\Phi$ such that $\epsilon = \mathcal{O}(\exp(-m))$ and $m = \mathcal{O}(\text{polylog}(T))$, then we can obtain sublinear regret. Using the properties of QFF from earlier, we can obtain a specific bound as follows.

**Corollary 7.1.** *Fix $m = 2(6 \log T)^d$ and let $k$ be any kernel that obeys Assumption 7.1. Algorithm 12 run with $m$-dimensional NQFF and noise $\mathbf{H}_t, \mathbf{h}_t$ that are $\zeta/2T$-accurate with constants $\lambda_{\max}, \lambda_{\min}$ and $\kappa$ obtains with probability at least $1 - \zeta$, cumulative pseudoregret:*

$$\mathfrak{R}(T) = \mathcal{O}\left(\sqrt{T\gamma_T}\left(\frac{B\sqrt{\lambda_{\max}}}{\rho} + \sqrt{\log\frac{1}{\zeta} + (\log T^6)^{d+1} + \frac{\kappa}{\rho}}\right)\right).$$

*Proof.* From Theorem 7.2, when $\bar{m} = 6 \log T$, and $\nu^2 < m$ we have that for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\sup_{\mathbf{x},\mathbf{y}\in\mathcal{D}} |k(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x})^\top \Phi(\mathbf{y})| \leq \frac{C_1}{T^6} = \epsilon$ for some constant $C_1$. Replacing this in Equation 7.6 we see that the terms dependent on $\epsilon$ are $o(1)$, giving us the final result. □

**Remark 7.3** (Selection of $m$)**.** Note that the analysis presents a bound in terms of the information gain of the true kernel $k$, and hence requires $m = 2(\log T^6)^d$ features. However, an alternate technique will be to bound the information gain of $\tilde{k}$, which can subsequently be bound with a term of $\mathcal{O}(\sqrt{m \log T})$. In this case, setting $m = 2(\log T^3)^d$ suffices for no-regret learning, however the obtained regret is (coarsely) $\mathcal{O}(\sqrt{T}(\log T)^{d+1})$, which can be loose if $\gamma_T = o((\log T)^{d+1})$ (e.g., when $k$ is low-rank).

**Remark 7.4** (Feasibility of Kernel Approximations)**.** The current framework requires $\epsilon = o(T^{-4})$ with $m = \mathcal{O}(\text{polylog}(T))$ to obtain a no-regret algorithm. Random Fourier Features, while capable of approximating a variety of stationary kernels, decay with $\epsilon = \mathcal{O}(m^{-1/2})$ which makes them infeasible. For finite $\mathcal{H}_k$, the results manifestly hold with $\epsilon = 0$.

**Remark 7.5** (Unknown $T$)**.** When $T$ is unknown, we can use a doubling scheme to calculate $m$ and $\epsilon$. To calculate $\epsilon$, we assume $T = 1$ for the first round, then assume $T = 2$ for the next, and then assume $T = 4$ for the next 2 rounds, $T = 8$ for the next 4 rounds and so on, and set $\epsilon = \mathcal{O}(t^{-5})$, for instance, within each "period" of length $t$ between doubling of $T$ to calculate $m$. We see that the regret is at most $\widetilde{\mathcal{O}}(\sqrt{t})$ for this period. Since there are at most $\mathcal{O}(\log T)$ such periods, and $t \leq T$, the total regret is $\mathcal{O}((\log T)\sqrt{T})$.

## 7.5 GP-UCB with Differential Privacy

We now present the mechanism to ensure Algorithm 12 is differentially private. Proceeding with the standard definition of differential privacy for the streaming setting, however, is infeasible (i.e., leading to linear regret, see Claim 13 of Shariff & Sheffet (2018)). We therefore work with a modified notion of privacy that is the standard for sequential decision-making (Shariff & Sheffet, 2018; Vietri et al., 2020b).

**Definition 7.6** (Joint Differential Privacy (JDP))**.** *Let $S = (\mathcal{D}_i, y_i)_{i=1}^T$ and $S' = (\mathcal{D}'_i, y'_i)_{i=1}^T$ be two sequences such that $(\mathcal{D}_i, y_i) = (\mathcal{D}'_i, y'_i)$ for all $i \neq t$, and $\mathcal{S}_{-t} \subseteq \mathcal{D}_1 \times ... \times \mathcal{D}_{t-1} \times \mathcal{D}_{t+1} \times ... \times \mathcal{D}_T$ denote a sequence of actions except the $t^{th}$. An algorithm $\mathcal{A}$ is $(\varepsilon, \delta)$-JDP under continual observation if for any $t \in [T], S, S'$, it holds that $\mathbb{P}(\mathcal{A}(S) \in \mathcal{S}_{-t}) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(S') \in \mathcal{S}_{-t}) + \delta$.*

The only change in the JDP setting (compared to standard DP) is that the algorithm is allowed to be non-private at time $t$ with respect to $\mathcal{D}_t$ (i.e., the active decision set). This is crucial as standard DP would imply that for any two actions $\mathbf{x}, \mathbf{x}' \in \mathcal{D}_t, \mathbb{P}(a_t = \mathbf{x}) \approx \mathbb{P}(a_t = \mathbf{x}')$ and the algorithm would incur linear regret.

### 7.5.1 Approximate GP-UCB with JDP

Our approach involves perturbing $(\mathbf{S}_t, \mathbf{u}_t)$ by noise $(\mathbf{H}_t, \mathbf{h}_t)$ to ensure JDP, and it is summarized in Algorithm 13. Observe that the estimates $(\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t)$ are noisy cumulative sums of $\mathbf{S}_t = \sum_{\tau=1}^{t-1} \Phi(\mathbf{x}_\tau)\Phi(\mathbf{x}_\tau)^\top, \mathbf{u}_t = \sum_{\tau=1}^{t-1} y_\tau \cdot \Phi(\mathbf{x}_\tau)$. This additive structure naturally suggests that we utilize a matrix variant of the tree-based mechanism (Dwork & Smith, 2010; Shariff & Sheffet, 2018) to maintain $(\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t)$. We consider the matrix $\mathbf{N}_t = [\Phi(\mathbf{X}_t),\ \mathbf{y}_t]^\top [\Phi(\mathbf{X}_t),\ \mathbf{y}_t] \in \mathbb{R}^{m+1 \times m+1}$ and compute this matrix via the tree-based mechanism. The advantage of maintaining $\mathbf{N}_t$ is that $\mathbf{N}_{t+1} = \mathbf{N}_t + [\Phi(\mathbf{x}_t),\ y_t]^\top[\Phi(\mathbf{x}_t),\ y_t]$ and the top $m \times m$ submatrix of $\mathbf{N}_t$ is $\mathbf{S}_t$ and the first $m$ entries of the last column of $\mathbf{N}_t$ is $\mathbf{u}_t$, giving us the required estimates.

**Tree-Based Mechanism**. The tree-based mechanism (Dwork & Smith, 2010) estimates the rolling sum of any series $\mathbf{n}_1, \mathbf{n}_2, ...$ via a binary tree. Let $P_{m+1}$ be a probability distribution over $\mathbb{R}^{m+1 \times m+1}$. A trusted entity (in our case, the Privatizer), maintains a binary tree $\mathcal{T}$ whose $t^{th}$ leaf node stores $\mathbf{n}_t = [\Phi(\mathbf{x}_t)\, y_t]^\top [\Phi(\mathbf{x}_t)\, y_t] + (1/\sqrt{2})\,(\boldsymbol{\upsilon}_t \top + \boldsymbol{\upsilon}_t)$, where $\boldsymbol{\upsilon}_t$ is a sample from $P_{m+1}$. Each parent node stores the sum of its children. Now, to compute $(\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t)$ we traverse $\mathcal{T}$ to the $t^{th}$ leaf node, and sum the values at each node. Since the path length traversed is $1 + \lceil \log_2 T \rceil$, we can rewrite $\widetilde{\mathbf{S}}_t = \mathbf{S}_t + \mathbf{H}_t$ where $\mathbf{H}_t$ is the sum of at most $n = 1 + \lceil \log_2 T \rceil$ samples from $P_{m+1}$. We now describe selecting $P_{m+1}$ to provide a JDP guarantee.

**Lemma 7.2** (JDP)**.** *Let $P_{m+1}$ be a composition of $(m+1)^2$ zero-mean normal variables with variance $\sigma^2_{\varepsilon,\delta}$. If $\sigma_{\varepsilon,\delta} > 16n(1 + B^2 + 2\rho^2 \log(8T/\delta)) \ln(10/\delta)^2/\varepsilon^2$, then Algorithm 12 with* Privatizer *following Algorithm 13 is $(\varepsilon, \delta)-$jointly differentially private.*

*Proof.* First note that since $y_t$ is sub-Gaussian with mean at most $B$ (since $\|f\|_k \leq B$), we have from Lemma 7.7, that with probability at least $1 - \delta/4$, for each $(y_\tau)_{\tau \in [T]}$ simultaneously,

$$|y_t|^2 \leq B^2 + 2\rho^2 \log \frac{8T}{\delta}.$$

The overall sensitivity $\Delta$ of each datum is then given by $\|\Phi(\mathbf{x}_t)\|_2 + |y_t|_2$, therefore $\Delta^2 \leq 1 + B^2 + 2\rho^2 \log \frac{8T}{\delta}$ with probability at least $1 - \delta/4$. Note now that we have the sum of at most $n = 1 + \lceil \log T \rceil$ noise variables. Therefore, to ensure $(\varepsilon, \delta)-$joint DP, we must ensure each noise variable preserves $(\varepsilon/\sqrt{8n \ln(2/\delta)}, \delta/2)$ privacy (based on zero-Concentrated DP (Bun & Steinke, 2016)).

If $\sigma^2_{\varepsilon,\delta} > 16n(1 + B^2 + 2\rho^2 \log \frac{8T}{\delta})^2 \ln(\frac{10}{\delta})^2$ then we have by Lemma 7.8 that each noise term preserves $(\varepsilon/\sqrt{8n \ln(2/\delta)}, \delta/2)$-DP, proving the result. □

Recall that our regret bound (Corollary 7.1) scales with the parameters $\lambda_{\min}, \lambda_{\max}$ and $\kappa$. It remains to provide these quantities under the selected $P_{m+1}$ such that they are accurate (Definition 7.3), and provide final regret bounds based on the properties of $P_{m+1}$. As remarked in Shariff & Sheffet (2018), we must shift the noise matrix to ensure that all noise samples $\mathbf{H}_t$ are PSD.

**Lemma 7.3** (Accurate Spectrum under JDP)**.** *For any $\zeta > 0$, when $P_{m+1}$ is selected according to Lemma 7.2 and $\mathbf{H}_t, \mathbf{h}_t$ are constructed according to Algorithm 13, the following $\lambda_{\min}, \lambda_{\max}$ and $\kappa$ are $(\zeta/2T)-$accurate:*

$$\lambda_{\min} = \Lambda, \lambda_{\max} = 3\Lambda, \kappa = \sigma_{\varepsilon,\delta}\sqrt{\frac{n}{\Lambda}}\left(\sqrt{m} + \sqrt{2 \ln \frac{2T}{\zeta}}\right).$$

*Here $\Lambda = \sigma_{\varepsilon,\delta}\sqrt{2n}(4\sqrt{m} + 2\ln(2T/\zeta))$.*

*Proof.* Follows from Proposition 11 from Shariff & Sheffet (2018) with our noise model. □

**Corollary 7.2** ($(\varepsilon, \delta)$-JDP Regret Bound)**.** *Fix $m = 2(6 \log T)^d$ and let $k$ be any kernel that obeys Assumption 7.1. Algorithm 12 run with $m$-dimensional NQFF and noise such that it maintains $(\varepsilon, \delta)-$JDP obtains with probability at least $1 - \zeta$, cumulative pseudoregret:*

$$\mathfrak{R}(T) = \mathcal{O}\left(\sqrt{T\gamma_T}\left(\log(T)^{\frac{d+2}{4}}\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\log\frac{1}{\zeta}\right)^{\frac{1}{2}} + \log(T)^{\frac{d+1}{2}}\right)\right).$$

The proof for Corollary 7.2 follows directly by substituting the results from Lemma 7.3 into Corollary 7.1.

**Remark 7.6** (Dependence on $m$)**.** Since the factors $\lambda_{\min}, \lambda_{\max}$ and $\kappa$ admit a dependence of $\mathcal{O}(\sqrt{m})$ on the dimensionality of $\Phi$, we require $m = o(\sqrt{T})$ features to guarantee no-regret learning under our approach. This constraint is complementary to the constraint on $m$ from kernel approximation (Remark 7.4), and mandates that even when the approximation $\tilde{k}$ has small $\gamma_T$ (i.e., $\gamma_T = o(\text{polylog}(T))$, we require small $m$.

### 7.5.2 Approximate GP-UCB with LDP

In many settings, the existence of a trusted entity (e.g., Privatizer) is not possible. For instance, consider the task of a centralized server learning a bandit algorithm in the case when each user $t$ does not wish $(\mathcal{D}_t, y_t)$ to be sent to the server at all (even to select $\mathbf{x}_t$). We can select $\mathbf{x}_t$, however, by sending the algorithm's (privatized) parameters to each user individually and collecting updated parameters after $\mathbf{x}_t$ has been played by the user $t$. Here, we employ an alternative definition of privacy known as local differential privacy (LDP).

**Definition 7.7** (Local Differential Privacy (LDP))**.** *A mechanism $g : \mathcal{X} \to \mathcal{Z}$ is $(\varepsilon, \delta)$-locally differentially private (Bebensee, 2019) (LDP) if for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbb{P}(g(\mathbf{x}) \in \mathcal{Z}) \leq e^{\varepsilon}\mathbb{P}(g(\mathbf{x}') \in \mathcal{Z}) + \delta$. For any sequence $(\mathcal{D}_t, y_t)_{t=1}^T$, an algorithm $\mathcal{A}$ protects locally joint differentially privacy (LDP) if for any $t$, $\mathcal{A}$ is locally differentially private with respect to each $(\mathcal{D}_\tau, y_\tau)$ simultaneously where $\tau \neq t$.*

This definition combines joint differential privacy (operating globally) with local differential privacy (operating individually). It is a stronger privacy guarantee than JDP, since it requires $\mathcal{A}$ to be private to each user simultaneously.

**Lemma 7.4** (LDP implies JDP)**.** *Any* $(\varepsilon, \delta)-$*LDP algorithm* $\mathcal{A}$ *protects* $(\varepsilon, \delta)$*-JDP* $\forall t \in [T]$*.*

*Proof (Sketch).* For any $t \in [T]$, any two $t$-neighboring sequences $S$ and $S'$ only differ in the $t^{th}$ entries $(\mathcal{D}_t, y_t)$ and $(\mathcal{D}'_t, y'_t)$. Since $\mathcal{A}$ is $(\varepsilon, \delta)-$locally JDP, $\mathbb{P}(a_{t'}(\mathcal{D}_t, y_t)) \in \mathcal{S}_{t'}) \leq e^{\varepsilon}\mathbb{P}(a_{t'}(\mathcal{D}'_t, y'_t)) \in \mathcal{S}_{t'}) + \delta$ for all $t' \neq t$, from which the result follows. The complete proof is provided in Section 7.8.2 □

Since a trusted entity does not exist, learning is done by sending the parameters directly to the users (ref. clients). We outline a server-client protocol and associated algorithm for $(\varepsilon, \delta)$-LDP Gaussian Process bandit optimization in Algorithm 14. This algorithm requires noise added individually to $(\Phi(\mathbf{x}_t), y_t)$ (instead of $(\mathbf{S}_t, \mathbf{u}_t)$). We achieve this by perturbing $\mathbf{S}_t$ and $\mathbf{u}_t$ separately with $\mathbf{N}_t \in \mathbb{R}^{m \times m}$ where $\mathbf{N}_t(i,j) \sim \mathcal{N}(0, \sigma_X^2)$ for $i \geq j$ and $\mathbf{N}_t(i,j) = \mathbf{N}_t(j,i)$ otherwise and $\mathbf{n}_t \in \mathbb{R}^m$ is such that $\mathbf{n}_t(i) \sim \mathcal{N}(0, \sigma_u^2)$. The variances $\sigma_X^2$ and $\sigma_u^2$ are chosen to ensure $(\varepsilon/2, \delta/2)$ respectively, securing $(\varepsilon, \delta)$-LDP.

**Lemma 7.5** (Noise for LDP)**.** *Algorithm 14 is* $(\varepsilon, \delta)-$*locally JDP whenever,*

$$\sigma_X^2 \geq \frac{8}{\varepsilon^2} \ln \frac{5}{2\delta}, \; \sigma_u^2 \geq \frac{8}{\varepsilon^2}\left(B^2 + 2\ln\frac{8m}{\beta}\right) \ln \frac{5}{\delta}.$$

*Proof.* We first note that the $L_2-$sensitivity of each element within $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_t)$ is 1 by the fact that $\|\Phi(\mathbf{x})\| \leq 1$. Next, note that the $L_2-$ sensitivity of each element of $y_t\Phi(\mathbf{x}_t)$ is with probability at least $1 - \delta/4$ at most $B + \rho\sqrt{2\log\frac{8m}{\beta}}$ ($y_t$ is Gaussian with mean at most $B$). Now, by the Gaussian mechanism for local DP (Dwork & Roth, 2014), we have that for $\sigma_x^2 \geq \frac{8\ln(2.5/\delta)}{\varepsilon^2}$ and $\sigma_u \geq \frac{8(B^2+2\log\frac{8m}{\beta})\ln(5/\delta)}{\varepsilon}$, both $\Phi(\mathbf{x}_t)^\top\Phi(\mathbf{x}_t) + \mathbf{N}_t$ and $y_t\Phi(\mathbf{x}_t) + \mathbf{n}_t$ are $(\varepsilon/2, \delta/2)-$locally DP. □

It remains to bound the spectral parameters ($\lambda_{\min}, \lambda_{\max}$ and $\kappa$) in order to obtain regret bounds.

**Lemma 7.6.** *(Spectrum for LDP) For any* $\zeta > 0$*, fix* $\Lambda = \sqrt{T}(4\sqrt{m} + 2\ln(2T/\zeta))$*. When* $P_{m+1}$ *is selected according to Lemma 7.5 and* $\mathbf{H}_t, \mathbf{h}_t$ *are constructed according to Algorithm 14, the following are* $(\zeta/2T)-$*accurate:*

$$\lambda_{\min} = \sigma_x\Lambda, \lambda_{\max} = 3\sigma_x\Lambda \; and, \kappa = \sigma_u\sqrt{mT\Lambda^{-1}}.$$

*Proof (Sketch).* The proof is identical to Lemma 7.3 except critically that in this case, $\mathbf{H}_t$ (resp. $\mathbf{h}_t$) is the sum of $t$ matrices $\mathbf{N}_t$ (resp. $\mathbf{n}_t$), with total variance $t\sigma_X^2$ (resp. $t\sigma_u^2$). Therefore, we can bound $\|\mathbf{H}_t\|_2 \leq \sigma_X\sqrt{T}(4\sqrt{m} + 2\ln(2T/\zeta))$ and $\|\mathbf{h}_t\|_2 \leq \sigma_u\sqrt{mT}$, which gives the result identical to Lemma 7.3. □

**Corollary 7.3** ($(\varepsilon,\delta)-$LDP Regret Bound)**.** *Fix* $m = (6\log T)^d$ *and let* $k$ *be any kernel that obeys Assumption 7.1. Algorithm 12 run with NQFF and noise* $\mathbf{H}_t, \mathbf{h}_t$ *that maintains* $(\varepsilon,\delta)-$*LDP obtains with probability at least* $1-\zeta$*, cumulative pseudoregret:*

$$\mathfrak{R}(T) = \mathcal{O}\left(T^{\frac{3}{4}}\log(T)^{\frac{d+2}{4}}\sqrt{\frac{\gamma_T}{\varepsilon}\ln\frac{1}{\delta}\ln\frac{1}{\zeta}}\right).$$

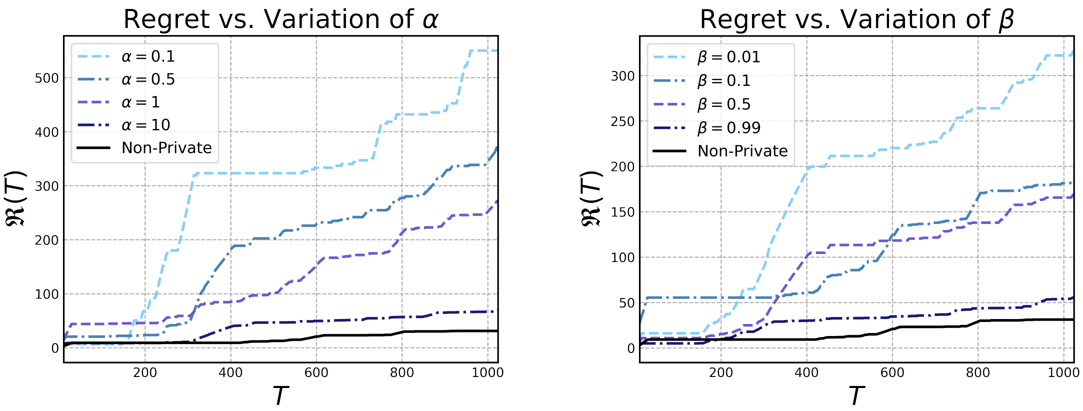
The proof for Corollary 7.3 follows directly by substituting the results from Lemma 7.6 into Corollary 7.1.

**Remark 7.7** (JDP vs. Locally JDP Regret)**.** Our algorithm for the locally JDP setting obtains $\widetilde{\mathcal{O}}(T^{3/4})$ regret in contrast to the JDP regret, which is close to the minimax optimal rate of $\widetilde{\Omega}(\sqrt{T})$ for squared-exponential and Matérn kernels (Scarlett et al., 2017). It is evident that this suboptimality is introduced by the $\widetilde{\mathcal{O}}(T)$ noise added via $\mathbf{H}_t$. However, we conjecture that in the absence of any known structure between the chosen actions $\mathbf{x}_1, ..., \mathbf{x}_{t-1}$, it is impossible to add correlated noise samples (i.e., such that the overall variance is $o(T)$) while maintaing local DP, as typically the environment selects $\mathcal{D}_t$ independently of $\mathcal{D}_{t-1}$.

## 7.6 Experiments

We conduct experiments primarily around the noisy Quadrature features for GP optimization, and consider the Joint DP setting. For more experimental results on the approximation guarantees of QFF, please refer to the appendix and experimental section of Mutny & Krause (2018), that analyse the efficiency of quadrature features in approximating stationary kernels.

We conduct experiments with input dimensionality $d = 2$ and selecting the squared-exponential kernel with variance 1, i.e., $k(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|_2^2/2)$ for simplicity. This choice was made as we essentially wish to demonstrate that the algorithms are private in practice for toy experiments, as larger dimensionalities $(d > 5)$ are rarely seen in prac-



(a) Ablation for the privacy budget $\varepsilon$.

(b) Ablations for the failure probability $\delta$.

Figure 7-1: Experimental comparisons for GP-UCB with privacy.

tice (Mutny & Krause, 2018) and would require additive assumptions for efficient inference (Munkhoeva et al., 2018).

**Experimental Setup**. We construct $f$ by randomly sampling a set of points $\mathcal{I}$ from $\mathcal{B}_d(2)$ such that $|\mathcal{I}| = 4$ and randomly generate $\boldsymbol{\alpha}$ from the unit $L_1$ ball $\mathcal{B}_d(1)$ (therefore, we assume $B = 1$). For any input point $\mathbf{x}$, $f(\mathbf{x})$ can then be denoted as $f(\mathbf{x}) = \sum_{i=1}^{|\mathcal{I}|} \alpha_i k(\mathbf{x}_i, \mathbf{x})$, where $\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{I}|}$ belong to $\mathcal{I}$. We consider $\mathcal{D}$ to be a random sample of size $n$ drawn from $\mathcal{B}_d(2)$ ($n$ may be variable, but is specified prior to each experiment). We draw $\mathcal{D}_t$ such that at least 1 sample $\mathbf{x}$ from $\mathcal{D}_t$ satisfies $f(\mathbf{x}) \geq 0.8$ and the others satisfy $f(\mathbf{x}) \leq 0.6$, ensuring a suboptimality gap of at least 0.2 (this is implemented somewhat crudely by iterative sampling). At each round $t$, the agent is presented with a random $\mathcal{D}_t$ and it obtains a reward $y_t$ drawn from the distribution $\text{Ber}(f(\mathbf{x}))$ and hence $|\xi_t| \leq 1$ and $\mathbb{E}[y_t] = f(\mathbf{x})$. Additionally, we see that the variance $\rho^2 = f(\mathbf{x})(1 - f(\mathbf{x}))$ for this case, but that is bounded from above by $1/4$. For simplicity, we restrict ourselves to Bernoulli rewards. This model, while ensuring sub-Gaussianity, also ensures that the rewards are bounded, and hence removes an additional logarithmic factor from the sensitivity analysis for the JDP setting. This can be observed by directly applying $L_2$-sensitivity to the JDP noise (Lemma 7.2), and ignoring the probabilistic argument.

**Effect of** $\varepsilon$. We first examine the effect of adjusting the privacy level $\varepsilon$. We fix $n = 25$, $\delta = 0.1$ and set $T = 1024$ (similar to Mutny & Krause (2018)). We run 20 trials and compare the performance at $\varepsilon = 0.1, 0.5, 1, 10$ (averaged over 20 trials). The regret scales as predicted with decreasing $\varepsilon$ (Figure 7-1a).

| Algorithm | camel | styb | mw |
|---|---|---|---|
| Non-Private | 519 | 885 | 901 |
| $\varepsilon = 10$ | 775 | 1667 | 1558 |
| $\varepsilon = 1$ | 1029 | 2680 | 2883 |
| $\varepsilon = 0.1$ | 3324 | 4493 | 5002 |

Table 7.1: Cumulative regret at $T = 10K$ averaged over 10 trials on functions from Mutny & Krause (2018) for a $\delta = 0.01$.

**Effect of** $\delta$. Next, we examine the effect of adjusting the privacy failure probability $\delta$. We fix $n = 25$, $\varepsilon = 1$ and set $T = 1024$ (similar to Mutny & Krause (2018)). We run 20 trials and compare the performance at $\delta = 0.01, 0.1, 0.5, 0.99$ (averaged over 20 trials). The regret increases with decreasing $\delta$, summarized in Figure 7-1b.

**Additional Benchmarks**. In addition to the environment proposed earlier, we additionally evaluate the JDP algorithm on previous benchmark environments. We consider the functional environments for the Camelback (`camel`), Stybtang-20 (`styb`) and Michalewicz-10 (`mw`) benchmarks from (Mutny & Krause, 2018). We observe a consistent increase in regret as the privacy budget ($\varepsilon$) is reduced (Table 7.1). While the bound predicts a $\varepsilon^{-\frac{1}{2}}$ deterioration, we observe a larger effect, which suggests that stronger analyses can close the gap.

## 7.7 Discussion and Concluding Remarks

In this paper, we presented the first *no-regret* algorithmic framework for differentially-private Gaussian Process bandit optimization for a class of stationary kernels in both the joint DP and local DP settings, extending the literature on private bandit estimation beyond multi-armed (Mishra & Thakurta, 2015) and linear (Shariff & Sheffet, 2018) problems. We rigorously analyse the proposed algorithms and demonstrate their provable efficiency in terms of regret, computation and privacy. Our work additionally introduces several new avenues for further research - while the dependence of the achieved pseudoregret on $T$ is near-optimal in the JDP setting, the LDP setting introduces an additional $\mathcal{O}(T^{1/4})$ which we conjecture is necessary owing to the nested estimation problems involved (Remark 7.7). Additionally, developing lower bounds on private GP regret and efficient kernel approximations for non-stationary kernels are valuable pursuits of inquiry.

## 7.8 Omitted Proofs

### 7.8.1 Intermediate Results

**Lemma 7.7** (Chernoff with Maximum Mean Bound)**.** *Let $X$ be any $\sigma$-sub-Gaussian random variable with mean $\mu \leq \mu^*$ for some constant $\mu^*$. Then, with probability at least $1-\beta$,*

$$|X| \leq \mu^* + \sigma\sqrt{2\ln\left(\frac{2}{\beta}\right)}.$$

*Proof.* $X$ is sub-Gaussian with variance, therefore by a Chernoff bound,

$$\begin{aligned} &\mathbb{P}\left(X - \mu > t\right) \leq \exp\left(-\frac{t^2}{2\sigma^2}\right) \\ \implies &\mathbb{P}\left(X > t + \mu\right) \leq \exp\left(-\frac{t^2}{2\sigma^2}\right) \end{aligned}$$

Subsitituing $t' = t + \mu$,

$$\begin{aligned} \implies \mathbb{P}\left(X > t'\right) &\leq \exp\left(-\frac{(t'-\mu)^2}{2\sigma^2}\right) \\ \implies X &\leq \mu + \sigma\sqrt{2\ln\left(\frac{1}{\beta}\right)}. && \text{(With probability at least } 1-\beta\text{)} \end{aligned}$$

The same can be derived for the other tail. By combining both statements with a union bound we get the result. □

**Lemma 7.8** (DP with probabilistic $L_2$ sensitivity)**.** *Let $f : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^m$ be an arbitrary d-dimensional real-valued function with $L_2-$sensitivity $\Delta$ with probability at least $1-\beta'$, and $\varepsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\beta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta/\varepsilon$ is $(\varepsilon, \beta + \beta')-$differentially private.*

*Proof.* Denote two adjacent samples in $\mathcal{X}$ as $\mathbf{x}, \mathbf{x}'$. We release $\mathbf{y} = f(\mathbf{x}) + \eta$ and $\mathbf{y}' = f(\mathbf{x}') + \eta$, where $\eta$ is sampled from the corresponding Gaussian. Let $\Delta_2(f)$ denote the sensitivity of $f$. For any arbitrary subset $S$ of $\mathbb{R}^m$,

$$\begin{aligned} \mathbb{P}(\mathbf{y} \in S) &= \mathbb{P}(\mathbf{y} \in S;\ \Delta_2(f) \text{ is } \Delta) + \mathbb{P}(\mathbf{y} \in S; \Delta_2(f) \text{ is not } \Delta) \\ &= \mathbb{P}(\mathbf{y} \in S|\ \Delta_2(f) \text{ is } \Delta)\mathbb{P}(\Delta_2(f) \text{ is } \Delta) + \beta'\mathbb{P}(\mathbf{y} \in S|\ \Delta_2(f) \text{ is not } \Delta) \\ &\leq \mathbb{P}(\mathbf{y} \in S|\ \Delta_2(f) \text{ is } \Delta)\mathbb{P}(\Delta_2(f) \text{ is } \Delta) + \beta' \end{aligned}$$

$$
\begin{aligned}
&\leq e^{\varepsilon}\left[\mathbb{P}(\mathbf{y}' \in S |\ \Delta_2(f) \text{ is } \Delta) + \beta\right] \mathbb{P}(\Delta_2(f) \text{ is } \Delta) + \beta' \\
&= e^{\varepsilon}\mathbb{P}(\mathbf{y}' \in S |\ \Delta_2(f) \text{ is } \Delta)\mathbb{P}(\Delta_2(f) \text{ is } \Delta) + \beta\mathbb{P}(\Delta_2(f) \text{ is } \Delta) + \beta' \\
&\leq e^{\varepsilon}\mathbb{P}(\mathbf{y}' \in S;\ \Delta_2(f) \text{ is } \Delta) + \beta + \beta' \\
&\leq e^{\varepsilon}\mathbb{P}(\mathbf{y}' \in S) + \beta + \beta'.
\end{aligned}
$$

The second inequality is obtained by the Gaussian Mechanism (Theorem A.1 of Dwork and Roth (Dwork & Roth, 2014)). □

**Lemma 7.9** (Existence of Proximal Space (Lemma 4 of (Mutny & Krause, 2018)))**.** *Let $k$ be a kernel defining $\mathcal{H}_k$ and $f \in \mathcal{H}_k$, its RKHS, such that the spectral characteristic function is bounded by $B$. Assuming that the defining points of $f$ come from the set $\mathcal{D}$, let $\mathcal{F}_m$ be an approximating space with a mapping $\mathbf{\Phi}$ such that this mapping is an $\epsilon$-approximation to the kernel $k$. Then there exists $\widehat{\mu} \in \mathcal{F}_m$ (with corresponding feature $\widehat{\boldsymbol{\theta}}$ such that $\widehat{\mu}(\mathbf{x}) = \langle \widehat{\boldsymbol{\theta}}, \Phi(\mathbf{x}) \rangle$), such that $\sup_{\mathbf{x}\in\mathcal{D}} |\widehat{\mu}(\mathbf{x}) - f(\mathbf{x})| \leq B\epsilon$.*

*Assuming the spectral characteristic function for $f$ is given by $\boldsymbol{\alpha}(\omega) = \sum_{j\in\mathcal{I}} \alpha_j \exp(i\omega^\top \mathbf{x}_j)$, then $\widehat{\mu}(\mathbf{x}) = \sum_{j\in\mathcal{I}} \alpha_j \Phi(\mathbf{x})^\top \Phi(\mathbf{x}_j)$ and the corresponding $\widehat{\boldsymbol{\theta}} = \sum_{j\in\mathcal{I}} \alpha_j \phi(\mathbf{x}_j)$ for the index set $\mathcal{I}$ defining $f$.*

**Lemma 7.10** (Norm Bound for Proximal Function)**.** *Let $\widehat{\mu} \in \mathcal{F}_m$ denote the $\epsilon$-approximation of $f$ given by Lemma 7.9 and $\widehat{\boldsymbol{\theta}}$ denote the corresponding feature representation. Then $\|\widehat{\boldsymbol{\theta}}\|_2 \leq B$.*

*Proof.* Recall that by the Representer Theorem, $\widehat{\boldsymbol{\theta}} = \sum_{i\in\mathcal{I}} \alpha_i \Phi(\mathbf{x}_i)$ for some (possibly infinite) index set $\mathcal{I} \subseteq \mathcal{D}$. Then, we can write $\|\widehat{\boldsymbol{\theta}}\|_2^2 = \left\langle \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}} \right\rangle_{\mathcal{F}_m} = \sum_{i,j\in\mathcal{I}^2} \alpha_i\alpha_j \left(\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)\right)$. Then, we can utilize the property that $\widehat{\boldsymbol{\theta}}$ is van $\epsilon$-approximation of $\mu_t \in \mathcal{H}_k$:

$$
\begin{aligned}
\|\widehat{\boldsymbol{\theta}}\|_2^2 = \sum_{i,j\in\mathcal{I}^2} \alpha_i\alpha_j \left(\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)\right) &\leq \sum_{i,j\in\mathcal{I}^2} \alpha_i\alpha_j && (\|\Phi(\mathbf{x})\| \leq 1) \\
&\leq \max_{\omega} |\boldsymbol{\alpha}(\omega)|^2 && \text{(Lemma 4 of (Mutny \& Krause, 2018))} \\
&\leq B^2.
\end{aligned}
$$

Taking the square root gives us the final form. □

**Lemma 7.11** (Variance Approximation)**.** *Let $\sigma_t(\mathbf{x}) = \mathbf{k}_t(\mathbf{x})^\top (\mathbf{K}_t + (\lambda + \lambda_{\min})\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x})$ where the quantities in $\mathbf{k}_t$ and $\mathbf{K}_t$ are determined by $k$ via Equation 7.1. Then for all $t$ and $\mathbf{x} \in \mathcal{D}$, we have* *that,*

$$\widetilde{\sigma}_t(\mathbf{x}) \leq \sigma_t(\mathbf{x}) + \frac{2t^2\sqrt{\epsilon}}{\rho}.$$

*Proof.* First note that $\widetilde{\sigma}_t(\mathbf{x}) = \rho\|\Phi(\mathbf{x})\|_{\mathbf{V}_t^{-1}} \leq \rho\|\Phi(\mathbf{x})\|_{(\mathbf{G}_t+(\lambda+\lambda_{\min})\mathbf{I})^{-1}} = 1 - \widetilde{\mathbf{k}}_t(\mathbf{x})^\top(\widetilde{\mathbf{K}}_t + (\lambda+\lambda_{\min})\mathbf{I})^{-1}\widetilde{\mathbf{k}}_t(\mathbf{x})$. Now, we will bound the second quantity on the RHS by $\sigma_t^2(\mathbf{x})$.

$$\widetilde{\sigma}_t(\mathbf{x}) \leq \sigma_t(\mathbf{x}) + \mathbf{k}_t(\mathbf{x})^\top(\mathbf{K}_t + (\lambda+\lambda_{\min})\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x}) - \widetilde{\mathbf{k}}_t(\mathbf{x})^\top(\widetilde{\mathbf{K}}_t + (\lambda+\lambda_{\min})\mathbf{I})^{-1}\widetilde{\mathbf{k}}_t(\mathbf{x}).$$

Following identically the steps in Proposition 1 (by approximating the difference in terms of the Frobenius norm of $\widetilde{\mathbf{K}}_t - \mathbf{K}_t$ in terms of $\epsilon$) of Mutny *et al.*(Mutny & Krause, 2018) (appendix), we obtain the remainder of the proof. □

### 7.8.2 Proof of Lemma 7.4

Note that the output of the algorithm at any instant $t$ is merely $\mathbf{x}_t$, and the input data at any instant $t$ is $(\mathbf{x}_\tau, y_\tau)_{\tau<t}$. Therefore, we need to bound the ratio of probabilities for any two $t$-neighboring sequences $S$ and $S'$, for all $\tau \neq t$ and subset $\mathcal{S}_{-t} = \mathcal{S}_1 \times \mathcal{S}_2 \times ... \times \mathcal{S}_{t-1} \times \mathcal{S}_{t+1} \times ... \times \mathcal{S}_T \subset \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_{t-1} \times \mathcal{D}_{t+1} \times ... \times \mathcal{D}_T$. Consider the actions taken under $S$ as $\mathbf{x}_1, ..., \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, ..., \mathbf{x}_T$ and under $S'$ as $\mathbf{x}'_1, ..., \mathbf{x}'_{t-1}, \mathbf{x}'_{t+1}, ..., \mathbf{x}'_T$. Then, we have,

$$\frac{\mathbb{P}(\mathbf{x}_1, ..., \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, ..., \mathbf{x}_T \in \mathcal{S}_{-t})}{\mathbb{P}(\mathbf{x}'_1, ..., \mathbf{x}'_{t-1}, \mathbf{x}'_{t+1}, ..., \mathbf{x}'_T \in \mathcal{S}_{-t})} = \frac{\prod_{\tau=1,\tau\neq t}^{T} \mathbb{P}(\mathbf{x}_\tau \in \mathcal{S}_\tau | (\mathbf{x}_i, y_i)_{i=1}^{\tau} \in \mathcal{S}_{<t})}{\prod_{\tau=1,\tau\neq t}^{T} \mathbb{P}(\mathbf{x}'_\tau \in \mathcal{S}_\tau | (\mathbf{x}'_i, y'_i)_{i=1}^{\tau} \in \mathcal{S}_{<t})}$$

Since $S$ and $S'$ only differ in $\mathcal{D}_t$ and for identical subsequences up to instant $t$, $\mathcal{A}$ is not stochastic. Therefore,

$$\begin{aligned} &= \frac{\prod_{\tau>t} \mathbb{P}(\mathbf{x}_\tau \in \mathcal{S}_\tau | (\mathbf{x}_i, y_i)_{i=1}^{\tau} \in \mathcal{S}_{<\tau})}{\prod_{\tau>t} \mathbb{P}(\mathbf{x}'_\tau \in \mathcal{S}_\tau | (\mathbf{x}'_i, y'_i)_{i=1}^{\tau} \in \mathcal{S}_{<\tau})} \\ &= \frac{\mathbb{P}(\mathbf{x}_{t+1} \in \mathcal{S}_{t+1} | (\mathbf{x}_i, y_i)_{i=1}^{t+1} \in \mathcal{S}_{<t+1})}{\mathbb{P}(\mathbf{x}'_{t+1} \in \mathcal{S}_{t+1} | (\mathbf{x}'_i, y'_i)_{i=1}^{t+1} \in \mathcal{S}_{<t+1})} \\ &\leq e^{\varepsilon} + \delta \end{aligned}$$

Here, the last inequality follows from the fact that $S$ and $S'$ differ only in $\mathcal{D}_t$ and that $\mathcal{A}$ is $(\varepsilon, \delta)-$LDP for all $t$.

## 7.9 Algorithm Pseudocode

**Algorithm 12** Approximate GP-UCB

```
Input: $m, \Phi$ that $\epsilon$-uniformly approximates $k$.
Privatizer Initialize: $\mathbf{S}_1 = \mathbf{0}, \mathbf{u}_1 = \mathbf{0}$.
for round $t = 1, 2, ..., T$ do
    Server:
    Receive $\mathcal{D}_t$ from environment.
    Receive $\widetilde{\mathbf{S}}_t = \mathbf{S}_t + \mathbf{H}_t, \widetilde{\mathbf{u}}_t = \mathbf{u}_t + \mathbf{h}_t \leftarrow$Privatizer.
    Set $\mathbf{V}_t \leftarrow \widetilde{\mathbf{S}}_t + \lambda\mathbf{I}, \widetilde{\boldsymbol{\theta}}_t \leftarrow \mathbf{V}_t^{-1}\widetilde{\mathbf{u}}_t$.
    Compute $v_t$ based on Theorem 7.1.
    Select $\mathbf{x}_t \leftarrow \arg\max_{\mathbf{x}\in\mathcal{D}_t} \langle \widetilde{\boldsymbol{\theta}}_t, \Phi(\mathbf{x}) \rangle + v_t \|\Phi(\mathbf{x})\|_{\mathbf{V}_t^{-1}}$.
    Play arm $\mathbf{x}_t$ and obtain $y_t$.
    Send $(\Phi(\mathbf{x}_t), y_t) \rightarrow$Privatizer.
    Privatizer:
    Sending parameters:
    Obtain $\mathbf{H}_t, \mathbf{h}_t$ based on Section 7.5.
    Send $\widetilde{\mathbf{S}}_t = \mathbf{S}_t + \mathbf{H}_t, \widetilde{\mathbf{u}}_t = \mathbf{u}_t + \mathbf{h}_t \rightarrow$Server
    Updating parameters:
    Receive $\mathbf{x}_t, y_t \leftarrow$Server.
    Securely update $\mathbf{S}_{t+1} \leftarrow \mathbf{S}_t + \Phi(\mathbf{x})\Phi(\mathbf{x})^\top$ (Sec. 7.5).
    Securely update $\mathbf{u}_{t+1} \leftarrow \mathbf{u}_t + y_i\Phi(\mathbf{x})$ (Section 7.5).
end for
```

**Algorithm 13** Privatizer under JDP

```
Initialize: Binary tree $\mathcal{T}$.
for round $t = 1, 2, ..., T$ do
    Sending parameters:
    Obtain $\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t$ by traversing $\mathcal{T}$ to node $t$.
    Send $\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t \rightarrow$Server.
    Updating parameters:
    Receive $\mathbf{x}_t, y_t \leftarrow$Server.
    Insert $[\Phi(\mathbf{x}_t), y_t]^\top[\Phi(\mathbf{x}_t), y_t]$ into $\mathcal{T}$.
    Update noise values $\mathbf{n}$ on the inserted path $\mathcal{T}$.
end for
```

**Algorithm 14** GP-UCB with LDP

```
SERVER:
  **Initialize**: $\mathbf{S}_1 = \mathbf{0}, \mathbf{u}_1 = \mathbf{0}$.
  **for** round $t = 1, 2, ..., T$ **do**
    Send $\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t \rightarrow$CLIENT$(t)$.
    Receive updated $\widetilde{\mathbf{S}}_{t+1}, \widetilde{\mathbf{u}}_{t+1} \leftarrow$CLIENT$(t)$.
  **end for**
CLIENT$(t)$:
  Initialize $\sigma_X^2$ and $\sigma_u^2$ according to Lemma 7.5.
  Receive $\mathcal{D}_t$ from environment.
  Receive $\widetilde{\mathbf{S}}_t, \widetilde{\mathbf{u}}_t \leftarrow$SERVER.
  Set $\mathbf{V}_t \leftarrow \widetilde{\mathbf{S}}_t + \lambda \mathbf{I}, \widetilde{\boldsymbol{\theta}}_t \leftarrow \mathbf{V}_t^{-1} \widetilde{\mathbf{u}}_t$.
  Compute $v_t$ based on Theorem 7.1.
  Select $\mathbf{x}_t \leftarrow \arg\max_{\mathbf{x} \in \mathcal{D}_t} \langle \widetilde{\boldsymbol{\theta}}_t, \Phi(\mathbf{x}) \rangle + v_t \|\Phi(\mathbf{x})\|_{\mathbf{V}_t^{-1}}$.
  Play arm $\mathbf{x}_t$ and obtain $y_t$.
  Sample $\mathbf{N}_t, \mathbf{n}_t$ using $\sigma_X^2, \sigma_u^2$.
  Send $\widetilde{\mathbf{S}}_{t+1} \rightarrow \mathbf{S}_t + \Phi(\mathbf{x}_t)\Phi(\mathbf{x}_t)^\top + \mathbf{N}_t \rightarrow$SERVER.
  Send $\widetilde{\mathbf{u}}_{t+1} \rightarrow \mathbf{u}_t + y_t \Phi(\mathbf{x}_t) + \mathbf{n}_t \rightarrow$SERVER.
```

# Chapter 8

# Kernel Methods for Federated Decision-Making

## 8.1 Introduction

Up to this point, we have studied federated decision-making problems that are *homogeneous*, i.e., all agents are interacting with identical decision-making environments. We now consider the *heterogeneous* federated learning setting (Ghosh et al., 2019; Yu et al., 2020a), which, in a decision-making context, implies that each agent is present in a potentially unique environment that is "similar" to the environments of other agents[1].

Moreover, the assumption of homogeneity does not hold for most decentralized decision-making problems in practice (Boldrini et al., 2018). For instance, in a decentralized supply chain network (Thadakamaila et al., 2004), agents interact with similar but non-identical decision problems, since loads are generally distributed non-uniformly. In this setting, naïvely incorporating observations from neighboring agents may not be beneficial, and federated algorithms must be carefully designed to efficiently leverage network feedback.

A related problem is the online *social network clustering* of bandits, where, at every trial, a randomly selected agent interacts with the bandit (Cesa-Bianchi et al., 2013; Gentile et al., 2014, 2017; Li et al., 2016, 2019). In this formulation, a fixed (but unknown) clustering over the agents is assumed, where agents within a cluster have identical context functions. While the assumptions of linearity and clustering are feasible in the context

[1] This requirement of "similarity" is somewhat intuitive, as we cannot expect to be able to transfer information from entirely different bandit instances.

of social networks (Al Mamunur Rashid et al., 2006), these assumptions may not hold for general multi-agent environments, such as geographically-distributed computational clusters (Cano et al., 2016). In the case when each agent has its own unique decision problem, the clustering approach leads to a worst-case $\mathcal{O}(M)$ multiplicative increase in the group regret. Moreover, the *social network clustering* problem is *single-agent*, since at any trial, only one agent interacts with the bandit. Multi-agent settings have been considered for *social network clustering* (Korda et al., 2016), but without delayed feedback.

In this chapter, we assume, in contrast to the previous chapters, that each agent $v \in \mathcal{V}$ interacts with a separate bandit function $f_v$, where all functions $f_v,\ v \in \mathcal{V}$ have small norm in a known reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ (Schölkopf & Smola, 2005) specified by a fixed kernel $K_x$. This is a more general setting compared to the existing *clustering* or identical (i.e., homogeneous) settings in the earlier chapters, and allows us to propose a technique to measure the similarity between the functions $f_v$ via an agent-based similarity kernel, which can be learnt online when it is unknown. Under this formulation, we present FedUCB-Kernel, an algorithm for the multi-agent contextual bandit problem on networks. FedUCB-Kernel uses a "network" kernel $K_z$, to measure similarity between agent reward functions $f_1, ..., f_V$. When $K_z$ is known (such as, e.g., in cases when agents correspond to users in a social network), we can use $K_z$ to construct a product kernel $K = K_z \odot K_x$, and use $K$ (instead of $K_x$) to construct upper confidence bounds.

We consider the case when the decision sets presented to any agent can be infinite or continuum action spaces, and $\forall\ v \in \mathcal{V}, \|f_v\|_{\mathcal{H}} \leq B$ for some known constant $B$. In this kernelized bandit setting, a relevant single-agent baseline is the single-agent IGP-UCB (Chowdhury & Gopalan, 2017) algorithm, which, for example, obtains a regret of $\widetilde{\mathcal{O}}(\sqrt{MT}(B\sqrt{\gamma^x_{MT}} + \gamma^x_{MT}))$ when run for a total of $MT$ rounds, where $\gamma^x_{MT}$ is the *information gain* after $MT$ rounds, the structural complexity of the RKHS specified by $K_x$, as defined in the previous chapter (Definition 7.1). We demonstrate that FedUCB-Kernel obtains a regret of

$$\widetilde{\mathcal{O}}\left(\sqrt{MT \cdot \bar{\chi}(G_d)}\left(B\sqrt{\gamma^x_{MT}\gamma_z} + \gamma^x_{MT}\gamma_z\right)\right).$$

Here, $\gamma_z$ determines the similarity between functions $f_v$ via the network kernel $K_z$, and $\bar{\chi}(G_d)$[2] is a term accounting for the delayed propagation of information in the network

[2] In contrast to prior chapters, we use the notation $d$ to refer to the message life parameter instead of $\gamma$ to avoid confusion with the *information gain*.

$G$[3]. This bound is achieved by utilizing graph partitions to control the deviation in the confidence bound for each agent. We can further see that *information gain* via the contexts $\mathcal{X} \subset \mathbb{R}^n$ typically grows as $\mathcal{O}((\log T)^n)$ for popularly employed kernels such as the squared-exponential kernel, and the network similarity $1 \leq \gamma_z \leq M$ grows as the decision problems faced by the agents progressively become dissimilar.

In comparison to prior bounds derived in Chapter 6 for the decentralized case, we see that this bound indeed is better by a factor of $\sqrt{d}$. We achieve this rate by assuming communication occurs every round and hence we do not require a *time-dependent* communication protocol. Furthermore, we demonstrate that this constant can be improved to the independence number $\alpha(G_d)$ if the agents follow a leader-follower strategy in the homogeneous setting, extending the results from Chapter 6 to the RKHS setting.

Our bound is reminiscent of single-agent bounds with additional contexts (Deshmukh et al., 2017; Krause & Ong, 2011), which rely on a *known* $K_z$. However, in many cases, $K_z$ is (unknown) and requires estimation. For this case, we provide an alternative algorithm (without regret guarantees) via kernel mean embeddings (Christmann & Steinwart, 2010). Against state-of-the-art methods on a variety of real-world and synthetic multi-agent networks, our algorithm exhibits superior performance. Moreover, we present a variant, FedUCB-Eager, of our algorithm (without regret bounds) that comfortably outperforms FedUCB-Kernel and other benchmarks. This extends the current literature of federated bandit estimation from the stochastic multi-armed problem (Landgren et al., 2018; Martínez-Rubio et al., 2019; Landgren et al., 2016a) to a more general class of functions, and provides a technique to determine task similarity over arbitrary federated settings.

## 8.2 Preliminaries

**Problem Setup**. We consider a multi-agent setting of $M$ agents sitting on the vertices of a network represented by an undirected and connected graph $G = (\mathcal{V}, \mathcal{E})$[4]. We assume that agents each solve unique instances of kernelised contextual bandit problems. At each step $t = 1, 2, ...$ each agent $v \in \mathcal{V}$ obtains, at time $t$, a decision set $\mathcal{D}_{v,t} \subseteq \mathcal{X} \subset \mathbb{R}^n$, where $\mathcal{X}$ is a compact subset of $\mathbb{R}^n$. In this chapter, unlike previous chapters, we assume $\mathcal{D}_{v,t}$ to even be

[3] $\bar{\chi}(G_d)$ denotes the minimum clique number of the $d^{th}$ power of graph $G$, i.e., $G_d$ has an edge $(i, j)$ if there is a path of length at most $d$ between $i$ and $j$ in $G$.

[4] Our results and algorithm can be trivially extended to directed or disconnected case, by considering each connected subgraph individually, and considering statistics of the directed graph instead.

an infinite set or a continuum of actions, however, we discuss the case when it is a finite set of contexts $\mathbf{x}_{v,t}^{(1)}, \mathbf{x}_{v,t}^{(2)}, \dots$ later on, for which tighter regret bounds can be obtained. At each trial $t$, each agent selects an action $\mathbf{x}_{v,t} \in \mathcal{D}_{v,t}$, and receives a reward $y_{v,t} = f_v(\mathbf{x}_{v,t}) + \varepsilon_{v,t}$. Where $f_v : \widetilde{\mathcal{X}} \to \mathbb{R}$ is a fixed (but unknown) function, and $\varepsilon_{v,t}$ is additive noise such that the noise sequence $\{\varepsilon_{v,t}\}_{t=1,v\in\mathcal{V}}^{\infty}$ is conditionally $R$-sub-Gaussian.

**Single-Agent Kernelized UCB**. Our approach builds on the existing research for upper confidence bounds for bandit kernel learning, a line of research that has seen a lot of interest (Srinivas et al., 2009; Krause & Ong, 2011; Chowdhury & Gopalan, 2017; Valko et al., 2013). The central idea across all these approaches is to construct an upper confidence bound (UCB) envelope for the true function $f(\cdot)$ using an estimate $\hat{f}_t$, and then chooses an action $\mathbf{x}_t \in \mathcal{D}_t$ that maximizes this upper confidence bound, i.e., for some estimate $\hat{f}_t$ of $f$,

$$\mathbf{x}_t = \underset{\mathbf{x}\in\mathcal{D}_t}{\arg\max} \left[\hat{f}_t(\mathbf{x}) + \sqrt{\beta_t}\sigma_{t-1}(\mathbf{x})\right].$$

Here, $\beta_t$ is an appropriately chosen "exploration" parameter, and $\sigma_{t-1}$ can be thought of as the "variance" in the estimate $\hat{f}_t$. Existing UCB-based approaches aim to construct a sequence $(\beta_t)_t$ to ensure a near-optimal tradeoff between exploration and exploitation. The natural choice for $\hat{f}_t$ is the solution to the kernelised ridge regression. Given $\lambda \geq 0$ and $\mathbf{X}_{<t} = (\mathbf{x}_i, y_i)_{i=1}^t$,

$$\hat{f}_t = \underset{f\in\mathcal{H}}{\arg\min} \frac{1}{t} \sum_{(\mathbf{x},y)\in\mathbf{X}_{<t}} (f(\mathbf{x}) - y))^2 + \lambda\|f\|_{\mathcal{H}}^2. \tag{8.1}$$

The solution to the above problem (8.1) can be written as the following (Valko et al., 2013) (for $\boldsymbol{\kappa}_t(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}_i))_{i=1}^t, \mathbf{y}_t = (y_i)_{i=1}^t$ and $\mathbf{K}_t = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j\in[t]}$):

$$\hat{f}_t(\mathbf{x}) = \boldsymbol{\kappa}_t(\mathbf{x})^\top (\mathbf{K}_t + \lambda\mathbf{I})^{-1} \mathbf{y}_t.$$

For any particular choice of the sequence $(\beta_t)_t$, various algorithms can be obtained, with different regret guarantees. We now present the regret bounds obtained by various algorithms based on the *maximum information gain* (Definition 7.1).

**Remark 8.1** (UCB Regret for Single-Agent Algorithms)**.** Let $\delta \in (0, 1]$. For continuum-armed $\mathcal{D}_t$, choosing $\beta_t = 2B + 300\gamma_{t-1}^x \log^3(t/\delta)$ guarantees with probability at least $1 - \delta$

a regret of

$$\widetilde{\mathcal{O}}\left(\sqrt{T}\left(B\sqrt{\gamma_T^x}+\gamma_T^x\log^{3/2}(T)\right)\right).$$

This result was demonstrated in the seminal work of (Srinivas et al., 2009) (GP-UCB). In the work of Chowdhury & Gopalan (2017), this bound was improved to

$$\widetilde{\mathcal{O}}\left(\sqrt{T}\left(B\sqrt{\gamma_T^x}+\gamma_T^x\right)\right).$$

The key element of the approach was a new self-normalized martingale inequality for infinite-dimensional spaces with the choice of $\beta_t = B + R\sqrt{2\left(\gamma_{t-1}^x + 1 + \log\left(\frac{1}{\delta}\right)\right)}$. A frequentist regret bound of $\widetilde{\mathcal{O}}(\sqrt{\tilde{d}T})$ was provided via the Sup-KernelUCB algorithm of Valko et al. (2013) (for finite-armed $\mathcal{D}_t$), where $\tilde{d}$ is the **effective dimension** of $K_x$, a measure of the intrinsic dimensionality of the RKHS $\mathcal{H}$. $\tilde{d}$ is related to $\gamma^x$ as $\gamma^x \geq \Omega(\tilde{d}\ln\ln T)$. Further work has focused on improving bounds for various families of kernels, e.g., see Janz et al. (2020); Scarlett et al. (2017).

The primary goal in the federated learning setting is to provide each agent with stronger estimators that leverage observations from neighboring agents. A suitable baseline, therefore, in this setting, would be that of a centralized agent pulling $MT$ arms in a round-robin manner. Indeed, in the next section we present a lower bound for federated kernelized bandits for certain kernel families. More generally, however, existing single-agent algorithms propose a regret bound of $\widetilde{\mathcal{O}}(\gamma_{MT}^x\sqrt{MT})$ in this setting, and this is the comparative regret bound we wish to match. We do not focus on stronger controls for specific kernels or of the information gain $\gamma^x$, and for that we refer the reader to references in Remark 8.1.

## 8.3 Lower Bounds for Kernelized Federated Bandits

We now present lower bounds for the kernelized bandit in homogeneous federated settings, for both squared-exponential and Matérn kernels.

**Theorem 8.1.** *Fix $B > 0$. Then, for any $1-$sub-Gaussian bandit environment with $T$ rounds, every $M$ agent federated algorithm $\mathcal{A}$ interacting with any function $f \in \mathcal{F}(B)$ endowed with kernel $K_x$ must incur expected regret after $T$ rounds such that*

$$\mathfrak{R}(T;\mathcal{A}) = \Omega\left(\log\left(B^2 M(T-\tilde{d}(G))\right)^{\frac{n}{2}}\sqrt{M(T+\tilde{d}(G))}\right), \text{ if } K_x \text{ is sq-exp},$$

$$\mathfrak{R}(T;\mathcal{A}) = \Omega\left(B^{\frac{n}{2\nu+n}}(MT)^{\frac{n+\nu}{n+2\nu}}\right), \text{ if } K_x \text{ is Matérn with parameter } \nu.$$

*Furthermore, if $\mathcal{A}$ is a decentralized agnostic policy, then we have,*

$$\mathfrak{R}(T;\mathcal{A}) = \Omega\left(\log\left(\alpha_\star(G_\gamma)MTB^2\right)^{\frac{n}{2}}\sqrt{\alpha_\star(G_\gamma)MT}\right), \text{ if } K_x \text{ is sq-exp},$$
$$\mathfrak{R}(T;\mathcal{A}) = \Omega\left(B^{\frac{n}{2\nu+n}}(\alpha_\star(G_\gamma)MT)^{\frac{n+\nu}{n+2\nu}}\right), \text{ if } K_x \text{ is Matérn with parameter } \nu.$$

*Where $\tilde{d}(G)$ denotes the average delay incurred by any message across the network $G$, and $\alpha^\star(G_\gamma) = \frac{N}{1+\bar{d}_\gamma}$ is Turan's lower bound (Turán, 1941) on $\alpha(G_\gamma)$.*

*Proof.* The proof follows the argument in Scarlett et al. (2017) almost exactly except for the network argument presented in Theorem 2.5. Scarlett et al. (2017) present a construction of $K$ functions $f_1, ..., f_K$ to uniformly approximate the unknown function $f$, i.e., $\mathbb{E}[f] = \sum_{k=1}^{K} f_k$. In summary, for any gap $\varepsilon$, they select $K$ functions whose RKHS norm is bounded by $B$, with peak values $2\varepsilon$ such that the $\varepsilon-$optimal points for each function are disjoint. This proof can be found in Lemma 2 of Scarlett et al. (2017). Under this construction, the pseudoregret for GP bandit optimization can be reduced to a $K$ armed bandit (see Theorem 2 of Scarlett et al. (2017)). Observe that by Theorem 2.5, for any arbitrary multi-agent networked policy, we have that the expected regret for a $\varepsilon-$different policy over $K$ arms is bounded as,

$$\mathbb{E}_k\left[\sum_{t=1}^{T}\left(M\cdot r_k(t) - \sum_{i\in\mathcal{V}} r_{A_i}(t)\right)\right] \geq \varepsilon MT\left(0.5 - 4\varepsilon\sqrt{\frac{M(T-\tilde{d}(G))}{K}}\right)$$

From the above we have that if the horizon $T$ satisfies,

$$T \leq \frac{K}{256M\varepsilon^2} + \tilde{d}(G),$$

Then the cumulative regret is at least $\frac{\varepsilon MT}{4}$. Otherwise stated, the equivalent control on $\varepsilon$ is as follows.

$$\varepsilon \leq \sqrt{\frac{K}{256M(T-\tilde{d}(G))}}.$$

What we want is to select $\epsilon$ and $K$ such that it is at least as large as the above constraint.

However, we have that $\varepsilon$ is a function of $K$ and vice versa. Now, for the squared exponential kernel, we have from Equation (23) of Scarlett et al. (2017) that the number of viable arms for a squared-exponential kernel over $n$ dimensions is $K = \Theta\left(\log\left(\frac{B}{\varepsilon}\right)^{n/2}\right)$. Therefore, using the analysis directly from Scarlett et al. (2017) (Equations 78-83), we can determine that the appropriate $\varepsilon = \Theta\left(\frac{1}{256M(T-\tilde{d}(G))}\log\left(\frac{B^2M(T-\tilde{d}(G))}{256}\right)^{n/2}\right)$. Replacing this value of $\varepsilon$ and $K$ provides us the bound.

Similarly, for Matérn kernels, we select $K = \Theta\left(\left(\frac{B}{\varepsilon}\right)^{n/\nu}\right)$ and repeat the analysis. The second part follows identically for decentralized agnostic policies, starting from the regret decomposition in Theorem 2.5. □

## 8.4 Cooperative Kernelized Bandits

**Network Contexts.** Recall that for any agent $v$, the rewards $y_v$ are generated following $y_{v,t} = f_v(\mathbf{x}_{v,t}) + \varepsilon_{v,t}$. To provide a relationship between different $f_v$, we assume that the functions $f_v, v \in \mathcal{V}$ are parameteric functionals of some function $F : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ for a known *network context* space $\mathcal{Z}$ such that $\forall\, v \in \mathcal{V}, \exists\, \mathbf{z}_v \in \mathcal{Z}$ such that $\forall \mathbf{x} \in \mathcal{X}$,

$$f_v(\mathbf{x}) = F(\mathbf{x}, \mathbf{z}_v).$$

**Kernel Assumptions**. We denote the space $\mathcal{X} \times \mathcal{Z}$ as $\widetilde{\mathcal{X}}$, and the overall input $(\mathbf{x}, \mathbf{z})$ as $\tilde{\mathbf{x}}$. Furthermore, we assume that the function $F$ has a small norm in the reproducing kernel Hilbert space (RKHS, Schölkopf & Smola (2005)) $\mathcal{H}_K$ associated with a PSD kernel $K : \widetilde{\mathcal{X}} \times \widetilde{\mathcal{X}} \to \mathbb{R}$. $\mathcal{H}_K$ is completely specified by the kernel $K(\cdot, \cdot)$, and via an inner product $\langle \cdot, \cdot \rangle_K$ following the reproducing property. As is typical with the kernelized bandit literature, we assume a known bound on the RKHS norm of $F$, i.e. $\|F\|_K \leq B$, and we assume that the kernel has finite variance, i.e. $K(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \leq 1, \forall\, \tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$[5].

Finally, we must impose constraints on the interaction of the inputs $\mathbf{x}$ and $\mathbf{z}$ via two kernels $K_x(\cdot, \cdot)$ and $K_z(\cdot, \cdot)$. We assume that $K$ is a composition of two separate positive-semidefinite kernels, $K_z$ and $K_x$ such that $K_z : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, i.e., operating on the network contexts, and $K_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ operates on the action contexts. Our regret bounds assume

[5] These are typically made assumptions in the contextual bandit literature, and avoid scaling of the regret bounds. In the linear case, the first assumption corresponds to having a bound on the norm of the context vectors (Chowdhury & Gopalan, 2017), and the second is to ensure the methods are scale-free (Agrawal & Goyal, 2012).

that the overall kernel $K$ is formed via the Hadamard product of $K_x$ and $K_z$:

$$K\left((\mathbf{z},\mathbf{x}),(\mathbf{z}',\mathbf{x}')\right) = K_x(\mathbf{x},\mathbf{x}')K_z(\mathbf{z},\mathbf{z}').$$

**Remark 8.2** (Kernel Compositions)**.** For the development in the paper, we restrict ourselves to the Hadamard composition, however, it is important to note that this is not a limitation of our technique, and other compositions can be explored. See Section 8.10 for details on the sum ($K_z \oplus K_x$), and Kronecker ($K_z \otimes K_x$) compositions.

**Remark 8.3** (Independent vs. Pooled Modeling)**.** When $\mathcal{D}_{v,t}$ are countably finite, an alternate formulation is the "independent" assumption (Li et al., 2010), where a separate model is considered for each "arm". We assume the "pooled" environment (Abbasi-Yadkori et al., 2011), (i.e., where all "arms" are modeled together), however it is easy to extend results to the former setting, by assuming arm-dependent network contexts (see Section 8.10).

The *network* kernel, $K_z$, determines how "similar" agent functions $f_v$ are. For example, if all agents solve the same bandit problem, i.e., $f_v = f\ \forall v \in \mathcal{V}$, then the appropriate choice for this is to set $\mathcal{Z} = \{1\}$, and $\mathbf{z}_v = 1$ for all $v \in \mathcal{V}$, and hence, $K = K_x$. Alternatively, in many internet applications, users (which may correspond to agents) are arranged in an online social network (say, $G_{\text{net}}$), and $\mathbf{z}_v$ can be a network embedding of user $v$ in $G_{\text{net}}$. Typically, however, $\mathcal{Z}$ can be defined more generally with a corresponding positive semi-definite (PSD) kernel $K_z$.

### 8.4.1 Peer-to-Peer Communication

In this chapter as well, we will be operating in the *decentralized* setting (see Chapter 2 for a detailed description of the protocol), where agents communicate via peer-to-peer messages. We remark that our results can easily be extended to the distributed setting without significant changes to the algorithm. The protocol assumes that pulling a bandit arm and communication occur sequentially within each trial $t$, i.e., first, each agent $v \in \mathcal{V}$ pulls an arm $\tilde{\mathbf{x}}_{v,t}$ and receives a reward $y_{v,t}$ from the respective bandit environment. The agent then sends the message $\boldsymbol{m}_{v,t} = \langle t, v, \tilde{\mathbf{x}}_{v,t}, y_{v,t}\rangle$ to its neighbors in $G$. This message is forwarded from agent to agent $d$ times (taking one trial of the bandit problem each between forwards), after which it is dropped. The *time-to-live* (delay) parameter $d$ is a common technique to control communication complexity in this setting. Each agent $v \in \mathcal{V}$ therefore also receives messages $\boldsymbol{m}_{v',t-d(v,v')}$ from all the nodes $v'$ such that $d(v,v') \leq d$.

### 8.4.2 The FedUCB-Kernel Algorithm

In this section we present the primary algorithm, FedUCB-Kernel. The central ideas in the development of the algorithm are (a) to leverage the similarity of the agent kernels (as specified by $K_z$) and (b) to control the variance estimates $\sigma^2_{v,t-1}$ between agents by delayed diffusion of rewards.

**Using an Augmented Kernel**. For each agent $v \in \mathcal{V}$ we construct an upper confidence bound (UCB) envelope for the true function $f_v(\cdot) = F(\cdot, \mathbf{z}_v)$ over the space $\widetilde{\mathcal{X}}$. This is done by using the composition kernel $K$ instead of the action kernel $K_x$, which allows us to take the network context $\mathbf{z}_v$ into account. The agent then chooses an action that maximizes the upper confidence bound, following the typical approach in UCB-based algorithms. For any $v \in \mathcal{V}, \mathbf{x} \in \mathcal{D}_{v,t}$, the UCB can be given by,

$$\mathbf{x}_{v,t} = \underset{\mathbf{x} \in \mathcal{D}_{v,t}}{\arg\max} \left[ \widehat{F}_{v,t}\left(\mathbf{z}_v, \mathbf{x}\right) + \sqrt{\beta_{v,t}} \sigma_{v,t-1}\left(\mathbf{z}_v, \mathbf{x}\right) \right].$$

Here $\widehat{F}_{v,t}(\cdot, \mathbf{z}_v)$ is the agent's estimate for $f_v$ at time $t$, and the second term denotes the exploration bonus. Using $\mathbf{x}_{v,t}$, the agent can construct the *aggregate* optimal context $\tilde{\mathbf{x}}_{v,t} = (\mathbf{z}_v, \mathbf{x}_{v,t})$. $\widehat{F}_{v,t}$ is obtained by solving:

$$\widehat{F}_{v,t} = \underset{f \in \mathcal{H}_K}{\arg\min} \left( \sum_{(\tilde{\mathbf{x}}, y) \in \widetilde{\mathbf{X}}_{v,t}} \left(f(\tilde{\mathbf{x}}) - y\right)^2 \right) + \lambda \|f\|^2_{\mathcal{H}_K}. \tag{8.2}$$

Here, $\widetilde{\mathbf{X}}_{v,t} = (\tilde{\mathbf{x}}_i, y_i)_{i=1}^{n_v(t)}$ denotes the $n_v(t)$ total action-reward pairs available at time $t$. Note that this comprises not just personal observations, but additional observations available via the messages received until that time. The solution to the above problem (8.2) is given as:

$$\widehat{F}_{v,t}(\tilde{\mathbf{x}}) = \boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}})^\top \left(\mathbf{K}_{v,t} + \lambda \mathbf{I}\right)^{-1} \mathbf{y}_{v,t}.$$

Here, $\boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}}) = \left(K\left(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_i\right)\right)_{i=1}^{n_v(t)}$ denotes the vector of kernel values between the input vector $\mathbf{x}$ and all previously stored data by agent $v$, and similarly $\mathbf{y}_{v,t} = (y_{v,i})_{i=1}^{n_v(t)}$ denotes the vector of rewards. The matrix $\mathbf{K}_{v,t}$ denotes the $n_v(t) \times n_v(t)$ matrix of kernel evaluations of every pair of samples $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\mathbf{X}}_{v,t}$ possessed by agent $v$. To construct the sequence $(\beta_{v,t})_t$, following result motivates the upper confidence bound.

**Lemma 8.1.** *Let $\widetilde{\mathcal{X}} \subset \mathbb{R}^n$, and $F : \widetilde{\mathcal{X}} \to \mathbb{R}$ be a member of the RKHS of real-valued functions on $\widetilde{\mathcal{X}}$ with kernel $K$, and RKHS norm bounded by $B$. Then, with probability at least $1-\delta$, the following holds for all $\tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$, and simultaneously for all $t \geq 1, v \in \mathcal{V}$:*

$$\left|F(\tilde{\mathbf{x}}) - \widehat{F}_{v,t}(\tilde{\mathbf{x}})\right| \leq \sigma^2_{v,t-1}(\tilde{\mathbf{x}}) \left(B + R\sqrt{\ln \frac{\det\left(\lambda \mathbf{I} + \mathbf{K}_{v,t}\right)}{\delta^2} + 2\log(M)}\right).$$

The proof for this confidence bound can be derived by a union bound over all $M$ agents of the kernelized self-normalized concentration inequality from Chowdhury & Gopalan (2017) (Theorem 1). This result holds *simultaneously* for all $t \geq 1$, and hence prevents the second logarithmic term, in contrast to Srinivas et al. (2009) for continuum-armed $\mathcal{D}_{v,t}$. We denote the "variance" proxy for the UCB as

$$\sigma^2_{v,t-1}(\tilde{\mathbf{x}}) = K\left(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}\right) - \boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}})^\top \left(\mathbf{K}_{v,t} + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}}).$$

**Controlling Drift via Clique Partitions**. The fundamental idea in controlling regret is to bound the per-round regret incurred by any agent by the UCB "variance" term $\sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t})$, and the algorithm attempts to bound $\sum_{v\in\mathcal{V}} \sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t})$ by a quantity smaller than $\mathcal{O}(\sqrt{M})$ (i.e., improve over non-cooperative behavior). Our approach is to obtain this rate by partitioning $G$ into $g$ subgraphs $G'_1, ..., G'_g$, and ensuring that the variance terms are similar for each agent within a subgraph for all $t$. Our partitioning solution is a conservative one: let $\mathbf{C}$ be a clique covering of the $d$-power of $G$. For any clique $\mathcal{C} \in \mathbf{C}$, we restrict each agent $v \in \mathcal{C}$ to only accept observations from agents that belong to $\mathcal{C}$ as well. This ensures that at any $t$, any agent $v \in \mathcal{C}$ has an upper bound on $\sigma^2_{v,t-1}$ that depends only on $\mathcal{C}$. Therefore we can control the group regret within each clique $\mathcal{C}$, leading to a factor of $\sqrt{\bar{\chi}(G_d)}$ (instead of $M$) in the regret, where $\bar{\chi}(\cdot)$ is the clique number.

**Remark 8.4** (Computational complexity)**.** As outlined in Valko et al. (2013), it is possible to perform an $\mathcal{O}(1)$ update of the Gram matrix, via the Schur decomposition (l. 30-34, Algorithm 15). This update can also be applied when $K_z$ is unknown and approximated online, see Section 8.4.3.

Algorithm 15 presents the FedUCB-Kernel algorithm with a tunable exploration parameter $\eta$, which can be different from the parameter $\beta_{v,t}$ used in the analysis, as is typical in this setting (Gentile et al., 2014; Chowdhury & Gopalan, 2017). We now present a regret bound for this algorithm.

**Theorem 8.2** (Group Regret under Delayed Communication)**.** *Let* $\mathbf{C}$ *be a minimal clique covering of* $G_d$*. When* $\mathcal{D}_{v,t}$ *is continuum-armed, Algorithm 15 incurs a per-agent average regret that satisfies, with probability at least* $1-\delta$*,*

$$\mathfrak{R}(T) = \mathcal{O}\left(\sqrt{\bar{\chi}(G_d)\cdot\frac{T}{M}}\left(R\cdot\widehat{\gamma}_T + \sqrt{\widehat{\gamma}_T}\left(B + R\sqrt{2\log\frac{M\lambda}{\delta}}\right)\right)\right).$$

*Here* $\widehat{\gamma}_T = \max_{\mathcal{C}\in\boldsymbol{C}}\left[\log\det\left(\frac{1}{\lambda}\mathbf{K}_{\mathcal{C},T}+\mathbf{I}\right)\right]$ *is the overall information gain, and for any clique* $\mathcal{C}\in\mathbf{C}$*, the matrix* $\mathbf{K}_{\mathcal{C},T}$ *is the Gram matrix formed by actions from all agents within* $\mathcal{C}$ *until time* $T$*, i.e.* $(\tilde{\mathbf{x}}_{v,t})_{v\in\mathcal{C},t\in[T]}$*.*

The proof has been deferred to Section 8.9.2 for brevity. Note that the proof technique used in this result differs from the prior approaches in Chapters 6 and 7 as we introduce an additional parameter space $\mathcal{Z}$ corresponding to the network contexts, despite the leading terms in the bound being similar.

We first discuss the leading factors in the bound. Compared to single-agent bounds, a coarse approximation of our rate reveals an additional factor of $\mathcal{O}\left(\sqrt{\bar{\chi}(G_d)}\right)$. This factor arises from the delayed spread of information, and is equal to the minimum clique number of the $d$ power graph of the communication network. When $G$ is $d$-complete (i.e., $G_d$ is complete), $\bar{\chi}(G_d) = 1$, providing us the best rate. An example topology when this condition is realized include $d/2$-star graphs (one node at the center, and 'spikes' of $d/2$ nodes). Conversely, since we assume $G$ is connected, in the worst-case graph, $\bar{\chi}(G_d) = \lceil V/d \rceil$. in which case the regret bound is equivalent to that obtained when all agents run in isolation. This is achieved, for example, in a line graph. Now, we formalize the idea of heterogeneity among agents.

**Definition 8.1** (Heterogeneity)**.** *In a multi-agent setting with* $M$ *agents and context vector space* $\mathcal{Z}$ *with kernel* $K_z$*, let* $\mathbf{K}_z$ *be the matrix of pairwise interactions, i.e.,* $\mathbf{K}_z = (K(\mathbf{z}_v,\mathbf{z}_{v'}))_{v,v'\in\mathcal{V}}$*. Then, the corresponding* ***heterogeneity*** $\gamma_z$ *for this setting is defined as* $\gamma_z = rank(\mathbf{K}_z)$*.*

For the composition considered in our paper, we can derive a regret bound in terms of

$\gamma^x_{MT}$, i.e., the information gain from $MT$ actions $\mathbf{x}$ across agents and heterogeneity $\gamma_z$.

**Corollary 8.1.** *When $K = K_z \odot K_x$, Algorithm 15 incurs the following per-agent average regret, with probability at least $1 - \delta$,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left(\gamma_z \cdot \gamma_{MT} \cdot \sqrt{\bar{\chi}(G_d) \cdot \frac{T}{M} \cdot \log\left(\frac{M\lambda}{\delta}\right)}\right).$$

**Remark 8.5.** The regret bound displays a smooth interaction between the network structure, communication delays and agent similarity[6]. Corollary 8.1 implies that the communication effectively acts as a "mask" on the underlying performance, which is controlled by the proximity of the network contexts. For instance, when network contexts are identical (*homogeneous*), $\gamma_z = 1$, and then the network structure entirely determines the regret (via $\bar{\chi}(G_d)$). Conversely, if $\mathbf{K}_z$ is full-rank, then $\gamma_z = M$, and agents cannot leverage cooperation. In this case, no improvement can be obtained regardless of the density of $G$.

**Examples**. We now provide a few examples to illustrate the problem setting. Consider the case when $K_x$ and $K_z$ are both linear. In this case, the algorithm can be understood as a weighted variant of the linear UCB algorithm: for an observation $\mathbf{x}_{v,t}$ from an agent $v$ to $v'$, the "weighted" observation is given by $(\mathbf{z}_v^\top \mathbf{z}_{v'}) \cdot \mathbf{x}_{v,t}^\top \mathbf{x}_{v,t}$, and hence the "weight" is $(\mathbf{z}_v^\top \mathbf{z}_{v'})$. In an ideal implementation, the vectors $\mathbf{z} \in \mathbb{S}_{n-1}(1)$, i.e., the unit sphere in $n$ dimensions, such that $\mathbf{z}^\top \mathbf{z} = 1$, ensuring that personal observations are given a weight of 1. Alternatively, when both $K_z$ and $K_x$ are RBF kernels, we observe an additive effect, i.e., $K = \exp\left(-\frac{\|\mathbf{z}_v - \mathbf{z}_{v'}\|^2}{2\sigma_z^2} - \frac{\|\mathbf{x}_v - \mathbf{x}_{v'}\|^2}{2\sigma_x^2}\right) = \exp\left(-\frac{\|\hat{\mathbf{x}}_v - \hat{\mathbf{x}}_{v'}\|^2}{2}\right)$, where $\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x}/\sigma_x \\ \mathbf{z}_v/\sigma_z \end{bmatrix}$. Note that the additional factor incurred in comparison to single-agent learning is $\gamma_z = \mathcal{O}(n \log(M))$ for sq-exp $K_z$, and for any action or network kernel, the regret can be obtained via Remark 8.1.

### 8.4.3 Approximating Network Contexts

The previous analysis assumes the availability of the underlying *network context* vectors $\mathbf{z}_v$ for each agent (or at least oracle access to the kernel $K_z$), however, for many applications, this information is not available, and must be estimated from the contexts themselves. Our

[6] We demonstrate this smooth relationship for the product kernel, i.e. $K = K_z \odot K_x$, however, alternate relationships are worth exploring. For more details and results on some forms of kernel compositions, see Section 8.10.

approach is based on *kernel mean embeddings* (Blanchard et al., 2011; Christmann & Steinwart, 2010; Deshmukh et al., 2017).

Consider the network context space $\mathcal{Z}$ to be the RKHS $\mathcal{H}_{K_x}$, and we assume that the contexts $\mathbf{x}_{v,t}$ for each agent are drawn from an underlying probability density $\mathcal{P}_v$. The idea is to use $\mathbf{z}_v$ as a representation of $\mathcal{P}_v$, so that we can (with an appropriate metric), use $K_z$ as a measure of "similarity" of the context distributions. For this, we look towards *kernel mean embeddings* of the distributions $\mathcal{P}_v$ in the RKHS $\mathcal{H}_{K_x}$. This implies that the augmented context $\tilde{\mathbf{x}}_{v,t}$ at any time $t$ for any agent $v \in \mathcal{V}$ is $(\Psi(\mathcal{P}_v), \mathbf{x}_{v,t})$, where $\Psi(\mathcal{P}_v) = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_v}[\phi_x(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_v}[K_x(\cdot, \mathbf{x})]$ is the kernel mean embedding of $\mathcal{P}_v$ in $\mathcal{H}_{K_x}$. Using this, we can define the kernel $K_z$ as follows.

$$K_z\left(\Psi(\mathcal{P}_v), \Psi(\mathcal{P}_{v'})\right) = \exp\left(-\frac{\|\Psi(P_v) - \Psi(P_{v'})\|_2}{2\sigma_z^2}\right).$$

Here the variance $\sigma_z$ can be tuned via experimentation. We can estimate this kernel from the available context via the *empirical mean kernel embedding*. Note, however, that in order to ensure that the estimator is unbiased, for each round $t$, each agent $v$ must sample the empirical mean from i.i.d. samples (and not samples that depend on the bandit policy). Therefore, we compute the empirical mean embedding $\widehat{\Psi}_t(\mathcal{P}_v) = \frac{1}{t}\sum_{i=1}^{t} K_x\left(\cdot, \mathbf{x}'_{v,i}\right)$, where $\mathbf{x}'_{v,i}$ is a random sample from $\mathcal{D}_{v,i}$. Assuming that the decision sets $\mathcal{D}_{v,i}$ are composed of samples drawn i.i.d. from the space $\mathcal{P}_v$, this is an unbiased estimator of the true embedding. Now, we can calculate the *empirical kernel approximation* $\widehat{K}_{z,t}(\cdot, \cdot)$ at time $t$:

$$\widehat{K}_{z,t}(\mathcal{P}_v, \mathcal{P}_{v'}) = \exp\left(-\frac{\mathrm{MMD}_{\mathcal{H}}(\widehat{\Psi}_t(\mathcal{P}_v), \widehat{\Psi}_t(\mathcal{P}_{v'}))}{2\sigma_z^2}\right),$$

The empirical maximum mean discrepancy (MMD) (Gretton et al., 2012) is the measure employed to measure the divergence of the embeddings in $\mathcal{H}_{K_x}$, and is given by:

$$\mathrm{MMD}^2_{\mathcal{H}}(\widehat{\Psi}_t(\mathcal{P}_v), \widehat{\Psi}_t(\mathcal{P}_{v'})) = \sum_{\tau,\tau'}^{t,t}\left[K_x(\mathbf{x}'_{v,\tau}, \mathbf{x}'_{v,\tau'}) + K_x(\mathbf{x}'_{v',\tau}, \mathbf{x}'_{v',\tau'}) - 2K_x(\mathbf{x}'_{v,\tau}, \mathbf{x}'_{v',\tau'})\right].$$

Our next result describes how the approximation (constructed from $t$ samples each) $\widehat{K}_t = \widehat{K}_{z,t} \odot K_x$ deviates from the true kernel $K = K_z \odot K_x$ under this model.

**Lemma 8.2.** *For an RKHS $\mathcal{H}$, assume that $\|f\|_\infty \leq d$ for all $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$. Then, the* *following is true with probability at least $1 - \delta$ for all $\mathbf{x}, \mathbf{x}' \in \widetilde{\mathcal{X}}$:*

$$\left| \log \left( \frac{\widehat{K}_{z,t}\left(\mathbf{x}, \mathbf{x}'\right)}{K_z\left(\mathbf{x}, \mathbf{x}'\right)} \right) \right| \leq \frac{1}{\sigma_z^2} \left( \sup_{\mathcal{P} \in \mathfrak{P}_{\mathcal{X}}} \mathfrak{R}_t(\mathcal{H}, \mathcal{P}) + 2d\sqrt{\frac{1}{2t} \log \frac{1}{\delta}} \right).$$

*Here $\mathfrak{R}_t(\mathcal{H}, \mathcal{P}_v)$ denotes the t-sample Rademacher average (Bartlett & Mendelson, 2002) of $\mathcal{H}$ under $\mathcal{P}_v \in \mathfrak{P}_{\mathcal{X}}$.*

We prove this in Section 8.9.4. Lemma 8.2 implies the *consistency* of the empirical Kernel estimator, i.e., for any $v, v' \in \mathcal{V}$, $\widehat{K}_{z,t} \to K_z$ with probability 1 as $t \to \infty$. To obtain $K_z$ we can employ any other PSD kernel $K_P$ on $\mathcal{X}$ besides $K_x$ as well.

**Remark 8.6** (Regret of Simultaneous Estimation)**.** At any instant, the empirical heterogeneity is locally controlled, i.e., for a clique cover $\mathbf{C}$ of $G_d$, $\gamma_z \leq \max_{\mathcal{C} \in \mathbf{C}} |\mathcal{C}|^2$. This follows directly from Theorem 2 of Krause & Ong (2011) and the fact that for any agent in a clique $\mathcal{C}$, the empirical kernel approximation only takes $1/2\left(|\mathcal{C}| \cdot (|\mathcal{C}| - 1)\right)$ distinct values at any instant. This implies that sparse network settings can easily be shown to benefit from cooperation (i.e., when $|\mathcal{C}| = \mathcal{O}(M^{1/4})$), but future work can address stronger controls on the group regret.

## 8.5 Extensions

**Homogeneous Setting**. In the homogeneous federated setting, the network contexts for all agents are identical, and hence each agent is solving the same contextual bandit problem, a generalization of the setting in Part I and Chapter 6 to the RKHS setting. When the decision set is fixed (i.e. $D_{v,t} = D \subset \mathcal{X}$ for all $v, t$, Cesa-Bianchi et al. (2019b)) we can derive a variant of FedUCB-Kernel that provides near-optimal performance. The central idea of the algorithm is for centrally-positioned agents to essentially follow Algorithm 15, however, the agents that are positioned peripherally in $G$ mimic the actions (obtained after the appropriate delay) of these centrally positioned agents (Algorithm 17). We partition the set of agents $G$ into “central” and “peripheral” agents such that each peripheral agent is connected to at least one central agent in $G_d$. This algorithm is defined as DistUCB-Kernel as this algorithm is not decentralized. It just remains to define the partition.

We set the “central” agents $\mathcal{V}_C$ of $G$ as the maximal weighted independent set of $G_d$ (where, for any node $v \in \mathcal{V}$, the weight $w_v = N_d(v)$), and set the complement $\mathcal{V}_P = \mathcal{V} \setminus \mathcal{V}_C$

as the "peripheral" set. Each peripheral agent $p$ is assigned the central agent it is connected to (denoted as $\text{cent}(p)$), and in case any peripheral agent is connected to more than one central agent, we assign it to the central agent with maximum degree in $G_d$. The set of peripheral agents assigned to a central agent $c$ is denoted by $\pi(c)$. We can then make the following claim about regret incurred in this setting.

**Theorem 8.3.** *If $\mathcal{D}_{v,t}$ is continuum-armed,* DistUCB-Kernel *incurs a per-agent average regret that satisfies, with probability at least $1-\delta$,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left(\gamma_z \cdot \gamma_{MT} \cdot \sqrt{\alpha(G_d) \cdot \frac{T}{M} \cdot \log\left(\frac{M\lambda}{\delta}\right)}\right).$$

*Here, $\alpha(G_d)$ refers to the independence number of $G_d$.*

The proof is presented in Section 8.9.5. The central concept utilized in this case is to partition the network in a manner that allows for a group of agents to make identical (albeit delayed) decisions. The regret analysis uses the property that the vertex cover of the elements of $\mathcal{V}_C$ spans $G_d$, and one can bound the regret by bounding the regret incurred by each "central" agent, since for any "peripheral" agent $v$, the regret incurred is only a constant larger than the correponding regret incurred by the "central" agent, i.e., $\mathcal{R}_v(T) \leq \widetilde{\mathcal{O}}(\sqrt{d}) + \mathcal{R}_{\text{cent}(v)}(T)$. Note that the proof of this case involves a more delicate analysis compared to the analysis present in Chapter 4 for the partially-decentralized algorithm, as we have to bound the change in the variance of the Gaussian process due to the delays, whereas in the earlier analysis, we only had to bound the change in arm pulls.

**Remark 8.7.** In addition to the tighter average per-agent regret (since $\alpha(G_d) \leq \bar{\chi}(G_d)$), we can make a stronger claim about the *individual* regret for any agent as well. Both these regret bounds match the rates mentioned for the context-free case in (Cesa-Bianchi et al., 2019b; Bar-On & Mansour, 2019). Moreover, when $d = 1$, then the bound on the group regret matches the lower bound shown in the nonstochastic case (Cesa-Bianchi et al., 2019a).

In comparison to the lower bound presented in Theorem 8.1, we see that up to polylogarithmic factors, we match the obtained rate in $T$ and $M$ for both squared-exponential and Matérn kernels. Comparing the network term, we see that while it is known that Turan's bound is not tight, we believe that the suboptimality is in the upper bound, where future work can improve the rates by a more careful federated exploration strategy.

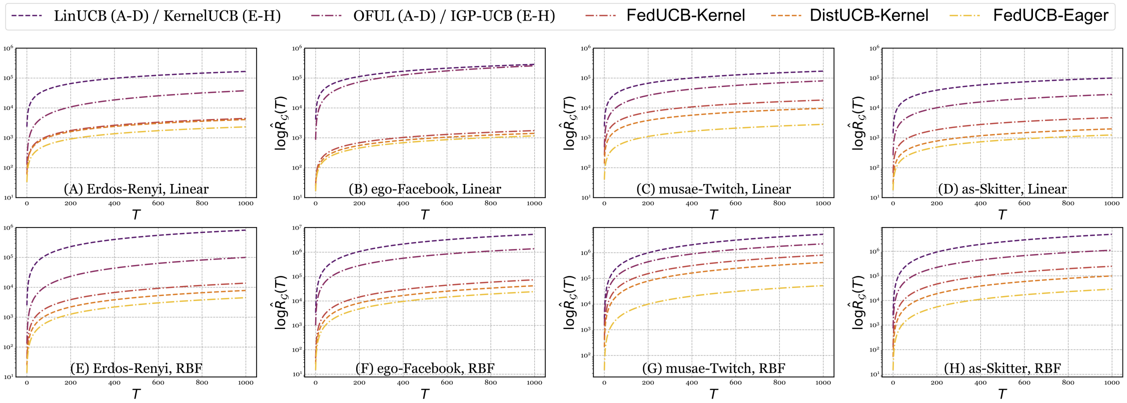


Figure 8-1: An experimental comparison of FedUCB-Kernel and its variants with benchmark techniques for contextual bandits. Each experiment is averaged over 100 trials. The top row denotes the linear kernel, and the bottom row denotes the RBF kernel.

**Eager Estimation**. In addition to the algorithms mentioned above, we also consider a (potentially stronger) variant of FedUCB-Kernel that does not take delays into account at all, and simply updates its observation set as soon as it obtains any new information (from any communicating agent). Consequently, for any arbitrary $G$ and $d$, this can lead to significant *drift* in the Gram matrices for any pair of agents, making this algorithm (dubbed FedUCB-Eager) significantly more challenging to analyze. We defer the analysis therefore to future work and present empirical evaluations of this variant in this paper. This algorithm can be understood as Algorithm 15 run with all observations (i.e., lines 20-22 in Algorithm 15 are ignored), and we present the complete pseudocode in Algorithm 16.

## 8.6 Experiments

The central aspect we wish to experimentally understand is the behavior of the algorithm with respect to network structures and delay in federated learning (alternatively, a detailed experimental comparison of single-agent KernelUCB under Gaussian noise can be found in (Srinivas et al., 2009; Krause & Ong, 2011)), and hence our experimental benchmark setup focuses on these aspects as well. We conduct two major lines of experimentation, the first on synthetically generated random networks, and the second on real-world networks subsampled from the SNAP network datasets (Leskovec & Sosič, 2016).

**Comparison Environments**. We compare in two benchmark setups. In order to compare performance with linear methods, our first setup assumes $K_x$ is the linear kernel, and $K_z$ is a clustering of the agents given by the independent sets of the $d^{th}$ power of the underlying connectivity graph $G$ ($K_z$ not known to the algorithms *a priori*, and $d = \text{diameter}(G)/2$). This is done to motivate the central application scenario where the network connectivity and task similarity are correlated. The second setup is where $K_x$ and $K_z$ are both randomly initialized Gaussian kernels (where $K_z$ is again unknown to our method). We run both setups on graphs of $M = 200$ nodes, $\mathcal{D}_{v,t}$ is a set of 8 randomly generated contexts for all $v \in \mathcal{V}, t \in T$ and dimensionality $d = 10$ for $\mathcal{X}$ and $\mathcal{Z}$ (for setup 2). For the kernel estimation task, we set $\sigma_z = 1$, and we set $\lambda = 1$.

**Network Structures**. We run experiments on two network structures - (a) synthetic, randomly generated networks and (b) real-world networks. For the synthetic networks, we generate random connected Erdos-Renyi networks (Erdős & Rényi, 1960) of size $M = 200$ with $p = 0.7$. For the synthetic networks, we subsample $M$ nodes and their corresponding edges (for $M = 200$) from the `ego-Facebook`, `musae-Twitch`, and `as-Skitter` networks, in order to represent a diverse set of networks found in social networks, peer-to-peer distribution and autonomous systems.

**Benchmark Methods**. In the linear setting, we compare against single-agent LinUCB (Li et al., 2010) (where every agent runs LinUCB independently), OFUL (Abbasi-Yadkori et al., 2011) and FedUCB-Kernel and FedUCB-Eager. In the kernel setting, we compare against single-agent KernelUCB (Valko et al., 2013), IGP-UCB (Chowdhury & Gopalan, 2017). Additionally, an important benchmark we compare against is *Naive Cooperation*, where agents run IGP-UCB (kernel) and LinUCB (linear) but include observations from neighbors as their own (without reweighting).

**Results Summary**. Figure 8-1 describes the regret achieved on each of the 4 benchmark networks for both linear and RBF (Gaussian) settings. Each plot is obtained after averaging the results for 100 trials, where the bandit contexts were refreshed every trial. We first highlight the general trend observed. Since the baseline techniques do not utilize cooperation at all, we expect them to provide a per-agent regret that scales linearly, instead of the $\mathcal{O}(1/\sqrt{M})$ dependence for our algorithms, which is obtained in our results as well. Among our algorithms, we see that FedUCB-Kernel and DistUCB-Kernel perform similarly for the Erdos-Renyi outperforms *Naive Cooperation* in both the linear and kernel settings, which can be attributed to the fact that naive cooperation does not take agent similarities into account. We observe that FedUCB-Eager consistently outperforms other algorithms, across all benchmark tasks.

Our motivation for this algorithm stems from work in the delayed feedback regime for the stochastic (context-free) bandit (Joulani et al., 2013), which suggests that incorporating observations *as soon as* they are available can provide optimal regret. While it is challenging to derive a provably optimal algorithm in the contextual setting (and more challenging in the multi-agent case), we simply extended the "as soon as" heuristic in FedUCB-Eager. The observed empirical regret suggests to us that FedUCB-Eager obtains $O(\sqrt{\frac{T}{M}(d+\alpha(G_d))}$ regret, lower than the other variants of FedUCB-Kernel. The other variations between different graph families can be attributed to the difference in connectivity (instead of the kernel approximation).

## 8.7 Discussion and Related Work

This paper is inspired by and draws from concepts in several (often disparate) subfields within the literature. We discuss our contributions with respect to these areas sequentially.

**Cooperative Multi-Agent Learning**. Cooperative bandit learning with delays has maintained the setting that all agents solve the same bandit problem (i.e., fully cooperative), which our work generalizes as a first step. In the nonstochastic (multi-armed) case (without delays), this problem was first studied in the work of Awerbuch & Kleinberg (2008), where they proposed an algorithm with a per-agent regret bound of $O(\sqrt{(1+KM^{-1})\ln T})$, which matches (up to logarithmic factors) our version of the bound in the same setting (with contexts). In the multi-armed case, (Landgren et al., 2016a,b, 2018; Martínez-Rubio et al., 2019) provide algorithms whose regret scales as a function of the graph Laplacian of $G$, using a *consensus* protocol (Bracha & Toueg, 1985). Our algorithms are based on a message-passing framework (i.e., Local), which maintains the same communication complexity, while providing significantly better regret guarantees. Moreover, we can express the consensus protocol as an instance (albeit restricted) of our algorithm, when $K_x(i,j) = \mu_i \mathbf{1}\{i=j\}, \mu_i \in \mathbb{R}$ is a scaled simplex, and $K_z(i,j) = A_{ij}^{d(i,j)}$ is the power of the graph Laplacian. Algorithms for the nonstochastic non-contextual case with delays have been developed in (Cesa-Bianchi et al., 2019b; Bar-On & Mansour, 2019), that propose algorithms with per-agent average regret scaling as $\widetilde{\mathcal{O}}(\sqrt{\alpha(G_d)TM^{-1}})$ and individual regret (for agent $v \in \mathcal{V}$) scaling as $\widetilde{\mathcal{O}}(\sqrt{(1+K|\pi(\text{cent}(v))|^{-1})T})$, which match the regret achieved by DistUCB-Kernel in the fully cooperative *contextual* setting. A minimax regret bound for the nonstochastic context-free of $O(\sqrt{(d+K)T})$ is also provided in (Cesa-Bianchi et al., 2019b), improving on the work of Neu et al. (2010), which our work improves up to smaller network factors $(\sqrt{\bar{\chi}(G_d)})$. When we compare our regret bounds with the algorithm-agnostic delayed feedback regret bounds provided by (Joulani et al., 2013) for the single-agent case, we observe the same relationship.

**Leveraging Social Contexts**. There has been extensive research in leveraging *social* side-observations across the bandit literature. Cesa-Bianchi et al. (2013) provide an algorithm called *GoB.Lin* that assumes an outer-product relationship between information flow in the network (via the graph Laplacian) and context information. This is exactly an instance of our framework, where the kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ is described by $\tilde{\phi}(\mathbf{x}_i)A_{\otimes}^{-1}\tilde{\phi}(\mathbf{x}_j)$ (in their notation), extended to the (kernel) multi-agent case with delayed feedback. In their setting, our regret bounds match exactly those of GoB.Lin. The clustering formulation can also be seen as a variant of the kernel framework, where $K_z(\mathbf{z}_v, \mathbf{z}_{v'}) = 1$ if agents belong to the same cluster, and 0 otherwise. The clustering is not known *a priori*, however, and the work of (Gentile et al., 2014, 2017; Li & Zhang, 2018) provides algorithms with tight regret guarantees for this case (our kernel embedding technique is similar in this regard). Again, we highlight that while multi-agent decision-making has been studied in the social network case (with non-identical contexts) (Korda et al., 2016; Wang et al., 2020c), none, to our knowledge, consider general graph communication with delays.

**Kernel Methods for Bandit Optimization**. A theoretical treatment of kernelised bandit learning was first explored in the work of Valko et al. (2013), which was built on the LinUCB (Li et al., 2010) and SupLinUCB (Chu et al., 2011), that were in turn inspired by the early work of Auer et al. (2002a). Our work is an improvement on the single-agent algorithms provided by Valko et al. (2013) owing to the martingale inequality presented in (Chowdhury & Gopalan, 2017), who use their result to construct improved versions single-agent Gaussian Process bandit algorithms (Krause & Ong, 2011; Srinivas et al., 2009). Our work also improves on the multi-task framework introduced by (Deshmukh et al., 2017) to the multi-agent setting with delays, along with a stronger regret bound, and an approximation guarantee for the *kernel mean embedding* approach to estimate task similarity. Recent results (Calandriello et al., 2019; Janz et al., 2020) on bandit optimization for certain kernel families can certainly be used to construct algorithmic variants with stronger guarantees on the context kernel $K_x$.

## 8.8 Conclusion

In this paper we presented FedUCB-Kernel, an kernelized algorithm for decentralized, multi-agent federated contextual bandits and proved regret bounds of $\widetilde{O}(\sqrt{\bar{\chi}(G_d)T/M})$ on the average pseudo-regret, and supported our theoretical developments with experimental performance. However, there are several aspects of the kernelised federated bandit problem that are left as open problems. An interesting first direction is to establish suitable *lower bounds* on the group regret in federated decision-making with delays. An $\Omega(\sqrt{MT})$ lower bound (Bubeck et al., 2012) can be derived for a single-agent playing $MT$ trials sequentially, however each of the $v \in \mathcal{V}$ agents can greatly reduce their uncertainty at every trial when cooperating, hence understanding the limits of cooperation is an interesting endeavor. Next, we presented a variant FedUCB-Eager of our algorithm that does not attempt to control the drift between the agent Gram matrices, that outperforms our main algorithm. Additionally, we presented matching lower bound for the homogeneous setting as well. We conjecture that a regret guarantee of the order $\widetilde{O}(\sqrt{(d+\alpha(G_d))T/M})$ exists for this algorithm in the linear case, and proving this under suitable assumptions is an interesting future direction as well. Finally, extending this line of research into the Bayesian case is also worth exploring.

## 8.9 Full Proofs

### 8.9.1 Intermediate Lemmas

**Theorem 8.4** (Theorem 2 of (Chowdhury & Gopalan, 2017)). *Let $D \subset \mathbb{R}^n$, and $f : D \to \mathbb{R}$ be a member of the RKHS of real-valued functions on $D$ with kernel $K$, and RKHS norm bounded* *by B. Then, with probability at least $1-\delta$, the following holds for all $\mathbf{x} \in D$, and $t \geq 1$:*

$$\left| f(\mathbf{x}) - \hat{f}_t(\mathbf{x}) \right| \leq s_t(\mathbf{x}) \left( B + R\sqrt{2 \ln \frac{\sqrt{\det\left((1+\eta)\mathbf{I}_t + \mathbf{K}_t\right)}}{\delta}} \right)$$

**Corollary 8.2.** *Let $\widetilde{\mathcal{X}} \subset \mathbb{R}^n$, and $F : \widetilde{\mathcal{X}} \to \mathbb{R}$ be a member of the RKHS of real-valued functions on $\widetilde{\mathcal{X}}$ with kernel $K$, and RKHS norm bounded by $B$. Then, with probability at least $1-\delta$, the following holds for all $\tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$, and simultaneously for all $t \geq 1, v \in \mathcal{V}$:*

$$\Delta_{v,t}(\tilde{\mathbf{x}}) \leq \sigma^2_{v,t-1}(\tilde{\mathbf{x}}) \left( B + R\sqrt{\ln \frac{\det\left(\lambda \mathbf{I} + \mathbf{K}_{v,t}\right)}{\delta^2} + 2\log(M)} \right).$$

*Proof.* This follows from Theorem 8.4 with probability $\delta/V$ for each agent $v \in \mathcal{V}$, and replacing $\lambda = 1 + \eta$.' □

**Theorem 8.5** (Theorem 2.1 of (ZI), Characterization of Schur Decomposition)**.** *Let $A$ be a Hermitian matrix given by*

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \textit{then, } A_{33} - A_{32}A_{22}^{-1}A_{23} \geq A_{33} - (A_{31}, A_{32}) \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} \begin{pmatrix} A_{13} \\ A_{23} \end{pmatrix}.$$

The central observation in the regret bound is the control of the "variance" terms in each clique directly in terms of the corresponding clique Gram matrix. We describe this result in the following lemma.

**Lemma 8.3** (Per-Clique Variance Bound)**.** *Let $\mathcal{C}$ be a clique in $G_d$ and the clique Gram matrix $\mathbf{K}_{\mathcal{C},T}$ be given by:*

$$\mathbf{K}_{\mathcal{C},T} = \begin{pmatrix} K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{1,1}) & ... & K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{|\mathcal{C}|,T}) \\ \vdots & \ddots & \vdots \\ K(\tilde{\mathbf{x}}_{|\mathcal{C}|,T}, \tilde{\mathbf{x}}_{1,1}) & \dots & K(\tilde{\mathbf{x}}_{|\mathcal{C}|,T}, \tilde{\mathbf{x}}_{|\mathcal{C}|,T}) \end{pmatrix}.$$

*Then, for any $T \geq d$,*

$$\sum_{t=d}^{T} \sum_{v \in \mathcal{C}} \sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq d|\mathcal{C}|B + \max(1, \frac{1}{\lambda}) \log\det\left(\frac{1}{\lambda}\mathbf{K}_{\mathcal{C},T} + \mathbf{I}\right).$$

*Proof.* Consider a hypothetical agent that pulls arms in a round-robin fashion for all agents in $\mathcal{C}$, i.e., let the agents within the clique $\mathcal{C}$ be indexed (without loss of generality) as $1, 2, ..., |\mathcal{C}|$, and the agent pulls arms $\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{2,1}, ..., \tilde{\mathbf{x}}_{1,2}, \tilde{\mathbf{x}}_{2,2}, ..., \tilde{\mathbf{x}}_{|\mathcal{C}|-1,T}, \tilde{\mathbf{x}}_{|\mathcal{C}|,T}$. Therefore, the agent will pull a total of $|\mathcal{C}|T$ arms. At any time $t \in [|\mathcal{C}|T]$, let the corresponding KernelUCB parameters for this agent be given by:

$$\mathbf{K}_{\mathcal{C},t} = \begin{pmatrix} K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{1,1}) & ... & K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{t \bmod |\mathcal{C}|, \lfloor t/|\mathcal{C}| \rfloor}) \\ \vdots & \ddots & \vdots \\ K(\tilde{\mathbf{x}}_{t \bmod |\mathcal{C}|, t}, \tilde{\mathbf{x}}_{1,1}) & \dots & K(\tilde{\mathbf{x}}_{t \bmod |\mathcal{C}|, \lfloor t/|\mathcal{C}| \rfloor}, \tilde{\mathbf{x}}_{t \bmod |\mathcal{C}|, \lfloor t/|\mathcal{C}| \rfloor}), \end{pmatrix}$$

$$\boldsymbol{\kappa}_{\mathcal{C},t}(\mathbf{x}) = \left[K(\mathbf{x}, \tilde{\mathbf{x}}_{1,1}), \dots, K(\mathbf{x}, \tilde{\mathbf{x}}_{t \bmod |\mathcal{C}|, \lfloor t/|\mathcal{C}| \rfloor})\right].$$

Consider the variance functional for any agent $v \in \mathcal{C}$ at time $t \in \sigma_\tau$:

$$\sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) = K\left(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v,t}\right) - \boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}}_{v,t})^\top \left(\mathbf{K}_{v,t} + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\kappa}_{v,t}(\tilde{\mathbf{x}}_{v,t})$$

Let $\tau$ be the instance at which the round-robin agent pulls arm $\tilde{\mathbf{x}}_{v,t-d}$. By Theorem 8.5, we have for $t \geq d$,

$$\leq K\left(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v,t}\right) - \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d})^\top \left(\mathbf{K}_{\mathcal{C},\tau} + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d}).$$

Therefore,

$$\sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq K\left(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v,t}\right) \\ - K\left(\tilde{\mathbf{x}}_{v,t-d}, \tilde{\mathbf{x}}_{v,t-d}\right) + K\left(\tilde{\mathbf{x}}_{v,t-d}, \tilde{\mathbf{x}}_{v,t-d}\right) - \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d})^\top \left(\mathbf{K}_{\mathcal{C},\tau} + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d}).$$

Let $\sigma^2_{\mathcal{C},\tau} = K\left(\tilde{\mathbf{x}}_{v,t-d}, \tilde{\mathbf{x}}_{v,t-d}\right) - \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d})^\top \left(\mathbf{K}_{\mathcal{C},\tau} + \lambda \mathbf{I}\right)^{-1} \boldsymbol{\kappa}_{\mathcal{C},\tau}(\tilde{\mathbf{x}}_{v,t-d})$. Summing up over all

$v \in \mathcal{C}$ and $t \geq d$:

$$\sum_{t=d}^{T} \sum_{v \in \mathcal{C}} \sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) = \sum_{t'=T-d}^{T} \sum_{v \in \mathcal{C}} K\left(\tilde{\mathbf{x}}_{v,t'}, \tilde{\mathbf{x}}_{v,t'}\right) + \sum_{\tau=1}^{|\mathcal{C}|(T-d)} \sigma^2_{\mathcal{C},\tau} \leq d|\mathcal{C}|B + \log\left(\prod_{\tau=1}^{|\mathcal{C}|T}(1+\sigma^2_{\mathcal{C},\tau})\right).$$

Here the inequality follows since the kernel $K$ is bounded by $B$ and $\sigma^2_{\mathcal{C},\tau} \leq \log(1+\sigma^2_{\mathcal{C},\tau})$. Lemma 7 of (Deshmukh et al., 2017) provides the following relationship for sequential pulls $\tilde{\mathbf{x}}_{v,t}, t \in [T]$ and their associated variance terms $\sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t})$ :

$$\prod_{t \in [T]} (1+\sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v,t})) = \frac{\det(\mathbf{K}_{\mathcal{C},T} + \lambda I)}{\lambda^{|\mathcal{C}|T+1}} = \det(\frac{1}{\lambda}\mathbf{K}_{\mathcal{C},T} + \lambda I).$$

This result is obtained using the determinant identity of the Schur decomposition provided by (ZI). Applying this result to $\mathbf{K}_{\mathcal{C},T}$ and variance terms $\sigma^2_{\mathcal{C},\tau}$ gives us the final result (since the round-robin agent pulls arms sequentially). □

### 8.9.2 Proof of Theorem 8.2

Consider the group pseudoregret at any instant $T$.

$$\mathfrak{R}(T) = \sum_{v \in G}\left(\sum_{t=1}^{T} r_{v,t}\right)$$

Let us examine the individual regret $r_{v,t}$ of agent $v \in \mathcal{V}$ at time $t$. From Theorem 8.1 and FedUCB-Kernel, we know that, for each agent $v \in \mathcal{V}$, $\beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + \hat{f}_{v,t}\left(\tilde{\mathbf{x}}_{v,t}\right) \geq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}^*_{v,t}) + \hat{f}_{v,t}\left(\tilde{\mathbf{x}}^*_{v,t}\right), f_v(\tilde{\mathbf{x}}^*_{v,t}) \leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}^*_{v,t}) + \hat{f}_{v,t}\left(\tilde{\mathbf{x}}^*_{v,t}\right)$ and $\hat{f}_v(\tilde{\mathbf{x}}_{v,t}) \leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + f_v\left(\tilde{\mathbf{x}}_{v,t}\right)$. Therefore for all $t \geq 1$ with probability at least $1-\delta$,

$$\begin{aligned} r_{v,t} &= f_v(\tilde{\mathbf{x}}^*_{v,t}) - f_v(\tilde{\mathbf{x}}_{v,t}) \\ &\leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + \hat{f}_{v,t}\left(\tilde{\mathbf{x}}_{v,t}\right) - f_v(\tilde{\mathbf{x}}_{v,t}) \\ &\leq 2\beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}). \end{aligned}$$

Therefore, for agent $v$, we have (since $\beta_{v,t} > \beta_{v,t-1}$ (Auer et al., 2002a)),

$$\sum_{t=1}^{T} r_{v,t} \leq 2\beta_{v,T}\sum_{t=1}^{T}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq 2d\sqrt{B}\beta_{v,d} + 2\beta_{v,T}\sum_{t=d}^{T}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t})$$

The second inequality follows from the fact that for all $t \leq d, \beta_{v,t} \leq \beta_{v,d}$ and that for all $v, t, \sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq \sqrt{B}$. We can now sum up the second term for the entire group of agents. Setting $\beta^*_T = \max_{v \in \mathcal{V}} \beta_{v,T}$, we get,

$$
\begin{aligned}
\sum_{t=d}^{T} \sum_{v \in \mathcal{V}} r_{v,t} &\leq 2\beta^*_T \left( \sum_{t=d}^{T} \sum_{v \in \mathcal{V}} \sigma_{v,t-1}\left(\tilde{\mathbf{x}}_{v,t}\right) \right) \\
&\leq 2\beta^*_T \sqrt{M(T-d)\left( \sum_{t=d}^{T} \sum_{v \in \mathcal{V}} \sigma^2_{v,t-1}\left(\tilde{\mathbf{x}}_{v,t}\right) \right)} \\
&\leq 2\beta^*_T \sqrt{M(T-d) \sum_{\mathcal{C} \in \mathbf{C}_d} \left( \sum_{t=d}^{T} \sum_{v \in \mathcal{C}} \sigma^2_{v,t-1}\left(\tilde{\mathbf{x}}_{v,t}\right) \right)} \\
&\overset{(a)}{\leq} 2\beta^*_T \sqrt{M(T-d) \sum_{\mathcal{C} \in \mathbf{C}_d} \left( d|\mathcal{C}|B + \max(1, \frac{1}{\lambda}) \log\left( \frac{\det(\mathbf{K}_{\mathcal{C},T} + \lambda \mathbf{I})}{\lambda^{|\mathcal{C}|T+1}} \right) \right)} \\
&\leq 2\beta^*_T \sqrt{M(T-d) \cdot \bar{\chi}(G_d) \cdot \max(1, \frac{1}{\lambda}) \left( dBV + \max_{\mathcal{C} \in \mathbf{C}} \left( \log\det(\mathbf{K}_{\mathcal{C},T} + \lambda \mathbf{I}) \right) \right)} \\
&\leq \beta^*_T \cdot \mathcal{O}\left( \sqrt{\bar{\chi}(G_d) \cdot MT \cdot \widehat{\gamma}_T} \right).
\end{aligned}
$$

Here, $(a)$ follows from Lemma 8.3. Now, from the definition of $\beta_{v,T}$ (Lemma 8.1), we know that, for all $v \in \mathcal{V}$ (where $v$ belongs to clique $\mathcal{C}$),

$$
\begin{aligned}
\beta_{v,T} &= B + R\sqrt{\lambda^{-1}} \sqrt{\log\left(\det\left(\mathbf{K}_{v,T} + \lambda \mathbf{I}\right)\right) + \log \frac{2M}{\delta}} \\
&\leq B + R\sqrt{\lambda^{-1}} \sqrt{\log\left(\det\left(\mathbf{K}_{\mathcal{C},T} + \lambda \mathbf{I}\right)\right) + \log \frac{2M}{\delta}} \\
&\leq B + R\sqrt{\lambda^{-1}} \sqrt{\widehat{\gamma}_T + \log \frac{2M\lambda}{\delta}} \\
\therefore \beta^*_T &= B + R\sqrt{\lambda^{-1}} \sqrt{\widehat{\gamma}_T + \log \frac{2M\lambda}{\delta}} \\
&= \mathcal{O}\left( B + R\sqrt{\widehat{\gamma}_T + \log \frac{2M\lambda}{\delta}} \right).
\end{aligned}
$$

Using this result in the earlier derivation, and then averaging over the number of agents $M$ gives us the final result.

### 8.9.3 Proof of Corollary 8.1

The proof follows directly from the following result being applied to Theorem 1.

**Lemma 8.4.** *Let* $\gamma_z = rk\,(\mathbf{K}_z)$*, where* $\mathbf{K}_z = (K_z(\mathbf{z}_v, \mathbf{z}_{v'}))_{v,v' \in \mathcal{V}}$*. When* $K = K_z \odot K_x$*,* $\widehat{\gamma}_T = 2\gamma_z\,(\gamma_T^x + \log(T))$*. When* $K = K_z \oplus K_x$*,* $\widehat{\gamma}_T = 2\,(\gamma_z \log(T) + \gamma_T^x)$ *.*

*Proof.* We first note that $\widehat{\gamma}_T \leq \log\det\left(\frac{1}{\lambda}\mathbf{K}_T + \mathbf{I}\right)$, where $\mathbf{K}_T = (K(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v',t'}))_{v,v' \in \mathcal{V}, t,t' \in [T]}$. Furthermore, note that (a) $\text{rk}(\mathbf{K}_T^z) = \text{rk}(\mathbf{K}_z)$, where $\mathbf{K}_T^z = (K_z(\mathbf{z}_{v,t}, \mathbf{z}_{v',t'}))_{v,v' \in \mathcal{V}, t,t' \in [T]}$, since $\mathbf{K}_T^z$ is composed entirely by tiling $T^2$ copies of $\mathbf{K}_z$. Now, to prove the first part, we simply use Theorem 2 of (Krause & Ong, 2011) on $\log\det\left(\frac{1}{\lambda}\mathbf{K}_T + \mathbf{I}\right)$. For the second part, we apply Theorem 3 of (Krause & Ong, 2011). □

### 8.9.4 Proof of Lemma 8.2

We first state a concentration result for the kernel mean embedding obtained by Smola et al. (2007).

**Lemma 8.5** (Smola et al. (2007))**.** *For an RKHS* $\mathcal{H}$*, assume that* $\|f\|_\infty \leq d$ *for all* $f \in \mathcal{H}$ *with* $\|f\|_{\mathcal{H}} \leq 1$*. Then, the following is true with probability at least* $1 - \delta$ *for any* $P_v \in \mathcal{P}_{\mathcal{X}}$*:*

$$\|\Psi(P_v) - \widehat{\Psi}_T(P_v)\| \leq 2\mathfrak{R}_T(\mathcal{H}, \mathcal{P}_{\mathcal{X}}) + d\sqrt{\frac{1}{T}\log(1/\delta)}.$$

We now begin the proof for Lemma 2 by analysing the absolute log-ratio of the estimated kernel and true kernel at any time instant $T$. Consider two samples $\mathbf{x}_i, \mathbf{x}_j \in \widetilde{\mathcal{X}}$ at any instant $t$. The ratio of the approximated and true kernel can be given by,

$$\begin{aligned}
\left|\log\left(\frac{\widehat{K}_t(\mathbf{x}_i, \mathbf{x}_j)}{K(\mathbf{x}_i, \mathbf{x}_j)}\right)\right| &= \frac{1}{2\sigma^2}\left|\|\Psi(P_i) - \Psi(P_j)\| - \|\widehat{\Psi}_T(P_i) - \widehat{\Psi}_T(P_j)\|\right| \\
&\leq \frac{1}{2\sigma^2}\left\|\Psi(P_i) - \widehat{\Psi}_T(P_i) - \Psi(P_j) + \widehat{\Psi}_T(P_j)\right\| \\
&\leq \frac{1}{2\sigma^2}\left(\left\|\Psi(P_i) - \widehat{\Psi}_T(P_i)\right\| + \left\|\Psi(P_j) - \widehat{\Psi}_T(P_j)\right\|\right).
\end{aligned}$$

Here, the first inequality is obtained via the reverse triangle inequality, and the second is obtained by Cauchy-Schwarz. Applying Lemma 8.5 with probability $\delta/2$ on each term in the RHS, and replacing the Rademacher average for a specific $P$ with the sup completes the proof.

### 8.9.5 Proof of Theorem 8.3

We begin with a few observations. Let the independent set used be given by $\mathcal{V}^* \subset \mathcal{V}$. For any agent $v \in \mathcal{V} \setminus \mathcal{V}^*$, let $c(v)$ denote the corresponding "center" agent that $v$ will mimic. Then, we first notice that for any $t \geq d(v, c(v))$, $\mathbf{x}_{v,t} = \mathbf{x}_{c(v),t-d(v,c(v))}$. We will continue with the notation used in the proof for Theorem 1.

**Lemma 8.6.** *Let $v \in \mathcal{V}^*$ be a "center" agent, and $N_d(v)$ denote its gamma neighborhood (including itself). Without loss of generality, consider an ordering $1, 2, ..., |N_d(v)|$ over the agents in $N_d(v)$. Now, we define the neigborhood Gram matrix $\mathbf{K}_{v,T}$ as:*

$$\mathbf{K}_{v,T} = \begin{pmatrix} K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{1,1}) & ... & K(\tilde{\mathbf{x}}_{1,1}, \tilde{\mathbf{x}}_{|N_d(v)|,T}) \\ \vdots & \ddots & \vdots \\ K(\tilde{\mathbf{x}}_{|N_d(v)|,T}, \tilde{\mathbf{x}}_{1,1}) & \dots & K(\tilde{\mathbf{x}}_{|N_d(v)|,T}, \tilde{\mathbf{x}}_{|N_d(v)|,T}) \end{pmatrix}.$$

*Assume all agents $\mathcal{V}$ follow* DistUCB-Kernel*. Then, for any agent $v \in \mathcal{V}^*$ and for any $T \geq d$,*

$$\sum_{t=d}^{T} \sum_{v' \in \mathcal{N}_d(v)} \sigma^2_{v,t-1}(\tilde{\mathbf{x}}_{v',t}) \leq Bd|N_d(v)| + \max(1, \frac{1}{\lambda}) \log\det\left(\frac{1}{\lambda}\mathbf{K}_{\mathcal{C},T} + \mathbf{I}\right).$$

*Proof.* The proof is obtained in a manner similar to Lemma 8.3, with the trivial modification that each agent $v \in \mathcal{V}^*$ considers observations from its entire neighborhood $N_d(v)$ and not just its parent clique. □

Now, consider the group pseudoregret at any instant $T$.

$$\mathfrak{R}(T) = \sum_{v \in G} \left( \sum_{t=1}^{T} r_{v,t} \right)$$

Let us examine the individual regret $r_{v,t}$ of agent $v \in \mathcal{V}$ at time $t$. From Theorem 8.1 and FedUCB-Kernel, we know that, for each agent $v \in \mathcal{V}$, $\beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + \hat{f}_{v,t}(\tilde{\mathbf{x}}_{v,t}) \geq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}^*_{v,t}) + \hat{f}_{v,t}(\tilde{\mathbf{x}}^*_{v,t})$, $f_v(\tilde{\mathbf{x}}^*_{v,t}) \leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}^*_{v,t}) + \hat{f}_{v,t}(\tilde{\mathbf{x}}^*_{v,t})$ and $\hat{f}_v(\tilde{\mathbf{x}}_{v,t}) \leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + f_v(\tilde{\mathbf{x}}_{v,t})$. Therefore for all $t \geq 1$ with probability at least $1 - \delta$,

$$\begin{aligned} r_{v,t} &= f_v(\tilde{\mathbf{x}}^*_{v,t}) - f_v(\tilde{\mathbf{x}}_{v,t}) \\ &\leq \beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) + \hat{f}_{v,t}(\tilde{\mathbf{x}}_{v,t}) - f_v(\tilde{\mathbf{x}}_{v,t}) \end{aligned}$$

$$\leq 2\beta_{v,t}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}).$$

Therefore, for agent $v$, we have (since $\beta_{v,t} > \beta_{v,t-1}$),

$$\sum_{t=1}^{T} r_{v,t} \leq 2\beta_{v,T}\sum_{t=1}^{T}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq 2d\sqrt{B}\beta_{v,d} + 2\beta_{v,T}\sum_{t=d}^{T}\sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t})$$

The second inequality follows from the fact that for all $t \leq d, \beta_{v,t} \leq \beta_{v,d}$ and that for all $v, t, \sigma_{v,t-1}(\tilde{\mathbf{x}}_{v,t}) \leq \sqrt{B}$. We can now sum up the second term for the entire group of agents. Setting $\beta_T^* = \max_{v\in\mathcal{V}}\beta_{v,T}$, we get,

$$\begin{aligned}
\sum_{t=d}^{T}\sum_{v\in\mathcal{V}} r_{v,t} &\leq 2\beta_T^*\left(\sum_{t=d}^{T}\sum_{v\in\mathcal{V}}\sigma_{v,t-1}\left(\tilde{\mathbf{x}}_{v,t}\right)\right) \\
&\leq 2\beta_T^*\sqrt{M(T-d)\left(\sum_{t=d}^{T}\sum_{v\in\mathcal{V}}\sigma^2_{v,t-1}\left(\tilde{\mathbf{x}}_{v,t}\right)\right)} \\
&\leq 2\beta_T^*\sqrt{M(T-d)\sum_{v\in\mathcal{V}^*}\left(\sum_{t=d}^{T}\sum_{v'\in N_d(v)}\sigma^2_{v',t-1}\left(\tilde{\mathbf{x}}_{v,t}\right)\right)} \\
&\overset{(a)}{\leq} 2\beta_T^*\sqrt{M(T-d)\sum_{v\in\mathcal{V}^*}\left(Bd|N_d(v)| + \max(1,\frac{1}{\lambda})\log\left(\frac{\det(\mathbf{K}_{v,T}+\lambda\mathbf{I})}{\lambda^{|\mathcal{C}|T+1}}\right)\right)} \\
&\leq 2\beta_T^*\sqrt{M(T-d)\cdot\alpha(G_d)\cdot\max(1,\frac{1}{\lambda})\left(dBV + \max_{v\in\mathcal{V}^*}\left(\log\det(\frac{1}{\lambda}\mathbf{K}_{v,T}+\lambda\mathbf{I})\right)\right)} \\
&\leq \beta_T^*\cdot\mathcal{O}\left(\sqrt{\alpha(G_d)\cdot MT\cdot\widehat{\gamma}_T^D}\right).
\end{aligned}$$

Note the alternate *information gain* quantity $\widehat{\gamma}_T^D = \max_{v\in\mathcal{V}^*}\log\det(\frac{1}{\lambda}\mathbf{K}_{v,T}+\lambda\mathbf{I})$. Here, $(a)$ follows from Lemma 8.6. Now, from the definition of $\beta_{v,T}$ (Lemma 8.1), we know that, for all $v \in \mathcal{V}^*$,

$$\begin{aligned}
\beta_{v,T} &= B + R\sqrt{\lambda^{-1}}\sqrt{\log\left(\det\left(\mathbf{K}_{v,T}+\lambda\mathbf{I}\right)\right) + \log\frac{2M}{\delta}} \\
&\leq B + R\sqrt{\lambda^{-1}}\sqrt{\log\left(\det\left(\mathbf{K}_{v,T}+\lambda\mathbf{I}\right)\right) + \log\frac{2M}{\delta}} \\
&\leq B + R\sqrt{\lambda^{-1}}\sqrt{\widehat{\gamma}_T^D + \log\frac{2M\lambda}{\delta}} \\
\therefore \beta_T^* &= B + R\sqrt{\lambda^{-1}}\sqrt{\widehat{\gamma}_T^D + \log\frac{2M\lambda}{\delta}}
\end{aligned}$$

$$= \mathcal{O}\left(B + R\sqrt{\widehat{\gamma}_T^D + \log\frac{2M\lambda}{\delta}}\right).$$

The above bound on $\beta_{v,T}$ holds even for agents not in $\mathcal{V}^*$ since they simply mimic one agent within $\mathcal{V}^*$, each for whom the above bound holds. Finally, applying the identical arguments as Lemma 8.4, we can bound $\widehat{\gamma}_T^D$ in terms of $\gamma_z$ and $\gamma_{MT}^x$. Dividing by the number of agents $M$ gives us the final result.

## 8.10 Alternative Models

### 8.10.1 "Independent" vs "Pooled" Settings

While we consider the pooled setting (Abbasi-Yadkori et al., 2011), we can easily extend our algorithm to the independent case (i.e., one bandit algorithm for each arm), by running $K$ different bandit algorithms in tandem (one for each arm), as specified in (Deshmukh et al., 2017). In order to leverage observations between arms, we must specify an additional kernel $K_{\text{arm}}$ and *arm contexts* for each arm. The overall kernel can then be given by,

$$\widetilde{K}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = K_{\text{arm}}(\boldsymbol{t}_1, \boldsymbol{t}_2) K_z(\mathbf{z}_1, \mathbf{z}_2) K_x(\mathbf{x}_1, \mathbf{x}_2)$$

Here, $\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{z}, \boldsymbol{t})$ is the augmented context that now contains both the task-based similarity context and the network-based similarity context in addition to the typical context vector $\mathbf{x}$. Alternatively, one can consider a joint kernel (where the arms and network contexts are intertwined), as follows.

$$\widetilde{K}(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = K_{\text{arm, network}}((\boldsymbol{t}_1 \mathbf{z}_1), (\boldsymbol{t}_2, \mathbf{z}_2)) K_x(\mathbf{x}_1, \mathbf{x}_2)$$

These modifications will only increase the regret at most by a factor of $\sqrt{K \cdot \text{rank}(K_{\text{arm}})}$ for all algorithms presented in this paper, by simply considering the latter case and following a similar analysis as the previous theorems.

### 8.10.2 Alternative Compositions

In this paper, we explore composition kernels of the Hadamard form, i.e., $\widetilde{\mathbf{K}} = \mathbf{K}_z \odot \mathbf{K}_x$. However, alternate formulations may be considered as well, first of which is the additive kernel, i.e., $\widetilde{\mathbf{K}} = \mathbf{K}_z \oplus \mathbf{K}_x$. For this case, we can rely on the following rank decomposition (Horn & Johnson, 2012):

$$\text{rank}(\mathbf{K}_z \oplus \mathbf{K}_x) \leq \text{rank}(\mathbf{K}_z) + \text{rank}(\mathbf{K}_x).$$

Alternatively, when one considers the Kronecker product, i.e., $\widetilde{\mathbf{K}} = \mathbf{K}_z \otimes \mathbf{K}_x$, we can use the following result from Schake (2004)(KRON 16) to bound the rank of the overall Gram matrix:

$$\text{rank}(\mathbf{K}_z \otimes \mathbf{K}_x) = \text{rank}(\mathbf{K}_z)\text{rank}(\mathbf{K}_x).$$

We omit these two compositions, however, as we found the Hadamard composition to work best in practice.

## 8.11 Algorithm Pseudocode

**Algorithm 15** FedUCB-Kernel

```
Input: Graph $G_d$ with clique cover $\boldsymbol{C}_d$, kernels $K_x(\cdot,\cdot), K_z(\cdot,\cdot)$, $\lambda$, explore param. $\eta$, buffers $\boldsymbol{B}_v = \phi$.
for For each iteration $t \in [T]$ do
    for For each agent $v \in \mathcal{V}$ do
        if $t = 1$ then
            $\mathbf{x}_{v,t} \leftarrow \textsc{Random}(D_{v,t})$.
        else
            $\mathbf{x}_{v,t} \leftarrow \underset{\mathbf{x} \in D_{v,t}}{\arg\max} \left( \hat{f}_{v,t}(\mathbf{z}_v, \mathbf{x}) + \frac{\eta}{\sqrt{\lambda}} \sigma_{v,t-1}(\mathbf{z}_v, \mathbf{x}) \right)$.
        end if
        $\tilde{\mathbf{x}}_{v,t} \leftarrow (\mathbf{z}_v, \mathbf{x}_{v,t}), y_{v,t} \leftarrow \textsc{Pull}(\tilde{\mathbf{x}}_{v,t})$.
        if $t = 1$ then
            $(\mathbf{K}_{v,t})^{-1} \leftarrow 1/K(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v,t}) + \lambda$.
            $\mathbf{y}_v \leftarrow [y_{v,0}]$.
            $\boldsymbol{\kappa}_v = (K(\cdot, \tilde{\mathbf{x}}_{v,t}))$.
        else
            $\boldsymbol{B}_v \leftarrow \boldsymbol{B}_v \cup (\tilde{\mathbf{x}}_{v,t}, y_{v,t})$.
        end if
        $\boldsymbol{m}_{v,t} \leftarrow \langle t, v, \tilde{\mathbf{x}}_{v,t}, y_{v,t} \rangle$.
        $\textsc{SendMessage}(\boldsymbol{m}_{v,t})$.
        for $\langle t', v', \tilde{\mathbf{x}}', y' \rangle$ in $\textsc{RecvMessages}(v, t)$ do
            if $v' \in \textsc{Clique}(v, \mathbf{C}_d)$ then
                $\boldsymbol{B}_v \leftarrow \boldsymbol{B}_v \cup (\tilde{\mathbf{x}}', y')$.
            end if
        end for
        for $(\tilde{\mathbf{x}}', y') \in \boldsymbol{B}_v$ do
            $\mathbf{y}_v \leftarrow [\mathbf{y}_v, y']$.
            $\boldsymbol{\kappa}_v = (\boldsymbol{\kappa}_v, K(\cdot, \tilde{\mathbf{x}}'))$.
            $\mathbf{K}_{22} \leftarrow \left( K(\tilde{\mathbf{x}}', \tilde{\mathbf{x}}') + \lambda - (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v \right)^{-1}$.
            $\mathbf{K}_{11} \leftarrow \left( (\mathbf{K}_{v,t})^{-1} + \mathbf{K}_{22} (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \right)$.
            $\mathbf{K}_{12} \leftarrow -\mathbf{K}_{22} (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v^\tau$.
            $\mathbf{K}_{21} \leftarrow -\mathbf{K}_{22} (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1}$.
            $(\mathbf{K}_{v,t})^{-1} \leftarrow [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}]$.
        end for
        $\boldsymbol{B}_v = \phi$.
        $\hat{f}_{v,t+1} \leftarrow (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \mathbf{y}_v$.
        $s_{v,\rho+1} \leftarrow \sqrt{K(\cdot,\cdot) - (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v}$.
    end for
end for
```

**Algorithm 16** FedUCB-Eager

```
Input: Graph $G_d$ with clique cover $\boldsymbol{C}_d$, kernels $K_x(\cdot,\cdot), K_z(\cdot,\cdot)$, $\lambda$, explore param. $\eta$, buffers $\boldsymbol{B}_v = \phi$.
for For each iteration $t \in [T]$ do
    for For each agent $v \in \mathcal{V}$ do
        if $t = 1$ then
            $\mathbf{x}_{v,t} \leftarrow$Random$(D_{v,t})$.
        else
            $\mathbf{x}_{v,t} \leftarrow \underset{\mathbf{x} \in D_{v,t}}{\arg\max} \left( \hat{f}_{v,t}(\mathbf{z}_v, \mathbf{x}) + \frac{\eta}{\sqrt{\lambda}} \sigma_{v,t-1}(\mathbf{z}_v, \mathbf{x}) \right)$.
        end if
        $\tilde{\mathbf{x}}_{v,t} \leftarrow (\mathbf{z}_v, \mathbf{x}_{v,t}), y_{v,t} \leftarrow$Pull$(\tilde{\mathbf{x}}_{v,t})$.
        if $t = 1$ then
            $(\mathbf{K}_{v,t})^{-1} \leftarrow 1/K(\tilde{\mathbf{x}}_{v,t}, \tilde{\mathbf{x}}_{v,t}) + \lambda$.
            $\mathbf{y}_v \leftarrow [y_{v,0}]$.
            $\boldsymbol{\kappa}_v = (K(\cdot, \tilde{\mathbf{x}}_{v,t}))$.
        else
            $\boldsymbol{B}_v \leftarrow \boldsymbol{B}_v \cup (\tilde{\mathbf{x}}_{v,t}, y_{v,t})$.
        end if
        $\boldsymbol{m}_{v,t} \leftarrow \langle t, v, \tilde{\mathbf{x}}_{v,t}, y_{v,t} \rangle$.
        SendMessage$(\boldsymbol{m}_{v,t})$.
        for $\langle t', v', \tilde{\mathbf{x}}', y' \rangle$ in RecvMessages$(v, t)$ do
            $\boldsymbol{B}_v \leftarrow \boldsymbol{B}_v \cup (\tilde{\mathbf{x}}', y')$.
        end for
        for $(\tilde{\mathbf{x}}', y') \in \boldsymbol{B}_v$ do
            $\mathbf{y}_v \leftarrow [\mathbf{y}_v, y']$.
            $\boldsymbol{\kappa}_v = (\boldsymbol{\kappa}_v, K(\cdot, \tilde{\mathbf{x}}'))$.
            $\mathbf{K}_{22} \leftarrow \left( K(\tilde{\mathbf{x}}', \tilde{\mathbf{x}}') + \lambda - (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v \right)^{-1}$.
            $\mathbf{K}_{11} \leftarrow \left( (\mathbf{K}_{v,t})^{-1} + \mathbf{K}_{22} (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \right)$.
            $\mathbf{K}_{12} \leftarrow -\mathbf{K}_{22} (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v$.
            $\mathbf{K}_{21} \leftarrow -\mathbf{K}_{22} (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1}$.
            $(\mathbf{K}_{v,t})^{-1} \leftarrow [\mathbf{K}_{11}, \mathbf{K}_{12}; \mathbf{K}_{21}, \mathbf{K}_{22}]$.
        end for
        $\boldsymbol{B}_v = \phi$.
        $\hat{f}_{v,t+1} \leftarrow (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \mathbf{y}_v$.
        $\sigma_{v,t+1} \leftarrow \sqrt{K(\cdot,\cdot) - (\boldsymbol{\kappa}_v)^\top (\mathbf{K}_{v,t})^{-1} \boldsymbol{\kappa}_v}$.
    end for
end for
```

**Algorithm 17** DistUCB-Kernel

```
**Input**: Graph $G_d$ with clique cover $\boldsymbol{C}$, kernels $K_x(\cdot,\cdot), K_z(\cdot,\cdot), \lambda, \eta$, buffer $\boldsymbol{B}_v = \phi \forall v \in \mathcal{V}$.
**for** For each iteration $t \in [T]$ **do**
    **for** For each agent $v \in \mathcal{V}$ **do**
        **if** $v \in \mathcal{V}_C$ **then**
            $\tilde{\mathbf{x}}_{v,t}, y_{v,t} \leftarrow$ Run lines 4-18 from Algorithm 15.
        **else**
            **if** $t \leq d(v, \text{cent}(v))$ **then**
                $\tilde{\mathbf{x}}_{v,t}, y_{v,t} \leftarrow$ KernelUCB (Valko et al., 2013) or IGP-UCB (Chowdhury & Gopalan, 2017).
            **else**
                $\tilde{\mathbf{x}}_{v,t}, y_{v,t} \leftarrow \textsc{PullLastStoredArm}(\text{cent}(v))$.
            **end if**
        **end if**
        $\boldsymbol{m}_{v,t} \leftarrow \langle t, v, \tilde{\mathbf{x}}_{v,t}, y_{v,t} \rangle$.
        $\textsc{SendMessage}(\boldsymbol{m}_{v,t})$.
        **if** $v \in \mathcal{V}_C$ **then**
            **for** $\langle t', v', \tilde{\mathbf{x}}', y' \rangle$ in $\textsc{RecvMessages}(v, t)$ **do**
                $\boldsymbol{B}_v \leftarrow \boldsymbol{B}_v \cup (\tilde{\mathbf{x}}', y')$.
            **end for**
            Run lines 22-33 in Algorithm 15.
        **else**
            $\textsc{UpdateLastStoredArm}(\text{cent}(v))$.
        **end if**
    **end for**
**end for**
```

# Part III

# Reinforcement Learning

# Chapter 9

# Federated Reinforcement Learning with Function Approximation

While the earlier chapters focused on federated decision-making in the bandit setting, many decision-making applications, e.g., in distributed robotics, are better formulated using Markov Decision Processes (MDPs), motivating us to study federated reinforcement learning. Recent research in the statistical learning community has focused on cooperative multi-agent decision-making algorithms with provable guarantees (Zhang et al., 2018b; Wai et al., 2018; Zhang et al., 2018a). However, prior work focuses on algorithms that, while are decentralized, provide guarantees on convergence (e.g., Zhang et al. (2018b)) but no finite-sample guarantees for regret, in contrast to efficient algorithms with function approximation proposed for single-agent RL (e.g., Jin et al. (2018, 2020); Yang et al. (2020b)). Moreover, optimization in the decentralized multi-agent setting is also known to be non-convergent without strong assumptions (Tan, 1993). Given its immediate widespread applicability in federated settings such as the internet of things and distributed robotics, regret minimization in federated reinforcement learning is an important real-world problem.

This chapter addresses reinforcement learning in the *independent* federated setting, i.e., when agents interact with isolated MDPs, and communicate among each other to improve convergence. The "isolation" here refers to the fact that the reward and transition function for any agent is independent of the state and action of other agents, similar to the federated bandit. In Chapter 10, we will consider the alternative, where a group of agents are placed in the same environment, and hence must be modeled together. From a technical perspective, there are several challenges beyond the federated bandit setting that arise in federated reinforcement learning. For instance, in contrast to the contextual and multi-armed bandit, where the size of each message is $\mathcal{O}(K)$ (where $K$ denotes the number of "arms"), reinforcement learning involves maintaining an $\mathcal{O}(K)-$sized statistic for each state transition, requiring $\mathcal{O}(K|\mathcal{S}|^2)-$sized messages (where $\mathcal{S}$ denotes the state space) to naively extend bandit algorithms to RL. This requirement on communication is prohibitively expensive for real-world applications, where the state space $\mathcal{S}$ can be extremely large. Furthermore, most existing work on federated reinforcement learning either provides no performance guarantees (Zhuo et al., 2019; Yu et al., 2020b) or provide guarantees only in the *tabular* setting with homogeneous environments (Agarwal et al., 2021).

In this chapter, we propose decentralized algorithms for federated reinforcement learning that are provably efficient with limited communication. We consider specifically the *low-rank* MDP, i.e., a Markov decision process that can be described (up to constant factors) by a $d-$dimensional linear representation. We discuss the federated problem of learning *low-rank* MDPs and provide several characterizations of *heterogeneity* or "non i.i.d.-ness" that correspond to real-world federated environments. We then present a federated algorithm for solving low-rank MDPs with $M$ agents that obtains competitive performance with bounded $\mathcal{O}(M^3)$ total rounds of communication.

Existing regret bounds for single-agent episodic RL in this *low-rank* or *linear* MDP setting scale as $\widetilde{\mathcal{O}}(H^2\sqrt{d^3T})$ for $T$ episodes of length $H$ each[1], leading to a cumulative regret of $\widetilde{\mathcal{O}}(MH^2\sqrt{d^3T})$ if $M$ agents operate in isolation. Similarly, an agent running for $MT$ episodes will consequently obtain $\widetilde{\mathcal{O}}(H^2\sqrt{d^3MT})$ regret. In comparison, we provide an algorithm built on least-squares value iteration (LSVI) titled FedLSVI, which obtains a group regret of $\widetilde{\mathcal{O}}((d+k)H^2\sqrt{(d+\Gamma)MT})$, where $\Gamma$ is a measure of heterogeneity between different MDPs, and $k$ is the size of the ambient space used to model this heterogeneity. When the MDPs are homogenous, our rate matches the *centralized* single-agent regret. We introduces several new aspects in the analysis of linear MDPs: first, we analyse stochastic communication and function approximation in the federated setting, presenting a novel concentration argument to bound the per-step estimation error. Next, we present two modes of communication, each with varying message sizes and associated regret bounds. For both approaches

[1] The $\widetilde{\mathcal{O}}$ notation ignores logarithmic factors and failure probability, and $d$ is the dimensionality of the ambient feature space. See, e.g., Yang et al. (2020b) and Jin et al. (2018) for the precise bounds.

we provide rigorous analyses of regret and a lower bound on the group regret for learning federated MDPs as well.

## 9.1 Federated Markov Decision Processes

Parallel MDPs (Sucar, 2007; Kretchmar, 2002) are a set of discrete time Markov decision processes that are executed in parallel, where a different agent interacts with the MDP in isolation. We consider a generalization of the parallel setting, which we call federated MDPs, in which each agent interacts with a potentially unique MDP and agents can occassionally communicate via a synchronization server[2]. Each agent interacts with their respective MDPs, each with identical (but disjoint) action and state spaces, but possibly unique reward functions and transition probabilities.

We denote the group of agents as $\mathcal{M}$ where the MDP for any agent $m \in \mathcal{M}$ is given by MDP$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}_m, \mathbf{r}_m)$. Here the state and action spaces are given by $\mathcal{S}$ and $\mathcal{A}$ respectively (which are assumed to be common across all agents), and the reward functions $\mathbf{r}_m = \{r_{m,h}\}_{h\in[H]}, r_{m,h} : \mathcal{S} \times \mathcal{A} \to [0,1]$[3], and transition probabilities $\mathbb{P}_m = \{\mathbb{P}_{m,h}\}_{h\in[H]}, \mathbb{P}_{m,h} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, i.e., $\mathbb{P}_{m,h}(x'|x,a)$ denotes the probability of the agent moving to state $x'$ if at step $h$ it selects action $a$ from state $x$. We assume that $\mathcal{S}$ is measurable with possibly infinite elements, and that $\mathcal{A}$ is finite with some size $A$. For any agent $m$, the policy $\pi_m$ is a set of $H$ functions $\pi_m = \{\pi_{m,h}\}_{m\in[M]}, \pi_{m,h} : \mathcal{S} \to \mathcal{A}$ such that $\sum_{a\in\mathcal{A}} \pi_{m,h}(a|x) = 1 \ \forall \ x \in \mathcal{S}$ and $\pi_{m,h}(a|x)$ is the probability of agent $m$ taking action $a$ from state $x$ at step $h$.

The problem proceeds as follows. At every episode $t = 1, 2, ...,$ each agent $m \in \mathcal{M}$ fixes a policy $\pi^t_m = \{\pi^t_{m,h}\}_{h\in[H]}$, and starts in an initial state $x^t_{m,1}$ picked arbitrarily by the environment. For each step $h \in [H]$ of the episode, each agent observes its state $x^t_{m,h}$, selects an action $a^t_{m,h} \sim \pi^t_{m,h}(\cdot|x^t_{m,h})$, obtains a reward $r_{m,h}(x^t_{m,h}, a^t_{m,h})$, and transitions to state $x^t_{m,h+1}$ sampled according to $\mathbb{P}_{m,h}(\cdot|x^t_{m,h}, a^t_{m,h})$. The episode terminates at step $H+1$ where agents receive 0 reward. After termination, the agents can communicate among themselves via a server, if required. The performance of any policy $\pi$ in the $m^{th}$ MDP is measured by the

[2] We leave the peer-to-peer communication setting as future work.

[3] We consider $r_{m,h}$ to be deterministic and bounded for simplicity. Our results can easily be extended to random rewards with sub-Gaussian densities.

value function $V^{\pi}_{m,h}(x) : \mathcal{S} \to \mathbb{R}$, defined $\forall\ x \in \mathcal{S}, h \in [H], m \in \mathcal{M}$ as,

$$V^{\pi}_{m,h}(x) \triangleq \mathbb{E}_{\pi}\left[\sum_{i=h}^{H} r_{m,i}(x_i, a_i) \;\middle|\; x_{m,h} = x\right].$$

The expectation is taken with respect to the random trajectory followed by the agent in the $m^{th}$ MDP under policy $\pi$. A related function $Q^{\pi}_{m,h} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ determines the total expected reward from any action-state pair at step $h$ for the $m^{th}$ MDP for any state $x \in \mathcal{S}$ and action $a \in \mathcal{A}$:

$$Q^{\pi}_{m,h}(x,a) \triangleq \mathbb{E}_{\pi}\left[\sum_{i=h}^{H} r_{m,i}(x_i, a_i) \;\middle|\; (x_{m,h}, a_{m,h}) = (x, a)\right].$$

Let $\pi^{\star}_{m}$ denote the optimal policy for the $m^{th}$ MDP, i.e., the policy that gives the maximum value, $V^{\star}_{m,h}(x) = \sup_{\pi} V^{\pi}_{m,h}(x)$, for all $x \in \mathcal{S}, h \in [H]$. We can see that with the current set of assumptions, the optimal policy for each agent is possibly unique. For $T$ episodes, the cumulative *group* regret (in expectation), is defined as,

$$\mathfrak{R}(T) \triangleq \sum_{t=1}^{T}\left[\sum_{m \in \mathcal{M}}\left[V^{\star}_{m,1}(x^{t}_{m,1}) - V^{\pi_{m,t}}_{m,1}(x^{t}_{m,1})\right]\right].$$

## 9.2 Least-Squares Value Iteration

Least-squares value iteration is a popular approach that recovers the optimal policy by finding the optimal value function (Sutton & Barto, 2018), and can be shown to converge to the optimal value function with probability 1 (Bellman & Kalaba, 1965). For any agent $m$, value iteration proceeds by obtaining the optimal Q-values $\{Q^{\star}_{m,h}\}_{h\in[H], m \in \mathcal{M}}$ by recursively applying the Bellman equation. Specifically, each agent $m \in \mathcal{M}$ constructs a sequence of action-value functions $\{Q_{m,h}\}_{h\in[H]}$ as, for each $x \in \mathcal{S}, a \in \mathcal{A}, m \in \mathcal{M}$,

$$Q_{m,h}(x,a) \leftarrow r_{m,h}(x,a) + \mathbb{P}_h[V_{m,h+1}](x,a),\ V_{m,h+1}(x,a) \leftarrow \max_{b \in \mathcal{A}}\left[Q_{m,h+1}(x,b)\right].$$

Where $\mathbb{P}_h[V(x,a)] = \mathbb{E}_{z\sim\mathbb{P}_h(\cdot|x,a)}\left[V(z)\right]$. Asymptotic convergence of the value iteration algorithm has been studied extensively in the optimal control and dynamic programming community (Williams & Baird, 1993). Further, it is known that value iteration reaches the optimal policy within a finite number of steps even if the value function itself has not converged (Bertsekas, 1987). It has been seen in practice that value iteration finds the optimal policy in a small number of steps, making it an appealing algorithm to analyse.

To obtain finite-sample regret guarantees (insted of asymptotic convergence), a recent line of work investigates value iteration in the parametric setting, i.e., when the MDP can be represented "approximately" by some function class $\mathcal{F}$, e.g., linear (Jin et al., 2020; He et al., 2021). We follow a similar approach with a key modification: we construct estimates using *multi-agent* historical data. For any function class $\mathcal{F}$, assume that any agent $m \in [M]$ has observed $k$ transition tuples $\{x_h^\tau, a_h^\tau, x_{h+1}^\tau\}_{\tau \in [k]}$ for any step $h \in [H]$. Then, the agent estimates the optimal Q-value for any step by solving the following regularized least-squares regression:

$$\widehat{Q}_{m,h}^t \leftarrow \underset{f \in \mathcal{F}}{\arg\min} \left\{ \sum_{\tau \in [k]} \left[ r_h(x_h^\tau, a_h^\tau) + V_{m,h+1}^t(x_{h+1}^\tau) - f(x_h^\tau, a_h^\tau) \right]^2 + \|f\|^2 \right\}. \tag{9.1}$$

Here, the targets $y_h^\tau = r_h(x_h^\tau, a_h^\tau) + V_{m,h+1}^t(x_{h+1}^\tau)$ denote the empirical value from specific transitions possessed by the agent, and $\|f\|$ denotes an appropriate regularization term based on the capacity of $f$ and the class $\mathcal{F}$. To foster exploration, an additional bonus $\sigma_{m,h}^t : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ term is added, inspired by the principle of optimism in the face of uncertainty, giving the final $Q$-value for any state-action $(x, a) \in \mathcal{S} \times \mathcal{A}$ as,

$$Q_{m,h}^t(x, a) = \min \left\{ \widehat{Q}_{m,h}^t(x, a) + \beta_{m,h}^t \cdot \sigma_{m,h}^t(x, a), H - h + 1 \right\}, \tag{9.2}$$

Since the policy is greedy with respect to the above $Q-$values, the value function is given as, for any state $x \in \mathcal{S}$,

$$V_{m,h}^t(x) = \max_{a \in \mathcal{A}} Q_{m,h}^t(x, a). \tag{9.3}$$

Here $\{\beta_{m,h}^t\}_{t \in [T], m \in \mathcal{M}}$ is a sequence selected appropriately to ensure that the estimated $Q$ values bound the optimal $Q$ values with high probability, similar to the exploration bonuses derived for bandit algorithms. For episode $t$, we denote $\boldsymbol{\pi}^t = \{\pi_m^t\}_{m \in \mathcal{M}}$ as the (joint) greedy policy with respect to the $Q$-values $\{Q_{m,h}^t\}_{h \in [H]}$ for each agent. While this describes the algorithm abstractly for any general function class $\mathcal{F}$, we first describe the *homogeneous* federated setting, where we assume $\mathcal{F}$ to be linear in $d$ dimensions, called the *linear* MDP (Jin et al. (2020), also see Bradtke & Barto (1996) and Melo & Ribeiro (2007)).

**Definition 9.1** (Linear MDP). *An MDP$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, \mathbf{r})$ is a linear MDP with feature map $\phi$ :*

*$\mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist $d$ unknown (signed) measures $\boldsymbol{\mu}_h = (\mu_h^1, ..., \mu_h^d)$ over $\mathcal{S}$ and an unknown vector $\boldsymbol{\theta}_h \in \mathbb{R}^d$ such that for any $(x, a) \in \mathcal{S} \times \mathcal{A}$,*

$$\mathbb{P}_h(\cdot|x, a) = \langle \boldsymbol{\phi}(x, a), \boldsymbol{\mu}_h(\cdot) \rangle, r_h(x, a) = \langle \boldsymbol{\phi}(x, a), \boldsymbol{\theta}_h \rangle$$

*We assume without loss of generality, $\|\boldsymbol{\phi}(x, a)\| \leqslant 1$ and $\max\{\|\boldsymbol{\mu}_h(\mathcal{S})\|, \|\boldsymbol{\theta}_h\|\} \leqslant \sqrt{d}$.*

We now present the algorithm FedLSVI in various environments.

## 9.3 FedLSVI in Homogeneous Federated Environments

As a warm up, we first describe FedLSVI in the homogenous setting, i.e., when the transition functions $\mathbb{P}_{m,h}$ and reward functions $r_{m,h}$ are identical for all agents and can be given by $\mathbb{P}_h$ and $r_h$ respectively for any episode $h \in [H]$. Corresponding to Eq. 9.1, we assume $\mathcal{F}$ to be the class of linear functions in $d$ dimensions over a *known* feature map $\boldsymbol{\phi}$, i.e., $f(\cdot) = \mathbf{w}^\top \boldsymbol{\phi}(\cdot), \mathbf{w} \in \mathbb{R}^d$, and set the ridge norm $\|\mathbf{w}\|_2^2$ as the regularizer. Furthermore, we fix a threshold constant $S$ that determines the amount of communication between the agents.

In a nutshell, the algorithm operates by each agent executing a local linear least-squares value iteration and then synchronizing observations between other agents if the threshold condition is met every episode. Specifically, for each $t \in [T]$, each agent $m \in \mathcal{M}$ obtains a sequence of value functions $\{Q_{m,h}^t\}_{h \in [H]}$ by iteratively performing linear least-squares ridge regression from the *multi-agent* history available from the previous $t - 1$ episodes. Note that any transition at some step $h$ can be described as $(n, x, a, z)$ where $n$ denotes the agent identity, $x$ denotes the initial state, $a$ denotes the action taken by the agent and $z$ denotes the new state the agent has transitioned to. During synchronization, it is assumed that a mechanism exists that allows all agents to share their personal transitions upto that round with all other agents. Now, if we assume that at any round $t$, the previous synchronization round occured after episode $k_t$, then, the set of transitions available to agent $m \in \mathcal{M}$ for any step $h$ before episode $t$ can be given by,

$$\mathcal{U}_h^m(t) = \left\{ \cup \left(n, x_{n,h}^\tau, a_{n,h}^\tau, x_{n,h+1}^\tau\right)_{n \in \mathcal{M}, \tau \in [k_t]} \right\} \bigcup \left\{ \cup \left(m, x_{m,h}^\tau, a_{m,h}^\tau, x_{m,h+1}^\tau\right)_{\tau = k_t + 1}^{t-1} \right\}.$$

For exposition denote $\Psi$ as an ordering of $\mathcal{U}_h^m(t)$, and $U = |\mathcal{U}_h^m(t)|$. We have that $\mathcal{U}_h^m$ is a set of $U$ elements, where each element is a set of the form $(n, x, a, z)$ as described earlier. Each agent $m$ first sets $Q^t_{m,H+1}$ to be $\mathbf{0}_d$, and for $h = H, ..., 1$, iteratively solves $H$ regressions:

$$\widehat{Q}^t_{m,h} \leftarrow \arg\min_{\mathbf{w}} \left\{ \sum_{(n,x,a,z) \in \mathcal{U}^m_h(t)} \left[ r_h(x,a) + V^t_{m,h+1}(z) - \mathbf{w}^\top \phi(x,a) \right]^2 + \lambda \|\mathbf{w}\|_2^2 \right\}. \tag{9.4}$$

Here $\lambda > 0$ is a regularizer. Next, $Q^t_{m,h}$ and $V^t_{m,h}$ are obtained via Equations 9.2 and 9.3, that can be defined completely after introducing some additional notation. The solution to Equation 9.8 can be given as $Q^t_{m,h}(x,a) = \phi(x,a)^\top \widehat{\mathbf{w}}^t_{m,h}$, where,

$$\widehat{\mathbf{w}}^t_{m,h} = \left(\mathbf{\Lambda}^t_{m,h}\right)^{-1} \mathbf{u}^t_{m,h}, \mathbf{\Lambda}^t_{m,h} = \sum_{\tau \in \Psi} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda \mathbf{I}_d, \text{ and } \mathbf{u}^t_{m,h} = \sum_{\tau \in \Psi} y_\tau \phi(x_\tau, a_\tau). \tag{9.5}$$

The solution presented above corresponds to the ridge regression solution computed over the covariates $\phi$ and targets $y$. Here, we denote the targets as $y_\tau = y_{m,h}(x_\tau, a_\tau, z_\tau) = r_h(x_\tau, a_\tau) + V^t_{m,h+1}(z_\tau)$. To obtain the confidence bonus, we first present the key lemma bounding the deviation of these predicted $Q$ value from the optimal one. We first make the following substitution. For any $(x,a) \in \mathcal{S} \times \mathcal{A}$,

$$\sigma^t_{m,h}(x,a) = \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} = \left(\phi(x,a)^\top (\mathbf{\Lambda}^t_{m,h})^{-1} \phi(x,a)\right)^{\frac{1}{2}}. \tag{9.6}$$

**Lemma 9.1.** *There exists an absolute constant $c_\beta$ such that when $\beta^t_{m,h} = c_\beta \cdot dH\sqrt{\log(dMHT/\delta')}$ for any policy $\pi$, for each $x \in \mathcal{S}, a \in \mathcal{A}$ we have for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously, with probability at least $1 - \delta'/2$ that,*

$$\left|\langle \phi(x,a), \widehat{\mathbf{w}}^t_{m,h} - \mathbf{w}^\pi_h \rangle\right| \leqslant \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) + c_\beta \cdot \sigma^t_{m,h}(x,a) \cdot dH \cdot \sqrt{\log\left(\frac{dMTH}{\delta'}\right)}.$$

Observe that if we set the $Q-$values of the greedy policy as, for some $\beta^t_{m,h} = c_\beta \cdot dH\sqrt{\log(dMHT/\delta')}$ where $c_\beta > 0$ is an absolute constant,

$$Q^t_{m,h} \leftarrow \underbrace{\langle \phi(x,a), \widehat{\mathbf{w}}^t_{m,h} \rangle}_{\widehat{Q}^t_{m,h}} + \underbrace{c_\beta \cdot \sigma^t_{m,h}(x,a) \cdot dH \cdot \sqrt{\log\left(\frac{dMTH}{\delta'}\right)}}_{\text{confidence bonus}},$$

we have that the estimated $Q-$values upper bound the corresponding $Q-$ values for *any* policy $\pi$, including the optimal policy, up to the factor $\mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a)$ which we

Table 9.1: Comparison of communication complexity and regret (homogeneous).

| Algorithm | Threshold $S$ | Communication Cost | Regret Speed-up |
|---|---|---|---|
| FedLSVI<br>(Dense) | $\infty$ | $0$ | $1$ |
| | $\mathcal{O}(1)$ | $\mathcal{O}(H^2M\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert\sqrt{dT\log(MT)})$ | $\mathcal{O}(\sqrt{M})$ |
| | $\mathcal{O}\left(\frac{T}{dM^2\log(MT)}\right)$ | $\mathcal{O}(dH^2M\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert\log(MT))$ | $\mathcal{O}\left(\sqrt{\frac{M}{\log(MT)}}\right)$ |
| | $\mathcal{O}\left(\frac{T\log(MT)}{dM^2}\right)$ | $\mathcal{O}(dH^2M^3\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert)$ | $\mathcal{O}\left(\frac{\sqrt{M}}{\log(MT)}\right)$ |
| FedLSVI<br>(Rare) | $c > 1$ | $\mathcal{O}\left(d^2H^2M(\frac{\log(MT)}{\log(c)})\right)$ | $\mathcal{O}\left(\sqrt{\frac{M}{c}}\right)$ |
| | $\mathcal{O}\left(\log(MT)\right)$ | $\mathcal{O}(d^3H^2M^3)$ | $o(1)$ |

show to be small later on. The strength of this result lies within the fact that this holds *simultaneously* for all agents *and* arbitrary communication protocols, which is a generalization of prior results, e.g., in Jin et al. (2020) to the federated setting. This is non-trivial, as in contrast to the single-agent setting, where the error term can directly be bound by a self-normalizing concentration argument, our terms are more complex: the error and communication protocol are not necessarily independent, causing the cumulative error to lose its martingale structure. We adopt a "worst-case communication" approach, which considers the maximum deviation under any arbitrary communication protocol and can then bound the worst-case deviation with only a constant $\sqrt{2}$ increase in the bound. The complete proof can be found in Section 9.8.

The corresponding exploration bonus ($\beta^t_{m,h} \cdot \sigma^t_{m,h}$) is similar to that of Gaussian process (GP) optimization (Srinivas et al., 2009) and linear bandit (Abbasi-Yadkori et al., 2011) algorithms, and it can be interpreted as the posterior variance of a Gaussian process regression. The motivation for adding the confidence term is similar to that in the bandit and GP case, to adequately overestimate the uncertainty in the ridge regression solution.

Once $c_\beta$ is fixed, the algorithm is straightforward. Given a set of transitions, each agent computes the above least-squares value iteration, and executes the greedy policy with respect to the obtained $Q^t_{m,h}$. The only element left to discuss is the synchronization of transitions. Observe that each transition is of the form $(n, x, a, z)$, and if $t$ episodes pass between successive synchronizations, one will require $\mathcal{O}(4t)$ bits per agent to communicate each transition, leading to an overall complexity of $\mathcal{O}(4MT)$ communication regardless of the schedule used to communicate. Alternatively, if $\mathcal{S}$ and $\mathcal{A}$ are countably finite such that $|\mathcal{S}|^2|\mathcal{A}| \ll T$, we can simply store the visitation "counts", providing a total communication complexity of $\mathcal{O}(n \cdot M|\mathcal{S}^2||\mathcal{A}|)$, if we have $n$ rounds of communication. We denote this as the first communication protocol, dubbed Dense.

This requirement of explicitly communicating transitions is necessary because of Equation 9.8. Observe that one requires, for each transition, the target state $z$ in order to compute the value function during a policy update. We can get away by communicating sufficient statistics instead of communicating transitions, if we restrict the policy updates to only in communicating rounds. This will reduce the communication complexity to $\mathcal{O}(nd^2H)$ per message, which will typically be much smaller than $|\mathcal{S}|^2|\mathcal{A}|$, at a cost to the regret. We denote this protocol as Rare. We first analyse Dense.

### 9.3.1 Dense Communication

In this protocol, each agent maintains two sets of parameters. The first is $\mathbf{S}^t_{m,h}$ which refers to the parameters that have been updated with the other agents via the synchronization, and the second is $\delta\mathbf{S}^t_{m,h}$, which refers to the parameters that are not synchronized. Now, in each step of any episode $t$, after computing $Q-$values (Eq. 9.2), each agent executes the greedy policy with respect to the $Q-$values, i.e., $a^t_{m,h} = \arg\max_{a\in\mathcal{A}} Q^t_{m,h}(x^t_{m,h}, a)$, and updates the unsyncrhonized parameters $\delta\mathbf{S}^t_{m,h}$. If any agent's new unsynchronized parameters satisfy the determinant condition with threshold $S$, i.e., if

$$\log \frac{\det\left(\mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h} + \lambda\mathbf{I}_d\right)}{\det\left(\mathbf{S}^t_{m,h} + \lambda\mathbf{I}_d\right)} \geqslant \frac{S}{(t-k_t)}, \tag{9.7}$$

the agent signals a synchronization with the server, and messages are exchanged. Here $k_t$ denotes the episode after which the previous round of synchronization took place (step 19 of Algorithm 18). The algorithm is summarized in Algorithm 18. The next result bounds communication complexity as a function of $S$.

**Lemma 9.2** (Communication Complexity). *If Algorithm 18 is run with threshold $S$, then the total number of episodes with communication $n \leqslant 2H\sqrt{d(T/S)\log(MT)} + 4H$. The total communication complexity under Dense is therefore $\mathcal{O}(H^2M|\mathcal{S}|^2|\mathcal{A}|\sqrt{d(T/S)\log(MT)})$ bits.*

The above result demonstrates that when $S = o(T)$, it is possible to ensure that the agents communicate only in a constant number of episodes, regardless of $T$. We now present the regret guarantee for the homogenous setting.

**Theorem 9.1** (Homogenous Regret with DENSE Communication)**.** *Algorithm 18 when run on $M$ agents with communication threshold $S$, $\beta^t_{m,h} = \mathcal{O}(H\sqrt{d\log(tMH)})$ and $\lambda = 1$ obtains the following cumulative regret after $T$ episodes, with probability at least $1-\alpha$,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left(dH^2\left(dM\sqrt{S} + \sqrt{dMT}\right)\sqrt{\log\left(\frac{1}{\alpha}\right)}\right).$$

The proof is presented in Section 9.8.

**Remark 9.1.** Theorem 9.1 claims that appropriately chosen $\beta$ and $\lambda$ ensures sublinear group regret. Similar to the single-agent analysis in linear (Jin et al., 2020) and kernel (Yang et al., 2020b) function approximation settings, our analysis admits a dependence on the (linear) function class via the $\ell_\infty-$covering number, which we simplify in Theorem 9.1 by selecting appropriate values of the parameters. Generally, the regret scales as

$$\mathcal{O}\left(H^2\left(M\sqrt{S} + \sqrt{MT\log\mathcal{N}_\infty(\epsilon^\star)}\right)\sqrt{\log\left(\frac{1}{\alpha}\right)}\right),$$

where $\mathcal{N}_\infty(\epsilon)$ is the $\epsilon$-covering number of the set of linear value functions under the $\ell_\infty$ norm, and $\epsilon^\star = \mathcal{O}(dH/T)$. We elaborate on this connection in the full proof.

**Remark 9.2** (Multi-Agent Analysis)**.** The aspect central to the multi-agent analysis is the dependence on the communication parameter $S$. If the agents communicate every round, i.e., $S = \mathcal{O}(1)$, we observe that the cumulative regret is $\widetilde{\mathcal{O}}(d^{\frac{3}{2}}H^2\sqrt{MT})$, matching the centralized setting. With no communication, the agents simply operate independently and the regret incurred is $\widetilde{\mathcal{O}}(M\sqrt{T})$, matching the group regret incurred by isolated agents. Furthermore, with $S = \mathcal{O}(T\log(MT)/dM^2)$ we observe that with a total of $\mathcal{O}(dHM^3)$ episodes with communication, we recover a group regret of $\mathcal{O}(d^{\frac{3}{2}}H^2\sqrt{MT}(\log MT))$, which matches the optimal rate (in terms of $T$). This dependence is shown in Table 9.1.

### 9.3.2 Rare Communication

The Rare communication protocol has two key differences from the Dense protocol: first, each agent only uses the *synchronized* transition functions to compute the policies, i.e., it does not use the personal transitions observed after any communication round until they have been synchronized with all other agents. Next, the earlier Dense setting communicated a vector of size $\mathcal{O}(H|\mathcal{S}|^2|\mathcal{A}|)$, where each element $(h, x, a, z)$ denoted the number of times the agent transitioned from state $x$ to state $z$ by taking action $a$ in step $h$. Instead, in this setting, the agents will communicate the sufficient statistics required for the least-squares value iteration, and the *server* computes the common policy.

Formally, we have that during each synchronization round, the server constructs a sequence of value functions $\{Q_h^t\}_{h\in[H]}$ by iteratively performing linear least-squares ridge regression from the *synchronized* multi-agent history available from the previous $t-1$ episodes. Now, if we assume that synchronization occurs after episode $t$, then, the cumulative set of transitions available can be given by,

$$\mathcal{U}_h(t) = \left\{ \cup \left(n, x_{n,h}^{\tau}, a_{n,h}^{\tau}, x_{n,h+1}^{\tau}\right)_{n\in\mathcal{M},\tau\in[t]} \right\}.$$

Let us denote the set of *personal* observations for any agent $m$ until time $t$ as $\mathcal{Z}_{m,h}(t)$, i.e.,

$$\mathcal{Z}_{m,h}(t) = \left\{ \cup \left(m, x_{m,h}^{\tau}, a_{m,h}^{\tau}, x_{n,h+1}^{\tau}\right)_{\tau\in[t-1]} \right\}.$$

Denote $\Psi$ as an ordering of $\mathcal{U}_h(t)$, and $U = |\mathcal{U}_h(t)|$ as before. The server first sets $Q_{m,H+1}^t$ to be $\mathbf{0}_d$, and for $h = H, ..., 1$, iteratively solves $H$ regressions, similar to the previous case in a decentralized manner:

$$\widehat{Q}_{m,h}^t \leftarrow \arg\min_{\mathbf{w}} \left\{ \sum_{(n,x,a,z)\in\mathcal{U}_h(t)} \left[ r_h(x,a) + V_{m,h+1}^t(z) - \mathbf{w}^\top \phi(x,a) \right]^2 + \lambda\|\mathbf{w}\|_2^2 \right\}. \quad (9.8)$$

We handle the computation of regressions separately for $h = H$ and otherwise.

**<u>Case</u>** $\underline{h = H}$. When $h = H$, the solution can be given as,

$$\widehat{\mathbf{w}}_H^t = \left(\mathbf{\Lambda}_H^t\right)^{-1} \mathbf{u}_H^t, \text{ where}$$

$$\mathbf{\Lambda}_H^t = \sum_{\tau \in \Psi} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda \mathbf{I}_d, \text{ and } \mathbf{u}_H^t = \sum_{\tau \in \Psi} r_H(x_\tau, a_\tau)\phi(x_\tau, a_\tau).$$

This can be further decomposed as follows.

$$\mathbf{\Lambda}_H^t = \sum_{m \in \mathcal{M}} \left(\mathbf{\Lambda}_{m,H}^t\right) + \lambda \mathbf{I}_d \text{ and } \mathbf{u}_H^t = \sum_{m \in \mathcal{M}} \mathbf{u}_{m,H}^t, \text{ where}$$

$$\mathbf{\Lambda}_{m,H}^t = \sum_{(x,a,z) \in \mathcal{Z}_{m,H}(t)} \phi(x,a)\phi(x,a)^\top, \mathbf{u}_{m,H}^t = \sum_{(x,a,z) \in \mathcal{Z}_{m,H}(t)} r_H(x,a)\phi(x,a).$$

Therefore, if each agent simply communicates their corresponding $\mathbf{\Lambda}_{m,H}^t$ and $\mathbf{u}_{m,H}^t$ then the server can compute $\mathbf{\Lambda}_H^t$ and $\mathbf{u}_H^t$ and relay it back to each agent with only $\mathcal{O}(d^2 + d)$ bits of total communication per agent. This will allow each agent to compute the final $Q-$values following Equation 9.2 and then compute the value functions following Equation 9.3.

<u>**Case** $h < H$</u>. When synchronization is taking place, the agents begin with $h = H$ following the above protocol. Next, we synchronize for $h = H - 1$ and progressively go to $h = 1$. Now, for $h = H - 1$, the agents compute the value functions $V_H^t(z)$ for each $(x, a, z) \in \mathcal{Z}_{m,H}(t)$ using Equation 9.3. Observe that in this case, the global parameter for $\mathbf{u}_{H-1}^t$ is given by,

$$\mathbf{u}_{H-1}^t = \sum_{m \in \mathcal{M}} (\mathbf{u}_{m,H-1}^t + \mathbf{v}_{m,H-1}^t), \text{ where } \mathbf{v}_{m,H-1}^t = \sum_{(x,a,z) \in \mathcal{Z}_{m,H-1}(t)} V_H^t(z)\phi(x,a).$$

Note that the second term is absent when $h = H$ since the value function is by convention assumed to be 0 at $h = H + 1$. However, $\mathbf{v}_{m,H-1}^t$ can easily be computed in a decentralized manner by each agent without any communication, and $\mathbf{u}_{m,H-1}^t$ and $\mathbf{\Lambda}_{m,H-1}^t$ can be computed as done for $h = H$. Once these quantities are computed, the server can once again aggregate them and transmit $\mathbf{\Lambda}_{H-1}^t$ and $\mathbf{u}_{H-1}^t$ back to the agents with a total communication complexity of $\mathcal{O}(d^2 + 2d)$ bits per agent. One can repeat this process for $h = H - 1, ..., 1$. The total communication complexity, therefore, per round, is $\mathcal{O}(2d^2 HM)$.

While the remainder of the algorithm remains the same (i.e., computing $Q-$values and executing the greedy policy with respect to the $Q-$values, the question still remains on when to synchronize. Similar to the Dense case, the synchronization criterion has a tunable parameter $S$, but the criterion itself is different. As earlier, each agent $m$ maintains two sets of parameters. The first is $\mathbf{\Lambda}_h^t$ obtained during the last synchronization round $k_t$, and the second is $\delta \mathbf{S}_{m,h}^t = \sum_{tau=k_t+1}^{t-1} \phi(x_{m,\tau}^h, a_{m,\tau}^h)\phi(x_{m,\tau}^h, a_{m,\tau}^h)^\top$, which refers to the "variance" that is not synchronized. Now, in each step of any episode $t$, after the agent executes the greedy policy with respect to the $Q-$values, i.e., $a^t_{m,h} = \arg\max_{a \in \mathcal{A}} Q^t_h(x^t_{m,h}, a)$, it updates the unsyncrhonized parameters $\delta \mathbf{S}^t_{m,h}$ with the observed transition. If any agent's new unsynchronized parameters satisfy the determinant condition with threshold $S$, i.e., if

$$\frac{\det\left(\mathbf{\Lambda}^t_h + \delta \mathbf{S}^t_{m,h}\right)}{\det\left(\mathbf{\Lambda}^t_h\right)} \geqslant S, \tag{9.9}$$

the agent signals a synchronization with the server, and messages are exchanged. The algorithm is summarized in Algorithm 19. This communication threshold, compared to the prior one, omits the $(t - k_t)$ term in the denominator. This can be understood as the agents must have more communication rounds in order to keep their policy updated, as the agents only update policies during synchronization rounds. The next result bounds communication complexity as a function of $S$ for the Rare protocol.

**Lemma 9.3** (Complexity of Rare communication)**.** *If Algorithm 19 is run with threshold $S$, then the total number of episodes with communication $n \leqslant 2H(\log_S(1 + \frac{MT}{d}) + 1)$. The total communication complexity under* Rare *is therefore $\mathcal{O}(d^2 MH(\log_S(1 + \frac{MT}{d}))$ bits.*

*Proof.* Let the number of communication rounds triggered by step $h$ be given by $n_h$. Observe that communication occurs whenever $\log\det\left(\mathbf{\Lambda}^t_h + \delta \mathbf{S}^t_{m,h}\right) - \log\det\left(\mathbf{\Lambda}^t_h\right) \geqslant S$. Since $\log\det(\mathbf{\Lambda}^T_H) \leqslant d\log(1 + (MT)/d)$ (since $\|\phi(x,a)\|_2 \leqslant 1$), and $\log\det(\mathbf{\Lambda}^t_h) \geqslant d\lambda = d$ (by regularization), we have that $n_h \log(S) \leqslant \log(1 + \frac{MT}{d})$. Since communication can be triggered by any of the $h$ steps satisfying the criterion, we have by summing over $h$ that $n \cdot \log(S) = \log(S) \cdot \sum_h n_h \leqslant H\log(1 + \frac{MT}{d})$. The next part is obtained by multiplying the total $\mathcal{O}(d^2 M)$ bits sent per communication round. □

**Remark 9.3** (Dense vs. Rare Protocols)**.** The Rare protocol exhibits a worse dependence on the communication threshold $S$ compared with the Dense protocol, primarily as the Rare protocol does not allow agents to update their policies without synchronization. A detailed comparison of the communication-regret tradeoff is summarized in Table 9.1.

We now present the regret obtained by FedLSVI under the Rare protocol.

**Theorem 9.2** (Homogenous Regret with Rare Communication)**.** *Algorithm 19 when run on $M$ agents with communication threshold $S$, $\beta^t_{m,h} = \mathcal{O}(H\sqrt{d\log(tMH)})$ and $\lambda = 1$ obtains the* *following cumulative regret after T episodes, with probability at least* $1-\alpha$*,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left(d^{3/2}H^2\sqrt{SMT\log\left(\frac{1}{\alpha}\right)}\right).$$

This is proved in Section 9.9. We now present a lower bound for the *homogeneous* federated MDP.

## 9.4 Lower Bounds for Federated MDPs

In this section we discuss lower bounds for the problem before moving to the *heterogeneous* environments. We can trivially bound the performance of non-communicating agents using a single-agent bound for tabular MDPs.

**Lemma 9.4.** *There exists a federated linear MDP instance such that any set of M agents suffer* $\Omega(\sqrt{dH^3MT})$ *regret.*

*Proof.* We consider a tabular environment $\mathcal{T}$ with state and action spaces $\mathcal{S}, \mathcal{A}$ such that $\phi(x,a) = \boldsymbol{e}_{(x,a)}$, i.e., the standard basis in $d-$dimensions. Therefore, by the single-agent analysis in Domingues et al. (2021), we have that the regret $\mathfrak{R}_{\mathcal{T}}$ suffered by any agent in $\mathcal{T}$ over $MT$ episodes is $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|H^3T}) = \Omega(\sqrt{dH^3MT})$. We will now demonstrate that any federated reinforcement learning policy must incur at least the expected regret incurred by a single agent running for $MT$ episodes. Note that while the high-level approach to the problem appears conceptually similar to the lower bound for federated bandits (Theorem 6.5), the approach involves a different set of arguments.

We consider the set of all possible single-agent episodic policies for $T$ episodes over the declared environment as $\mathbf{\Pi}$. Now, observe that the set of all possible multi-agent policies over $M$ agents can be given as $\widetilde{\mathbf{\Pi}} = (\mathbf{\Pi})^M$ (we can form an arbitrary multi-agent policy only by the product of individual single-agent policies). Therefore, any policy $\boldsymbol{\pi} \in \widetilde{\mathbf{\Pi}}$ can be written as $\boldsymbol{\pi} = \pi_1 \times \cdots \times \pi_M$, where $\pi_m \in \mathbf{\Pi}$. Observe that by the above decomposition, the group regret for any $\boldsymbol{\pi}$ follows.

$$\mathfrak{R}_{\mathcal{T}}(T;\boldsymbol{\pi}) \geqslant \sum_{m=1}^{M}\mathfrak{R}_m(T;\pi_m).$$

Where $\mathfrak{R}_m$ denotes the regret incurred by the $m^{th}$ agent in $\mathcal{T}$. We set all possible federated policies considered in the networked setting as $\widetilde{\mathbf{\Pi}}_f \subseteq \widetilde{\mathbf{\Pi}}$. Consider now the set of all possible single-agent episodic policies for $MT$ episodes $\mathbf{\Pi}_M$. We can see that $\mathbf{\Pi} \subset \mathbf{\Pi}_M$.

We now consider $M$ modified single-agent environments $\widetilde{\mathcal{T}}_1, \dots, \widetilde{\mathcal{T}}_M$ as follows. Recall that $\mathcal{T}$ is constructed by the environment arbitrarily sampling an initial state $x^t_{m,1}$ for each of the $m$ agents and $t \leq T$ episodes. Since the transitions in each MDP are independent of the others, we can consider the $\mathcal{T}$ to be composed of $M$ independent MDPs running in parallel. We select $\widetilde{\mathcal{T}}_m$ to be the environment constructed by repeating the initial state $x^t_{m,1}$ for $M$ episodes for each $t \in [T]$ with the state space $\mathcal{S}$, action space $\mathcal{A}$ and transition probabilites $\{\mathbb{P}_h\}_{h=1}^H$, same as $\mathcal{T}$ for any individual agent.

Now, for any policy $\pi \in \mathbf{\Pi}$, consider the compounded policy $\tilde{\pi} \in \mathbf{\Pi}_M$ such that $\tilde{\pi}^t_h(x) = \pi_h^{\lceil \frac{t}{M} \rceil}(x)$ for each $h \in H, x \in \mathcal{S}$, i.e., the policy is created by repeating the policy at any episode $t$ for $M$ episodes. Then, we have that for any agent $m$

$$\mathfrak{R}_m(T; \pi_m) = \frac{1}{M}\mathfrak{R}_{\widetilde{\mathcal{T}}_m}(MT; \tilde{\pi}_m).$$

The above holds by the design of the environment $\widetilde{\mathcal{T}}_m$ and the policy $\tilde{\pi}_m$. We can now bound the regret for any multi-agent policy $\boldsymbol{\pi} \in \widetilde{\mathbf{\Pi}}_f$ as follows.

$$\begin{aligned}
\mathfrak{R}_{\mathcal{T}}(T; \boldsymbol{\pi}) &\geqslant \inf_{\boldsymbol{\pi} \in \widetilde{\mathbf{\Pi}}_f} \mathfrak{R}_{\mathcal{T}}(T; \boldsymbol{\pi}) \geqslant \inf_{\boldsymbol{\pi} \in \widetilde{\mathbf{\Pi}}} \mathfrak{R}_{\mathcal{T}}(T; \boldsymbol{\pi}) \\
&\geqslant \sum_{m=1}^{M} \inf_{\pi_m \in \mathbf{\Pi}} \mathfrak{R}_m(T; \pi_m) = \frac{1}{M} \sum_{m=1}^{M} \inf_{\tilde{\pi}_m \in \mathbf{\Pi}_M} \mathfrak{R}_{\widetilde{\mathcal{T}}_m}(T; \tilde{\pi}_m) \\
&\geqslant \frac{1}{M} \sum_{m=1}^{M} \sqrt{dH^3MT} = \Omega\left(\sqrt{dH^3MT}\right).
\end{aligned}$$

□

## 9.5 Heterogeneous Federated MDPs

Now that we have established lower bounds for the homogeneous federated bandit problem, we move on to two approaches for handling heterogeneity in federated reinforcement learning. For simplicity, we will only present the analysis under the Dense communication protocol, as the techniques used will apply to the Rare protocol as well. Note that even for heterogeneous settings, our algorithms require assumptions on the nature of heterogeneity in order to benefit from cooperative estimation.

### 9.5.1 Robustness to "Small" Heterogeneity

The first heterogeneous setting we consider is when the deviations between MDPs are much smaller than the horizon $T$, which allows Algorithm 18 to be no-regret as long as an upper bound on the heterogeneity is known.

**Assumption 9.1** ("Small" deviations)**.** *For any $\xi = o(T^{-\alpha}) < 1, \alpha > 0$, a federated MDP setting demonstrates "small deviations" if for any $m, m' \in \mathcal{M}$, the corresponding linear MDPs defined in Definition 9.1 obey the following for all $(x, a) \in \mathcal{S} \times \mathcal{A}$:*

$$\mathbb{D}_{\mathsf{TV}}(\mathbb{P}_{m,h}(\cdot|x,a), \mathbb{P}_{m',h}(\cdot|x,a)) \leqslant \xi, \text{ and } |(r_{m,h} - r_{m',h})(x,a)| \leqslant \xi.$$

Under this assumption, when an upper bound on $\xi$ is known, we do not have to modify Algorithm 18 as the confidence intervals employed by FedLSVI are robust to small deviations. We formalize this with the following regret bound.

**Theorem 9.3.** *Algorithm 18 when run on $M$ agents with parameter $S$ in the small deviation setting (Assumption 9.1), with $\beta^t_{m,h} = \mathcal{O}(H\sqrt{d\log(tMH)} + \xi\sqrt{dMT})$ and $\lambda = 1$ obtains the following cumulative regret after $T$ episodes, with probability at least $1 - \alpha$,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left(dH^2\left(dM\sqrt{S} + \sqrt{dMT}\right)\left(\sqrt{\log\left(\frac{1}{\alpha}\right)} + 2\xi\sqrt{dMT}\right)\right).$$

The proof is presented in Section 9.10.

**Remark 9.4** (Comparison with Misspecification)**.** While this demonstrates that FedLSVI is robust to small deviations in the different MDPs, the analysis can be extended to the case when the MDPs are "approximately" linear, as done in Jin et al. (2020) (Theorem 3.2). A key distinction in the above result and the standard bound in the misspecification setting is that in the general misspecified linear MDP there are two aspects to the anlaysis - the first being the (adversarial) error introduced from the linear approximation, and the second being the error introduced by executing a policy following the misspecified linear approximation. In our case, the second term does not exist as the policy is valid within each agents' own MDP, but the misspecification error still remains.

### 9.5.2 Parametric Approach for Large Heterogeneity

For the large heterogeneity case, in order to transfer knowledge from a different agents' MDP, we assume that each agent $m \in \mathcal{M}$ possesses an additional contextual description $\boldsymbol{\kappa}(m) \in \mathbb{R}^k$ for some $k > 0$ that describes the heterogeneity linearly.

**Definition 9.2** (Heterogenous Linear MDP)**.** *A heterogeneous federated MDP*$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, R)$ *is a set of linear MDPs with two feature maps* $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ *and* $\boldsymbol{\kappa} : \mathcal{M} \to \mathbb{R}^k$*, if for any* $h \in [H]$*, there exist* $d$ *unknown (signed) measures* $\boldsymbol{\mu}_h = (\mu_h^1, ..., \mu_h^d)$ *over* $\mathcal{S}$*, an unknown vector* $\boldsymbol{\theta}_h \in \mathbb{R}^d$*,* $k$ *unknown (signed) measures* $\boldsymbol{\nu}_h = (\nu_h^1, ..., \nu_h^k)$ *over* $\mathcal{S}$ *and an unknown vector* $\boldsymbol{\alpha}_h \in \mathbb{R}^k$ *such that for any* $(x, a) \in \mathcal{S} \times \mathcal{A}, m \in \mathcal{M}$ *and target state* $z \in \mathcal{Z}$

$$\mathbb{P}_{m,h}(z|x,a) = \begin{bmatrix} \boldsymbol{\phi}(x,a) \\ \boldsymbol{\kappa}(m) \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\mu}_h(z) \\ \boldsymbol{\nu}_h(z) \end{bmatrix}, r_{m,h}(x,a) = \begin{bmatrix} \boldsymbol{\phi}(x,a) \\ \boldsymbol{\kappa}(m) \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\theta}_h \\ \boldsymbol{\alpha}_h \end{bmatrix}.$$

*We denote the combined features via the shorthand:*

$$\widetilde{\boldsymbol{\phi}}(m,x,a) = \left[\boldsymbol{\phi}(x,a)^\top, \boldsymbol{\kappa}(m)^\top\right]^\top, \widetilde{\boldsymbol{\mu}}_h(z) = \left[\boldsymbol{\mu}_h(z)^\top, \boldsymbol{\nu}_h(z)^\top\right]^\top, \widetilde{\boldsymbol{\theta}}_h = \left[\boldsymbol{\theta}_h^\top, \boldsymbol{\alpha}_h^\top\right]^\top.$$

*We assume, that* $\|\widetilde{\boldsymbol{\phi}}(m,x,a)\| \leqslant 1 \; \forall \; (m,x,a) \in \mathcal{M} \times \mathcal{S} \times \mathcal{A}$, $\max\{\|\boldsymbol{\mu}_h(\mathcal{S})\|, \|\boldsymbol{\theta}_h\|\} \leqslant \sqrt{d}$*, and* $\max\{\|\boldsymbol{\nu}_h(\mathcal{S})\|, \|\boldsymbol{\alpha}_h\|\} \leqslant \sqrt{k}$*.*

**Remark 9.5** (Parametric Modeling of Heterogeneity)**.** A similar heterogeneous model presented for multi-armed bandits has been presented in Chapter 8. Definition 9.2 encapsulates a general parametric approach to model the heterogeneity between agent MDPs by assuming additional contextual information. Such an approach has been extensively employed in the contextual bandit literature (Krause & Ong, 2011; Deshmukh et al., 2017; Dubey & Pentland, 2020c). Note, however, that we utilize a linear model to account for differences between agents. For example, for any two agents $m, n \in \mathcal{M}$, we have that,

$$|r_{m,h}(x,a) - r_{n,h}(x,a)| = \boldsymbol{\alpha}_h^\top (\boldsymbol{\kappa}(m) - \boldsymbol{\kappa}(n)) \leqslant 2\|\boldsymbol{\alpha}_h\|_2 \leqslant 2\sqrt{k}.$$

This allows us to model larger deviations between agents (by selecting $k > d$) by incorporating additional contextual information, in contrast to the earlier setting where we assume the maximum deviation to be small, i.e., $\varepsilon = o(T^{-\alpha})$ for no-regret learning. Similarly, we have for the transition functions,

$$\begin{aligned}\mathbb{D}_{\mathsf{TV}}(\mathbb{P}_{m,h}(\cdot|x,a),\mathbb{P}_{m,h}(\cdot|x,a)) &= \left|\int_{z\in\mathcal{S}} \mathbb{P}_{m,h}(z|x,a) - \mathbb{P}_{n,h}(z|x,a)dz\right| \\ &= \left|\int_{z\in\mathcal{S}} \boldsymbol{\nu}_h(z)^\top\left(\boldsymbol{\kappa}(m)-\boldsymbol{\kappa}(n)\right)dz\right| \\ &= \left|\left(\int_{z\in\mathcal{S}} \boldsymbol{\nu}_h(z)dz\right)^\top \left(\boldsymbol{\kappa}(m)-\boldsymbol{\kappa}(n)\right)\right| \\ &\leqslant 2\|\boldsymbol{\nu}_h(\mathcal{S})\|_2 \leqslant 2\sqrt{k}.\end{aligned}$$

Again, the parameteric setup eventually implies a similar bound on the transition functions, but allows for greater flexibility.

Concretely, we assume that for each MDP, the discrepancies between both the transition and reward functions can be explained as a linear function of the underlying agent-specific contexts, i.e., $\boldsymbol{\kappa}$, which is independent of the state and action pair $(x,a)$[4]. In contrast to the homogeneous setting, here, each agent predicts an *agent-specific* $Q$ and value function. Specifically, for each $t\in[T]$, each agent $m\in\mathcal{M}$ obtains a sequence of value functions $\{Q^t_{m,h}\}_{h\in[H]}$ by iteratively performing linear least-squares ridge regression from the *multi-agent* history available from the previous $t-1$ episodes, but in contrast to the homogenous case, it now learns a $Q-$function over $\mathcal{M}\times\mathcal{S}\times\mathcal{A}$ and value function over $\mathcal{M}\times\mathcal{A}$. Each agent $m$ first sets $Q^t_{m,H+1}$ to be a zero function, and for any $h\in[H]$, solves the regression problem in $\mathbb{R}^{d+k}$ to obtain $Q-$values.

$$\widehat{Q}^t_{m,h} \leftarrow \underset{\mathbf{w}\in\mathbb{R}^{d+k}}{\arg\min}\left\{\sum_{(n,x,a,z)\in\mathcal{U}^m_h(t)}\left[r_{n,h}(x,a)+V^t_{m,h+1}(n,x')-\mathbf{w}^\top\widetilde{\phi}(n,x,a)\right]^2+\lambda\|\mathbf{w}\|_2^2\right\}. \tag{9.10}$$

Here, $\lambda>0$ is a regularizer, and $n\in\mathcal{M}$ denotes the agent whose $Q$ function the agent $m$ is estimating, and $V^t_{m,h+1}(n,x)=\max_{a\in\mathcal{A}}Q(n,x,a)$, where the $Q$ values are given by, for any $n,x,a\in\mathcal{M}\times\mathcal{S}\times\mathcal{A}$,

$$Q(n,x,a) = \underbrace{\widehat{Q}^t_{m,h}(n,x,a)}_{\text{“agent-specific” regressor}} + \underbrace{\beta^t_{m,h}\cdot\left(\widetilde{\phi}(n,x,a)^\top\left(\widetilde{\boldsymbol{\Lambda}}^t_{m,h}\right)^{-1}\widetilde{\phi}(n,x,a)\right)^{1/2}}_{\text{confidence bonus}}.$$

[4] Intricate models can be assumed that exploit interdependence in a sophisticated manner (see, e.g., Chapter 8), however, we leave that for future work.

The regressed $Q-$values $\widehat{Q}^t_{m,h}$ can be computed by the least-squares solution described as follows. Let $\Psi$ be an ordering of $\mathcal{U}^m_h(t)$ (the set of all transitions available to any agent $m$), and $U = |\mathcal{U}^m_h(t)|$. The regression involves the covariates $\widetilde{\phi}(n, x, a)$, which, in contrast to the previous setting, are in fact functions of both the state-action space via $\phi(x, a)$ and the agent context via $\boldsymbol{\kappa}(n)$. The targets are, for any $\tau \in \Psi$, $y_\tau = y_{m,h}(n_\tau, x_\tau, a_\tau, z_\tau)$, where,

$$y_\tau = y_{m,h}(n_\tau, x_\tau, a_\tau, z_\tau) = r_{n_\tau,h}(x_\tau, a_\tau) + V^t_{m,h+1}(n_\tau, z_\tau).$$

The above equation essentially has two terms for any transition $(n_\tau, x_\tau, a_\tau, z_\tau)$ available to agent $m$ - the first is the reward that the $n^{th}_\tau$ agent obtains in step $h$ by selecting action $a_\tau$ in state $x_\tau$, and the second is the *agent-specific* value function, i.e., the value function that the agent $m$ predicts corresponding to the $n^{th}_\tau$ MDP for state $z$ at step $h+1$. This step essentially introduces the heterogeneous modeling aspect: any agent ($m$), at all times will estimate the value for *all other agents* based on their respective MDPs in order to leverage the federated data. As in the homogeneous case, the ridge regression solution is given by,

$$\widehat{Q}^t_{m,h}(n, x, a) = \widetilde{\phi}(n, x, a)^\top \widetilde{\mathbf{w}}^t_{m,h} \text{ s.t. } \mathbf{w}^t_{m,h} = \left(\widetilde{\boldsymbol{\Lambda}}^t_{m,h}\right)^{-1} \widetilde{\mathbf{u}}^t_{m,h}, \text{ where,} \tag{9.11}$$

$$\widetilde{\boldsymbol{\Lambda}}^t_{m,h} = \sum_{\tau\in\Psi} \widetilde{\phi}(n_\tau, x_\tau, a_\tau)\widetilde{\phi}(n_\tau, x_\tau, a_\tau)^\top + \lambda \mathbf{I}_{d+k}, \text{ and } \widetilde{\mathbf{u}}^t_{m,h} = \sum_{\tau\in\Psi} y_\tau \widetilde{\phi}(n_\tau, x_\tau, a_\tau). \tag{9.12}$$

At any episode $t$ and step $h$, each agent $m$ then follows the greedy policy with respect to $Q(m, x^t_{m,h})$. The remainder of the algorithm is identical to Algorithm 18, and is presented in Algorithm 20. To present the regret bound, we first define coefficient of heterogeneity $\Gamma$, and then present the regret bound in terms of this coefficient.

**Definition 9.3** (Coefficient of Heterogeneity). *In a heterogeneous federated MDP (Definition 9.2), let* $\mathbf{K} \in \mathbb{R}^{M\times M}$ *be a symmetric positive-semidefinite matrix such that*

$$\mathbf{K} = \begin{bmatrix} \boldsymbol{\kappa}(1)^\top\boldsymbol{\kappa}(1) & \dots & \boldsymbol{\kappa}(M)^\top\boldsymbol{\kappa}(1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\kappa}(1)^\top\boldsymbol{\kappa}(M) & \dots & \boldsymbol{\kappa}(M)^\top\boldsymbol{\kappa}(M) \end{bmatrix}.$$

*The coefficient of heterogeneity is defined as* $\Gamma = \mathit{rank}(\mathbf{K}) \leqslant \min\{M, k\}$.

**Remark 9.6** (Coefficient of Heterogeneity). The coefficient of heterogeneity $\Gamma$ encapsulates the difference in the respective MDPs for each agent. For example, to model the *homoge-*

*neous* case, one can simply set $\boldsymbol{\kappa}(1) = \cdots = \boldsymbol{\kappa}(M)$ which provides us that $\Gamma = 1$. In the worst case, since $\mathbf{K}$ is a Gram matrix of $M$ elements in $\mathbb{R}^k$, we have that $\Gamma = \text{rank}(\mathbf{K}) = \text{rank}(\sum_{m=1}^{M} \boldsymbol{\kappa}(m)\boldsymbol{\kappa}(m)^\top) \leqslant k$. This design, therefore, is useful only when the differences between the respective MDPs are in fact explainable by $k \ll M$ unique features. To make this remark more precise, consider the lower bound for the single-agent linear MDP.

We have that for any algorithm, there exists an MDP such that the algorithm incurs $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|H^3T})$ regret. If we therefore consider modeling this by a linear MDP in $d+k$ dimensions we have the bound $\Omega(\sqrt{(d+k)H^3MT})$ by Lemma 9.4, making the parametric model vacuous when $k \geqslant M$ since we are better off by modeling each MDP separately in $d-$dimensions in this case.

**Theorem 9.4.** *Algorithm 20 when run on $M$ agents with parameter $S$ in the heterogeneous setting (Definition 9.2), with $\beta_t = \mathcal{O}(H\sqrt{(d+k)\log(tMH)})$ and $\lambda = 1$ obtains the following cumulative regret after $T$ episodes, with probability at least $1-\alpha$,*

$$\mathfrak{R}(T) = \widetilde{\mathcal{O}}\left((d+k)H^2\left(M(d+\Gamma)\sqrt{S} + \sqrt{(d+\Gamma)MT}\right)\sqrt{\log\left(\frac{1}{\alpha}\right)}\right).$$

This is proved in Section 9.11

**Remark 9.7** (Optimality Discussion)**.** Heterogeneous FedLSVI regret is bounded by the similarity in the agents' MDPs. In the case when the agents' have identical MDPs, $\Gamma = 1$, which implies that the heterogenous variant has a worse regret by a factor of $(1+\frac{k}{d})\sqrt{1+\frac{1}{d}}$, which arises from the fact that we use a model that lies in $\mathbb{R}^{d+k}$. Indeed, we can see that by Lemma 9.4 we can construct an MDP setting such that any federated algorithm incurs $\Omega\left(\sqrt{*d+k)H^3MT}\right)$ regret. Nevertheless, this suboptimality is indeed an artifact of our regret analysis, particularly introduced by the covering number of linear functions in $\mathbb{R}^{d+k}$, and future work can address modifications to ensure tightness. Alternatively, in the worst case, $\Gamma = k$, which matches the linear federated MDP in $d+k$ dimensions, which ensures that no suboptimality has been introduced by the heterogeneous analysis. Under this model however, one can only observe improvements when $k = o(\sqrt{M})$ suffices for modeling the heterogeneity within the federated MDP.

## 9.6 Related Work and Discussion

Our work builds on the body of recent work in (single-agent) reinforcement learning with function approximation. Classical work in this line of research, e.g., Bradtke & Barto (1996); Melo & Ribeiro (2007) provide algorithms, however, with no polynomial-time sample efficiency guarantees. In the presence of a simulator Yang & Wang (2020) provide a sample-efficient algorithm under linear function approximation. For the linear MDP assumption studied in this paper, our algorithms build on the seminal work of Jin et al. (2020), that present an efficient (i.e., no-regret) algorithm. This research was further extended to kernel and neural function approximation in the recent work of Yang et al. (2020b); Wang et al. (2020a). Other approaches in this approximation setting are either computationally intractable (Krishnamurthy et al., 2016; Dann et al., 2018; Dong et al., 2020) or require strong assumptions on the transition model (Wen & Van Roy, 2017).

Parallel reinforcement learning is a very relevant practical setting for reinforcement learning in large-scale and distributed systems, studied first in (Kretchmar, 2002). A variant of the SARSA was presented for parallel RL in Grounds & Kudenko (2005), that provides an efficient algorithm but with no regret guarantees. Modern deep-learning based approaches (with no regret guarantees) have been studied recently as well (e.g., Clemente et al. (2017); Espeholt et al. (2018); Horgan et al. (2018); Nair et al. (2015)). In the federated setting, which corresponds to a decentralized variant of parallel reinforcement learning, there has been recent interested from application domains as well (Yu et al., 2020b; Zhuo et al., 2019).

## 9.7 Proof of Lemma 9.3

Denote an epoch as the number of episodes between two rounds of communication. Let $q = \sqrt{\frac{ST}{d\log(1+T/d)}} + 1$. There can be at most $\lceil T/q \rceil$ rounds of communication such that they occur after an epoch of length $q$. On the other hand, if there is any round of communication succeeding an epoch (that begins, say at time $t$) of length $< n'$, then for that epoch, $\log \frac{\det\left(\mathbf{S}^t_{m,h}+\delta\mathbf{S}^t_{m,h}+\lambda\mathbf{I}_d\right)}{\det\left(\mathbf{S}^t_{m,h}+\lambda\mathbf{I}_d\right)} \geqslant \frac{S}{q}$. Let the communication occur at a set of episodes $t'_1, ..., t'_n$. Now, since:

$$\sum_{i=1}^{n-1} \log \frac{\det\left(\mathbf{S}^{t_{i+1}}_{m,h}\right)}{\det\left(\mathbf{S}^{t_i}_{m,h}\right)} = \log \frac{\det\left(\mathbf{\Lambda}^T_h\right)}{\det\left(\mathbf{\Lambda}^0_h\right)} \leqslant d\log(1+T/(d)),$$

We have that the total number of communication rounds succeeding epochs of length less than $n'$ is upper bounded by $\log \frac{\det\left(\mathbf{\Lambda}^T_h\right)}{\det\left(\mathbf{\Lambda}^0_h\right)} \leqslant d\log(1+T/(d)) \cdot (q/S)$. Combining both the results together, we have the total rounds of communication as:

$$\begin{aligned} n &\leqslant \lceil T/q \rceil + \lceil d\log(1+T/(d)) \cdot (q/S) \rceil & (9.13) \\ &\leqslant T/q + d\log(1+T/(d)) \cdot (q/S) + 2 & (9.14) \end{aligned}$$

Replacing $q$ from earlier and summing over $h \in [H]$ (as communication may be triggered by any of the steps satisfying the condition) gives us the final result.

## 9.8 Proof of Theorem 9.1

We first present our primary concentration result to bound the error in the least-squares value iteration.

**Lemma 9.5.** *Under the setting of Therorem 9.1, let $c_\beta$ be the constant defining $\beta$, and $\mathbf{S}^t_{m,h}$ and $\mathbf{\Lambda}^k_t$ be defined as follows.*

$$\mathbf{S}^t_{m,h} = \sum_{n=1}^{M}\sum_{\tau=1}^{k_t} \phi(x^\tau_{n,h}, a^\tau_{n,h}) \left[V^t_{m,h+1}(x^\tau_{n,h+1}) - (\mathbb{P}_h V^t_{m,h+1})(x^\tau_{n,h}, a^\tau_{n,h})\right] \\ + \sum_{\tau=k_t+1}^{t-1} \phi(x^\tau_{m,h}, a^\tau_{m,h}) \left[V^t_{m,h+1}(x^\tau_{m,h+1}) - (\mathbb{P}_h V^t_{m,h+1})(x^\tau_{m,h}, a^\tau_{m,h})\right],$$

$$\mathbf{\Lambda}^t_{m,h} = \sum_{n=1}^{M}\sum_{\tau=1}^{k_t} \phi(x^\tau_{n,h}, a^\tau_{n,h})\phi(x^\tau_{n,h}, a^\tau_{n,h})^\top + \sum_{\tau=k_t+1}^{t-1} \phi(x^\tau_{m,h}, a^\tau_{m,h})\phi(x^\tau_{m,h}, a^\tau_{m,h})^\top + \lambda \mathbf{I}_d.$$

*Where $V \in \mathcal{V}$ and $\mathcal{N}_\epsilon$ denotes the $\epsilon-$covering of the value function space $\mathcal{V}$. Then, there exists an absolute constant $c_\beta$ independent of $M, T, H, d$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,*

$$\left\|\mathbf{S}^t_{m,h}\right\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \leqslant c_\beta \cdot dH\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

*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**. Note that for any agent $m$, the function $V^t_{m,h+1}$ depends on the historical data from all $M$ agents from the first $k_t$ episodes, and the personal historical data for the first $(t-1)$ episodes, and depends on

$$\mathcal{U}^m_h(t) = \left(\cup_{n\in[M],\tau\in[k_t]}\{(x^\tau_{n,h}, a^\tau_{n,h}, x^\tau_{n,h+1})\}\right) \bigcup \left(\cup_{\tau\in[k_t+1,t-1}\{(x^\tau_{m,h}, a^\tau_{m,h}, x^\tau_{m,h+1})\}\right).$$

To bound the term we will construct an appropriate filtration to use a self-normalized concentration defined on elements of $\mathcal{U}^m_h(t)$. We highlight that in the multi-agent case with stochastic communication, it is not straightforward to provide a uniform martingale concentration that holds for all $t \in [T]$ simultaneously (as is done in the single-agent case), as the stochasticity in the environment dictates when communication will take place, and subsequently the quantity considered within self-normalization will depend on this communication itself. To circumvent this issue, we will first fix $k_t \leqslant t$ and obtain a filtration for a fixed $k_t$. Then, we will take a union bound over all $k_t \in [t]$ to provide the final self-normalized bound. We first fix $k_t$ and define the following mappings where $i \in [M(t-1)], l \in [t-1]$, and $n \in [M]$.

$$\mu(i) = \left\lceil \frac{i}{M} \right\rceil, \nu(i) = i(\text{mod } M), \text{ and}, \eta(l,n) = l \cdot (M+1) + n - 1.$$

Now, for a fixed $k_t$, consider the stochastic processes $\{\tilde{x}_\tau\}_{\tau=1}^\infty$ and $\{\tilde{\phi}_\tau\}_{\tau=1}^\infty$, where,

$$\tilde{\phi}_i = \phi(x_{\mu(i),h+1}^{\nu(i)}) \otimes \mathbb{1}_d \{(\mu(i) = m) \vee (\nu(i) \leqslant k_t)\}$$

Here $\otimes$ denotes the Hadamard product, and $\mathbb{1}_d$ is the indicator function in $\mathbb{R}^d$. Consider now the filtration $\{\mathcal{F}_\tau\}_{\tau=0}^\infty$, where $\mathcal{F}_0$ is empty, and $\mathcal{F}_\tau = \sigma\left(\left\{\bigcup(\tilde{x}_i, \tilde{\phi}_i)\right\}_{i \leqslant \tau}\right)$, where $\sigma(\cdot)$ denotes the corresponding $\sigma-$algebra formed by the set.

At any instant $t$ for any agent $m$, the function $V_{m,h+1}^t$ and features $\phi(x_{m,h}^t, a_{m,h}^t)$ depend only on historical data from all other agents $[M] \setminus \{m\}$ up to the last episode of synchronization $k_t \leqslant t-1$ and depend on the personal data up to episode $t-1$. Hence, both are $V_{m,h+1}^t$ and $\phi(x_{m,h}^t, a_{m,h}^t)$ are measurable with respect to

$$\sigma\left(\left\{\bigcup_{l=1}^{k_t}\bigcup_{n=1}^{M}(\tilde{x}_{\eta(l,n)}, \tilde{\phi}_{\eta(l,n)})\right\} \bigcup \left\{\bigcup_{l=k_t+1}^{t-1}(\tilde{x}_{\eta(l,m)}, \tilde{\phi}_{\eta(l,m)})\right\}\right).$$

This is a subset of $\mathcal{F}_{\eta(t,m)}$. Therefore $V_{m,h+1}^t$ is $\mathcal{F}_{\eta(t,m)}-$measurable for fixed $k_t$. Now, consider $\mathcal{U}_h^m(\tau)$, the set of features available to agent $m$ at episode $\tau \leqslant t$. We therefore have that, for any value function $V$,

$$\begin{aligned}
&\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_{m,h}(\tau)\left\{V(\tilde{x}_\tau) - \mathbb{E}[V(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\} \\
&= \sum_{\tau=1}^{M(t-1)} \left[\phi(x_{\mu(i),h+1}^{\nu(i)}) \otimes \mathbb{1}_d\{(\mu(i) = m) \vee (\nu(i) \leqslant k_t)\}\right]\left\{V(\tilde{x}_\tau) - \mathbb{E}\left[V(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}\right]\right\} \\
&= \sum_{(x_\tau, a_\tau, x'_\tau) \in \mathcal{U}_h^m(t)} \phi(x_\tau, a_\tau)\left\{V(x'_\tau) - \mathbb{E}[V(x'_\tau)|\mathcal{F}_{\tau-1}]\right\}.
\end{aligned}$$

Now, when $V = V_{m,h+1}^t$, we have from the above,

$$\begin{aligned}
&\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_\tau \left\{V_{h,m+1}^t(\tilde{x}_\tau) - \mathbb{E}[V_{h,m+1}^t(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\} \\
&= \sum_{(n_\tau, x_\tau, a_\tau, x'_\tau) \in \mathcal{U}_h^m(t)} \phi(x_\tau, a_\tau)\left\{V_{m,h+1}^t(x'_\tau) - \mathbb{E}[V_{m,h+1}^t(x'_\tau)|\mathcal{F}_{\tau-1}]\right\} = \mathbf{S}_{m,h}^t.
\end{aligned}$$

Furthermore, consider $\widetilde{\mathbf{\Lambda}}_{m,h}^t = \lambda\mathbf{I}_d + \sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_\tau \widetilde{\phi}_\tau^\top$ and let $\mathbb{1}(t)$ denote the shorthand for

$\mathbb{1}_d \{\mu(i) = m \vee \nu(i) \leqslant k_t\}$. For the second term, we have,

$$\begin{aligned}
\widetilde{\boldsymbol{\Lambda}}^t_{m,h} &= \lambda \mathbf{I}_d + \sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_\tau \widetilde{\phi}_\tau^\top \\
=&\lambda \mathbf{I}_d + \sum_{\tau=1}^{M(t-1)} \left[\phi(x^{\nu(i)}_{\mu(i),h+1}) \otimes \mathbb{1}(t)\right] \left[\phi(x^{\nu(i)}_{\mu(i),h+1}) \otimes \mathbb{1}(t)\right]^\top \\
=&\lambda \mathbf{I}_d + \sum_{(n_\tau, x_\tau, a_\tau, x'_\tau) \in \mathcal{U}^m_h(t)} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top = \boldsymbol{\Lambda}^t_{m,h}.
\end{aligned}$$

Next, we bound $\left\|\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_{m,h}(\tau) \left\{V^t_{h,m+1}(\tilde{x}_\tau) - \mathbb{E}[V^t_{h,m+1}(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\}\right\|_{(\widetilde{\boldsymbol{\Lambda}}^t_{m,h})^{-1}}$ over all $k_t \in [t]$. We proceed following a self-normalized martingale bound and a covering argument, as done in Yang et al. (2020b).

Applying Lemma 9.20 to $\left\|\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_{m,h}(\tau) \left\{V^t_{h,m+1}(\tilde{x}_\tau) - \mathbb{E}[V^t_{h,m+1}(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\}\right\|_{(\widetilde{\boldsymbol{\Lambda}}^t_{m,h})^{-1}}$ under the filtration $\{\mathcal{F}_\tau\}_{\tau=0}^\infty$ described earlier, we have that with probability at least $1-\delta'$,

$$\begin{aligned}
\left\|\mathbf{S}^t_{m,h}\right\|^2_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} &= \left\|\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_\tau \left\{V^t_{h,m+1}(\tilde{x}_\tau) - \mathbb{E}[V^t_{h,m+1}(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\}\right\|^2_{(\widetilde{\boldsymbol{\Lambda}}^t_{m,h})^{-1}} \\
&\leqslant \sup_{V \in \mathcal{V}} \left\|\sum_{\tau=1}^{M(t-1)} \widetilde{\phi}_\tau \left\{V(\tilde{x}_\tau) - \mathbb{E}[V(\tilde{x}_\tau)|\mathcal{F}_{\tau-1}]\right\}\right\|^2_{(\widetilde{\boldsymbol{\Lambda}}^t_{m,h})^{-1}} \\
&\leqslant 4H^2 \cdot \log \frac{\det\left(\widetilde{\boldsymbol{\Lambda}}^t_{m,h}\right)}{\det\left(\lambda \mathbf{I}_d\right)} + 8H^2 \log(|\mathcal{N}_\epsilon|/\delta') + 8M^2 t^2 \epsilon^2/\lambda.
\end{aligned}$$

Where $\mathcal{N}_\epsilon$ is an $\epsilon-$covering of $\mathcal{V}$. Therefore, we have that, with probability at least $1-\delta'$, for any fixed $k_t \leqslant t$,

$$\left\|\mathbf{S}^t_{m,h}\right\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \leqslant 2H \sqrt{\log\left(\frac{\det\left(\widetilde{\boldsymbol{\Lambda}}^t_{m,h}\right)}{\det\left(\lambda \mathbf{I}_d\right)}\right) + 2\log\left(\frac{|\mathcal{N}_\epsilon|}{\delta}\right) + \frac{2M^2t^2\epsilon^2}{H^2\lambda}}.$$

Taking a union bound over all $k_t \in [t]$, $m \in \mathcal{M}, t \in [T], h \in [H]$ and replacing $\delta' = \delta/(MHT^2)$ gives us that with probability at least $1-\delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$\left\|\mathbf{S}^t_{m,h}\right\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \leqslant 2H \sqrt{\log\left(\frac{\det\left(\widetilde{\boldsymbol{\Lambda}}^t_{m,h}\right)}{\det\left(\lambda \mathbf{I}_d\right)}\right) + \log\left(MHT^2 \cdot \frac{|\mathcal{N}_\epsilon|}{\delta'}\right) + \frac{2M^2t^2\epsilon^2}{H^2\lambda}}.$$

$$\leqslant 2H\sqrt{\log\left(\frac{\det\left(\mathbf{\Lambda}_h^t\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)+2\log\left(\frac{MHT^2|\mathcal{N}_\epsilon|}{\delta'}\right)+\frac{2t^2\epsilon^2}{H^2\lambda}}$$

$$\le 2H\sqrt{d\log\frac{t+\lambda}{\lambda}+4\log(MHT)+2\log\left(\frac{|\mathcal{N}_\epsilon|}{\delta'}\right)+\frac{2t^2\epsilon^2}{H^2\lambda}}$$

(AM $\geqslant$ GM; determinant-trace inequality)

**Step 2**. Here $\mathcal{N}_\epsilon$ is an $\epsilon-$covering of the function class $\mathcal{V}_{\text{UCB}}$ for any $h \in [H], m \in [M]$ or $t \in [T]$ under the distance function $\text{dist}(V, V') = \sup_{x\in\mathcal{S}} |V(x) - V'(x)|$. To bound this quantity by the appropriate covering number, we first observe that for any $V \in \mathcal{V}_{\text{UCB}}$, we have that the policy weights are bounded as $2H\sqrt{dMT/\lambda}$ (Lemma 9.22). Therefore, by Lemma 9.23 we have for any constant $B$ such that $\beta^t_{m,h} \leqslant B$,

$$\log|\mathcal{N}_\varepsilon| \leqslant d\log\left(1+8H\sqrt{\frac{dMT}{\lambda\epsilon^2}}\right)+d^2\log\left(1+\frac{8d^{1/2}B^2}{\lambda\epsilon^2}\right).$$

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

$$\log|\mathcal{N}_\varepsilon| \leqslant d\log\left(1+8\sqrt{\frac{MT^3}{d}}\right)+d^2\log\left(1+8c_\beta d^{1/2}T^2\log(TMH)\right).$$

Therefore, for some absolute constant $C'$ independent of $M, T, H, d$ and $c_\beta$, we have,

$$\log|\mathcal{N}_\varepsilon| \leqslant C'd^2\log\left(CdT\log(TMH)\right).$$

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

$$\begin{aligned}&\left\|\mathbf{S}^t_{m,h}\right\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}\\ &\leqslant 2H\sqrt{(d+2)\log\frac{t+\lambda}{\lambda}+2\log\left(\frac{1}{\alpha}\right)+C'd^2\log\left(c_\beta dT\log(TMH)\right)+2+4\log(TMH)}.\end{aligned}$$

This implies that there exists an absolute constant $C$ independent of $M, T, H, d$ and $c_\beta$, such that, with probability at least $1-\delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$\left\|\mathbf{S}_{m,h}^t\right\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \leqslant C \cdot dH\sqrt{2\log\left(\frac{(c_\beta+2)dMTH}{\delta'}\right)}.$$

Now, following the procedure in Lemma B.4 of Jin et al. (2020), we can select $c_\beta$ such that we have,

$$\left\|\mathbf{S}_{m,h}^t\right\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \leqslant c_\beta \cdot dH\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

This finishes the proof. □

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 $\pi$.

**Lemma 9.6.** *There exists an absolute constant $c_\beta$ such that for $\beta_{m,h}^t = c_\beta \cdot dH\sqrt{\log(2dMHT/\delta')}$ for any policy $\pi$, such that for each $x \in \mathcal{S}, a \in \mathcal{A}$ we have for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously, with probability at least $1-\delta'/2$,*

$$\begin{aligned}&\left|\langle\phi(x,a), \mathbf{w}_{m,h}^t - \mathbf{w}_h^\pi\rangle\right| \\ &\qquad\leqslant \mathbb{P}_h(V_{m,h+1}^t - V_{m,h+1}^\pi)(x,a) + c_\beta \cdot dH \cdot \|\phi(z)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.\end{aligned}$$

*Proof.* By the Bellman equation and the assumption of the linear MDP (Definition 9.1), we have that for any policy $\boldsymbol{\pi}$, there exist weights $\mathbf{w}_h^\pi$ such that, for all $z \in \mathcal{Z}$,

$$\langle\phi(z), \mathbf{w}_h^\pi\rangle = r_h(z) + \mathbb{P}_h V_{h+1}^\pi(z).$$

The set of all observations available to any agent at instant $t$ is given by $\mathcal{U}_h^m(t)$ for step $h$, with the cardinality of this set being $U_m^h(t)$. For convenience, let us assume an ordering $\tau = 1, ..., U_m^h(t)$ over this set and use the shorthand $U_m = U_m^h(t)$. Therefore, we have, for any $m \in \mathcal{M}$,

$$\begin{aligned}\mathbf{w}_{m,h}^t - \mathbf{w}_h^\pi &= (\mathbf{\Lambda}_{m,h}^t)^{-1}\sum_{\tau=1}^{U_m}\left[\phi_\tau[(r_h + V_{m,h+1}^t)(x_\tau)]\right] - \mathbf{w}_h^\pi \\ &= (\mathbf{\Lambda}_{m,h}^t)^{-1}\left\{-\lambda\mathbf{w}_h^\pi + \sum_{\tau=1}^{U_m}\left[\phi_\tau[V_{m,h+1}^t(x'_\tau) - \mathbb{P}_h V_{m,h+1}^\pi(x_\tau, a_\tau)]\right]\right\}.\end{aligned}$$

$$\implies \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_h = \underbrace{-\lambda(\mathbf{\Lambda}^t_{m,h})^{-1}\mathbf{w}^\pi_h}_{\mathbf{v}_1} + \underbrace{(\mathbf{\Lambda}^t_{m,h})^{-1}\left\{\sum_{\tau=1}^{U_m}\left[\boldsymbol{\phi}_\tau[V^t_{m,h+1}(x'_\tau) - \mathbb{P}_h V^t_{m,h+1}(z_\tau)]\right]\right\}}_{\mathbf{v}_2}$$
$$+ \underbrace{(\mathbf{\Lambda}^t_{m,h})^{-1}\left\{\sum_{\tau=1}^{U_m}\left[\boldsymbol{\phi}_\tau[\mathbb{P}_h V^t_{m,h+1} - \mathbb{P}_h V^\pi_{m,h+1})(z_\tau)]\right]\right\}}_{\mathbf{v}_3}.$$

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

$$|\langle \boldsymbol{\phi}(z), \mathbf{v}_1\rangle| \leqslant \lambda \left|\langle \boldsymbol{\phi}(z), \mathbf{\Lambda}^t_{m,h})^{-1}\mathbf{w}^\pi_h\rangle\right| \leqslant \lambda \cdot \|\mathbf{w}^\pi_h\| \|\boldsymbol{\phi}(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \leqslant 2H\lambda\sqrt{d}\|\boldsymbol{\phi}(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}$$

Here the last inequality follows from Lemma 9.21. For the next term, we have by Lemma 9.5 that there exists an absolute constant $C$ independent of $M, T, H, d$ and $c_\beta$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$|\langle \boldsymbol{\phi}(z), \mathbf{v}_2\rangle| \leqslant \|\boldsymbol{\phi}(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \cdot c_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

We can bound the last term as follows. We make the substitution $\Delta_V = (V^t_{m,h+1} - V^\pi_{m,h+1})$ for brevity.

$$\begin{aligned}
|\langle \boldsymbol{\phi}(z), \mathbf{v}_3\rangle| &= \left\langle \boldsymbol{\phi}(z), (\mathbf{\Lambda}^t_{m,h})^{-1}\left\{\sum_{\tau=1}^{U_m}\left[\boldsymbol{\phi}_\tau[\mathbb{P}_h V^t_{m,h+1} - \mathbb{P}_h V^\pi_{m,h+1})(z_\tau)]\right]\right\}\right\rangle \\
&= \left\langle \boldsymbol{\phi}(z), (\mathbf{\Lambda}^t_{m,h})^{-1}\sum_{\tau=1}^{U_m}\left[\boldsymbol{\phi}_\tau\boldsymbol{\phi}_\tau^\top \int \Delta(V)(x')d\boldsymbol{\mu}_h(x')\right]\right\rangle \\
&= \left\langle \boldsymbol{\phi}(z), \int \Delta(V)(x')d\boldsymbol{\mu}_h(x')\right\rangle - \lambda\left\langle \boldsymbol{\phi}(z), (\mathbf{\Lambda}^t_{m,h})^{-1}\int \Delta(V)(x')d\boldsymbol{\mu}_h(x')\right\rangle \\
&= \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) - \lambda\left\langle \boldsymbol{\phi}(z), (\mathbf{\Lambda}^t_{m,h})^{-1}\int \Delta(V)(x')d\boldsymbol{\mu}_h(x')\right\rangle \\
&= \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) + 2H\sqrt{d\lambda}\|\boldsymbol{\phi}(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}.
\end{aligned}$$

Putting it all together, we have that since $\langle \boldsymbol{\phi}(z), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_h\rangle = \langle \boldsymbol{\phi}(z), \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3\rangle$, there exists an absolute constant $C$ independent of $M, T, H, d$ and $c_\beta$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$\left|\langle \boldsymbol{\phi}(x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_h\rangle\right| \leqslant \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a)$$

$$+ \|\phi(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \left( C \cdot dH \cdot \sqrt{2\log\left((c_\beta + 2)\frac{dMTH}{\delta'}\right)} + 2H\sqrt{d\lambda} + 2H\lambda\sqrt{d} \right)$$

Since $\lambda \leqslant 1$ and since $C$ is independent of $c_\beta$, we can select $c_\beta$ such that we have the following for any $(x, a) \in \mathcal{S} \times \mathcal{A}$ with probability $1 - \delta'/2$ simultaneously for all $h \in [H], m \in \mathcal{M}, t \in [T]$,

$$\begin{aligned}
&\left|\langle \phi(x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_h \rangle\right| \\
&\qquad \leqslant \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) + c_\beta \cdot dH \cdot \|\phi(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.
\end{aligned}$$

□

**Lemma 9.7** (UCB in the Homogenous Setting)**.** *With probability at least $1 - \delta'/2$, we have that for all $(x, a, h, t, m) \in \mathcal{S} \times \mathcal{A} \times [H] \times [T] \times \mathcal{M}$, $Q^t_{m,h}(x, a) \geqslant Q^\star_{m,h}(x, a)$.*

*Proof.* The proof is done by induction, identical to the proof in Lemma B.5 of Jin et al. (2020), and we urge the reader to refer to the aforementioned source. □

**Lemma 9.8** (Recursive Relation in Homogenous Settings)**.** *Let $\delta^t_{m,h} = V^t_{m,h}(x^t_{m,h}) - V^{\pi_t}_{m,h}(x^t_{m,h})$, and $\xi^t_{m,h+1} = \mathbb{E}\left[\delta^t_{m,h} | x^t_{m,h}, a^t_{m,h}\right] - \delta^t_{m,h}$. Then, with probability at least $1 - \alpha$, for all $(t, m, h) \in [T] \times \mathcal{M} \times [H]$ simultaneously,*

$$\delta^t_{m,h} \leqslant \delta^t_{m,h+1} + \xi^t_{m,h+1} + 2\left\|\phi(x^t_{m,h}, a^t_{m,h})\right\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \cdot c_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}.$$

*Proof.* By Lemma 9.6, we have that for any $(x, a, h, m, t) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{M} \times [T]$ with probability at least $1 - \alpha/2$,

$$\begin{aligned}
Q^t_{m,h}(x,a) - Q^{\pi_t}_{m,h}(x,a) &\leqslant \mathbb{P}_h(V^t_{m,h+1} - V^{\pi_t}_{m,h})(x,a) \\
&\qquad + 2\|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \cdot c_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}.
\end{aligned}$$

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

**Lemma 9.9.** *For $\xi^t_{m,h}$ as defined in Lemma 9.8 and any $\delta \in (0, 1)$, we have with probability at least*

$1 - \delta/2$,

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \xi^t_{m,h} \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)}.$$

*Proof.* We demonstrate that the overall sums can be written as bounded martingale difference sequences with respect to an appropriately chosen filtration. For any $(t, m, h) \in [T] \times [M] \times [H]$, we define the $\sigma$-algebra $\mathcal{F}_{t,m,h}$ as,

$$\mathcal{F}_{t,m,h} = \sigma\left(\left\{\left(x^\tau_{l,i}, a^\tau_{l,i}\right)\right\}_{(\tau,l,i)\in[t-1]\times[M]\times[H]} \cup \left\{\left(x^t_{l,i}, a^t_{l,i}\right)\right\}_{(i,l)\in[h]\times[m-1]} \cup \left\{\left(x^t_{m,i}, a^t_{m,i}\right)\right\}_{i\in[h]}\right)$$

Where we denote the $\sigma-$algebra generated by a finite set by $\sigma(\cdot)$. For any $t \in [T], m \in [M], h \in [H]$, we can define the timestamp index $\tau(t, m, h)$ as $\tau(t, m, h) = (t-1)\cdot HM + h(m-1) + (h-1)$. We see that this ordering ensures that the $\sigma-$algebras from earlier form a filtration. We can see that for any agent $m \in [M]$, $Q^t_{m,h}$ and $V^t_{m,h}$ are both obtained based on the trajectories of the first $(t-1)$ episodes, and are both measurable with respect to $\mathcal{F}_{t,1,1}$ (which is a subset of $\mathcal{F}_{t,m,h}$ for all $h \in [H]$ and $m \in [M]$). Moreover, note that $a^t_{m,h} \sim \pi_{m,t}(\cdot|x^t_{m,h})$ and $x^t_{m,h+1} \sim \mathbb{P}_{m,h}(\cdot|x^t_{m,h}, a^t_{m,h})$. Therefore,

$$\mathbb{E}_{\mathbb{P}_{m,h}}[\xi^t_{m,h}|\mathcal{F}_{t,m,h}] = 0.$$

where we set $\mathcal{F}_{1,0,0}$ with the empty set. We define the martingale $\{U_{t,m,h}\}_{(t,h,m)\in[T]\times[M]\times[H]}$ indexed by $\tau(t, m, h)$ defined earlier, as follows. For any $(t, m, h) \in [T] \times [M] \times [H]$, we define

$$U_{t,m,h} = \left\{\sum_{(a,b,c)} \xi^a_{b,c} : \tau(a, b, c) \leqslant \tau(t, m, h)\right\},$$

Additionally, we have that

$$U_{T,M,H} = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \xi^t_{m,h}.$$

Now, we have that for each $m \in \mathcal{M}$, $V^t_{m,h}, Q^t_{m,h}, V^{\pi_{m,t}}_{m,h}, Q^{\pi_{m,t}}_{m,h}$ take values in $[0, H]$. Therefore, wh have that $\xi^t_{m,h} \leqslant 2H$ for all $(t, m, h) \in [T] \times [M] \times [H]$. This allows us to apply the Azuma-Hoeffding inequality (Azuma, 1967) to $U_{T,M,H}$. We therefore obtain that for all

$\tau > 0$,

$$\mathbb{P}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H}\xi_{m,h}^{t} > \tau\right) \leqslant \exp\left(\frac{-\tau^2}{2H^3MT}\right).$$

Setting the RHS as $\alpha/2$, we obtain that with probability at least $1-\alpha/2$,

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H}\xi_{m,h}^{t} \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)}.$$

□

**Lemma 9.10** (Variance control via DENSE communication). *Let Algorithm 18 be run for any $T > 0$ and $M \geqslant 1$, with $S$ as the communication control factor. Then, the following holds for the cumulative variance.*

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\left\|\phi(z_{m,h}^{t})\right\|_{(\mathbf{\Lambda}_{m,h}^{t})^{-1}} \leqslant 2\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)\left(\frac{M}{\log 2}\right)\sqrt{S} + 2\sqrt{2MT\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)}.$$

*Proof.* Consider the following mappings $\nu_M, \nu_T : [MT] \to [M]\times[T]$.

$$\nu_M(\tau) = \tau(\text{mod } M), \text{and } \nu_T = \left\lceil\frac{\tau}{M}\right\rceil.$$

Now, consider $\bar{\mathbf{\Lambda}}_h^{\tau} = \lambda\mathbf{I}_d + \sum_{u=1}^{\tau}\phi\left(z_{\nu_M(u),h}^{\nu_T(u)}\right)\phi\left(z_{\nu_M(u),h}^{\nu_T(u)}\right)^{\top}$ for $\tau > 0$ and $\bar{\mathbf{\Lambda}}_h^{0} = \lambda\mathbf{I}_d$. Furthermore, assume that global synchronizations occur at round $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)$ where there are a total of $n-1$ rounds of synchronization and $\sigma_i \in [T] \forall\ i \in [N-1]$ and $\sigma_n = T$, i.e., the final round. Let $R_h = \left\lceil\log\left(\frac{\det\left(\bar{\mathbf{\Lambda}}_h^T\right)}{\det(\lambda\mathbf{I}_d)}\right)\right\rceil$. It follows that there exist at most $R_h$ periods between synchronization (i.e., intervals $\sigma_{k-1}$ to $\sigma_k$ for $k \in [N]$) in which the following does not hold true:

$$1 \leqslant \frac{\det(\bar{\mathbf{\Lambda}}_h^{\sigma_k})}{\det(\bar{\mathbf{\Lambda}}_h^{k-1})} \leqslant 2. \tag{9.15}$$

Let us denote the event when Equation (9.15) does holds for an interval $\sigma_{k-1}$ to $\sigma_k$ as $E$. Now, for any $t \in [\sigma_{k-1}, \sigma_k]$, we have, for any $m \in [M]$,

$$\left\|\phi(z_{m,h}^{t})\right\|_{(\mathbf{\Lambda}_{m,h}^{t})^{-1}} \leqslant \left\|\phi(z_{m,h}^{t})\right\|_{(\bar{\mathbf{\Lambda}}_h^{t})^{-1}}\sqrt{\frac{\det\left(\bar{\mathbf{\Lambda}}_h^{t}\right)}{\det\left(\mathbf{\Lambda}_{m,h}^{t}\right)}} \leqslant \left\|\phi(z_{m,h}^{t})\right\|_{(\bar{\mathbf{\Lambda}}_h^{t})^{-1}}\sqrt{\frac{\det\left(\bar{\mathbf{\Lambda}}_h^{\sigma_k}\right)}{\det\left(\bar{\mathbf{\Lambda}}_h^{\sigma_{k-1}}\right)}}$$

$$\leqslant 2\left\|\phi(z^t_{m,h})\right\|_{(\bar{\boldsymbol{\Lambda}}^t_h)^{-1}}.$$

Here, the first inequality follows from the fact that $\boldsymbol{\Lambda}^t_{m,h} \preccurlyeq \bar{\boldsymbol{\Lambda}}^t_h$, the second inequality follows from the fact that $\boldsymbol{\Lambda}^t_{m,h} \preccurlyeq \bar{\boldsymbol{\Lambda}}^{\sigma_k}_h \implies \det(\boldsymbol{\Lambda}^t_{m,h}) \leqslant \det(\bar{\boldsymbol{\Lambda}}^{\sigma_k}_h)$, and $\boldsymbol{\Lambda}^t_{m,h} \succcurlyeq \bar{\boldsymbol{\Lambda}}^{\sigma_{k-1}}_h \implies \det(\boldsymbol{\Lambda}^t_{m,h}) \geqslant \det(\bar{\boldsymbol{\Lambda}}^{\sigma_{k-1}}_h)$; and the final inequality follows from the fact that event $E$ holds. Now, we can consider the partial sums only in the intervals for which event $E$ holds. For any $t \in [T]$, consider $\sigma(t) = \max_{i\in[N]}\{\sigma_i | \sigma_i \leqslant t\}$ denote the last round of synchronization prior to episode $t$. Then,

$$\sum_{t:E\text{ is true}}^{T}\sum_{m=1}^{M}\left\|\phi(z^t_{m,h})\right\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \leqslant \sqrt{MT\sum_{m=1}^{M}\sum_{t:E\text{ is true}}^{T}\left\|\phi(z^t_{m,h})\right\|^2_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}}}$$
$$\leqslant 2\sqrt{MT\sum_{m=1}^{M}\sum_{t:E\text{ is true}}^{T}\left\|\phi(z^t_{m,h})\right\|^2_{(\bar{\boldsymbol{\Lambda}}^t_h)^{-1}}} \leqslant 2\sqrt{MT\sum_{m=1}^{M}\sum_{t=1}^{T}\left\|\phi(z^t_{m,h})\right\|^2_{(\bar{\boldsymbol{\Lambda}}^t_h)^{-1}}}$$
$$= 2\sqrt{MT\sum_{m=1}^{M}\sum_{\tau=1}^{T}\left\|\phi(z^{\nu_T(\tau)}_{\nu_M(\tau),h})\right\|^2_{(\bar{\boldsymbol{\Lambda}}^t_h)^{-1}}} = 2\sqrt{MT\log\left(\frac{\det\left(\boldsymbol{\Lambda}^T_h\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)}.$$

Here, the first inequality is Cauchy-Schwarz, the second inequality follows from the fact that event $E$ holds, and the final equality follows from Lemma 10.12. Now, we sum up the cumulative sum for episodes when $E$ does not hold. Consider an interval $\sigma_{k-1}$ to $\sigma_k$ for $k \in [N]$ of length $\Delta_k = \sigma_k - \sigma_{k-1}$ in which $E$ does not hold. We have that,

$$\sum_{m=1}^{M}\sum_{t=\sigma_{k-1}}^{\sigma_k}\left\|\phi(z^t_{m,h})\right\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \leqslant \sum_{m=1}^{M}\sqrt{\Delta_{k,h}\sum_{t=\sigma_{k-1}}^{\sigma_k}\left\|\phi(z^t_{m,h})\right\|^2_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}}}$$
$$\leqslant \sum_{m=1}^{M}\sqrt{\Delta_{k,h}\cdot\log_\omega\left(\frac{\det(\boldsymbol{\Lambda}^{\sigma_k}_{m,h})}{\det(\boldsymbol{\Lambda}^{\sigma_{k-1}}_{m,h})}\right)} \leqslant \sum_{m=1}^{M}\sqrt{\Delta_{k,h}\cdot\log_\omega\left(\frac{\det(\bar{\boldsymbol{\Lambda}}^{\sigma_k}_h)}{\det(\bar{\boldsymbol{\Lambda}}^{\sigma_{k-1}}_h)}\right)} \leqslant M\sqrt{S}.$$

The last inequality follows from the synchronization criterion. Now, note that there are at most $R_h$ periods in which event $E$ does not hold, and hence the total sum in this period can be bound as,

$$\sum_{(t:E\text{ is not true})}^{T}\sum_{m=1}^{M}\left\|\phi(z^t_{m,h})\right\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \leqslant R_h M\sqrt{S} \leqslant \left(\log\left(\frac{\det\left(\boldsymbol{\Lambda}^T_h\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)+1\right)M\sqrt{S}.$$

Therefore, we can bound the total variance as,

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\|\phi(z_{m,h}^t)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \leqslant \left(\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)+1\right)M\sqrt{S}+2\sqrt{MT\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)}$$
$$\leqslant 2\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)\left(\frac{M}{\log 2}\right)\sqrt{S}+2\sqrt{2MT\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)}.$$

□

We are now ready to prove Theorem 9.1. We have by the definition of group regret:

$$\mathfrak{R}(T)=\sum_{m=1}^{M}\sum_{t=1}^{T}V_{m,1}^{\star}(x_{m,1}^t)-V_{m,1}^{\pi_t}(x_{m,1}^t)\leqslant\sum_{m=1}^{M}\sum_{t=1}^{T}\delta_{m,1}^t$$
$$\leqslant\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{h=1}^{H}\xi_{m,h}^t+2c_\beta\cdot dH\cdot\sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H}\|\phi(x,a)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}}\right).$$

Where the last inequality holds with probability at least $1-\alpha/2$, via Lemmas 9.7 and 9.8. Next, we can bound the first term via Lemma 9.9. We have with probability at least $1-\alpha$, for some absolute constant $c_\beta$,

$$\mathfrak{R}(T)\leqslant\sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)}+2c_\beta\cdot dH\cdot\sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H}\|\phi(x,a)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}}\right).$$

Finally, to bound the summation, we use Lemma 9.10. We have that,

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H}\|\phi(x,a)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}}$$
$$\leqslant 2\sum_{h=1}^{H}\left(\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)\left(\frac{M}{\log 2}\right)\sqrt{S}+2\sqrt{2MT\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda\mathbf{I}_d\right)}\right)}\right)$$
$$\leqslant 2H\log(dMT)M\sqrt{S}+2\sqrt{2dMT\log(MT)}.$$

Where the last inequality is an application of the determinant-trace inequality and using the fact that $\|\phi(\cdot)\|_2\leqslant 1$. Replacing this result, we have that with probability at least $1-\alpha$,

$$\mathfrak{R}(T)\leqslant\sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)}$$

$$+ 2c_\beta dH^2 \sqrt{2\log\left(\frac{dMTH}{\alpha}\right)} \left(2\log(dMT)M\sqrt{S} + 2\sqrt{2dMT\log(MT)}\right).$$

Simplifying the above finishes the proof.

## 9.9 Proof of Theorem 9.2

By the same arguments as Theorem 9.1, we have with probability at least $1-\alpha$, for some absolute constant $c_\beta$,

$$\mathfrak{R}(T) \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)} + 2c_\beta \cdot dH \cdot \sqrt{2\log\left(\frac{dMTH}{\alpha}\right)} \left(\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}\right).$$

We bound the term $\left(\sum_{t=1}^{T}\sum_{m=1}^{M} \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}\right)$ for each $h \in [H]$ via Lemma 12 of Abbasi-Yadkori et al. (2011). Assume that the penultimate synchronization round was given by $k_T$ and let

$$\widehat{\mathbf{\Lambda}}_h = \sum_{m=1}^{M}\sum_{t=1}^{k_T} \phi(x^t_{m,h}, a^t_{m,h})\phi(x^t_{m,h}, a^t_{m,h})^\top,$$

$$\overline{\mathbf{\Lambda}}_h = \sum_{m=1}^{M}\sum_{t=1}^{T} \phi(x^t_{m,h}, a^t_{m,h})\phi(x^t_{m,h}, a^t_{m,h})^\top$$

for each $h \in [H]$. We have for any $h$,

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{m=1}^{M} \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} &\leqslant \sqrt{\frac{\det(\mathbf{\Lambda}^t_{m,h})}{\det(\widehat{\mathbf{\Lambda}}_h)}} \cdot \left(\sum_{t=1}^{T}\sum_{m=1}^{M} \|\phi(x,a)\|_{(\widehat{\mathbf{\Lambda}}_h)^{-1}}\right) \\
&\leqslant \sqrt{\frac{\det(\overline{\mathbf{\Lambda}}_h)}{\det(\widehat{\mathbf{\Lambda}}_h)}} \cdot \left(\sum_{t=1}^{T}\sum_{m=1}^{M} \|\phi(x,a)\|_{(\widehat{\mathbf{\Lambda}}_h)^{-1}}\right) \\
&\leq \sqrt{S} \cdot \left(\sum_{t=1}^{T}\sum_{m=1}^{M} \|\phi(x,a)\|_{(\widehat{\mathbf{\Lambda}}_h)^{-1}}\right) \\
&= \mathcal{O}\left(\sqrt{SMT\log\left(\frac{MT}{d}\right)}\right).
\end{aligned}$$

Where the last inequality is an application of the determinant-trace inequality and using the fact that $\|\phi(\cdot)\|_2 \leqslant 1$. Replacing this result, we have that with probability at least $1-\alpha$, there exists an independent constant $C$ such that

$$\mathfrak{R}(T) \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)} + 2C \cdot c_\beta dH^2\sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}\left(\sqrt{SMT\log\left(\frac{MT}{d}\right)}\right).$$

Simplifying the above finishes the proof.

## 9.10 Proof of Theorem 9.3

The proof for this algorithm largely follows the structure of Theorem 9.1 with several key modifications to handle the differences between MDPs as a case of model misspecification. First, we must bound the difference in the projected $Q-$values for any pair of MDPs under the small heterogeneity condition.

**Lemma 9.11.** *Under the small heterogeneity condition (Assumption 9.1), for any policy* $\boldsymbol{\pi}$ *over* $\mathcal{S} \times \mathcal{A}$*, let the corresponding weights at step* $h$ *for two MDPs* $m, m' \in \mathcal{M}$ *be given by* $\mathbf{w}^\pi_{m,h}, \mathbf{w}^\pi_{m',h}$ *respectively, i.e.,* $Q^\pi_{m,h}(x,a) = \langle \phi(x,a), \mathbf{w}^\pi_{m,h}\rangle$ *and* $Q^\pi_{m',h}(x,a) = \langle \phi(x,a), \mathbf{w}^\pi_{m',h}\rangle$*. Then, we have for any* $x, a \in \mathcal{S} \times \mathcal{A}$*,*

$$\left|\langle \phi(x,a), \mathbf{w}^\pi_{m,h} - \mathbf{w}^\pi_{m',h}\rangle\right| \leqslant 2H\xi.$$

*Proof.* The proof follows from the fact that for any $h \in [H]$,

$$\begin{aligned}
&\left|\langle \phi(x,a), \mathbf{w}^\pi_{m,h} - \mathbf{w}^\pi_{m',h}\rangle\right| \\
&= \left|Q^\pi_{m,h}(x,a) - Q^\pi_{m',h}(x,a)\right| \\
&\leqslant |r_{m,h}(x,a) - r_{m',h}(x,a)| + \left|\mathbb{P}_{m,h}V^\pi_{m,h+1}(x,a) - \mathbb{P}_{m',h}V^\pi_{m',h+1}(x,a)\right| \\
&\leqslant |r_{m,h}(x,a) - r_{m',h}(x,a)| + \sup_{x' \in \mathcal{S}} |V^\pi_{m,h+1}(x') - V^\pi_{m',h+1}(x')| \cdot \|(\mathbb{P}_{m,h} - \mathbb{P}_{m',h})(x,a)\|_{\text{TV}} \\
&\leqslant |r_{m,h}(x,a) - r_{m',h}(x,a)| + H \cdot \|(\mathbb{P}_{m,h} - \mathbb{P}_{m',h})(x,a)\|_{\text{TV}} \\
&\leqslant 2H\xi.
\end{aligned}$$

Here the last inequality follows from Assumption 9.1. □

Now, we reproduce a general result bounding bias introduced by the potentially adversarial noise due to misspecification.

**Lemma 9.12.** *Let $\{\varepsilon_\tau\}_{\tau=1}^t$ be a sequence such that $|\varepsilon_\tau| \leqslant B$. We have, for any $(h, t, m) \in [H] \times [T] \times \mathcal{M}$, and $\boldsymbol{\phi} \in \mathbb{R}^d$,*

$$|\boldsymbol{\phi}^\top (\boldsymbol{\Lambda}_{m,h}^t)^{-1} \sum_{\tau=1}^{U_h^m(t)} \boldsymbol{\phi}_\tau \varepsilon_\tau| \leqslant B\sqrt{dMt}\|\boldsymbol{\phi}\|_{(\boldsymbol{\Lambda}_{m,h}^t)^{-1}}.$$

*Proof.* Recall that at any instant the collective set of observations possessed by an agent is given by $\mathcal{U}_h^m(t)$ with size $U_h^m(t) \leqslant Mt$. We have that,

$$\begin{aligned}
|\boldsymbol{\phi}^\top (\boldsymbol{\Lambda}_{m,h}^t)^{-1} \sum_{\tau=1}^{U_h^m(t)} \boldsymbol{\phi}_\tau \varepsilon_\tau| &\leqslant B \cdot |\boldsymbol{\phi}^\top (\boldsymbol{\Lambda}_{m,h}^t)^{-1} \sum_{\tau=1}^{U_h^m(t)} \boldsymbol{\phi}_\tau| \\
&\leqslant B \cdot \sqrt{\left[\sum_{\tau=1}^{U_h^m(t)} \boldsymbol{\phi}^\top (\boldsymbol{\Lambda}_{m,h}^t)^{-1} \boldsymbol{\phi}\right] \cdot \left[\sum_{\tau=1}^{U_h^m(t)} \boldsymbol{\phi}_\tau^\top (\boldsymbol{\Lambda}_{m,h}^t)^{-1} \boldsymbol{\phi}_\tau\right]} \\
&\leqslant B\sqrt{dMt}\|\boldsymbol{\phi}\|_{(\boldsymbol{\Lambda}_{m,h}^t)^{-1}}.
\end{aligned}$$

□

Now we present the primary concentration result for the small heterogeneity setting.

**Lemma 9.13.** *There exists an absolute constant $c_\beta$ such that for $\beta_{m,h}^t = c_\beta \cdot dH(\sqrt{\log(2dMHT/\delta')} + \xi\sqrt{dMT})$ for any policy $\pi$, there exists a constant $c_\beta$ such that for each $x \in \mathcal{S}, a \in \mathcal{A}$ we have for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously, with probability at least $1 - \delta'/2$,*

$$\begin{aligned}
&\left|\langle \boldsymbol{\phi}(x, a), \mathbf{w}_{m,h}^t - \mathbf{w}_{m,h}^\pi \rangle\right| \leqslant \\
&\mathbb{P}_{m,h}(V_{m,h+1}^t - V_{m,h+1}^\pi)(x, a) + c_\beta \cdot dH \cdot \|\boldsymbol{\phi}(z)\|_{(\boldsymbol{\Lambda}_{m,h}^t)^{-1}} \left(\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 2\xi\sqrt{dMT}\right).
\end{aligned}$$

*Proof.* By the Bellman equation and the assumption of the linear MDP (Definition 9.1), we have that for any policy $\boldsymbol{\pi}$, there exist weights $\mathbf{w}_{m,h}^\pi$ such that, for all $z \in \mathcal{Z}$,

$$\langle \boldsymbol{\phi}(z), \mathbf{w}_{m,h}^\pi \rangle = r_{m,h}(z) + \mathbb{P}_{m,h} V_{h+1}^\pi(z).$$

Recall that the set of all observations available to any agent at instant $t$ is given by $\mathcal{U}_h^m(t)$ for step $h$, with the cardinality of this set being $U_m^h(t)$. For convenience, let us assume an ordering $\tau = 1, ..., U_m^h(t)$ over this set and use the shorthand $U_m = U_m^h(t)$. Therefore, we have, for any $m \in \mathcal{M}$,

$$\begin{aligned}
&\mathbf{w}_{m,h}^t - \mathbf{w}_{m,h}^\pi \\
&= (\mathbf{\Lambda}_{m,h}^t)^{-1} \sum_{\tau=1}^{U_m} \left[\phi_\tau[(r_h + V_{m,h+1}^t)(x_\tau)]\right] - \mathbf{w}_{m,h}^\pi \\
&= (\mathbf{\Lambda}_{m,h}^t)^{-1} \left\{ -\lambda \mathbf{w}_h^\pi + \sum_{\tau=1}^{U_m} \left[\phi_\tau[V_{m,h+1}^t(x_\tau') - \mathbb{P}_{m_\tau,h} V_{m,h+1}^\pi(x_\tau, a_\tau)]\right] \right\}.
\end{aligned}$$

$$\begin{aligned}
\implies &\mathbf{w}_{m,h}^t - \mathbf{w}_{m,h}^\pi \\
&= \underbrace{-\lambda(\mathbf{\Lambda}_{m,h}^t)^{-1}\mathbf{w}_{m,h}^\pi}_{\mathbf{v}_1} + \underbrace{(\mathbf{\Lambda}_{m,h}^t)^{-1} \left\{ \sum_{\tau=1}^{U_m} \left[\phi_\tau[V_{m,h+1}^t(x_\tau') - \mathbb{P}_{m,h} V_{m,h+1}^t(z_\tau)]\right] \right\}}_{\mathbf{v}_2} \\
&\quad + \underbrace{(\mathbf{\Lambda}_{m,h}^t)^{-1} \left\{ \sum_{\tau=1}^{U_m} \left[\phi_\tau[\mathbb{P}_{m,h} V_{m,h+1}^t - \mathbb{P}_{m,h} V_{m,h+1}^\pi)(z_\tau)]\right] \right\}}_{\mathbf{v}_3} \\
&\quad + \underbrace{(\mathbf{\Lambda}_{m,h}^t)^{-1} \left\{ \sum_{\tau=1}^{U_m} \left[\phi_\tau[\mathbb{P}_{m,h} V_{m,h+1}^t - \mathbb{P}_{m_\tau,h} V_{m,h+1}^t)(z_\tau)]\right] \right\}}_{\mathbf{v}_4} \\
&\quad + \underbrace{(\mathbf{\Lambda}_{m,h}^t)^{-1} \left\{ \sum_{\tau=1}^{U_m} \left[\phi_\tau[\mathbb{P}_{m,h} V_{m,h+1}^\pi - \mathbb{P}_{m_\tau,h} V_{m,h+1}^\pi)(z_\tau)]\right] \right\}}_{\mathbf{v}_5}.
\end{aligned}$$

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

$$|\langle \phi(z), \mathbf{v}_1 \rangle| \leqslant \lambda \left| \langle \phi(z), \mathbf{\Lambda}_{m,h}^t)^{-1} \mathbf{w}_h^\pi \rangle \right| \leqslant \lambda \cdot \|\mathbf{w}_h^\pi\| \|\phi(z)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \leqslant 2H\lambda\sqrt{d} \|\phi(z)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}}$$

Here the last inequality follows from Lemma 9.21. For the second term, we have by Lemma 9.5 that there exists an absolute constant $C$ independent of $M, T, H, d$ and $c_\beta$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$|\langle \phi(z), \mathbf{v}_2 \rangle| \leqslant \|\phi(z)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \cdot c_\beta \cdot dH \cdot \sqrt{2 \log \left( \frac{dMTH}{\delta'} \right)}.$$

Let us make the substitution $\Delta_V = (V^t_{m,h+1} - V^\pi_{m,h+1})$. For the third term, note that,

$$\begin{aligned}
&|\langle \boldsymbol{\phi}(z), \mathbf{v}_3 \rangle| \\
&= \left\langle \boldsymbol{\phi}(z), (\boldsymbol{\Lambda}^t_{m,h})^{-1} \left\{ \sum_{\tau=1}^{U_m} \left[ \boldsymbol{\phi}_\tau [\mathbb{P}_h V^t_{m,h+1} - \mathbb{P}_h V^\pi_{m,h+1})(z_\tau)] \right] \right\} \right\rangle \\
&= \left\langle \boldsymbol{\phi}(z), (\boldsymbol{\Lambda}^t_{m,h})^{-1} \sum_{\tau=1}^{U_m} \left[ \boldsymbol{\phi}_\tau \boldsymbol{\phi}_\tau^\top \int \Delta_V(x') d\boldsymbol{\mu}_h(x') \right] \right\rangle \\
&= \left\langle \boldsymbol{\phi}(z), \int \Delta_V(x') d\boldsymbol{\mu}_h(x') \right\rangle - \lambda \left\langle \boldsymbol{\phi}(z), (\boldsymbol{\Lambda}^t_{m,h})^{-1} \int \Delta_V(x') d\boldsymbol{\mu}_h(x') \right\rangle \\
&= \mathbb{P}_h (V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) - \lambda \left\langle \boldsymbol{\phi}(z), (\boldsymbol{\Lambda}^t_{m,h})^{-1} \int \Delta_V(x') d\boldsymbol{\mu}_h(x') \right\rangle \\
&= \mathbb{P}_h (V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) + 2H\sqrt{d\lambda} \|\boldsymbol{\phi}(z)\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}}
\end{aligned}$$

For the remaining terms, we have that both $[\mathbb{P}_{m,h} V^\pi_{m,h+1} - \mathbb{P}_{m_\tau,h} V^\pi_{m,h+1})(z_\tau)]$ and $[\mathbb{P}_{m,h} V^t_{m,h+1} - \mathbb{P}_{m_\tau,h} V^t_{m,h+1})(z_\tau)]$ are bounded by $H\xi$ (from Assumption 9.1 and the fact that the value functions are always smaller than $H$). This gives us, by Lemma 9.12,

$$|\langle \boldsymbol{\phi}(z), \mathbf{v}_4 + \mathbf{v}_5 \rangle| \leqslant 2H\xi\sqrt{dMt} \|\boldsymbol{\phi}(z)\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}}$$

Putting it all together, we have that since $\langle \boldsymbol{\phi}(z), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_{m,h} \rangle = \langle \boldsymbol{\phi}(z), \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4 + \mathbf{v}_5 \rangle$, there exists an absolute constant $C$ independent of $M, T, H, d$ and $c_\beta$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,

$$\begin{aligned}
&\left| \langle \boldsymbol{\phi}(x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_{m,h} \rangle \right| \leqslant \mathbb{P}_{m,h} (V^t_{m,h+1} - V^\pi_{m,h+1})(x,a) + \\
&\|\boldsymbol{\phi}(z)\|_{(\boldsymbol{\Lambda}^t_{m,h})^{-1}} \left( C \cdot dH \cdot \sqrt{2 \log \left( (c_\beta + 2) \frac{dMTH}{\delta'} \right)} + 2H\sqrt{d\lambda} + 2H\lambda\sqrt{d} + 2H\xi\sqrt{dMT} \right)
\end{aligned}$$

Since $\lambda \leqslant 1$ and since $C$ is independent of $c_\beta$, we can select $c_\beta$ such that we have the following for any $(x,a) \in \mathcal{S} \times \mathcal{A}$ with probability $1 - \delta'/2$ simultaneously for all $h \in [H], m \in \mathcal{M}, t \in [T]$,

$$\left| \langle \boldsymbol{\phi}(x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_{m,h} \rangle \right| \leqslant$$

$$\mathbb{P}_{m,h}(V^t_{m,h+1} - V^{\pi}_{m,h+1})(x,a) + c_\beta \cdot dH \cdot \|\phi(z)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \left(\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 2\xi\sqrt{dMT}\right).$$

□

We now present an analogous recursive relationship in the small heterogeneity setting.

**Lemma 9.14** (Recursive Relation in Small Heterogeneous Settings). *Let $\delta^t_{m,h} = V^t_{m,h}(x^t_{m,h}) - V^{\pi_t}_{m,h}(x^t_{m,h})$, and $\xi^t_{m,h+1} = \mathbb{E}\left[\delta^t_{m,h}|x^t_{m,h}, a^t_{m,h}\right] - \delta^t_{m,h}$. Then, with probability at least $1-\alpha$, for all $(t,m,h) \in [T] \times \mathcal{M} \times [H]$ simultaneously,*

$$\delta^t_{m,h} \leqslant \delta^t_{m,h+1} + \xi^t_{m,h+1} + c_\beta \cdot dH \cdot \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \left(\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 2\xi\sqrt{dMT}\right).$$

*Proof.* By Lemma 9.13, we have that for any $(x,a,h,m,t) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{M} \times [T]$ with probability at least $1-\alpha/2$,

$$Q^t_{m,h}(x,a) - Q^{\pi_t}_{m,h}(x,a) = \left|\langle \phi(x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^{\pi}_{m,h}\rangle\right| \leqslant$$
$$\mathbb{P}_{m,h}(V^t_{m,h+1} - V^{\pi}_{m,h+1})(x,a) + c_\beta \cdot dH \cdot \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}} \left(\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 2\xi\sqrt{dMT}\right).$$

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

We are now ready to prove Theorem 9.3. We have by the definition of group regret:

$$\mathfrak{R}(T) \tag{9.16}$$
$$= \sum_{m=1}^{M}\sum_{t=1}^{T} V^\star_{m,1}(x^t_{m,1}) - V^{\pi_t}_{m,1}(x^t_{m,1}) \leqslant \sum_{m=1}^{M}\sum_{t=1}^{T} \delta^t_{m,1} \tag{9.17}$$
$$\leqslant \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{h=1}^{H} \xi^t_{m,h} + 4c_\beta \cdot dH \cdot \left(\sqrt{\log\left(\frac{dMTH}{\alpha}\right)} + \xi\sqrt{dMT}\right)\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \|\phi(x,a)\|_{(\mathbf{\Lambda}^t_{m,h})^{-1}}\right). \tag{9.18}$$

Where the last inequality holds with probability at least $1-\alpha/2$, via Lemma 9.14 and Lemma 9.7. Next, we can bound the first term via Lemma 9.9. We have with probability at least $1-\alpha$, for some absolute constant $c_\beta$,

$$\mathfrak{R}(T) \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)}$$

$$+ 4c_\beta \cdot dH \cdot \left( \sqrt{\log\left(\frac{dMTH}{\alpha}\right)} + \xi\sqrt{dMT} \right) \left( \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \|\phi(x,a)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \right).$$

Finally, to bound the summation, we use Lemma 9.10. We have that,

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{h=1}^{H} \|\phi(x,a)\|_{(\mathbf{\Lambda}_{m,h}^t)^{-1}} \\
&\leqslant 2\sum_{h=1}^{H}\left( \log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda \mathbf{I}_d\right)}\right)\left(\frac{M}{\log 2}\right)\sqrt{S} + 2\sqrt{2MT\log\left(\frac{\det\left(\mathbf{\Lambda}_h^T\right)}{\det\left(\lambda \mathbf{I}_d\right)}\right)} \right) \\
&\leqslant 2H\log(dMT)M\sqrt{S} + 2\sqrt{2dMT\log(MT)}.
\end{aligned}$$

Where the last inequality is an application of the determinant-trace inequality and using the fact that $\|\phi(\cdot)\|_2 \leqslant 1$. Replacing this result, we have that with probability at least $1-\alpha$,

$$\begin{aligned}
&\mathfrak{R}(T) \leqslant \sqrt{2H^3MT\log\left(\frac{2}{\alpha}\right)} \\
&+ 4c_\beta \cdot dH^2 \cdot \left( \sqrt{\log\left(\frac{dMTH}{\alpha}\right)} + \xi\sqrt{dMT} \right)\left( 2\log(dMT)M\sqrt{S} + 2\sqrt{2dMT\log(MT)} \right)
\end{aligned}$$

$$\implies \mathfrak{R}(T) = \widetilde{\mathcal{O}}\left( d^{3/2}H^2\left(M\sqrt{S} + \sqrt{MT}\right)\left( \sqrt{\log\left(\frac{1}{\alpha}\right)} + 2\xi\sqrt{dMT} \right) \right).$$

## 9.11 Proof for Theorem 9.4

The proof for this section is follows the strucutre of Theorem 9.1, however since we use the modified feature, the analysis differs in several key places. First we introduce the basic result which relates the variance with the coefficient of heterogeneity.

**Lemma 9.15** (Variance Decomposition)**.** *Under the heterogeneous federated MDP assumption (Definition 9.2) and coefficient of heterogeneity defined in Definition 9.3, we have that,*

$$\max_{h\in[H]} \log\det\left(\widetilde{\mathbf{\Lambda}}_h^T\right) \leqslant (d+\lambda+\Gamma)\log(MT).$$

*Proof.* We know, from the form of $\widetilde{\mathbf{\Lambda}}_h^T$ that,

$$\log\det\left(\widetilde{\mathbf{\Lambda}}_h^T\right) = \log\det\left((\widetilde{\mathbf{\Phi}}_h^T)^\top(\widetilde{\mathbf{\Phi}}_h^T\right) + \lambda\mathbf{I}_{d+k}) = \log\det\left((\widetilde{\mathbf{\Phi}}_h^T)(\widetilde{\mathbf{\Phi}}_h^T)^\top + \lambda\mathbf{I}_{MT}\right).$$

Here, $\widetilde{\mathbf{\Phi}}_h^T \in \mathbb{R}^{MT\times(d+k)}$ is the matrix of all features $\widetilde{\phi}(x,a,m)$ for step $h$ until episode $T$. Now, observe that the matrix $(\widetilde{\mathbf{\Phi}}_h^T)(\widetilde{\mathbf{\Phi}}_h^T)^\top$ can be rewritten as the sum of two matrices $(\widetilde{\mathbf{\Phi}}_h^T)(\widetilde{\mathbf{\Phi}}_h^T)^\top = (\mathbf{\Phi}_h^T)(\mathbf{\Phi}_h^T)^\top + \widetilde{\mathbf{K}}_h^T$, where $[\widetilde{\mathbf{K}}_h^T]_{i,j} = \boldsymbol{\nu}(m_i)^\top\boldsymbol{\nu}(m_j), \widetilde{\mathbf{K}}_h^T \in \mathbb{R}^{MT\times MT}$, i.e., the corresponding dot-product contribution from the agent-specific features between any pair of transitions, and $(\mathbf{\Phi}_h^T)(\mathbf{\Phi}_h^T)^\top$ refers to the regular (agent-agnostic) features, i.e., $[(\mathbf{\Phi}_h^T)(\mathbf{\Phi}_h^T)^\top]_{i,j} = \phi_i^\top\phi_j$. Now, from Theorem IV of Madiman (2008), we have that,

$$\begin{aligned}\log\det\left((\widetilde{\mathbf{\Phi}}_h^T)(\widetilde{\mathbf{\Phi}}_h^T)^\top + \lambda\mathbf{I}_{MT}\right) &\leqslant \log\det\left((\mathbf{\Phi}_h^T)(\mathbf{\Phi}_h^T)^\top + \lambda\mathbf{I}_{MT}\right) + \log\det\left(\widetilde{\mathbf{K}}_h^T + \lambda\mathbf{I}_{MT}\right)\\ &= \log\det\left((\mathbf{\Phi}_h^T)^\top(\mathbf{\Phi}_h^T) + \lambda\mathbf{I}_d\right) + \log\det\left(\widetilde{\mathbf{K}}_h^T + \lambda\mathbf{I}_{MT}\right)\\ &\leqslant d\log(MT) + \log\det\left(\widetilde{\mathbf{K}}_h^T + \lambda\mathbf{I}_{MT}\right)\\ &\leqslant d\log(MT) + \lambda\log(MT) + \text{rank}(\widetilde{\mathbf{K}}_h^T)\cdot\log(MT)\\ &= (d+\lambda)\log(MT) + \text{rank}(\mathbf{K}_h^\kappa)\log(MT).\end{aligned}$$

The second inequality follows from $\|\phi(\cdot)\| \leqslant 1$ and then applying an AM-GM inequality followed by the determinant-trace inequality (as is common in bandit analyses). The final equality follows by the fact that since $\widetilde{\mathbf{K}}_h^T$ is $T\times T$ tiles of $\mathbf{K}_h^\kappa$ followed by permutations, which implies that $\text{rank}(\widetilde{\mathbf{K}}_h^T) = \text{rank}(\mathbf{K}_h^\kappa)$. Taking the maximum over all $h\in[H]$ and gives us the result. □

Next, we present a variant of the previous concentration result to bound the least-squares value iteration error (analog of Lemma 9.5).

**Lemma 9.16.** *Under the setting of Theorem 9.4, let $c'_\beta$ be the constant defining $\beta$, and $\widetilde{\mathbf{S}}_{m,h}^t$ and $\widetilde{\mathbf{\Lambda}}_t^k$ be defined as follows.*

$$\begin{aligned}\widetilde{\mathbf{S}}_{m,h}^t = \sum_{n=1}^{M}\sum_{\tau=1}^{k_t}\widetilde{\phi}(n,x_{n,h}^\tau,a_{n,h}^\tau)\left[V_{m,h+1}^t(n,x_{n,h+1}^\tau) - (\mathbb{P}_{m,h}V_{m,h+1}^t)(n,x_{n,h}^\tau,a_{n,h}^\tau)\right]\\ + \sum_{\tau=k_t+1}^{t-1}\widetilde{\phi}(n,x_{m,h}^\tau,a_{m,h}^\tau)\left[V_{m,h+1}^t(m,x_{m,h+1}^\tau) - (\mathbb{P}_{m,h}V_{m,h+1}^t)(m,x_{m,h}^\tau,a_{m,h}^\tau)\right],\end{aligned}$$

$$\widetilde{\mathbf{\Lambda}}^t_{m,h} = \sum_{n=1}^{M} \sum_{\tau=1}^{k_t} \widetilde{\phi}(n, x^\tau_{n,h}, a^\tau_{n,h})\widetilde{\phi}(n, x^\tau_{n,h}, a^\tau_{n,h})^\top + \sum_{\tau=k_t+1}^{t-1} \widetilde{\phi}(n, x^\tau_{m,h}, a^\tau_{m,h})\widetilde{\phi}(n, x^\tau_{m,h}, a^\tau_{m,h})^\top + \lambda \mathbf{I}_{d+k}.$$

*Where $V \in \mathcal{V}$ and $\mathcal{N}_\epsilon$ denotes the $\epsilon-$covering of the value function space $\mathcal{V}$. Then, there exists an absolute constant $C$ independent of $M, T, H, d$ and $c'_\beta$, such that, with probability at least $1 - \delta'/2$ for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously,*

$$\left\| \widetilde{\mathbf{S}}^t_{m,h} \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \leqslant C \cdot (d+k)H \sqrt{2 \log \left( \frac{(c'_\beta + 2)(d+k)MTH}{\delta'} \right)}.$$

*Proof.* The proof is identical to that of Lemma 9.5, except that we utilize the combined features of dimensionality $(d+k)$, which requires us to select an alternative constant $c'_\beta$ in the bound. □

**Lemma 9.17.** *There exists a constant $c'_\beta$ such that for $\beta^t_{m,h} = c'_\beta \cdot dH\sqrt{\log(2(d+k)MHT/\delta')}$ for any policy $\pi$, there exists a constant $c_\beta$ such that for each $x \in \mathcal{S}, a \in \mathcal{A}$ we have for all $m \in \mathcal{M}, t \in [T], h \in [H]$ simultaneously, with probability at least $1 - \delta'/2$,*

$$\begin{aligned} \left| \langle \widetilde{\phi}(n,x,a), \mathbf{w}^t_{m,h} - \mathbf{w}^\pi_{m,h} \rangle \right| &\leqslant \mathbb{P}_{m,h}(V^t_{m,h+1} - V^\pi_{m,h+1})(n,x,a) \\ &\quad + c_\beta \cdot (d+k)H \cdot \|\widetilde{\phi}(n,z)\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \cdot \sqrt{2 \log \left( \frac{(d+k)MTH}{\delta'} \right)}. \end{aligned}$$

*Proof.* The proof for this is identical to Lemma 9.6, however we modify the application of Lemma 9.5 with Lemma 9.16 instead. □

**Lemma 9.18** (UCB in the Heterogeneous Setting)**.** *With probability at least $1 - \delta'/2$, we have that for all $(x, a, h, t, m) \in \mathcal{S} \times \mathcal{A} \times [H] \times [T] \times \mathcal{M}$, $Q^t_{m,h}(x,a) \geqslant Q^\star_{m,h}(x,a)$.*

*Proof.* The proof is done by induction, identical to the proof in Lemma B.5 of Jin et al. (2020), and we urge the reader to refer to the aforementioned source. □

**Lemma 9.19.** *Let $\delta^t_{m,h} = V^t_{m,h}(x^t_{m,h}) - V^{\pi_t}_{m,h}(x^t_{m,h})$, and $\xi^t_{m,h+1} = \mathbb{E}\left[\delta^t_{m,h} | x^t_{m,h}, a^t_{m,h}\right] - \delta^t_{m,h}$. Then, with probability at least $1 - \alpha$, for all $(t, m, h) \in [T] \times \mathcal{M} \times [H]$ simultaneously,*

$$\delta^t_{m,h} \leqslant \delta^t_{m,h+1} + \xi^t_{m,h+1} + 2 \left\| \widetilde{\phi}(m, x^t_{m,h}, a^t_{m,h}) \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \cdot c'_\beta \cdot (d+k)H \cdot \sqrt{2 \log \left( \frac{(d+k)MTH}{\alpha} \right)}.$$

*Proof.* By Lemma 9.17, we have that for any $(x, a, h, m, t) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{M} \times [T]$ with probability at least $1 - \alpha/2$,

$$Q^t_{m,h}(x,a) - Q^{\pi_t}_{m,h}(x,a) \leqslant \mathbb{P}_{m,h}(V^t_{m,h+1} - V^{\pi_t}_{m,h})(x,a) \\ + 2 \left\| \widetilde{\phi}(x,a) \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \cdot c'_\beta \cdot (d+k)H \cdot \sqrt{2 \log \left( \frac{(d+k)MTH}{\alpha} \right)}.$$

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

We are now ready to prove Theorem 9.4. We have by the definition of group regret:

$$\mathfrak{R}(T) = \sum_{m=1}^{M} \sum_{t=1}^{T} V^\star_{m,1}(x^t_{m,1}) - V^{\pi_t}_{m,1}(x^t_{m,1}) \leqslant \sum_{m=1}^{M} \sum_{t=1}^{T} \delta^t_{m,1} \\ \leqslant \sum_{m,t,h}^{M,T,H} \xi^t_{m,h} + 2c'_\beta \cdot (d+k)H \cdot \sqrt{2 \log \left( \frac{(d+k)MTH}{\alpha} \right)} \left( \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{h=1}^{H} \left\| \widetilde{\phi}(m,x,a) \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \right).$$

Where the last inequality holds with probability at least $1 - \alpha/2$, via Lemma 9.19 and Lemma 9.18. Next, we can bound the first term via Lemma 9.9. We have with probability at least $1 - \alpha$, for some absolute constant $c'_\beta$,

$$\mathfrak{R}(T) \leqslant \sqrt{2H^3 MT \log \left( \frac{2}{\alpha} \right)} \\ + 2c'_\beta \cdot (d+k)H \cdot \sqrt{2 \log \left( \frac{(d+k)MTH}{\alpha} \right)} \left( \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{h=1}^{H} \left\| \widetilde{\phi}(m,x,a) \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \right).$$

Finally, to bound the summation, we use Lemma 9.10. We have that,

$$\sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{h=1}^{H} \left\| \widetilde{\phi}(x,a) \right\|_{(\widetilde{\mathbf{\Lambda}}^t_{m,h})^{-1}} \\ \leqslant 2 \sum_{h=1}^{H} \left( \log \left( \frac{\det\left(\widetilde{\mathbf{\Lambda}}^T_h\right)}{\det\left(\lambda \mathbf{I}_d\right)} \right) \left( \frac{M}{\log 2} \right) \sqrt{S} + 2 \sqrt{2MT \log \left( \frac{\det\left(\widetilde{\mathbf{\Lambda}}^T_h\right)}{\det\left(\lambda \mathbf{I}_d\right)} \right)} \right) \\ \leqslant 2H(d+\Gamma) \log(MT) M\sqrt{S} + 2H\sqrt{2(d+\Gamma)MT \log(MT)}.$$

Where the last inequality is an application of the variance decomposition (Lemma 9.15) and using the fact that $\|\phi(\cdot)\|_2 \leqslant 1$. Replacing this result, we have that with probability at least

$1-\alpha$,

$$\mathfrak{R}(T) \leqslant \sqrt{2H^3 MT \log\left(\frac{2}{\alpha}\right)}$$
$$+ 8c'_\beta \cdot (d+k)H^2 \cdot \sqrt{\log\left(\frac{dMTH}{\alpha}\right)} \left(\log(MT)M(d+\Gamma)\sqrt{S} + \sqrt{(d+\Gamma)MT\log(MT)}\right).$$

$$\implies \mathfrak{R}(T) = \widetilde{\mathcal{O}}\left((d+k)H^2\left(M(d+\Gamma)\sqrt{S} + \sqrt{(d+\Gamma)MT}\right)\sqrt{\log\left(\frac{1}{\alpha}\right)}\right).$$

## 9.12 Omitted Results

**Lemma 9.20** (Lemma E.2 of Yang et al. (2020b), Lemma D.4 of Jin et al. (2018))**.** *Let $\{x_\tau\}_{\tau=1}^\infty$ and $\{\phi_\tau\}_{\tau=1}^\infty$ be an $\mathcal{S}$-valued and an $\mathcal{H}$-valued stochastic process adapted to filtration $\{\mathcal{F}_\tau\}_{\tau=0}^\infty$ respectively, where we assume that $\|\phi_\tau\|_2 \leq 1$ for all $\tau \geq 1$. Besides, for any $t \geq 1$, define $\mathbf{\Lambda}_t : \mathcal{H} \to \mathcal{H}$ as $\mathbf{\Lambda}_t = \lambda\mathbf{I}_d + \sum_{\tau=1}^t \phi_\tau\phi_\tau^\top$ with $\lambda > 1$. Then, for any $\delta > 0$ with probability at least $1-\delta$, we have,*

$$\sup_{V\in\mathcal{V}} \left\| \sum_{\tau=1}^t \phi_\tau \left\{V(x_\tau) - \mathbb{E}[V(x_\tau)|\mathcal{F}_{\tau-1}]\right\}\right\|^2_{\mathbf{\Lambda}_t^{-1}}$$
$$\leq 4H^2 \cdot \log\frac{\det(\mathbf{\Lambda}_t)}{\det(\lambda\mathbf{I}_d)} + 4H^2 t(\lambda-1) + 8H^2\log\left(\frac{|\mathcal{N}_\epsilon|}{\delta}\right) + \frac{8t^2\epsilon^2}{\lambda}.$$

**Lemma 9.21** (Bound on Weights of Homogenous Value Functions, Lemma B.1 of Jin et al. (2020))**.** *Under the linear MDP Assumption (Definition 9.1), for any fixed policy $\pi$, let $\{\mathbf{w}_h^\pi\}_{h\in[H]}$ be the weights such that $Q_h^\pi(x,a) = \langle\phi(x,a), \mathbf{w}_h^\pi\rangle$ for all $(x,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$ and $m \in \mathcal{M}$. Then, we have,*

$$\|\mathbf{w}_h^\pi\|_2 \leq 2H\sqrt{d}.$$

**Lemma 9.22** (Bound on Weights of FedLSVI Policy for MDPs)**.** *At any $t \in [T]$ for any $m \in \mathcal{M}$ and all $h \in [H]$, we have that the weights $\mathbf{w}_{m,h}^t$ of Algorithm 18 satisfy,*

$$\|\mathbf{w}_{m,h}^t\|_2 \leq 2H\sqrt{dMt/\lambda}.$$

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

$$\begin{aligned}
\left|\mathbf{v}^\top \widehat{\boldsymbol{\theta}}^t_{m,h}\right| &= \left|\mathbf{v}^\top \left(\boldsymbol{\Lambda}^t_{m,h}\right)^{-1} \left(\sum_{\tau=1}^{U_h^m(t)} \left[\phi(x_\tau, a_\tau)\left[r(x_\tau, a_\tau) + \max_a Q_{m,h+1}(x'_\tau, a)\right]\right]\right)\right| \\
&\leq 2H \cdot \left|\mathbf{v}^\top \left(\boldsymbol{\Lambda}^t_{m,h}\right)^{-1} \left(\sum_{\tau=1}^{U_h^m(t)} \phi(x_\tau, a_\tau)\right)\right| \\
&\leq 2H \cdot \sqrt{\left|\left(\sum_{\tau=1}^{U_h^m(t)} \|\mathbf{v}\|^2_{\left(\boldsymbol{\Lambda}^t_{m,h}\right)^{-1}} \|\phi(x_\tau, a_\tau)\|^2_{\left(\boldsymbol{\Lambda}^t_{m,h}\right)^{-1}}\right)\right|} \\
&\leq 2H\|\mathbf{v}\|\sqrt{dU_h^m(t)/\lambda} \leq 2H\sqrt{dMt/\lambda}.
\end{aligned}$$

The penultimate inequality follows from Lemma 10.12 and the final inequality follows from the fact that $U_h^m(t) \leq Mt$. The remainder of the proof follows from the fact that for any vector $\mathbf{w}$, $\|\mathbf{w}\| = \max_{\mathbf{v}:\|\mathbf{v}\|=1} |\mathbf{v}^\top \mathbf{w}|$. □

**Lemma 9.23** (Covering Number for UCB-style value functions, Lemma D.6 of Jin et al. (2020))**.** *Let $\mathcal{V}$ denote a class of functions mapping from $\mathcal{S}$ to $\mathbb{R}$ with the following parameteric form*

$$V(\cdot) = \min\left\{\max_{a\in\mathcal{A}}\left[\mathbf{w}^\top\phi(\cdot, a) + \beta\sqrt{\phi(\cdot, a)^\top \boldsymbol{\Lambda}^{-1}\phi(\cdot, a)}\right], H\right\},$$

*where the parameters $(\mathbf{w}, \beta, \boldsymbol{\Lambda})$ are such that $\|\mathbf{w}\| \leq L$, $\beta \in (0, B]$, $\|\phi(x, a)\| \leq 1 \ \forall (x, a) \in \mathcal{S} \times \mathcal{A}$, and the minimum eigenvalue of $\boldsymbol{\Lambda}$ satisfies $\lambda_{\min}(\boldsymbol{\Lambda}) \geq \lambda$. Let $\mathcal{N}_\varepsilon$ be the $\varepsilon-$covering number of $\mathcal{V}$ with respect to the distance $dist(V, V') = \sup_{x\in\mathcal{S}} |V(x) - V'(x)|$. Then,*

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

## 9.13 Algorithm Pseudocode

**Algorithm 18** FedLSVI with Dense Communication

```
Input: $T, \phi, H, S$, sequence $\beta_h = \{(\beta^t_{m,h})_{m,t}\}$.
Initialize: $\mathbf{S}^t_{m,h}, \delta\mathbf{S}^t_{m,h} = \mathbf{0}, \mathcal{U}^m_h, \mathcal{W}^m_h = \emptyset$.
for episode $t = 1, 2, ..., T$ do
    for agent $m \in \mathcal{M}$ do
        Receive initial state $x^t_{m,1}$.
        Set $V^t_{m,H+1}(\cdot) \leftarrow 0$.
        for step $h = H, ..., 1$ do
            Compute $\mathbf{\Lambda}^t_{m,h} \leftarrow \mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h}$.
            Compute $\widehat{Q}^t_{m,h}$ and $\sigma^t_{m,h}$ (Eqns. 9.5 and 9.6).
            Compute $Q^t_{m,h}(\cdot, \cdot)$ (Eqn. 9.2)
            Set $V^t_{m,h}(\cdot) \leftarrow \max_{a \in \mathcal{A}} Q^t_{m,h}(\cdot, a)$.
        end for
        for step $h = 1, ..., H$ do
            Take action $a^t_{m,h} \leftarrow \arg\max_{a \in \mathcal{A}} Q^t_{m,h}(x^t_{m,h}, a)$.
            Observe $r^t_{m,h}, x^t_{m,h+1}$.
            Update $\delta\mathbf{S}^t_{m,h} \leftarrow \delta\mathbf{S}^t_{m,h} + \phi(z^t_{m,h})\phi(z^t_{m,h})^\top$.
            Update $\mathcal{W}^m_h \leftarrow \mathcal{W}^m_h \cup (m, x, a, x')$.
            if $\log \frac{\det\left(\mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h} + \lambda\mathbf{I}\right)}{\det\left(\mathbf{S}^t_{m,h} + \lambda\mathbf{I}\right)} > \frac{S}{\Delta t_{m,h}}$ then
                Synchronize$\leftarrow$ True.
            end if
        end for
    end for
    if Synchronize then
        for step $h = H, ..., 1$ do
            [$\forall$ Agents] Send $\mathcal{W}^h_m$ $\rightarrow$Server.
            [Server] Aggregate $\mathcal{W}^h \rightarrow \cup_{m \in \mathcal{M}} \mathcal{W}^m_h$.
            [Server] Communicate $\mathcal{W}^h$ to each agent.
            [$\forall$ Agents] Set $\delta\mathbf{S}^t_h \leftarrow 0, \mathcal{W}^m_h \leftarrow \emptyset$.
            [$\forall$ Agents] Set $\mathbf{S}^t_h \leftarrow \mathbf{S}^t_h + \sum_{z \in \mathcal{W}^h} \phi(z)\phi(z)^\top$.
            [$\forall$ Agents] Set $\mathcal{U}^m_h \leftarrow \mathcal{U}^m_h \cup \mathcal{W}^m_h$
        end for
    end if
end for
```

**Algorithm 19** FedLSVI with Rare Communication

```
Input: $T, \phi, H, S$, sequence $\beta_h = \{(\beta^t_{m,h})_{m,t}\}$.
Initialize: $\mathbf{S}^t_{m,h}, \delta\mathbf{S}^t_{m,h} = \mathbf{0}$.
for episode $t = 1, 2, ..., T$ do
    for agent $m \in \mathcal{M}$ do
        Receive initial state $x^t_{m,1}$.
        for step $h = 1, ..., H$ do
            Take action $a^t_{m,h} \leftarrow \arg\max_{a\in\mathcal{A}} Q^t_{m,h}(x^t_{m,h}, a)$.
            Observe $r^t_{m,h}, x^t_{m,h+1}$.
            Update $\delta\mathbf{S}^t_{m,h} \leftarrow \delta\mathbf{S}^t_{m,h} + \phi(z^t_{m,h})\phi(z^t_{m,h})^\top$.
            if $\frac{\det\left(\mathbf{S}^t_{m,h}+\delta\mathbf{S}^t_{m,h}+\lambda\mathbf{I}\right)}{\det\left(\mathbf{S}^t_{m,h}+\lambda\mathbf{I}\right)} > S$ then
                Synchronize← True.
            end if
        end for
    end for
    if Synchronize then
        [∀ Agents] Set $V^t_{m,H+1}(\cdot) \leftarrow 0$.
        for step $h = H, ..., 1$ do
            [∀ Agents] Compute $\mathbf{u}^t_{m,h}$ and $\mathbf{v}^t_{m,h}$.
            [∀ Agents] Send $\delta\mathbf{S}^t_{m,h}, \mathbf{u}^t_{m,h}, \mathbf{v}^t_{m,h}$ →Server.
            [Server] Aggregate $\delta\mathbf{S}^t_h = \sum_m \delta\mathbf{S}^t_{m,h}, \mathbf{u}^t_h = \sum_m[\mathbf{u}^t_{m,h} + \mathbf{v}^t_{m,h}]$.
            [Server] Communicate $\delta\mathbf{S}^t_h, \mathbf{u}^t_h$ to each agent.
            [∀ Agents] Set $\delta\mathbf{S}^t_h \leftarrow 0$.
            [∀ Agents] Set $\mathbf{S}^t_h \leftarrow \mathbf{S}^t_h + \delta\mathbf{S}^t_h$.
            [∀ Agents] Set $\mathbf{u}^t_{m,h} \leftarrow \mathbf{u}^t_h$.
            [∀ Agents] Compute $\mathbf{\Lambda}^t_{m,h} \leftarrow \mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h}$.
            [∀ Agents] Compute $\widehat{Q}^t_{m,h}$ and $\sigma^t_{m,h}$ (Eqns. 9.5 and 9.6).
            [∀ Agents] Compute $Q^t_{m,h}(\cdot,\cdot)$ (Eqn. 9.2)
            [∀ Agents] Set $V^t_{m,h}(\cdot) \leftarrow \max_{a\in\mathcal{A}} Q^t_{m,h}(\cdot, a)$.
        end for
    end if
end for
```

**Algorithm 20** FedLSVI for Large Heterogeneity

```
Input: $T, \widetilde{\phi}, H, S$, sequence $\beta_h = \{(\beta^t_{m,h})_{m,t}\}$.
Initialize: $\mathbf{S}^t_{m,h}, \delta\mathbf{S}^t_{m,h} = \mathbf{0}, \mathcal{U}^m_h, \mathcal{W}^m_h = \emptyset$.
for episode $t = 1, 2, ..., T$ do
  for agent $m \in \mathcal{M}$ do
    Receive initial state $x^t_{m,1}$.
    Set $V^t_{m,H+1}(\cdot) \leftarrow 0$.
    for step $h = H, ..., 1$ do
      Compute $\widetilde{\mathbf{\Lambda}}^t_{m,h} \leftarrow \mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h}$.
      Compute $\widehat{Q}^t_{m,h}$ and $\sigma^t_{m,h}$ (Eqn. 9.10).
      Compute $Q^t_{m,h}(\cdot, \cdot, \cdot)$ (Eqn. ??)
      Set $V^t_{m,h}(\cdot) \leftarrow \max_{a \in \mathcal{A}} Q^t_{m,h}(\cdot, a)$.
    end for
    for step $h = 1, ..., H$ do
      Take action $a^t_{m,h} \leftarrow \arg\max_{a \in \mathcal{A}} Q^t_{m,h}(m, x^t_{m,h}, a)$.
      Observe $r^t_{m,h}, x^t_{m,h+1}$.
      Update $\delta\mathbf{S}^t_{m,h} \leftarrow \delta\mathbf{S}^t_{m,h} + \widetilde{\phi}(m, z^t_{m,h})\widetilde{\phi}(m, z^t_{m,h})^\top$.
      Update $\mathcal{W}^m_h \leftarrow \mathcal{W}^m_h \cup (m, x, a, x')$.
      if $\log \frac{\det\left(\mathbf{S}^t_{m,h} + \delta\mathbf{S}^t_{m,h} + \lambda\mathbf{I}\right)}{\det\left(\mathbf{S}^t_{m,h} + \lambda\mathbf{I}\right)} > \frac{S}{\Delta t_{m,h}}$ then
        Synchronize← True.
      end if
    end for
  end for
  if Synchronize then
    for step $h = H, ..., 1$ do
      [∀ Agents] Send $\mathcal{W}^h_m$ →Server.
      [Server] Aggregate $\mathcal{W}^h \rightarrow \cup_{m \in \mathcal{M}} \mathcal{W}^m_h$.
      [Server] Communicate $\mathcal{W}^h$ to each agent.
      [∀ Agents] Set $\delta\mathbf{\Lambda}^t_h \leftarrow 0, \mathcal{W}^m_h \leftarrow \emptyset$.
      [∀ Agents] Set $\mathbf{\Lambda}^t_h \leftarrow \mathbf{\Lambda}^t_h + \sum_{(n,x,a) \in \mathcal{W}^h} \widetilde{\phi}(n, x, a)\phi(n, x, a)^\top$.
      [∀ Agents] Set $\mathcal{U}^m_h \leftarrow \mathcal{U}^m_h \cup \mathcal{W}^m_h$
    end for
  end if
end for
```

## Chapter 10

# Provably Efficient Algorithms for Cooperative Low-Rank Markov Games

A fundamental characteristic of federated environments is that they are *independent*, i.e., the rewards obtained by an agent are independent of the actions and transitions of the other agents in the federated environment. In this chapter, we discuss the problem of multi-agent reinforcement learning in multi-agent MDPs, where, in contrast to previous settings, all agents are assumed to exist in the *same* environment, e.g., in applications such as distributed 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 $M$ 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 state-action space grows exponentially in $M$, introducing a curse of dimensionality that makes standard approaches intractable. Furthermore, designing a *globally* optimal policy is difficult in practice owing to communication and computational constraints.

In single-agent *tabular* reinforcement learning (RL), algorithms exist (such as the one discussed in the previous chapter) 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., 2020a; Yang et al., 2020b) 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 & 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 $M$ 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 chapter, 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 multi-agent reinforcement learning algorithms with function approximation?*

The above question has three aspects that introduce different technical challenges. First, we consider *tractablility*: as mentioned earlier, in a cooperative multi-agent setting with many agents, the joint state-action space increases *exponentially*, where designing a tractable policy requires careful localization assumptions (Qu et al., 2020) that are non-trivial to extend to general function approximation settings. Next, we have the issue of *scalability*: MARL algorithms require communication, which is expensive for large $M$ and rich environments (i.e., large $\mathcal{S}$ and $\mathcal{A}$). Finally, we desire *efficiency* with respect to both algorithm runtime as well as sample complexity, and seek bounds that scale in terms of the complexity of the approximating function class, and not the overall $\mathcal{A}$ and $\mathcal{S}$.

**Contributions.** We answer the former 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

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

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 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 $\mathcal{O}((|\mathcal{S}||\mathcal{A}|)^M)$ to $o(dM)$, where $d$ is the dimensionality of the approximating function class.

Next, we generalize the cooperative multi-agent reinforcement learning 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 MG-LSVI (Markov Game Least Squares Value Iteration), a decentralized *vector-valued* optimistic value iteration algorithm that even under partial observability conditions, obtains a cumulative *Bayes regret* of $\widetilde{\mathcal{O}}(\bar{\chi}(G)H^2\sqrt{d^3T})$ over $T$ episodes, where $\bar{\chi}(G)$ denotes the *clique covering* number of $G$. MG-LSVI runs in polynomial time and only requires a communication budget of $o(Md^2 \log T)$ rounds per agent in the worst case, which can be much smaller for sparse $G$. This ensures that MG-LSVI 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), MG-LSVI 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.

**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 & Riedmiller, 2000; Boutilier, 1996), control (Yoshikawa, 1978; Wang & 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 & 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 this chapter, 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., 2020a; Yang et al., 2020b; Yang & 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 in this chapter.

**Organization**. Section 10.2 presents assumptions about the Markov game considered. Section 10.3 presents our performance objective and recovery guarantees. Section 10.4 presents our algorithm and associated regret upper and lower bounds, followed by a brief discussion and experimental results.

## 10.2 Preliminaries

**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 $M$, the state and action spaces are factorized as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \mathcal{S}_M$ and $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \mathcal{A}_M$, where $\mathcal{S}_v$ and $\mathcal{A}_v$ denote the individual state and action space for agent $v$ respectively. The transition matrix $\mathbb{P} = \{\mathbb{P}_h\}_{h \in [H]}, \mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ determines how the joint state evolves given an existing joint state-action, and the reward function $\mathbf{R} = \{\mathbf{r}_h\}_{h=1}^{H}, \mathbf{r}_h = \{r_{v,h}\}_{v=1}^{n}, r_{v,h} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward obtained by each agent $v$ 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 $\mathbb{P}_h^{\mathcal{Z}} : \mathcal{S} \times \mathcal{A} \times (\prod_{i \in \mathcal{Z}} \mathcal{S}_i) \rightarrow [0, 1]$. Next, we consider a graphical model of influence in order to remove the exponential dependence on $M$ (a generalization of prior work, e.g., Qu & Li (2019); Qu et al. (2020)), as summarized below.

**Assumption 10.1** (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*

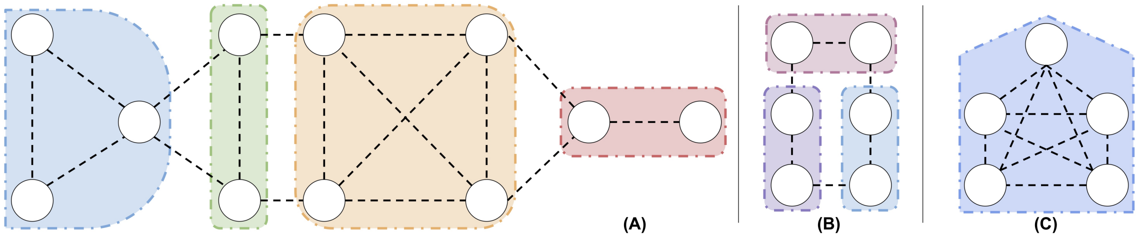


Figure 10-1: Example clique covering for (A) Typical graph with 11 nodes. (B) Circular graph with 6 nodes (worst case). (C) Fully-connected (complete) graph with 5 nodes (best case).

*$j$ (and vice-versa), and let $\mathcal{N}^+(v)$ denote, for any agent $v$ its neighborhood in $G$ (including itself). Alternatively, this implies that the reward for any agent $v$ obeys $r_{v,h} = r_{v,h}(\tilde{\boldsymbol{x}}_v, \tilde{\boldsymbol{a}}_v)$ where $\tilde{\boldsymbol{x}}_v = \{x_j\}_{j\in\mathcal{N}^+(v)}$ and $\tilde{\boldsymbol{a}}_v = \{x_j\}_{j\in\mathcal{N}^+(v)}$ denote the joint local state-action for agent $v$.*

**Remark 10.1** (Feasibility of Local Influence)**.** Networked influence assumptions similar to Assumption 10.1 have been explored extensively in the literature (Gu et al., 2020; Qu & 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.

Despite the above assumption, we are not quite yet equipped with a feasible learning model. This is evident as Assumption 10.1 in the worst case still leads to an exponential dependence on $M$, and combinatorial state-action spaces have been known to be intractable (Blondel & Tsitsiklis, 1999; Papadimitriou & Tsitsiklis, 1987). As a consequence, recent work has suggested additional conditions bounding the strength of interactions between agents to develop efficient policies (Qu & Li, 2019; Qu et al., 2020). We now describe an assumption to characterize dynamics in a similar vein.

**Definition 10.1** (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 10.2** (Clique-Dominant Dynamics)**.** *For the network $G$ defined in Assumption 10.1 let $\mathcal{C} = \cup_{l\in[k]} C_l$ be a known $k$-clique cover. For any subgraph $V \subseteq G$ and joint state-action pair $(\mathbf{x}, \mathbf{a})$, let $\mathbf{x}_V = \{\cup_{i\in V} x_i\}$ and $\mathbf{a}_V = \{\cup_{i\in V} a_i\}$ denote the joint state and action of all agents in $V$,* *and $\bar{\mathbf{x}}_V, \bar{\mathbf{a}}_V$ 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 $\widetilde{\mathbb{P}}_h^C : (\prod_{i \in C} \mathcal{S}_i) \times (\prod_{i \in C} \mathcal{A}_i) \times (\prod_{i \in C} \mathcal{S}_i) \to [0, 1]$, unknown functions $\{\widetilde{r}_{v,h}\}_{v=1}^{C}$ and a known nondecreasing function $\varepsilon(\cdot) : [1, M] \to [0, 1]$ such that for any joint state-action $(\mathbf{x}, \mathbf{a}) = (\{\mathbf{x}_C, \bar{\mathbf{x}}_C\}, \{\mathbf{a}_C, \bar{\mathbf{a}}_C\})$, we have for any $v \in C$, that,*

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

**Remark 10.2** (Feasibility of Clique-Dominance)**.** Assumption 10.2 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 $\varepsilon$ for all agents in $C$. Naturally, for a smaller clique-covering, a lower approximation error $\varepsilon$ can be expected. In fact, the minimal clique covering $\mathcal{C}^\star$ can incur zero error for certain $G$ (see Figure 10-1). Similar assumptions for local regularity have been made in prior work: Qu & Li (2019) introduce the $(c, \rho)-$exponential decay property that assumes a decay in the dependence of any agent on its neighborhood. Compared to the $(c, \rho)-$decay, 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 10.3** (Complexity of clique covering)**.** Assumption 10.2 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.

**Setting**. The game proceeds as follows. In each episode $t \in [T]$ each agent $v$ fixes a policy $\boldsymbol{\pi}_v(t) = \{\pi_v^h(t)\}_{h=1}^H$ in a (joint) initial state $\mathbf{x}_1(t) = \{x_v^1(t)\}_{v=1}^n$ picked arbitrarily by the environment. For each step $h \in [H]$ of the episode, each agent observes the local state $\tilde{\boldsymbol{x}}_v^h(t)$, selects an individual action $a_v^h(t) \sim \pi_v^h(\cdot|\tilde{\boldsymbol{x}}_v^h(t))$ (collectively the joint action $\mathbf{a}_h(t) = \{a_v^h(t)\}_{v=1}^n$), and obtains a reward $r_v^h(\tilde{\boldsymbol{x}}_v^h(t), \tilde{\boldsymbol{a}}_v^h(t))$ (collectively the joint reward $\mathbf{r}_h(\mathbf{x}_h(t), \mathbf{a}_h(t)) = \{r_v^h(\tilde{\boldsymbol{x}}_v^h(t), \tilde{\boldsymbol{a}}_v^h(t))\}_{v=1}^n$). All agents transition subsequently to a new joint state $\mathbf{x}_{h+1}(t) = \{x_v^{h+1}\}_{v=1}^n$ sampled according to $\mathbb{P}_h(\cdot|\mathbf{x}_h(t), \mathbf{a}_h(t))$. 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 $\boldsymbol{\pi} = \{\boldsymbol{\pi}_v\}_{v=1}^n$ denote a joint policy for all $M$ agents. We can define the vector-valued value function over all joint states $\mathbf{x} \in \mathcal{S}$ for a policy $\boldsymbol{\pi}$ and step $h$ as,

$$\mathbf{V}_h^{\boldsymbol{\pi}}(\mathbf{x}) \triangleq \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{i=h}^{H} \mathbf{r}_i(\mathbf{x}_i, \mathbf{a}_i) \;\middle|\; \mathbf{x}_h = \mathbf{x}\right].$$

Analogously, we define the vector-valued $Q$-function for a policy $\boldsymbol{\pi}$ and any $\mathbf{x} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, h \in [H]$,

$$\mathbf{Q}_h^{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{a}) \triangleq \mathbf{r}_h(\mathbf{x}, \mathbf{a}) + \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{i=h+1}^{H} \mathbf{r}_i(\mathbf{x}_i, \mathbf{a}_i) \;\middle|\; \mathbf{x}_h = \mathbf{x}, \mathbf{a}_h = \mathbf{a}\right].$$

At this point, we remark that Markov games are capable of modeling a variety of multi-agent decision processes, and are an instance of *stochastic games* (Shapley, 1953), which are closely related to the general framework for *repeated games* (Myerson, 1982). Repeated games are in turn generalizations of partially observable MDPs (POMDPs, Åström (1965)), and involve a variety of distinct challenges in competitive environments, most notably the conflict between individually optimal behavior and global objectives. While cooperative MARL has traditionally focused on recovering policies that maximize the *team average* reward, we now present a more general global performance objective.

## 10.3 Cooperative Behavior 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) which essentially are the policies that cannot improve any individual agent's reward without decreasing the reward of the other agents. Formally,

**Definition 10.2** (Pareto optimality, Paria et al. (2020))**.** *A policy $\boldsymbol{\pi}$ Pareto-dominates another policy $\boldsymbol{\pi}'$ if and only if $\mathbf{V}_1^{\boldsymbol{\pi}}(\mathbf{x}) \succeq \mathbf{V}_1^{\boldsymbol{\pi}'}(\mathbf{x}) \forall\ \mathbf{x} \in \mathcal{S}$. A policy is Pareto-optimal if it is not Pareto-dominated by any other policy. We denote the set of all policies by $\boldsymbol{\Pi}$, and the set of 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 $\boldsymbol{\Pi}^\star$. More broadly, the motivation to consider recovering the Pareto frontier is indeed derived from applications, e.g., in multi-agent EV charging protocols (Marinescu et al., 2014), smart grids (Chiu et al., 2019), and workflow optimization (Wang et al., 2019b).

**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* $\mathfrak{s}_{\boldsymbol{v}}(\mathbf{x}) = \boldsymbol{v}^\top \mathbf{x} : \mathbb{R}^M \to \mathbb{R}$ parameterized by $\boldsymbol{v}$ belonging to the set $\mathbf{Y} \subseteq \Delta^M$ (unit simplex in $M$ dimensions). We then have the *scalarized* value function $V_{\boldsymbol{v},h}^{\boldsymbol{\pi}}(\mathbf{x}) : \mathcal{S} \to \mathbb{R}$ and $Q-$function $Q_{\boldsymbol{v},h}^{\boldsymbol{\pi}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for some joint policy $\boldsymbol{\pi}$ as

$$V_{\boldsymbol{v},h}^{\boldsymbol{\pi}}(\mathbf{x}) \triangleq \mathfrak{s}_{\boldsymbol{v}}(\mathbf{V}_h^{\boldsymbol{\pi}}(\mathbf{x})) = \boldsymbol{v}^\top \mathbf{V}_h^{\boldsymbol{\pi}}(\mathbf{x})\ \text{, and } Q_{\boldsymbol{v},h}^{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{a}) \triangleq \mathfrak{s}_{\boldsymbol{v}}(\mathbf{Q}_h^{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{a})) = \boldsymbol{v}^\top \mathbf{Q}_h^{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{a}). \quad (10.1)$$

Since both $\mathcal{A} = \prod_i \mathcal{A}_i$ and $H$ are finite, there exists an optimal multi-agent policy for any fixed scalarization $\boldsymbol{v}$, which gives the value $V_{\boldsymbol{v},h}^\star = \sup_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} V_{\boldsymbol{v},h}^{\boldsymbol{\pi}}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}$ and $h \in [H]$. This policy coincides with the optimal policy for an MDP over the *joint* space $\mathcal{S} \times \mathcal{A}$, defined as follows.

**Proposition 10.1.** *For the scalarized value function given in Equation 10.1, the Bellman optimality conditions are given as, for all $h \in [H], \mathbf{x} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \boldsymbol{v} \in \mathbf{Y}$,*

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

*Proof.* We prove the above result by reducing the scalarized MMDP to an equivalent MDP. Observe that for any fixed $\boldsymbol{v} \in \mathbf{Y}$, the (vector-valued) rewards can be scalarized to a scalar reward. For any step $h \in [H]$, for any *fixed* $\boldsymbol{v} \in \mathbf{Y}$, consider the MDP with state space $\mathcal{S} = \mathcal{S}_1 \times ... \times \mathcal{S}_M$, action space $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_M$ and reward function $r'_h$ such that for all $(\mathbf{x}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}, r'_h(\mathbf{x}, \mathbf{a}) = \boldsymbol{v}^\top \mathbf{r}_h(\mathbf{x}, \mathbf{a})$. Therefore $r'_h(\mathbf{x}, \mathbf{a}) \in [0, 1]$ (since $\mathbf{r}_h$ lies on the $M-$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:

$$Q^\star_{\boldsymbol{v},h}(\mathbf{x}, \mathbf{a}) = r'_h(\mathbf{x}, \mathbf{a}) + \mathbb{P}_h V^\star_{\boldsymbol{v},h}(\mathbf{x}, \mathbf{a}), V^\star_{\boldsymbol{v},h}(\mathbf{x}) = \max_{\mathbf{a} \in \mathcal{A}} Q^\star_{\boldsymbol{v},h}(\mathbf{x}, \mathbf{a}), \text{and } V^\star_{\boldsymbol{v},H+1}(\mathbf{x}) = 0.$$

Replacing the reward function with the Sreward in terms of $\boldsymbol{v}$ provides us the result. □

The optimal policy for any fixed $\boldsymbol{v}$ is given by the greedy policy with respect to the Bellman-optimal scalarized $Q-$values. We denote this (unique) optimal policy by $\pi^\star_{\boldsymbol{v}}$. 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 $\boldsymbol{\Pi}^\star_{\mathbf{Y}} = \{\boldsymbol{\pi}^\star_{\boldsymbol{v}} | \boldsymbol{v} \in \Delta^M\}$ spans $\boldsymbol{\Pi}^\star$, and one can recover $\boldsymbol{\Pi}^\star$ by simply learning $\boldsymbol{\Pi}^\star_{\mathbf{Y}}$.

**Theorem 10.1.** *For any Markov game with finite $\mathcal{A}$ and $H$, $\boldsymbol{\Pi}^\star_{\mathbf{Y}} \subseteq \boldsymbol{\Pi}^\star$. If $\boldsymbol{\Pi}^\star$ is convex, $\boldsymbol{\Pi}^\star_{\mathbf{Y}} = \boldsymbol{\Pi}^\star$.*

*Proof.* First, we prove the forward direction, i.e., that $\boldsymbol{\Pi}^\star_{\mathbf{Y}} \subseteq \boldsymbol{\Pi}^\star$. The proof proceeds by contradiction. Assume that $\boldsymbol{\pi}^\star_{\boldsymbol{v}}$ does not lie in the Pareto frontier, then there exists a policy $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ such that $\mathbf{V}^{\boldsymbol{\pi}'}_1(\mathbf{x}) \succeq \mathbf{V}^{\boldsymbol{\pi}^\star_{\boldsymbol{v}}}_1(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}$ and $\boldsymbol{\pi} \neq \boldsymbol{\pi}^\star_{\boldsymbol{v}}$. Consider the final step $H$. Then, for any state $\mathbf{x} \in \mathcal{S}$, we have that if $\mathbf{V}^{\boldsymbol{\pi}'}_H(\mathbf{x}) \succeq \mathbf{V}^{\boldsymbol{\pi}^\star_{\boldsymbol{v}}}_H(\mathbf{x})$, then,

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

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

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

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

This is true only if $\boldsymbol{\pi}'_{H-1}(\mathbf{x}) = \boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{S}$, as $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}$ is the greedy policy with respect to $\mathfrak{s}_{\boldsymbol{\upsilon}} \mathbf{r}_{H-1}(\mathbf{x}, \mathbf{a}) + \mathbb{P}_H \mathbf{V}_H^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon},H}}(\mathbf{x}, \mathbf{a})$. Continuing this argument inductively for $h = H-2, H-3, ..., 1$ we obtain that $\mathbf{V}_1^{\boldsymbol{\pi}'}(\mathbf{x}) \succeq \mathbf{V}_1^{\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}}}(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{S}$ 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 $\boldsymbol{\Pi}^\star \subseteq \boldsymbol{\Pi}^\star_{\mathbf{Y}}$ for $\mathbf{Y} = \Delta^M$. 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 $\boldsymbol{\Pi}^\star_{\mathbf{Y}}$. Therefore, there does not exist any $\boldsymbol{\upsilon} \in \Delta^M$ such that $\boldsymbol{\pi}$ maximizes the value function of the scalarized MDP. Alternatively stated, for each $\boldsymbol{\upsilon} \in \mathbf{Y}$, there exists another policy $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}} \in \boldsymbol{\Pi}^\star_{\mathbf{Y}}$ 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 $\boldsymbol{\upsilon} \in \Delta^M$ 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 $[\boldsymbol{\upsilon}_\star]_i = \frac{[\mathbf{V}_1^{\boldsymbol{\pi}}]_i}{\|\mathbf{V}_1^{\boldsymbol{\pi}}\|_1} \in \Delta^M$. Now, by our assumption, there must exist an alternative policy $\boldsymbol{\pi}' \neq \boldsymbol{\pi}$ in $\boldsymbol{\Pi}^\star_{\mathbf{Y}}$, such that (by the convexity of scalarization), $\boldsymbol{\upsilon}_\star^\top(\mathbf{V}_1^{\boldsymbol{\pi}'} - \mathbf{V}_1^{\boldsymbol{\pi}}) \geq 0$. 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 $\boldsymbol{\pi} \in \boldsymbol{\Pi}^\star_{\mathbf{Y}}$. □

**Remark 10.4** (Limits of Scalarization)**.** Using scalarizations to recover $\boldsymbol{\Pi}^\star$ 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 convex scalarizations, however, we leave the extension to non-convex regions as future work, and assume $\mathbf{\Pi}^\star$ to be convex for simplicity.

**Bayes Regret.** Now, since we are in fact considering the recovery of a set of policies, it is unclear how regular *regret* in value approximation can provide a meaningful performance guarantee. Generally, algorithms for cooperative multi-agent RL consider maximizing the cumulative reward of all agents (Littman, 1994). Furthermore, in the fully-observable scenario (i.e., all agent observe the complete $(\mathbf{x}, \mathbf{a})$), the problem reduces to that of an MDP with fixed value and reward functions (given as the sum of individual value and rewards). Indeed by considering the scalarization $\boldsymbol{\upsilon}' = \frac{1}{M} \cdot \mathbf{1}_M$ we can observe that by Proposition 10.1, the optimal policy $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}'}$ corresponds to the optimal policy for the MDP defined over $\mathcal{S} \times \mathcal{A}$ with rewards given by the average of the rewards obtained by all agents. It is therefore straightforward to recover a no-regret policy using a single-agent algorithm (by making a linear MDP assumption over the joint state and action space $\mathcal{S} \times \mathcal{A}$, as in Jin et al. (2020)). Moreover, 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 $\mathbf{\Pi}^\star$ by learning a set of $T$ policies $\widehat{\mathbf{\Pi}}_T$ that minimize the Bayes regret, given by,

$$\mathfrak{R}_B(T) \triangleq \mathop{\mathbb{E}}_{\boldsymbol{\upsilon} \sim p_{\mathbf{Y}}} \left[ \max_{\mathbf{x} \in \mathcal{S}} \left[ V^\star_{\boldsymbol{\upsilon},1}(\mathbf{x}) - \max_{\boldsymbol{\pi} \in \widehat{\mathbf{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\mathbf{x}) \right] \right]. \tag{10.2}$$

Here $p_{\mathbf{Y}}$ is a distribution over $\mathbf{Y}$ that characterizes the nature of policies we wish to recover. For example, if we set $p_{\mathbf{Y}}$ as the uniform distribution over $\Delta^M$ 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 $\boldsymbol{\upsilon} \in \mathbf{Y}$, if $\boldsymbol{\pi}^\star_{\boldsymbol{\upsilon}} \in \widehat{\mathbf{\Pi}}_T$, then the regret incurred is 0. Hence, by collecting policies that minimize Bayes regret, we are effectively searching for policies that span dense regions of $\mathbf{\Pi}^\star$ (assuming convexity, see Remark 10.4). Consider now the *cumulative regret*:

$$\mathfrak{R}_C(T) \triangleq \sum_{t \in [T]} \mathop{\mathbb{E}}_{\boldsymbol{\upsilon}_t \sim p_{\mathbf{Y}}} \left[ \max_{\mathbf{x} \in \mathcal{S}} \left[ V^\star_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) \right] \right]. \tag{10.3}$$

Where $\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_T \sim p_{\mathbf{Y}}$ are sampled i.i.d. from $p_{\mathbf{Y}}$, and $\boldsymbol{\pi}_t$ refers to the joint policy at episode

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

$t$. Under suitable conditions on $\mathfrak{s}$ and $\mathbf{Y}$, we can bound the two quantities.

**Proposition 10.2.** *For $\mathfrak{s}$ that is Lipschitz and bounded $\mathbf{Y}$, we have that $\mathfrak{R}_B(T) \leq \frac{1}{T}\mathfrak{R}_C(T) + o(1)$.*

We minimize the regret $\mathfrak{R}_C(T)$, as any no-regret algorithm for $\mathfrak{R}_C$ bounds $\mathfrak{R}_B$.

## 10.4 An Efficient Algorithm with Linear Function Approximation

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

**Assumption 10.3** (Clique-dominant Linear Markov Game)**.** *Let $\mathcal{C}$ be a clique covering of $G$, and for any clique $C \in \mathcal{C}$, let $\mathcal{S}_C = \prod_{v \in C} \mathcal{S}_v$ and $\mathcal{A}_C = \prod_{v \in C} \mathcal{A}_v$ 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 10.2), and (b) for every $C \in \mathcal{C}, h \in [H]$, for a set of $|C| + 1$ features $\{\phi_v\}_{v \in C}, \phi_v : \mathcal{S}_C \times \mathcal{A}_C \to \mathbb{R}^d$ and $\boldsymbol{\psi}_C : \mathcal{S}_C \times \mathcal{A}_C \to \mathbb{R}^d$, there exist $d$ unknown measures $\boldsymbol{\mu}_{h,C}(\cdot) = \{\mu^1_{C,h}(\cdot), ..., \mu^d_{C,h}(\cdot)\}$ over $\mathcal{S}_C$ and an unknown vector $\boldsymbol{\theta}_{h,C} \in \mathbb{R}^d$ such that $\forall (\mathbf{x}, \mathbf{a}) \in \mathcal{S}_C \times \mathcal{A}_C$ and $v \in C$,*

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

*We denote the overall clique feature vector as $\boldsymbol{\Phi}_C(\cdot) \in \mathbb{R}^{d \times |C|}$, where, for any $\mathbf{x} \in \mathcal{S}_C, \mathbf{a} \in \mathcal{A}_C$, $\boldsymbol{\Phi}_C(\mathbf{x}, \mathbf{a}) = [[\phi_1(\mathbf{x}, \mathbf{a}), \boldsymbol{\psi}_C(\mathbf{x}, \mathbf{a})]^\top, ..., [\phi_{|C|}(\mathbf{x}, \mathbf{a}), \boldsymbol{\psi}_C(\mathbf{x}, \mathbf{a})]^\top]^\top$, and the overall approximate clique reward $\widetilde{\mathbf{r}}^C_h(\mathbf{x}, \mathbf{a}) = [\widetilde{r}_{1,h}(\mathbf{x}, \mathbf{a}), ..., \widetilde{r}_{|C|,h}(\mathbf{x}, \mathbf{a})]^\top$. Under this representation, we have that for any $\mathbf{x} \in \mathcal{S}_C, \mathbf{a} \in \mathcal{A}_C, h \in [H]$,*

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

*We assume, without loss of generality, that for each $C \in \mathcal{C}$ that $\|\boldsymbol{\Phi}_C(\mathbf{x}, \mathbf{a})\| \leq \sqrt{|C|}\ \forall\ (\mathbf{x}, \mathbf{a}) \in \mathcal{S}_C \times \mathcal{A}_C$, $\|\boldsymbol{\theta}_{h,C}\| \leq \sqrt{d}$ and $\|\boldsymbol{\mu}_{h,C}(\mathcal{S}_C)\| \leq \sqrt{d}$.*

Essentially, this assumption requires that once we are provided a clique covering, and the Markov game obeys the *clique-dominance* property (Assumption 10.2), the approximate rewards $\widetilde{r}_{v,h}$ are linear functions of a known feature vector $\phi_C$ evaluated on the joint state-action of the agents within its clique. Additionally, it assumes that the approximate marginal transition probabilities $\widetilde{\mathbb{P}}_h^C$ are linear functions of a known feature $\psi_C$. This, in fact, is a straightforward extension of the single-agent linear MDP parameterization (see Assumption A in Jin et al. (2020), developed from early work in Bradtke & Barto (1996); Melo & Ribeiro (2007)) to *clique-dominant* Markov games, as discussed below.

**Remark 10.5** (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., $r_{1,h} = ... = r_{M,h} \forall h \in [H]$, we have that assuming, for all agents that $\phi_1 = \phi_2 = ... = \phi_M$ satisfies the modeling requirement.

### 10.4.1 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 10.3 that this can be done in polynomial time with a 1.25 approximation of $\mathcal{C}^\star$. Since the game is clique-dominant (Assumption 10.2), 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 10.3 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 10.1** (Almost linear weights in Markov Games)**.** *Under Assumption 10.3 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 $\boldsymbol{\upsilon} = \{\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_k\} \in \mathbf{Y}$, there exist weights $\{\mathbf{w}_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}_\tau}\}_{h=1,\tau=1}^{H,k}$ such that*

$$\left| Q_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{a}) - \sum_{\tau=1}^{k} \boldsymbol{\upsilon}_\tau^\top \boldsymbol{\Phi}_\tau(\mathbf{x}_\tau, \mathbf{a}_\tau)^\top \mathbf{w}_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}_\tau} \right| \leq 2H\varepsilon(k) \ \forall (\mathbf{x}, \mathbf{a}, h),$$

*where* $\|\mathbf{w}_{\boldsymbol{\upsilon},h}^{\boldsymbol{\pi}_\tau}\|_2 \leq 2H\sqrt{d}, \ \forall \ \tau \in [k]$.

This result is proved in Section 10.8.2. 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 $\mathbf{\Pi}^\star_{\mathbf{Y}}$ recoverable by scalarization (see Remark 10.4), (b) next, we empirically approximate $\mathbf{\Pi}^\star_{\mathbf{Y}}$ with a collection of $T$ policies (one for each episode), such that the *Bayes Regret* is minimized (Proposition 10.2). In each episode $t \in [T]$, we sample a scalarization parameter $\boldsymbol{v}_t \sim p_{\mathbf{Y}}$, and run $k$ *vector-valued* decentralized linear least-squares regressions to approximate the optimal policy $\boldsymbol{\pi}^\star_{\boldsymbol{v}_t}$ with $k$ policies $\boldsymbol{\pi}_1(t), ...., \boldsymbol{\pi}_k(t)$ such that the resulting $Q-$values overestimate $Q^\star_{\boldsymbol{v}_t,h}$ with high probability. Then, each agent in clique $\tau$ selects the corresponding greedy action with respect to $\boldsymbol{\pi}_\tau(t)$. This approach is carried out via **vector-valued** 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 $\boldsymbol{v}(t) \in \mathbb{R}^M$, the $n_C$ values corresponding to agents in $\mathcal{C}$ is denoted by $\boldsymbol{v}^t_C$. Now, consider the MDP $\widetilde{\text{MDP}}_C$ formed by scalarizing the Markov game corresponding to the approximate rewards $\widetilde{\mathbf{r}}^C_h$ and transition dynamics $\widetilde{\mathbb{P}}^C_h$ with the parameter $\boldsymbol{v}_{C,t}$ (i.e., the reward function in $\widetilde{\text{MDP}}_C$ is given by $(\boldsymbol{v}^t_C)^\top \mathbf{r}^C_h$, 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 10.1, 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 $C$, we describe the procedure for an arbitrary agent in $C$.

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 $\{Q^t_{h,C}\}_{h\in[H]}$ 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 $\widehat{\mathbf{Q}}^t_{h,C}$ over $\mathbb{R}^{n_C}$, which is scalarized by $(\boldsymbol{v}^t_C)$ to obtain the $Q-$values as $Q^t_{h,C} = (\boldsymbol{v}^t_C)^\top \widehat{\mathbf{Q}}^t_{h,C}$. Each agent $m$ first sets $\widehat{\mathbf{Q}}^t_{h+1,C}$ 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,

$$
\begin{aligned}
&V^t_{h+1,C}(\mathbf{x}) \leftarrow \underset{\mathbf{a}\in\mathcal{A}}{\arg\max}\left[(\boldsymbol{v}^t_C)^\top\left((\mathbf{Q}^t_{h+1,C})^\top \mathbf{\Phi}_C(\mathbf{x},\mathbf{a})\right)\right] \ \forall\, \mathbf{x}\in\mathcal{S}_C,\\
&\widehat{\mathbf{Q}}^t_{h,C} \leftarrow \underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min}\left[\sum_{\tau\in[s^t_C]}\left\|\mathbf{y}^\tau_{h,C} - \mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})^\top\mathbf{w}\right\|_2^2 + \lambda\|\mathbf{w}\|_2^2\right],\\
&Q^t_{h,C}(\mathbf{x},\mathbf{a}) \leftarrow (\boldsymbol{v}^t_C)^\top\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})^\top\widehat{\mathbf{Q}}^t_{h,C} + \beta^t_{h,C}\cdot\left\|\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})^\top(\mathbf{\Lambda}^t_{h,C})^{-1}\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})\right\|_2.
\end{aligned}
$$

Where the last equation holds for any $(\mathbf{x},\mathbf{a})\in(\mathcal{S}_C\times\mathcal{A}_C)$ and the targets are defined as

$$
\mathbf{y}^\tau_{h,C} = \mathbf{r}^C_h(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C}) + \mathbf{1}_{n_C}\cdot V^t_{h+1,C}(\mathbf{x}^\tau_{h+1,C}),
$$

where $\mathbf{1}_{n_C}$ denotes the all-ones vector in $\mathbb{R}^{n_C}$, $\beta^t_{h,C}$ is selected such that with high probability the estimated $Q$-values overestimate the require $Q-$values, and $\mathbf{\Lambda}^t_{h,C}$ is described subsequently. Once all of these quantities are computed, each agent $v\in C$ selects the action $a^h_v(t) = \left[\arg\max_{\mathbf{a}\in\mathcal{A}_C} Q^t_{h,C}(\mathbf{x}^t_{h,C},\mathbf{a})\right]_v$ for each $h\in[H]$. Hence, the joint clique action $\mathbf{a}^t_{h,C} = \{a^h_v(t)\}_{v=1}^{n_C} = \arg\max_{\mathbf{a}\in\mathcal{A}_C} Q^h_C(\mathbf{x}^t_{h,C},\mathbf{a})$. Observe that while the computation of the policy is decentralized, the policies executed for all agents $v\in C$ coincide at all times by the modeling assumption and the periodic synchronizations between agents. We now present the closed form of $\widehat{\mathbf{Q}}^t_{h,C}$. Consider the contraction $\mathbf{z}^\tau_{h,C} = (\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})$ and the map $\mathbf{\Phi}^t_{h,C}:\mathbb{R}^d\rightarrow\mathbb{R}^{tn_C}$ such that for any $\boldsymbol{\theta}\in\mathbb{R}^d$,

$$
\mathbf{\Phi}^t_{h,C}\boldsymbol{\theta} \triangleq \left[(\mathbf{\Phi}_C(\mathbf{z}^1_{h,C})^\top\boldsymbol{\theta})^\top,\ldots,(\mathbf{\Phi}(\mathbf{z}^t_{h,C})^\top\boldsymbol{\theta})^\top\right]^\top.
$$

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

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

The algorithm is presented in Algorithm 21. 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 10.2). 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 $\mathbf{x}_C$ is visible to all agents (Assumption 10.1), 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 $\det(\mathbf{\Lambda}^t_{h,C}) \geq S \cdot \det(\mathbf{\Lambda}^{s^t_C}_{h,C})$, 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 $\mathcal{O}(M)$ messages by designating one agent per clique as the Server to aggregate messages.

**Lemma 10.2** (Communication complexity)**.** *Let the clique covering number be $\bar{\chi}(G)$ and let $n_{\max} \leq M$ denote the size of the largest clique of $G$. If we set $S > 1$, then the total number of communication episodes $\gamma \leq dH \cdot \bar{\chi}(G) \cdot \log_S \left(1 + \frac{Tn_{\max}}{d})\right) + \bar{\chi}(G) \cdot H$. When $S \leq 1$, $\gamma = T$.*

This result is proven in Section 10.8.3.

### 10.4.2 Regret Analysis

**Theorem 10.2.** *Algorithm 21 when run on a game with $M$ agents satisfying Assumptions 10.1, 10.2, 10.3 with error $\varepsilon_\star$, approximate clique covering $\widehat{\mathcal{C}}$, communication threshold $S$,*

$$\beta^t_{h,C} = \mathcal{O}(H\sqrt{d\log(MTH)} + \varepsilon_\star\sqrt{dT}) \;\forall\; C \in \widehat{\mathcal{C}},$$

*obtains, with probability at least $1-\alpha$, regret:*

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

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

The key technical challenges in the proof include deriving a martingale concentration result for multi-objective value iteration, a novel covering argument for vector-valued functions, and analysing the cost of rarely-switching policy updates for linear MDPs, all which may be of independent interest.

*Proof.* We first present a vector-valued concentration result which essentially extends the martingale analysis from the previous chapter (Lemmas 9.5 and 9.16) to a *vector* ridge regression problem.

**Lemma 10.3.** *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 $\mathbf{S}^t_{h,C}$ and $\boldsymbol{\Lambda}^t_{h,C}$ be defined as follows.*

$$\mathbf{S}^t_{h,C} = \sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C}) \left[\mathbf{v}^t_{\boldsymbol{v},h+1}(\mathbf{x}^\tau_{h+1,C}) - (\widetilde{\mathbb{P}}^C_h \mathbf{v}^t_{\boldsymbol{v},h+1})(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C})\right],$$

$$\boldsymbol{\Lambda}^t_{h,C} = \lambda \mathbf{I}_d + (\boldsymbol{\Phi}^{k_t}_{h,C})^\top (\boldsymbol{\Phi}^{k_t}_{h,C}).$$

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

$$\sup_{\boldsymbol{v}_C \in \mathbf{Y}_C} \left\|\mathbf{S}^t_{h,C}\right\|_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \leq B \cdot dH \sqrt{2\log\left(\frac{(C_\beta + 2)dMTH}{\delta'}\right)}.$$

*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 $\mathbf{S}^t_{h,C} = \sum_{\tau=1}^{k_t} \boldsymbol{\Phi}_C(\mathbf{z}^\tau_{h,C})[V^t_{\boldsymbol{v},h+1}(\mathbf{x}^\tau_{h+1,C}) - (\widetilde{\mathbb{P}}^C_h V^t_{\boldsymbol{v},h+1})(\mathbf{z}^\tau_{h,C})]$, where $\mathbf{v}^t_{\boldsymbol{v},h+1}$ is the vector with each entry being $V^t_{\boldsymbol{v},h+1}$. We have that

$$V^t_{\boldsymbol{v},h+1}(\mathbf{x}^\tau_{h+1,C}) - (\widetilde{\mathbb{P}}^C_h V^t_{\boldsymbol{v},h+1})(\mathbf{z}^\tau_{h,C}) = \mathbf{v}^t_{\boldsymbol{v},h+1} - \widetilde{\mathbb{P}}^C_h \mathbf{v}^t_{\boldsymbol{v},h+1}.$$

Consider the following distance metric $\text{dist}_{\mathbf{Y}_C}$,

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

Let $\mathcal{V}_{\mathbf{Y}_C}$ 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 $\mathcal{V}_{\mathbf{Y}_C}$ under $\text{dist}_{\mathbf{Y}_C}$, i.e., for every $\mathbf{v} \in \mathcal{V}_{\mathbf{Y}}$, there exists $\mathbf{v}' \in \mathcal{N}_\epsilon$ such that $\text{dist}_{\mathbf{Y}_C}(\mathbf{v}, \mathbf{v}') \leq \epsilon$. Now, here again, we adopt a similar strategy as the independent case. To bound the RHS, we decompose $\mathbf{S}^t_{h,C}$ in terms of the covering described earlier. We know that since $\mathcal{N}_\epsilon$ is an $\epsilon-$covering of $\mathcal{V}_{\mathbf{Y}_C}$, there exists a $\mathbf{v}' \in \mathcal{N}_\epsilon$ and $\mathbf{\Delta} = \mathbf{v}^t_{\boldsymbol{v},h+1} - \mathbf{v}'$ such that,

$$\mathbf{S}^t_{h,C} = \sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \left[\mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) - \widetilde{\mathbb{P}}^C_h \mathbf{v}'(\mathbf{z}^\tau_{h,C})\right] + \sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \left[\mathbf{\Delta}(\mathbf{x}^\tau_{h+1,C}) - \widetilde{\mathbb{P}}^C_h \mathbf{\Delta}(\mathbf{z}^\tau_{h,C})\right].$$

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

$$\begin{aligned}\left\|\mathbf{S}^t_{h,C}\right\|^2_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \leq 2\left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \left[\mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) - \widetilde{\mathbb{P}}^C_h \mathbf{v}'(\mathbf{z}^\tau_{h,C})\right]\right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \\ + \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}^\tau_C \bar{\varepsilon}_\tau\right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} + \frac{8Mt^2\epsilon^2}{\lambda}.\end{aligned}$$

Here $\bar{\varepsilon}_\tau$ denotes the maximum misspecification incurred from observing $\widetilde{\mathbb{P}}^h_C$ 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 $\varepsilon(k) \cdot \sqrt{dtM \log\left(\frac{\det(\mathbf{\Lambda}^t_{h,C})}{\lambda \mathbf{I}_d}\right)}$. To bound the first term on the RHS, we consider the substitution $\varepsilon^t_{\tau,h} = \mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) - \widetilde{\mathbb{P}}^C_h \mathbf{v}'(\mathbf{z}^\tau_h)$. 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 $\mathcal{F}_\tau = \sigma\left(\left\{\bigcup\left(\mathbf{x}^i_{h+1}, \mathbf{\Phi}_C(\mathbf{z}^i_h)\right)\right\}_{i \leq \tau}\right)$, and $\sigma$ denotes the $\sigma-$algebra generated by a finite set. Then, we have that,

$$\begin{aligned}&\left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \left[\mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) - \widetilde{\mathbb{P}}^C_h \mathbf{v}'(\mathbf{z}^\tau_{h,C})\right]\right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \\ &= \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \left[\mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) - \mathbb{E}\left[\mathbf{v}'(\mathbf{x}^\tau_{h+1,C}) | \mathcal{F}_{\tau-1}\right]\right]\right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \\ &= \left\|\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{z}^\tau_{h,C}) \varepsilon^t_{\tau,h}\right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}}.\end{aligned}$$

Note that for each $\varepsilon^t_{\tau,h}$, each entry is bounded by $2H$, and therefore we have that the vector

$\boldsymbol{\varepsilon}^t_{\tau,h}$ is $H-$sub-Gaussian. Then, applying Lemma 10.13, we have that,

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

Replacing this result for each $\mathbf{v} \in \mathcal{N}_\epsilon$, 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 $t \in [T], h \in [H]$,

$$\begin{aligned} \sup_{\boldsymbol{v}_t \in \mathbf{Y}, \mathbf{v} \in \mathcal{V}_\mathbf{Y}} \left\| \mathbf{S}^t_{h,C} \right\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} &\leq 2H \sqrt{\log \left( \frac{\det\left(\bar{\mathbf{\Lambda}}^t_h\right)}{\det\left(\lambda \mathbf{I}_d\right)} \right) + \log \left( \frac{HT|\mathcal{N}_\epsilon|}{\delta} \right) + \frac{2Mt^2\epsilon^2}{H^2 \lambda}} \\ &\leq 2H \sqrt{d \log \left( \frac{Mt + \lambda}{\lambda} \right) + \log \left( \frac{|\mathcal{N}_\epsilon|}{\delta} \right) + \log(HT) + \frac{2Mt^2\epsilon^2}{H^2 \lambda}}. \end{aligned}$$

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 $\mathcal{V}_{\mathbf{Y}_C}$ for any $h \in [H], m \in [M]$ or $t \in [T]$ under the distance function $\text{dist}_{\mathbf{Y}_C}(\mathbf{v}, \mathbf{v}') = \sup_{\mathbf{x} \in \mathcal{S}, \boldsymbol{v} \in \mathbf{Y}} \|\mathbf{v}(\mathbf{x}) - \mathbf{v}(\mathbf{x}')\|_1$. To bound this quantity by the appropriate covering number, we first observe that for any $V \in \mathcal{V}_{\mathbf{Y}_C}$, we have that the policy weights are bounded as $2HM\sqrt{dT/\lambda}$ (Lemma 10.11). Therefore, by Lemma 10.9 we have for any constant $B$ such that $\beta^t_h \leq B$,

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

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

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

Therefore, for some absolute constant $C'$ independent of $M, T, H, d$ and $C_\beta$, we have,

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

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,

$$\left\|\mathbf{S}^t_{h,C}\right\|^2_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \le 2H\Bigg((d+2+\varepsilon(k)dMT)\log\frac{MT+\lambda}{\lambda} + 2\log\left(\frac{1}{\delta'}\right) + C'd^2\log\left(C_\beta \cdot dMT\log(TMH)\right) + 2 + 4\log(TH)\Bigg).$$

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

$$\left\|\mathbf{S}^t_{h,C}\right\|_{(\boldsymbol{\Lambda}^t_h)^{-1}} \le B \cdot (dH + \varepsilon(k)H\sqrt{dMT})\sqrt{2\log\left(\frac{(C_\beta+2)dMTH}{\delta'}\right)}.$$

□

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 $\pi$.

**Lemma 10.4.** *Fix a clique $C \in \widehat{\mathcal{C}}$ such that $|C| = M$. For each $C$, there exists an absolute constant $c_\beta$ such that for $\beta^t_{h,C} = c_\beta \cdot (dH + \varepsilon(k) \cdot \sqrt{dtM})\sqrt{\log(2dMHt/\delta')}$ for any policy $\pi$, there exists a constant $C'_\beta$ 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 $1-\delta'/2$,*

$$\left|\left\langle \boldsymbol{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C), \mathbf{w}^t_{\boldsymbol{\upsilon}_C,h} - \mathbf{w}^\pi_{\boldsymbol{\upsilon}_C,h}\right\rangle\right| \le \mathbb{P}_h(V^t_{m,h+1} - V^\pi_{m,h+1})(\mathbf{x},\mathbf{a}) + 4H\varepsilon(k) + C'_\beta \cdot dH \cdot \|\boldsymbol{\Phi}_C(\mathbf{z}_C)\|_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

*Proof.* By the Bellman equation and Assumptions 10.1, 10.2, 10.3, we have that for any policy $\boldsymbol{\pi}$, and $\boldsymbol{\upsilon}_C \in \mathbf{Y}_C$, there exist weights $\mathbf{w}^\pi_{\boldsymbol{\upsilon}_C,h}$ such that, for all $\mathbf{z} = \{\mathbf{z}_C, \bar{\mathbf{z}}_C\} \in \mathcal{Z} = \mathcal{S} \times \mathcal{A}$,

$$\boldsymbol{\upsilon}_C^\top \boldsymbol{\Phi}_C(\mathbf{z}_C)^\top \mathbf{w}^\pi_{\boldsymbol{\upsilon}_C,h} = \boldsymbol{\upsilon}_C^\top \widetilde{\mathbf{r}}^C_h(\mathbf{z}_C) + \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{\upsilon}_C,h+1}(\mathbf{z}) = \boldsymbol{\upsilon}_C^\top\left(\widetilde{\mathbf{r}}^C_h(\mathbf{z}) + \mathbf{1}_M \cdot \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{\upsilon}_C,h+1}(\mathbf{z})\right).$$

We have,

$$\mathbf{w}^t_{\boldsymbol{\upsilon}_C,h} - \mathbf{w}^\pi_{\boldsymbol{\upsilon}_C,h}$$

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

$$\begin{aligned} \mathbf{w}^t_{\boldsymbol{v}_C,h} - \mathbf{w}^\pi_{\boldsymbol{v}_C,h} &= -\lambda (\boldsymbol{\Lambda}^t_{h,C})^{-1} \mathbf{w}^\pi_{\boldsymbol{v}_C,h} \\ &+ (\boldsymbol{\Lambda}^t_{h,C})^{-1} \left\{ \sum_{\tau=1}^{k_t} \left[ \boldsymbol{\Phi}_C(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C}) [\mathbf{1}_M \cdot (V^t_{\boldsymbol{v}_C,h+1}(\mathbf{x}'_\tau) - \widetilde{\mathbb{P}}^C_h V^\pi_{\boldsymbol{v}_C,h+1}(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C}))] \right] \right\}. \end{aligned}$$

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

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

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

Here the last inequality follows from Lemma 10.10. For the second term, we have by Lemma 10.3 that there exists an absolute constant $C_\beta$, independent of $M, T, H, d$ such that, with probability at least $1 - \delta'/2$ for all $t \in [T], h \in [H], \boldsymbol{v}_C \in \mathbf{Y}_C$ simultaneously,

$$\|\langle \boldsymbol{\Phi}_C(\mathbf{z}), \mathbf{v}_2 \rangle\|_2 \le \|\boldsymbol{\Phi}_C(\mathbf{z})\|_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \cdot C_\beta \cdot dH \cdot \sqrt{2 \log\left(\frac{dMTH}{\delta'}\right)}.$$

To bound the third term we make the substitution $\Delta_V = (V^t_{\boldsymbol{v}_C,h+1} - V^\pi_{\boldsymbol{v}_C,h+1})$ for brevity. We can bound it as follows.

$$\begin{aligned}
&\langle \mathbf{\Phi}_C(\mathbf{x},\mathbf{a}), \mathbf{v}_3\rangle \\
&= \left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\left\{\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})[\mathbf{1}_M\cdot(\mathbb{P}_h V^t_{\boldsymbol{v}_C,h+1} - \mathbb{P}_h V^\pi_{\boldsymbol{v}_C,h+1})(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})]\right\}\right\rangle \\
&= \left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\left\{\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})\mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})^\top \int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\}\right\rangle \\
&= \left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\left\{\sum_{\tau=1}^{k_t} \mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})\mathbf{\Phi}_C(\mathbf{x}^\tau_{h,C},\mathbf{a}^\tau_{h,C})^\top \int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\}\right\rangle \\
&= \left\langle \mathbf{\Phi}_C(\mathbf{z}), \int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\rangle - \lambda\left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\rangle \\
&= \int \Delta_V(\mathbf{x}')\left\langle \mathbf{\Phi}_C(\mathbf{z}), \boldsymbol{\mu}_h(\mathbf{x}')\right\rangle - \lambda\left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\rangle \\
&= \mathbf{1}_M\cdot\left(\widetilde{\mathbb{P}}^C_h\Delta_V(\mathbf{x},\mathbf{a})\right) - \lambda\left\langle \mathbf{\Phi}_C(\mathbf{z}), (\mathbf{\Lambda}^t_h)^{-1}\int \Delta_V(\mathbf{x}')d\boldsymbol{\mu}_h(\mathbf{x}')\right\rangle \\
&\le \mathbf{1}_M\cdot\left(\widetilde{\mathbb{P}}^C_h\Delta_V(\mathbf{x},\mathbf{a}) + 2H\sqrt{d\lambda}\|\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}}\right).
\end{aligned}$$

For the last two terms, we can bound them by a similar argument of misspecification as Lemma 10.3. We can bound both terms by $\mathbf{1}_M\cdot\left(\varepsilon(k)\cdot H\sqrt{dMT}\|\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}}\right)$. Putting it all together, we have that since $\langle \mathbf{\Phi}_C(\mathbf{x},\mathbf{a}), \mathbf{w}^t_{\boldsymbol{v}_C,h} - \mathbf{w}^\pi_{\boldsymbol{v}_C,h}\rangle = \langle \mathbf{\Phi}_C(\mathbf{x},\mathbf{a}), \mathbf{v}_1+\mathbf{v}_2+\mathbf{v}_3+\mathbf{v}_4+\mathbf{v}_5\rangle$, there exists an absolute constant $C_\beta$ independent of $M,T,H,d$, such that, with probability at least $1-\delta'/2$ for all $t\in[T], h\in[H], \boldsymbol{v}_C\in\mathbf{Y}_C$ simultaneously,

$$\begin{aligned}
\left|\langle \boldsymbol{v}_C^\top\mathbf{\Phi}_C(\mathbf{x},\mathbf{a}), \mathbf{w}^t_{\boldsymbol{v}_C,h} - \mathbf{w}^\pi_{\boldsymbol{v}_C,h}\rangle\right| &\le \boldsymbol{v}_C^\top\mathbf{1}_M\cdot\left(\mathbb{P}_h(V^t_{\boldsymbol{v}_C,h+1} - V^\pi_{\boldsymbol{v}_C,h+1})(\mathbf{x},\mathbf{a})\right) + 4H\varepsilon(k) \\
&\quad + \|\mathbf{\Phi}_C(\mathbf{x},\mathbf{a})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}}\Bigg(2H\sqrt{d\lambda} + C_\beta\cdot dH\cdot\sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} \\
&\qquad + 2HM\lambda\sqrt{d} + 2H\varepsilon(k)\sqrt{dMT}\Bigg)
\end{aligned}$$

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

$$\left|\langle \boldsymbol{v}_C^\top\mathbf{\Phi}_C(\mathbf{x},\mathbf{a}), \mathbf{w}^t_{\boldsymbol{v}_C,h} - \mathbf{w}^\pi_{\boldsymbol{v}_C,h}\rangle\right| \le \mathbb{P}_h(V^t_{\boldsymbol{v}_C,h+1} - V^\pi_{\boldsymbol{v}_C,h+1})(\mathbf{x},\mathbf{a}) + 4H\varepsilon(k)$$

$$+ C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \|\Phi(\mathbf{z})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.$$

□

**Lemma 10.5** (UCB in the Multiagent Setting). *For each $C \in \widehat{\mathcal{C}}$, with probability at least $1 - \delta'/2$, we have that for all $(\mathbf{x}_C, \mathbf{a}_C, h, t, \boldsymbol{v}_C) \in \mathcal{S}_C \times \mathcal{A}_C \times [H] \times [T] \times \mathbf{Y}_C$,*

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

*Proof.* We prove this result by induction. First, for the last step $H$, note that the statement holds as $Q^t_{\boldsymbol{v}_C,H}(\mathbf{x}_C, \mathbf{a}_C) \geq Q^\star_{\boldsymbol{v}_C,H}(\mathbf{x}_C, \mathbf{a}_C) - 4H\varepsilon(k)$ for all $\boldsymbol{v}_C$. Recall that the value function at step $H + 1$ is zero. Therefore, by Lemma 10.4, we have that, for any $\boldsymbol{v}_C \in \mathbf{Y}_C$,

$$\begin{aligned}&\left|\langle \boldsymbol{v}_C^\top \mathbf{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C), \mathbf{w}^t_{\boldsymbol{v}_C,H}\rangle - Q^\star_{\boldsymbol{v}_C,H}(\mathbf{x}_C, \mathbf{a}_C)\right| \\ &\qquad \leq C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \|\mathbf{\Phi}_C(\mathbf{z}_C)\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} + 4H\varepsilon(k).\end{aligned}$$

We have $Q^\star_{\boldsymbol{v}_C,H}(\mathbf{x}_C, \mathbf{a}_C) \leq \langle \boldsymbol{v}_C^\top \mathbf{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C), \mathbf{w}^t_{\boldsymbol{v}_C,H}\rangle + C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \|\Phi(\mathbf{z})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} = Q^t_{\boldsymbol{v}_C,H}(\mathbf{x}_C, \mathbf{a}_C)$. Now, for the inductive case, we have by Lemma 10.4 for any $h \in [H], \boldsymbol{v}_C \in \mathbf{Y}_C$,

$$\begin{aligned}&\left|\langle \boldsymbol{v}_C^\top \mathbf{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C), \mathbf{w}^t_{\boldsymbol{v}_C,h} - \mathbf{w}^\star_{\boldsymbol{v}_C,h}\rangle - \left(\mathbb{P}_h V^\star_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C) - \mathbb{P}_h V^t_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C)\right)\right| \\ &\leq C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \|\Phi(\mathbf{z})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)}.\end{aligned}$$

By the inductive assumption we have $Q^t_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C) \geq Q^\star_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C)$ implying

$$\mathbb{P}_h V^\star_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C) - \mathbb{P}_h V^t_{\boldsymbol{v}_C,h+1}(\mathbf{x}_C, \mathbf{a}_C) \geq 0.$$

Substituting the appropriate Q value formulations we have,

$$\begin{aligned}Q^\star_{\boldsymbol{v}_C,h} &\leq \langle \boldsymbol{v}_C^\top \mathbf{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C), \mathbf{w}^t_{\boldsymbol{v}_C,h}\rangle + 4H\varepsilon(k) \\ &\quad + C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \|\Phi(\mathbf{z})\|_{(\mathbf{\Lambda}^t_{h,C})^{-1}} \cdot \sqrt{2\log\left(\frac{dMTH}{\delta'}\right)} = Q^t_{\boldsymbol{v}_C,h}(\mathbf{x}_C, \mathbf{a}_C).\end{aligned}$$

□

**Lemma 10.6** (Recursive Relation in Multiagent MDP Settings)**.** *Fix a clique $C \in \widehat{\mathcal{C}}$ of size $M$. For any $\boldsymbol{v}_C \in \mathbf{Y}_C$, let $\delta^t_{\boldsymbol{v}_C,h} = V^t_{\boldsymbol{v}_C,h}(\mathbf{x}^t_{h,C}) - V^{\pi_t}_{\boldsymbol{v}_C,h}(\mathbf{x}^t_{h,C})$, and $\boldsymbol{\xi}^t_{\boldsymbol{v}_C,h+1} = \mathbb{E}\left[\delta^t_{\boldsymbol{v}_C,h}|\mathbf{x}^t_{h,C}, \mathbf{a}^t_{h,C}\right] - \delta^t_{\boldsymbol{v}_C,h}$. Then, with probability at least $1-\alpha$, for all $(t,h) \in [T] \times [H]$ simultaneously,*

$$\delta^t_{\boldsymbol{v}_C,h} \le \delta^t_{\boldsymbol{v}_C,h+1} + \boldsymbol{\xi}^t_{\boldsymbol{v}_C,h+1} + 4H\varepsilon(k) \\ + 2\left\|\boldsymbol{\Phi}_C(\mathbf{x}^t_{h,C}, \mathbf{a}^t_{h,C})\right\|_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \cdot C'_\beta \cdot \left(dH + \varepsilon(k)H\sqrt{dMT}\right) \cdot \sqrt{2\log\left(\frac{dMTH}{\alpha}\right)}.$$

*Proof.* By Lemma 10.4, we have that for any $(\mathbf{x}_C, \mathbf{a}_C, h, \boldsymbol{v}_C, t) \in \mathcal{S}_C \times \mathcal{A}_C \times [H] \times \mathbf{Y}_C \times [T]$ with probability at least $1-\alpha/2$,

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

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

**Lemma 10.7.** *For each clique $C \in \widehat{\mathcal{C}}$ and each $\boldsymbol{\xi}^t_{\boldsymbol{v}_C,h}$ as defined earlier and any $\delta \in (0,1)$, we have that with probability at least $1-\delta/2$,*

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

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

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

Rearranging provides us the final result. □

We are now ready to prove Theorem 10.2. We have by the definition of cumulative regret:

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

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

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

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

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

Where the last inequality holds with probability at least $1 - \alpha/2$, via Lemma 10.6 and Lemma 10.5. 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 $h \in [H]$

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

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

$$\sum_{t=1}^{T} \sum_{h=1}^{H} \left\| \boldsymbol{\Phi}_C(\mathbf{x}^t_{h,C}, \mathbf{a}^t_{h,C}) \right\|_{(\boldsymbol{\Lambda}^t_{h,C})^{-1}} \leq 2 \sum_{h=1}^{H} \left( \sqrt{S} \sum_{t=1}^{T} \left\| \boldsymbol{\Phi}_C(\mathbf{x}^t_{h,C}, \mathbf{a}^t_{h,C}) \right\|_{(\bar{\boldsymbol{\Lambda}}^t_{h,C})^{-1}} \right)$$

$$\leq 2H\sqrt{ST \cdot d \log \frac{MT+\lambda}{\lambda}}.$$

Where the last inequality is an application of Lemma 10.12 and using the fact that $\|\mathbf{\Phi}_C(\cdot)\|_2 \leq \sqrt{M}$. Replacing this result, we have that with probability at least $1-\alpha/2$, by a union bound over all cliques in $\widehat{\mathcal{C}}$,

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

We can bound the second term via Lemma 10.7. Taking expectation of the RHS over $\boldsymbol{v}_1, ..., \boldsymbol{v}_T$ gives us the final result (the $\widetilde{\mathcal{O}}$ notation hides polylogarithmic factors). □

**Remark 10.6** (Regret Bound)**.** Theorem 10.2 claims in conjunction with Proposition 10.2 that MG-LSVI obtains Bayes regret of $\widetilde{\mathcal{O}}(\bar{\chi}(G)\cdot\sqrt{T})$ even with limited communication. Note that for complete $G$, $\bar{\chi}(G) = 1$ 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 $p_{\mathbf{Y}}$ to be a point mass at the appropriate $\boldsymbol{v}$, with no increase in regret. Thirdly, we see that `MultOVI` can be simulated on a single agent with $M$ 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).

## 10.5 Lower Bound

The central observation in this setting is that under the clique-dominance assumption (Assumption 10.2), it is impossible to obtain regret that avoids the $\bar{\chi}(G)$ 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 $\text{MG}_C$ 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 $\boldsymbol{v}$ and is always $r_h^C$ and hence one can find the regret in $\mathbf{\Pi}^\star_{\mathbf{Y}}$ for the MG by simply choosing an arbitrary value of $\boldsymbol{v}$ 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 $\mathfrak{R}_C$, i.e., the total regret in the Markov game is $\Omega(\bar{\chi}(G)H\sqrt{dT})$.

This demonstrates that the $\bar{\chi}(G)$ term is unavoidable in general. Further, the utilization of "Bernstein-type" confidence bonuses can shave an additional factor of $\sqrt{H}$ in our regret (see discussion in Jin et al. (2020)). Regarding the dependence on communication, we conjecture that our bound is almost-optimal, as similar lower bounds exist for distributed exploration in multi-armed bandits (Hillel et al., 2013).

## 10.6 Experiments

We run experiments on a 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 $M = 8$ 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 $p_{\mathbf{Y}}$ is the uniform distribution over $\Delta_M$. The game runs in episodes of length `200` and $T = 5 \times 10^6$.

For each agent $c$ in clique $C$, the feature $\phi_c$ is given as the combined action of all visible agents (of dimensionality $8n'$) and $\psi_C$ is the joint state of all agents in $C$ (of dimensionality $100n'$) where $n' = 1$ in the self setting, $n' = \lfloor M/2 \rfloor + 1$ in the partial setting, $n' = M$ 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 $M$ individual Q-learning on the agents personal state space `Q-ind`, $M$ 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 | $20.885 \pm 2.833$ | $16.294 \pm 4.239$ | $13.202 \pm 4.887$ |
| DQN | $35.932 \pm 3.094$ | $27.587 \pm 5.059$ | $18.478 \pm 2.093$ |
| LSVI-ind | $17.439 \pm 2.192$ | $9.847 \pm 4.292$ | $7.340 \pm 3.778$ |
| MG-LSVI | $31.294 \pm 3.776$ | $22.119 \pm 5.882$ | $15.395 \pm 3.098$ |

Table 10.1: 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.

## 10.7 Discussion

**Remark 10.7** (Modeling influence and unknown dynamics)**.** For arbitrary influence graphs $G$, the misspecification $\varepsilon$ incurred by using a $k$-clique covering $\mathcal{C}$ of $G$ (Assumption 10.2) 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 10.8** (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 $\bar{\chi}(G)(dD+1)H$, which provides us a group regret of $\widetilde{\mathcal{O}}(\bar{\chi}(G) \cdot M^{\frac{1}{2D}} \cdot T^{\frac{1}{2}+\frac{1}{2D}})$. A balance can be obtained by setting $S = C'$ for some absolute constant $C'$, leading to a total $\mathcal{O}(\bar{\chi}(G) \cdot \log(n_{\max} T))$ rounds with $\widetilde{\mathcal{O}}(\bar{\chi}(G)\sqrt{T})$ regret.

**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.

## 10.8 Omitted Proofs

### 10.8.1 Proof of Proposition 10.2

Recall that $\mathbf{Y}$ is a bounded subset of $\mathbb{R}^M$. Now, we have that since $\mathfrak{s}_{\boldsymbol{v}}(\cdot) = \boldsymbol{v}^\top(\cdot)$, we have that $\mathfrak{s}_{\boldsymbol{v}}$ is Lipschitz with constant $M$ with respect to the $\ell_1-$norm, i.e., for any $\mathbf{y} \in \mathbb{R}^M$,

$$|\mathfrak{s}_{\boldsymbol{v}}(\mathbf{y}) - \mathfrak{s}_{\boldsymbol{v}'}(\mathbf{y})| \leq n\|\boldsymbol{v} - \boldsymbol{v}'\|_1.$$

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

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

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

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^T \max_{\mathbf{x}\in\mathcal{S}} \left[V_{\boldsymbol{v}_t,1}^{\boldsymbol{\pi}_t}(\mathbf{x}) - \mathbb{E}_{\boldsymbol{v}\in\mathbf{Y}}\left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V_{\boldsymbol{v},1}^{\boldsymbol{\pi}}(\mathbf{x})\right]\right] \\
&\leq \frac{1}{T}\sum_{t=1}^T \max_{\mathbf{x}\in\mathcal{S}} \left[V_{\boldsymbol{v}_t,1}^{\boldsymbol{\pi}_t}(\mathbf{x}) - \mathbb{E}_{\boldsymbol{v}\in\mathbf{Y}}\left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V_{\boldsymbol{v},1}^{\boldsymbol{\pi}}(\mathbf{x})\right]\right] \\
&\leq \mathbb{E}_{q(X,Y)}\left[\max_{\mathbf{x}\in\mathcal{S}}\left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V_{X,1}^{\boldsymbol{\pi}}(\mathbf{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V_{Y,1}^{\boldsymbol{\pi}}(\mathbf{x})\right]\right] \\
&\leq n \cdot \mathbb{E}_{q(X,Y)}\left[\|X - Y\|_1\right].
\end{aligned}$$

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

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

$$
\begin{aligned}
&= \mathbb{E}_{\boldsymbol{\upsilon}\in\mathbf{Y}}\left[\max_{\mathbf{x}\in\mathcal{S}}\left[V^{\star}_{\boldsymbol{\upsilon},1}(\mathbf{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\mathbf{x})\right]\right] - \mathbb{E}_{\mathcal{H}}\left[\frac{1}{T}\sum_{t=1}^{T}\max_{\mathbf{x}\in\mathcal{S}}\left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x})\right]\right] \\
&= \mathbb{E}_{\mathcal{H}}\left[\frac{1}{T}\sum_{t=1}^{T}\max_{\mathbf{x}\in\mathcal{S}}\left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) - \max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x})\right]\right] - \mathbb{E}_{\mathcal{H}}\left[\frac{1}{T}\sum_{t=1}^{T}\max_{\mathbf{x}\in\mathcal{S}}\left[V^{\star}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) - V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x})\right]\right] \\
&\leq \mathbb{E}_{\mathcal{H}}\left[\frac{1}{T}\sum_{t=1}^{T}\max_{\mathbf{x}\in\mathcal{S}}\left[V^{\boldsymbol{\pi}_t}_{\boldsymbol{\upsilon}_t,1}(\mathbf{x}) - \mathbb{E}_{\boldsymbol{\upsilon}\in\mathbf{Y}}\left[\max_{\boldsymbol{\pi}\in\widehat{\boldsymbol{\Pi}}_T} V^{\boldsymbol{\pi}}_{\boldsymbol{\upsilon},1}(\mathbf{x})\right]\right]\right] \\
&\leq n \cdot \mathbb{E}_{q(X,Y)}\left[\|X - Y\|_1\right].
\end{aligned}
$$

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 & Rosasco (2012).

### 10.8.2 Proof of Lemma 10.1

Follows from Lemma 10.10.

### 10.8.3 Proof of Lemma 10.2

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,

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

Where the last inequality follows from Lemma 10.12 and the fact that $\|\boldsymbol{\Phi}\| \leq \sqrt{n_{\max}} \leq \sqrt{M}$. This gives us that $n_h(C) \leq d \log_S\left(1 + n_{\max}T/d)\right) + 1$. Furthermore, by noticing that $\gamma \leq \sum_{C\in\widehat{\mathcal{C}}}\sum_{h=1}^{H} n_h(C)$, and that $|\widehat{\mathcal{C}}| \leq 1.25 \cdot \bar{\chi}(G)$, we have the final result.

### 10.8.4 Additional Lemmas

**Lemma 10.8** (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 10.9** (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 parametric form*

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

*where the parameters $(\mathbf{w}, \beta, \boldsymbol{\Lambda})$ are such that $\mathbf{w} \in \mathbb{R}^d$, $\|\mathbf{w}\|_2 \leq L$, $\beta \in (0, B]$, $\|\boldsymbol{\Phi}_C(\mathbf{x},\mathbf{a})\| \leq \sqrt{M}\ \forall(\mathbf{x},\mathbf{a}) \in \mathcal{S}\times\mathcal{A}$, and the minimum eigenvalue of $\boldsymbol{\Lambda}$ satisfies $\lambda_{\min}(\boldsymbol{\Lambda}) \geq \lambda$. Let $\mathcal{N}_\varepsilon$ be the $\varepsilon-$covering number of $\mathcal{V}$ with respect to the distance dist$(\mathbf{v},\mathbf{v}') = \sup_{\mathbf{x}\in\mathcal{S},\boldsymbol{\upsilon}\in\mathbf{Y}}|\mathbf{v}_{\boldsymbol{\upsilon}}(\mathbf{x}) - \mathbf{v}'_{\boldsymbol{\upsilon}}(\mathbf{x})|$. 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 $\mathbf{A}_1 = \beta^2\boldsymbol{\Lambda}_1^{-1}, \mathbf{A}_2 = \beta^2\boldsymbol{\Lambda}_2^{-1}$ and weight matrices $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$,

$$\begin{aligned}
&\sup_{\boldsymbol{\upsilon}\in\mathbf{Y},\mathbf{x}\in\mathcal{S}}\left|\mathbf{v}_{\boldsymbol{\upsilon}}(\mathbf{x}) - \mathbf{v}'_{\boldsymbol{\upsilon}}(\mathbf{x})\right|_1 = M\cdot\sup_{\mathbf{x}\in\mathcal{S},\boldsymbol{\upsilon}\in\mathbf{Y}}\left|\boldsymbol{\upsilon}^\top\mathbf{v}(\mathbf{x}) - \boldsymbol{\upsilon}^\top\mathbf{v}'(\mathbf{x})\right| \leq M\cdot\sup_{\mathbf{x}\in\mathcal{S}}\left|\mathbf{v}(\mathbf{x}) - \mathbf{v}'(\mathbf{x})\right|_1\\
&\leq M\cdot\sup_{\mathbf{x},\mathbf{a}}\left|\left[\mathbf{w}_1^\top\boldsymbol{\Phi}_C(\cdot,\mathbf{a}) + \left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_1\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2\right] - \left[\mathbf{w}_2^\top\boldsymbol{\Phi}_C(\cdot,\mathbf{a}) + \left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_2\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2\right]\right|_1\\
&\leq M\cdot\sup_{\mathbf{x},\mathbf{a}}\left|\left(\mathbf{w}_1 - \mathbf{w}_2\right)^\top\boldsymbol{\Phi}_C(\cdot,\mathbf{a}) + \left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_1\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2 - \left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_2\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2\right|_1\\
&\leq M\cdot\sup_{\mathbf{x},\mathbf{a}}\left|\left(\mathbf{w}_1 - \mathbf{w}_2\right)^\top\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right|_1 + M\cdot\sup_{\mathbf{x},\mathbf{a}}\left|\left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_1\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2 - \left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\mathbf{A}_2\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2\right|\\
&\leq M\cdot\sup_{\mathbf{x},\mathbf{a}}\left|\left(\mathbf{w}_1 - \mathbf{w}_2\right)^\top\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right|_1 + M\cdot\sup_{\mathbf{x},\mathbf{a}}\left\|\boldsymbol{\Phi}_C(\cdot,\mathbf{a})^\top\left(\mathbf{A}_1 - \mathbf{A}_2\right)\boldsymbol{\Phi}_C(\cdot,\mathbf{a})\right\|_2\\
&\leq M^{3/2}\cdot\sup_{\boldsymbol{\Phi}:\|\boldsymbol{\Phi}\|\leq\sqrt{M}}\left[\left\|\left(\mathbf{w}_1 - \mathbf{w}_2\right)^\top\boldsymbol{\Phi}\right\|_2\right] + M\cdot\sup_{\boldsymbol{\Phi}:\|\boldsymbol{\Phi}\|\leq\sqrt{M}}\left\|\boldsymbol{\Phi}^\top\left(\mathbf{A}_1 - \mathbf{A}_2\right)\boldsymbol{\Phi}\right\|_2\\
&\leq M^2\cdot\|\mathbf{w}_1 - \mathbf{w}_2\|_2 + M^2\|\mathbf{A}_1 - \mathbf{A}_2\|_2\\
&\leq M^2\cdot\|\mathbf{w}_1 - \mathbf{w}_2\|_2 + M^2\|\mathbf{A}_1 - \mathbf{A}_2\|_F
\end{aligned}$$

Now, let $\mathcal{C}_{\mathbf{w}}$ be an $\varepsilon/(2M^2)$ cover of $\left\{\mathbf{w}\in\mathbb{R}^d \mid \|\mathbf{w}\|_2 \leq L\right\}$ with respect to the Frobenius-norm, and $\mathcal{C}_{\mathbf{A}}$ be an $\varepsilon^2/4$ cover of $\left\{\mathbf{A}\in\mathbb{R}^{d\times d} | \|\mathbf{A}\|_F \leq (M^2d)^{1/2}B^2\lambda^{-1}\right\}$ with respect to the Frobenius norm. By Lemma 10.8 we have,

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

Therefore, we can select, for any $\mathbf{v}_{\boldsymbol{\upsilon}}(\cdot)$, corresponding weight $\mathbf{w}\in\mathcal{C}_{\mathbf{w}}$, and matrix $\mathbf{A}\in\mathcal{C}_{\mathbf{A}}$.

Therefore, $\mathcal{N}_\varepsilon \leq |\mathcal{C}_{\mathbf{A}}| \cdot |\mathcal{C}_{\mathbf{w}}|$. This gives us,

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

□

**Lemma 10.10** (Linearity of weights in Markov game)**.** *In a game with $M$ agents satisfying Assumptions 10.1, 10.2, 10.3, for any policy $\pi$, clique $C \in \widehat{\mathcal{C}}$ of size $M$, and $\boldsymbol{v}_C \in \mathbf{Y}_C$, there exists weights $\{\mathbf{w}^{\pi}_{\boldsymbol{v}_C,h}\}_{h\in[H]}$ such that $|Q^{\pi}_{\boldsymbol{v}_C,h}(\mathbf{x}_C, \mathbf{a}_C) - \boldsymbol{v}_C^\top \boldsymbol{\Phi}_C(\mathbf{x}_C, \mathbf{a}_C)^\top \mathbf{w}^{\pi}_{\boldsymbol{v}_C,h}| \leq 2H\varepsilon(k)$ for all $(\mathbf{x}_C, \mathbf{a}_C, h) \in \mathcal{S}_C \times \mathcal{A}_C \times [H]$, where $\|\mathbf{w}^{\pi}_{\boldsymbol{v}_C,h}\|_2 \leq 2H\sqrt{d}$.*

*Proof.* By the Bellman equation and Proposition 10.1, we have that for any MDP corresponding to the scalarization parameter $\boldsymbol{v}_C \in \mathbf{Y}_C$ and any policy $\boldsymbol{\pi}$, state $\mathbf{x} \in \mathcal{S}_C$, joint action $\mathbf{a} \in \mathcal{A}_C$,

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

The first inequality follows from Assumption 10.2. Here $\mathbf{w}^{\pi}_{\boldsymbol{v}_C,h} = \begin{bmatrix} \boldsymbol{\theta}_h \\ \int V^{\pi}_{\boldsymbol{v}_C,h+1}(\mathbf{x}'_C) d\boldsymbol{\mu}(\mathbf{x}'_C) d\mathbf{x}'_C \end{bmatrix}$. Therefore, since $\|\boldsymbol{\theta}_h\| \leq \sqrt{d}$ and $\|\int V^{\pi}_{\boldsymbol{v}_C,h+1}(\mathbf{x}'_C) d\boldsymbol{\mu}(\mathbf{x}'_C)\| \leq H\sqrt{d}$, the result follows. □

**Lemma 10.11** (Bound on Weights)**.** *For any $C \in \widehat{\mathcal{C}}, |C| = M, t \in [T], h \in [H], \boldsymbol{v} \in \mathbf{Y}$, the weights $\mathbf{w}^t_{\boldsymbol{v}_C,h}$ satisfy*

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

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

$$\left|\mathbf{v}^\top \mathbf{w}^t_{\boldsymbol{v},h}\right| = \left|\mathbf{v}^\top \left(\boldsymbol{\Lambda}^t_h\right)^{-1} \left(\sum_{\tau=1}^{k_t} \left[\boldsymbol{\Phi}_C(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C}) \left[\mathbf{r}_h(\mathbf{x}^\tau_{h,C}, \mathbf{a}^\tau_{h,C}) + \max_{\mathbf{a}\in\mathcal{A}} Q_{\boldsymbol{v},h+1}(\mathbf{x}'_\tau, \mathbf{a})\right]\right]\right)\right|$$

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

The penultimate inequality follows from Lemma 10.12 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 $\mathbf{w}$, $\|\mathbf{w}\| = \max_{\mathbf{v}:\|\mathbf{v}\|=1} |\mathbf{v}^\top \mathbf{w}|$. □

**Lemma 10.12** (Lemma 3 of Abbasi-Yadkori et al. (2011))**.** *Let $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{R}^d$ 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 $\nu > 1$,*

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

### 10.8.5 Multi-Task Concentration Bound

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

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

$\Gamma$ by

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

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 $\sigma$-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 $\ell^2$ space. Following Chowdhury & Gopalan (2020), consider now the map $\Phi_{\mathcal{X}_t} : \ell^2 \to \mathbb{R}^{Mt}$:

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

Additionally, denote $\mathbf{V}_t := \Phi_{\mathcal{X}_t}^\top \Phi_{\mathcal{X}_t}$ be a map from $\ell^2$ to itself, with $\mathbf{I}$ being the identity operator in $\ell^2$.

**Lemma 10.13** (Lemma 3 of Chowdhury & Gopalan (2020))**.** *Let the noise vectors $\{\varepsilon_t\}_{t \geqslant 1}$ be $\sigma$-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 $t \geqslant 1$:*

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

## 10.9 Algorithm Pseudocode

**Algorithm 21** MG-LSVI: Decentralized Learning in Low-Rank Cooperative Markov Games

```
Input: $T, \mathbf{\Phi}, H, S$, sequence $\beta_h = \{(\beta_h^t)_t\}$.
Initialize: $\mathbf{\Lambda}_h^C(t) = \lambda \mathbf{I}_d, \delta\mathbf{\Lambda}_{h,C}^t = \mathbf{0}, \mathcal{U}_\nu^h, \mathcal{W}_\nu^h = \emptyset$ for each $\nu \in G$, clique cover $\widehat{\mathcal{C}}$ of $G$.
for episode $t = 1, 2, ..., T$ do
    Sample $\boldsymbol{v}_t \sim p_{\mathbf{Y}}$ using public randomness.
    for clique $C \in \widehat{\mathcal{C}}$ do
        for agent $\nu \in C$ do
            Set $V_{h+1,C}^t(\cdot) \leftarrow 0$.
            for step $h = H, ..., 1$ do
                Compute $Q_{h,C}^t(\cdot, \cdot)$ using vector-valued least-squares regression on $\mathcal{U}_\nu^h$.
                Set $V_{h+1,C}^t(\cdot) \leftarrow \max_{\mathbf{a} \in \mathcal{A}_C} Q_{h,C}^t(\cdot, \mathbf{a})$.
            end for
            for step $h = 1, ..., H$ do
                Each agent observes partial state $x_v^h(t)$ creating clique state $\mathbf{x}_C^h(t) = \cup_{v \in C} x_v^h(t)$.
                Take action $a_\nu^h(t) \leftarrow [\arg\max_{a \in \mathcal{A}_C} Q_{h,C}^t(\mathbf{x}_C^h(t), \mathbf{a})]_\nu$.
                Observe $r_\nu^h(t), \tilde{\boldsymbol{x}}_\nu^{h+1}$.
                Update $\delta\mathbf{\Lambda}_{h,C}^t \leftarrow \delta\mathbf{\Lambda}_C^h(t-1) + \mathbf{\Phi}_C(\mathbf{z}_C^h(t))\mathbf{\Phi}_C(\mathbf{z}_C^h(t))^\top$.
                Update $\mathcal{W}_\nu^h \leftarrow \mathcal{W}_\nu^h \cup (\nu, x_\nu^h(t), r_\nu^h(t))$.
                if $\log \frac{\det\left(\mathbf{\Lambda}_{h,C}^t + \delta\mathbf{\Lambda}_{h,C}^t + \lambda\mathbf{I}\right)}{\det\left(\mathbf{\Lambda}_{h,C}^t + \lambda\mathbf{I}\right)} > S$ then
                    SYNCHRONIZE← 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
                [∀ AGENTS] Send $\mathcal{W}_\nu^h$ →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.
                [∀ AGENTS] Set $\delta\mathbf{\Lambda}_C^h(t+1) \leftarrow 0, \mathcal{W}_\nu^h \leftarrow \emptyset$.
                [∀ AGENTS] Set $\mathbf{\Lambda}_C^h(t+1) \leftarrow \mathbf{\Lambda}_{h,C}^t + \sum_{(\mathbf{x},\mathbf{a}) \in \mathcal{W}^h} \mathbf{\Phi}_C(\mathbf{x}, \mathbf{a})\mathbf{\Phi}_C(\mathbf{x}, \mathbf{a})^\top$.
                [∀ AGENTS] Set $\mathcal{U}_\nu^h \leftarrow \mathcal{U}_\nu^h \cup \mathcal{W}^h$
            end for
        end if
    end for
end for
```

# Chapter 11

# Concluding Remarks

This thesis discusses problems in sequential decision-making and online learning applied to the constraints and environments prevalent in the emerging paradigm of *federated learning*. While the federated setting itself is in its infancy with an initial focus primarily on enabling large-scale optimization of neural networks (Li et al., 2018, 2020; Konečnỳ et al., 2016; Yu et al., 2020b), federated decision-making will inevitably increase in relevance given the growing applications of active artificial intelligence.

While we considered the fundamental tradeoffs in decentralized and federated decision-making and the pursuit of optimality for such problems, there are numerous challenges that are yet to be explored within this problem domain. We will conclude by briefly summarizing these directions.

## 11.1 Computational Complexity

One of the most essential and relevant characterizations of an algorithm is its computational complexity, a topic that has largely been in the background of discussion within this thesis. For online learning and bandit problems, it is well-known (Lattimore & Szepesvári, 2020) that the computational complexity of most no-regret algorithms is heavily dependent on the space of actions (i.e., the decision set) for the agent. For most practical problems (e.g., in recommender systems), this space is typically finite (and often countable), i.e., a list of elements, or a subset of $\mathbb{R}^d$, in which case searching for the optimal action for many UCB-like or Thompson Sampling algorithms is relatively straightforward and can be achieved in polynomial time (see, e.g., Mutny & Krause (2018) for a discussion).

However, in some cases, the function approximation aspect of the problem itself can be expensive, e.g., when the unknown bandit function or RL reward function has a small norm within a reproducing kernel Hilbert space (Chowdhury & Gopalan, 2017), where approaches typically require the inversion of matrices of size $t \times t$ (at the $t^{th}$ round) in infinite-dimensional kernels (due to the kernel trick). This inversion can indeed be replaced by an online rank-1 matrix update, reducing the complexity to $\mathcal{O}(1)$ per round, but this comes at an immense cost to communication: it requires storing $\mathcal{O}(t^2)$-sized statistics which are prohibitively large to communicate in the federated setting. Interestingly, this dilemma can be solved by the "projection" approach outlined in Chapter 7, by projecting the infinite-dimensional "approximate" Hilbert space. While the discussion in Chapter 7 is focused on how this projection enables us to ensure differential privacy guarantees, this step enables us to side-step both computational and communication issues, indeed, this approach only requires a constant $\mathcal{O}(d^2)$ communication and computational complexity (where $d$ is the dimensionality of the approximating space). This hints towards a broader synergy between differential privacy and computational efficiency: if our algorithm requires operations on smaller statistics, then we can expect that (a) computational and communication requirements will be lower, and that (b) better guarantees for differential privacy can be achieved with smaller amounts of noise.

The projection approach, despite its merits, falls short in its generalizability. The algorithm presented in Chapter 7 only works for kernel functions obeying a certain symmetry, and additionally, performs poorly if the approximating dimensionality is low. Future work can alternatively consider some kind of private sampling approach to select a subset of interactions (or trajectories in RL) at random in order to reduce computational complexity. Alternatively, if it is known that the decision problem admits a certain sparsity structure, one can develop efficient approaches that can be provably efficient, e.g., as considered for single-agent bandits in Jamieson et al. (2015). We expect to see a similar synergy between computation and privacy in this regard as well.

Finally, we discuss the computational issues arising from the network structures present in decentralized decision-making. Several algorithms presented in earlier chapters (e.g., Chapter 10) require knowledge of the communication graph and specific partitions as well, e.g., the minimal clique cover or dominating set. These partitions are known to be NP-Hard to evaluate for arbitrary graphs (see Remark 10.3, for example). While we can indeed approximate them efficiently for several families of graphs, in applications where the number of agents is substantially large (e.g., comparable with $T$), it is prudent to deploy alternative communication protocols that do not rely on graph partitions. A potential approach to this problem is to construct a hierarchical arrangement of agents, where "server" agents are arranged in a decentralized network with a standard topology (e.g., a ring or star arrangement), and the remaining agents are connected directly to the servers, running a distributed optimization. This approach will potentially eliminate graph partitioning (or make it trivial), but will require a more nuanced communication protocol that allows for stochastic message-passing and asynchronous communication, as discussed next.

## 11.2 Asynchronous and Stochastic Communication

A notable restriction of the setting considered is that communication (both in the message-passing and distributed environments) occurs via *synchronization*, i.e., by ensuring that the arm pulls effectively follow the same "clock". While synchronous communication provides a cleaner and easier method to prove the efficiency of multi-agent algorithms, enabling synchronous communication in practice is not straightforward.

Asynchronous communication is long-known to be more efficient in distributed computation (Arjomandi et al., 1983), however the implementation of asynchronous communication in decision-making presents an additional challenge in proving regret bounds - it is non-trivial to bound the variation in arm pulls whenever the system clocks are not synchronized across agents. This is in contrast to federated optimization (Sprague et al., 2018; Chen et al., 2020b; Lu et al., 2019), where efficiency is measured relative to the optimal solution and one does not need exploration.

Therefore, extending approaches from the asynchronous optimization literature to decisionbmaking is a valuable line of pursuit. The primary hurdle in extending synchronization approaches directly to the asynchronous decision-making setting is that it is not straightforward to provide a control over the amount of exploration an agent does (i.e., number of sub-optimal pulls of an arm) without assuming stochasticity within the environment. For example, in the linear contextual bandit case, an assumption that would be sufficient is to assume that the contexts are drawn from a distribution $P$ such that $\lambda_{\min}\left(\mathbb{E}_{\mathbf{x}_t \sim P}[\mathbf{x}_t\mathbf{x}_t^\top | \mathcal{H}_{t-1}]\right) \geq \lambda_0$, i.e., the smallest eigenvalues of the context distribution are bounded away from zero with high probability. This ensures that even in the worst case, any individual agent would be guaranteed to explore the decision space sufficiently, ensuring a no-regret solution. Relaxing this assumption to adversarially-drawn contexts for asynchronous bandits is an interesting open problem.

As an alternative to asynchronous communication, one can consider the stochastic communication case, i.e., when the network $G$ is random and time-varying, such that the effective communication complexity decreases. While we have done a preliminary investigation in this direction in Madhushani et al. (2021), there are numerous open problems that can be addressed. As is in the case of asynchronous communication, it appears that the stochastic communication protocol also requires a "diversity" constraint to guarantee some exploration required for no-regret learning across agents, a constraint that is in sharp contrast to work on distributed optimization, where stochasticity can be used in synergy with optimization. Additionally, investigating the interactions with stochastic communication and decentralization is an interesting avenue as well. Particularly, one can intuit that leveraging fractional colorings of the communication graph might provide a mechanism to unify the network suboptimality gap present in deterministic communication.

## 11.3 Differential Privacy

There are several aspects worthy of inquiry in the context of differential privacy in federated online learning. Most notable of them is *local differential privacy*, where, in contrast to the (majority of) work presented in this thesis, we require that the privacy guarantee be valid for each outgoing message *separately*. While this may seem like a small change in terms of algorithm design, there are several open problems at the intersection of bandit learning and local differential privacy.

First, consider the local differential privacy rates obtained for Gaussian process and linear bandits in Chapter 7. We see that since each arm pull requires adding a new noise term in local privacy, the overall variance of noise added over $T$ rounds scales as $\mathcal{O}(T)$, which leads to a cumulative regret of $\mathcal{O}(T^{3/4})$. When the context distribution is stochastic, recent work has shown that this rate can be improved to the regular $\mathcal{O}(\sqrt{T})$ by leveraging the randomness of the contexts themselves to bypass explicit UCB-style bonuses (Han et al., 2021), however, it is still an open question whether this gap is necessary for *adversarially chosen* contexts. In follow-up work, we demonstrate that for "smoothed" adversarial context distributions, i.e., context distributions that are chosen adversarially but then are perturbed slightly (at random) by the environment (Kannan et al., 2018), once again, the well-known $\mathcal{O}(\sqrt{T})$ can be achieved. The case for deterministic adversarial contexts is yet to be solved.

In an alternate line of inquiry, one can argue that neither *joint* nor *local* differential privacy may not be the ideal privacy guarantee for federated environments, as the *joint* guarantee only provides privacy with respect to the individual sequence, and the *local* guarantee provides message-level privacy, which in fact is overkill for federated learning applications such as learning personalized recommender systems, where we would desire a guarantee that lies somewhere in between, i.e., that protects an entire user's history against adversaries, and not the complete sequence. Very recent work has investigated this in the context of federated optimization (Levy et al., 2021), it would be interesting to examine such a *user* level analysis for federated bandit systems in the future as well.

# Contributing Publications

The material in this thesis appears in full or in part in the following publications.

Dubey, A., & Pentland, A. (2020). Private and Byzantine-Proof Cooperative Decision-Making. In *Autonomous Agents and MultiAgent Systems* (pp. 357-365).

Dubey, A. & Pentland, A. (2020). Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning* (pp. 2730-2739). PMLR.

Dubey, A. & Pentland, A. (2020). Kernel methods for cooperative multi-agent contextual bandits. In *International Conference on Machine Learning* (pp. 2740-2750). PMLR.

Dubey, A., & Pentland, A. (2020). Differentially-Private Federated Linear Bandits. *Advances in Neural Information Processing Systems*, 33.

Dubey, A. (2021). No-Regret Algorithms for Private Gaussian Process Bandit Optimization. In *International Conference on Artificial Intelligence and Statistics* (pp. 2062-2070). PMLR.

Madhushani, U., Dubey, A., Leonard, N. & Pentland, A. (2021). Cooperative Bandits with Imperfect Communication. *Advances in Neural Information Processing Systems*, 34.

Dubey, A., & Pentland, A. (2021). Provably Efficient Cooperative Multi-Agent Reinforcement Learning with Function Approximation. arXiv preprint arXiv:2103.04972. Preliminary version in *Workshop on Communication-Efficient Distributed Optimization*.

Dubey, A., & Pentland, A. (2021). Provably Learning Pareto-Optimal Policies in Low-Rank Cooperative Markov Games. Preliminary version in *ICML Workshop on Reinforcement Learning Theory*.

# Bibliography

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

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

Agarwal, D., Basu, K., Ghosh, S., Xuan, Y., Yang, Y., and Zhang, L. Online parameter selection for web-based ranking problems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, pp. 23–32, 2018.

Agarwal, M., Ganguly, B., and Aggarwal, V. Communication efficient parallel reinforcement learning. *arXiv preprint arXiv:2102.10740*, 2021.

Agrawal, R. and Srikant, R. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 439–450, 2000.

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pp. 39–1, 2012.

Agrawal, S. and Goyal, N. Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pp. 99–107, 2013.

Agrawal, S., Juneja, S., and Glynn, P. Optimal delta-correct best-arm selection for heavy-tailed distributions. In *Algorithmic Learning Theory*, pp. 61–110. PMLR, 2020.

Al Mamunur Rashid, S. K. L., Karypis, G., and Riedl, J. Clustknn: a highly scalable hybrid model-& memory-based cf algorithm. 2006.

Albrecht, J. P. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

Altschuler, J., Brunel, V.-E., and Malek, A. Best arm identification for contaminated bandits. *Journal of Machine Learning Research*, 20(91):1–39, 2019.

Anantharam, V., Varaiya, P., and Walrand, J. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

Arjomandi, E., Fischer, M. J., and Lynch, N. A. Efficiency of synchronous versus asynchronous distributed systems. *Journal of the ACM (JACM)*, 30(3):449–456, 1983.

Åström, K. J. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Audibert, J.-Y., Bubeck, S., and Munos, R. Best arm identification in multi-armed bandits. In *COLT*, pp. 41–53, 2010.

Auer, P. and Chiang, C.-K. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pp. 116–120. PMLR, 2016.

Auer, P. and Ortner, R. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 49–56, 2007.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pp. 322–331. IEEE, 1995.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b.

Avron, H., Kapralov, M., Musco, C., Musco, C., Velingker, A., and Zandieh, A. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, pp. 253–262, 2017.

Awerbuch, B. and Kleinberg, R. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

Azuma, K. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

Bar-On, Y. and Mansour, Y. Individual regret in cooperative nonstochastic multi-armed bandits. *arXiv preprint arXiv:1907.03346*, 2019.

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

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Basu, D., Dimitrakakis, C., and Tossou, A. Differential privacy for multi-armed bandits: What is it and what is its cost?, 2020.

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

Bebensee, B. Local differential privacy: a tutorial. *arXiv preprint arXiv:1907.11908*, 2019.

Bellman, R. and Kalaba, R. E. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

Bertsekas, D. P. *Dynamic Programming: Determinist. and Stochast. Models*. Prentice-Hall, 1987.

Bistritz, I. and Leshem, A. Distributed multi-player bandits-a game of thrones approach. In *Advances in Neural Information Processing Systems*, pp. 7222–7232, 2018.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, pp. 2178–2186, 2011.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

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

Bochner, S. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108(1):378–410, 1933.

Boldrini, S., De Nardis, L., Caso, G., Le, M. T., Fiorina, J., and Di Benedetto, M.-G. mumab: A multi-armed bandit model for wireless network selection. *Algorithms*, 11(2):13, 2018.

Bollobás, B. and Riordan, O. M. Mathematical results on scale-free random graphs. *Handbook of graphs and networks: from the genome to the internet*, pp. 1–34, 2003.

Borak, S., Härdle, W., and Weron, R. Stable distributions. In *Statistical tools for finance and insurance*, pp. 21–44. Springer, 2005.

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

Bracha, G. and Toueg, S. Asynchronous consensus and broadcast protocols. *Journal of the ACM (JACM)*, 32(4):824–840, 1985.

Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

Brânzei, S. and Peres, Y. Multiplayer bandit learning, from competition to cooperation. *arXiv preprint arXiv:1908.01135*, 2019.

Briggs, C., Fan, Z., and Andras, P. A review of privacy-preserving federated learning for the internet-of-things. *Federated Learning Systems*, pp. 21–50, 2021.

Brisimi, T. S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I. C., and Shi, W. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Bubeck, S. *Bandits games and clustering foundations*. PhD thesis, 2010.

Bubeck, S. and Slivkins, A. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pp. 42–1. JMLR Workshop and Conference Proceedings, 2012.

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pp. 23–37. Springer, 2009.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

Bubeck, S., Li, Y., Peres, Y., and Sellke, M. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. *arXiv preprint arXiv:1904.12233*, 2019.

Buccapatnam, S., Eryilmaz, A., and Shroff, N. B. Stochastic bandits with side observations on networks. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, volume 42, pp. 289–300. ACM, 2014.

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

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

Burnetas, A. N. and Katehakis, M. N. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

Bush, R. R. and Mosteller, F. A stochastic model with applications to learning. *The Annals of Mathematical Statistics*, pp. 559–585, 1953.

Busoniu, L., Babuska, R., and De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Calandriello, D., Carratino, L., Lazaric, A., Valko, M., and Rosasco, L. Gaussian process optimization with adaptive sketching: Scalable and no regret. *arXiv preprint arXiv:1903.05594*, 2019.

Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W., and Shmatikov, V. " you might also like:" privacy risks of collaborative filtering. In *2011 IEEE symposium on security and privacy*, pp. 231–246. IEEE, 2011.

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

Cano, I., Weimer, M., Mahajan, D., Curino, C., and Fumarola, G. M. Towards geo-distributed machine learning. *arXiv preprint arXiv:1603.09035*, 2016.

Catoni, O. Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'IHP Probabilités et statistiques*, volume 48, pp. 1148–1185, 2012.

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

Cesa-Bianchi, N., Gentile, C., and Zappella, G. A gang of bandits. In *Advances in neural information processing systems*, pp. 737–745, 2013.

Cesa-Bianchi, N., Cesari, T. R., and Monteleoni, C. Cooperative online learning: Keeping your neighbors updated. *arXiv preprint arXiv:1901.08082*, 2019a.

Cesa-Bianchi, N., Gentile, C., and Mansour, Y. Delay and cooperation in nonstochastic bandits. *The Journal of Machine Learning Research*, 20(1):613–650, 2019b.

Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. In *International Colloquium on Automata, Languages, and Programming*, pp. 405–417. Springer, 2010.

Chan, T.-H. H., Shi, E., and Song, D. Privacy-preserving stream aggregation with fault tolerance. In *International Conference on Financial Cryptography and Data Security*, pp. 200–214. Springer, 2012.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Chen, S., Xue, D., Chuai, G., Yang, Q., and Liu, Q. Fl-qsar: a federated learning-based qsar prototype for collaborative drug discovery. *Bioinformatics*, 36(22-23):5492–5498, 2020a.

Chen, T., Jin, X., Sun, Y., and Yin, W. Vafl: a method of vertical asynchronous federated learning. *arXiv preprint arXiv:2007.06081*, 2020b.

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

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. *arXiv preprint arXiv:1704.00445*, pp. 844–853, 2017.

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

Christmann, A. and Steinwart, I. Universal kernels on non-standard input spaces. In *Advances in neural information processing systems*, pp. 406–414, 2010.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Clemente, A. V., Castejón, H. N., and Chandra, A. Efficient parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1705.04862*, 2017.

Costabal, F. S., Matsuno, K., Yao, J., Perdikaris, P., and Kuhl, E. Machine learning in drug development: Characterizing the effect of 30 drugs on the qt interval using gaussian process regression, sensitivity analysis, and uncertainty quantification. *Computer Methods in Applied Mechanics and Engineering*, 348:313–333, 2019.

Crescenzi, P., Kann, V., and Halldórsson, M. A compendium of np optimization problems, 1995.

Crovella, M. E., Taqqu, M. S., and Bestavros, A. Heavy-tailed probability distributions in the world wide web. *A practical guide to heavy tails*, 1:3–26, 1998.

Cummings, R. and Desai, D. The role of differential privacy in gdpr compliance. In *FAT'18: Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2018.

Dann, C., Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. On oracle-efficient pac rl with rich observations. *arXiv preprint arXiv:1803.00606*, 2018.

Dao, T., De Sa, C. M., and Ré, C. Gaussian quadrature for kernel features. In *Advances in neural information processing systems*, pp. 6107–6117, 2017.

De Vany, A. S., Lee, C., et al. *Information cascades in multi-agent models*. University of California, Irivine, Department of Economics, 1999.

Deshmukh, A. A., Dogan, U., and Scott, C. Multi-task learning for contextual bandits. *arXiv preprint arXiv:1705.08618*, pp. 4848–4856, 2017.

Devroye, L., Lerasle, M., Lugosi, G., and Oliveira, R. I. Sub-gaussian mean estimators. *The Annals of Statistics*, 44(6):2695–2725, 2016.

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

Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pp. 578–598. PMLR, 2021.

Dong, K., Peng, J., Wang, Y., and Zhou, Y. Root-n-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, pp. 1554–1557. PMLR, 2020.

Dubey, A. No-regret algorithms for private gaussian process bandit optimization. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2062–2070. PMLR, 13–15 Apr 2021. URL http://proceedings.mlr.press/v130/dubey21a.html.

Dubey, A. and Pentland, A. Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning*, pp. 2730–2739. PMLR, 2020a.

Dubey, A. and Pentland, A. Differentially-private federated linear bandits. *Neural Information Processing Systems (NeurIPS)*, 2020b.

Dubey, A. and Pentland, A. Kernel methods for cooperative multi-agent contextual bandits. In *International Conference on Machine Learning*, pp. 2740–2750. PMLR, 2020c.

Dubey, A. and Pentland, A. Private and byzantine-proof cooperative decision-making. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 357–365, 2020d.

Dubey, A. and Pentland, A. S. Thompson sampling on symmetric alpha-stable bandits. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5715–5721. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/792. URL https://doi.org/10.24963/ijcai.2019/792.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438. IEEE, 2013.

Dwork, C. Differential privacy. *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C. and Smith, A. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2):715–724, 2010.

Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60. IEEE, 2010.

Dwork, C., Smith, A., Steinke, T., and Ullman, J. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

Eom, Y.-H. and Jo, H.-H. Tail-scope: Using friends to estimate heavy tails of degree distributions in large-scale complex networks. *Scientific reports*, 5:09752, 2015.

Erdős, P. and Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1407–1416. PMLR, 2018.

Even-Dar, E., Mannor, S., Mansour, Y., and Mahadevan, S. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

Foster, D. J., Gentile, C., Mohri, M., and Zimmert, J. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Fraigniaud, P. Locality in distributed graph algorithms, 2016.

Garcelon, E., Perchet, V., Pike-Burke, C., and Pirotta, M. Local differentially private regret minimization in reinforcement learning. *arXiv preprint arXiv:2010.07778*, 2020.

Garivier, A. and Cappé, O. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pp. 359–376. JMLR Workshop and Conference Proceedings, 2011.

Gentile, C., Li, S., and Zappella, G. Online clustering of bandits. In *International Conference on Machine Learning*, pp. 757–765, 2014.

Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G., and Etrue, E. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1253–1262. JMLR. org, 2017.

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

Ghosh, A., Hong, J., Yin, D., and Ramchandran, K. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Glover, F. and Kochenberger, G. A tutorial on formulating qubo models. *arXiv preprint arXiv:1811.11538*, 2018.

Goddard, M. The eu general data protection regulation (gdpr): European regulation that has a global impact. *International Journal of Market Research*, 59(6):703–705, 2017.

Granter, S. R., Beck, A. H., and Papke Jr, D. J. Alphago, deep learning, and the future of the human microscopist. *Archives of pathology &amp; laboratory medicine*, 141(5):619–621, 2017.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

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

Grounds, M. and Kudenko, D. Parallel reinforcement learning with linear function approximation. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pp. 60–74. Springer, 2005.

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

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

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

Gündüz, D., de Kerret, P., Sidiropoulos, N. D., Gesbert, D., Murthy, C. R., and van der Schaar, M. Machine learning in the air. *IEEE Journal on Selected Areas in Communications*, 37(10):2184–2199, 2019.

Gupta, A., Koren, T., and Talwar, K. Better algorithms for stochastic bandits with adversarial corruptions. *arXiv preprint arXiv:1902.08647*, 2019.

Han, Y., Liang, Z., Wang, Y., and Zhang, J. Generalized linear bandits with local differential privacy. *arXiv preprint arXiv:2106.03365*, 2021.

Hannun, A., Knott, B., Sengupta, S., and van der Maaten, L. Privacy-preserving contextual bandits. *arXiv preprint arXiv:1910.05299*, 2019.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 61–70. IEEE, 2010.

He, J., Zhou, D., and Gu, Q. Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pp. 4171–4180. PMLR, 2021.

Hernandez-Campos, F., Marron, J., Samorodnitsky, G., and Smith, F. D. Variable heavy tails in internet traffic. *Performance Evaluation*, 58(2-3):261–284, 2004.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

Hildebrand, F. B. *Introduction to numerical analysis*. Courier Corporation, 1987.

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

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., Van Hasselt, H., and Silver, D. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Huber, P. J. A robust version of the probability ratio test. *The Annals of Mathematical Statistics*, pp. 1753–1758, 1965.

Huber, P. J. *Robust statistics*. Springer, 2011.

Iyengar, R., Near, J. P., Song, D., Thakkar, O., Thakurta, A., and Wang, L. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 299–316. IEEE, 2019.

Jain, P., Kothari, P., and Thakurta, A. Differentially private online learning. In *Conference on Learning Theory*, pp. 24–1, 2012.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Jamieson, K., Katariya, S., Deshpande, A., and Nowak, R. Sparse dueling bandits. In *Artificial Intelligence and Statistics*, pp. 416–424. PMLR, 2015.

Janz, D., Burt, D. R., and González, J. Bandit optimisation of functions in the mat\'ern kernel rkhs. *arXiv preprint arXiv:2001.10396*, 2020.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in neural information processing systems*, pp. 4863–4873, 2018.

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

Joulani, P., Gyorgy, A., and Szepesvári, C. Online learning under delayed feedback. In *International Conference on Machine Learning*, pp. 1453–1461, 2013.

Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. In *International conference on machine learning*, pp. 1376–1385. PMLR, 2015.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kandasamy, K., Krishnamurthy, A., Schneider, J., and Póczos, B. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 133–142. PMLR, 2018.

Kannan, S., Morgenstern, J., Roth, A., Waggoner, B., and Wu, Z. S. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. *arXiv preprint arXiv:1801.03423*, 2018.

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

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

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

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

Kocsis, L. and Szepesvári, C. Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293. Springer, 2006.

Kolla, R. K., Jagannathan, K., and Gopalan, A. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795, 2018.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Konovalov, A. Information cascades and experts institutions. *New Institutional Economics: Research Methods and Research Methods and Tools*, pp. 66, 2010.

Korda, N., Kaufmann, E., and Munos, R. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems*, pp. 1448–1456, 2013.

Korda, N., Szörényi, B., and Shuai, L. Distributed clustering of linear bandits in peer to peer networks. In *Journal of machine learning research workshop and conference proceedings*, volume 48, pp. 1301–1309. International Machine Learning Society, 2016.

Krause, A. and Ong, C. S. Contextual gaussian process bandit optimization. In *Advances in neural information processing systems*, pp. 2447–2455, 2011.

Kretchmar, R. M. Parallel reinforcement learning. Citeseer, 2002.

Krishnamurthy, A., Agarwal, A., and Langford, J. Pac reinforcement learning with rich observations. *arXiv preprint arXiv:1602.02722*, 2016.

Kusner, M., Gardner, J., Garnett, R., and Weinberger, K. Differentially private bayesian optimization. In *International conference on machine learning*, pp. 918–927, 2015.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Landgren, P., Srivastava, V., and Leonard, N. E. On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference (ECC)*, pp. 243–248. IEEE, 2016a.

Landgren, P., Srivastava, V., and Leonard, N. E. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 167–172. IEEE, 2016b.

Landgren, P., Srivastava, V., and Leonard, N. E. Social imitation in cooperative multiarmed bandits: Partition-based algorithms with strictly local information. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 5239–5244. IEEE, 2018.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Lattimore, T., Szepesvari, C., and Weisz, G. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pp. 5662–5670. PMLR, 2020.

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

Leskovec, J. and Sosič, R. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1–20, 2016.

Letham, B. and Bakshy, E. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019.

Levy, D., Sun, Z., Amin, K., Kale, S., Kulesza, A., Mohri, M., and Suresh, A. T. Learning with user-level privacy, 2021.

Lévy, P. Calcul des probabilités, vol. 9. *Gauthier-Villars Paris*, 1925.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Li, S. and Zhang, S. Online clustering of contextual cascading bandits. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Li, S., Karatzoglou, A., and Gentile, C. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 539–548, 2016.

Li, S., Chen, W., and Leung, K.-S. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Linial, N. Locality in distributed graph algorithms. *SIAM Journal on computing*, 21(1):193–201, 1992.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Liu, K. and Zhao, Q. Decentralized multi-armed bandit with multiple distributed players. In *2010 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2010a.

Liu, K. and Zhao, Q. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010b.

Liu, K. and Zhao, Q. Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 58, pp. 3010–3013. IEEE, IEEE, 2010c.

Liu, X. and Guillas, S. Dimension reduction for gaussian process emulation: An application to the influence of bathymetry on tsunami heights. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):787–812, 2017.

Lu, Y., Huang, X., Dai, Y., Maharjan, S., and Zhang, Y. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics*, 16(3):2134–2143, 2019.

Lucas, A. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

Lugosi, G. and Mendelson, S. Robust multivariate mean estimation: the optimality of trimmed mean. *arXiv preprint arXiv:1907.11391*, 2019.

Lykouris, T., Mirrokni, V., and Paes Leme, R. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 114–122, 2018.

Madhushani, U., Dubey, A., Leonard, N. E., and Pentland, A. Cooperative bandits with imperfect communication. *Advances in Neural Information Processing Systems 34*, 2021.

Madiman, M. On the entropy of sums. In *2008 IEEE Information Theory Workshop*, pp. 303–307. IEEE, 2008.

Malekzadeh, M., Athanasakis, D., Haddadi, H., and Livshits, B. Privacy-preserving bandits. *arXiv preprint arXiv:1909.04421*, 2019.

Mannor, S. and Tsitsiklis, J. N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

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

Martínez-Rubio, D., Kanade, V., and Rebeschini, P. Decentralized cooperative stochastic multi-armed bandits. In *Advances in Neural Information Processing Systems*, 2019.

Mary, J., Gaudel, R., and Preux, P. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pp. 325–336. Springer, 2015.

Medina, A. M. and Yang, S. No-regret algorithms for heavy-tailed linear bandits. In *International Conference on Machine Learning*, pp. 1642–1650, 2016.

Meeds, E. and Welling, M. Gps-abc: Gaussian process surrogate approximate bayesian computation. *arXiv preprint arXiv:1401.2838*, 2014.

Melo, F. S. and Ribeiro, M. I. Q-learning with linear function approximation. In *International Conference on Computational Learning Theory*, pp. 308–322. Springer, 2007.

Mishra, N. and Thakurta, A. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 592–601. AUAI Press, 2015.

Močkus, J. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pp. 400–404. Springer, 1975.

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

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

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

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems*, pp. 9147–9156, 2018.

Mutny, M. and Krause, A. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems*, pp. 9005–9016, 2018.

Myerson, R. B. Optimal coordination mechanisms in generalized principal–agent problems. *Journal of mathematical economics*, 10(1):67–81, 1982.

Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

Neu, G., Antos, A., György, A., and Szepesvári, C. Online markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2010.

Nikfar, B. and Vinck, A. H. Relay selection in cooperative power line communication: A multi-armed bandit approach. *Journal of Communications and Networks*, 19(1):1–9, 2017.

Ono, H. and Takahashi, T. Locally private distributed reinforcement learning. *arXiv preprint arXiv:2001.11718*, 2020.

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

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

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

Park, M., Nassar, M., and Vikalo, H. Bayesian active learning for drug combinations. *IEEE transactions on biomedical engineering*, 60(11):3248–3255, 2013.

Peterson, K., Rudovic, O., Guerrero, R., and Picard, R. W. Personalized gaussian processes for future prediction of alzheimer's disease progression. *arXiv preprint arXiv:1712.00181*, 2017.

Pichai, S. Privacy should not be a luxury good. *The New York Times*, 2019.

Pike-Burke, C., Agrawal, S., Szepesvari, C., and Grunewalder, S. Bandits with delayed, aggregated anonymous feedback. *arXiv preprint arXiv:1709.06853*, 2017.

Prasad, A., Balakrishnan, S., and Ravikumar, P. A unified approach to robust mean estimation. *arXiv preprint arXiv:1907.00927*, 2019.

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

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

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Rivest, R. L., Adleman, L., Dertouzos, M. L., et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

Rohde, D., Bonner, S., Dunlop, T., Vasile, F., and Karatzoglou, A. Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. *arXiv preprint arXiv:1808.00720*, 2018.

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

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Savazzi, S., Nicoli, M., Bennis, M., Kianoush, S., and Barbieri, L. Opportunities of federated learning in connected, cooperative, and automated industrial systems. *IEEE Communications Magazine*, 59(2):16–21, 2021.

Scarlett, J., Bogunovic, I., and Cevher, V. Lower bounds on regret for noisy gaussian process bandit optimization. *arXiv preprint arXiv:1706.00090*, 2017.

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

Schake, K. On the kronecker product. pp. 1==54, 2004.

Schölkopf, B. and Smola, A. Support vector machines and kernel algorithms. In *Encyclopedia of Biostatistics*, pp. 5328–5335. Wiley, 2005.

Seldin, Y. and Lugosi, G. An improved parametrization and analysis of the exp3++ algorithm for stochastic and adversarial bandits. In *Conference on Learning Theory*, pp. 1743–1759. PMLR, 2017.

Seldin, Y. and Slivkins, A. One practical algorithm for both stochastic and adversarial bandits. In *ICML*, pp. 1287–1295, 2014.

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

Shahrampour, S., Rakhlin, A., and Jadbabaie, A. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2786–2790. IEEE, 2017.

Shao, H., Yu, X., King, I., and Lyu, M. R. Almost optimal algorithms for linear stochastic bandits with heavy-tailed payoffs. In *Advances in Neural Information Processing Systems*, pp. 8430–8439, 2018.

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

Shariff, R. and Sheffet, O. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pp. 4296–4306, 2018.

Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R. R., et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Smith, M. T., Zwiessele, M., and Lawrence, N. D. Differentially private gaussian processes. *arXiv preprint arXiv:1606.00720*, 2016.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pp. 13–31. Springer, 2007.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.

Sprague, M. R., Jalalirad, A., Scavuzzo, M., Capota, C., Neun, M., Do, L., and Kopp, M. Asynchronous federated learning for geospatial applications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 21–28. Springer, 2018.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 1015–1022, 2010.

Sucar, L. E. Parallel markov decision processes. In *Advances in Probabilistic Graphical Models*, pp. 295–309. Springer, 2007.

Sui, Y., Yue, Y., and Burdick, J. W. Correlational dueling bandits with application to clinical treatment in large decision spaces. *arXiv preprint arXiv:1707.02375*, 2017.

Suomela, J. Survey of local algorithms. *ACM Computing Surveys (CSUR)*, 45(2):24, 2013.

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

Szepesvári, C. and Littman, M. L. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation*, 11(8):2017–2060, 1999.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. 1993.

Tankard, C. What the gdpr means for businesses. *Network Security*, 2016(6):5–8, 2016.

Tewari, A. and Murphy, S. A. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, pp. 495–517. Springer, 2017.

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

Thakurta, A. G. and Smith, A. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pp. 2733–2741, 2013.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Tossou, A. and Dimitrakakis, C. Algorithms for differentially private multi-armed bandits. *arXiv preprint arXiv:1511.08681*, 2015a.

Tossou, A. C. and Dimitrakakis, C. Differentially private, multi-agent multi-armed bandits. In *European Workshop on Reinforcement Learning (EWRL)*, 2015b.

Tossou, A. C. and Dimitrakakis, C. Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Turán, P. On an external problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941.

Vaid, A., Jaladanki, S. K., Xu, J., Teng, S., Kumar, A., Lee, S., Somani, S., Paranjpe, I., De Freitas, J. K., Wanyan, T., et al. Federated learning of electronic health records improves mortality prediction in patients hospitalized with covid-19. *medRxiv*, 2020.

Vakili, S., Liu, K., and Zhao, Q. Deterministic sequencing of exploration and exploitation for multi-armed bandit problems. *IEEE Journal of Selected Topics in Signal Processing*, 7(5): 759–767, 2013.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

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

Vanchinathan, H. P., Nikolic, I., De Bona, F., and Krause, A. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 225–232, 2014.

Vernade, C., Carpentier, A., Zappella, G., Ermis, B., and Brueckner, M. Contextual bandits under delayed feedback. *arXiv preprint arXiv:1807.02089*, 2018.

Vietri, G., Balle, B., Krishnamurthy, A., and Wu, Z. S. Private reinforcement learning with pac and regret guarantees. *arXiv preprint arXiv:2009.09052*, 2020a.

Vietri, G., Balle, B., Krishnamurthy, A., and Wu, Z. S. Private reinforcement learning with pac and regret guarantees. *arXiv preprint arXiv:2009.09052*, 2020b.

Wai, H.-T., Yang, Z., Wang, Z., and Hong, M. Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*, pp. 9649–9660, 2018.

Wang, B. and Hegde, N. Privacy-preserving q-learning with functional noise in continuous state spaces. *arXiv preprint arXiv:1901.10634*, 2019.

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

Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. A principled approach to data valuation for federated learning. In *Federated Learning*, pp. 153–167. Springer, 2020b.

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

Wang, Y., Hu, J., Chen, X., and Wang, L. Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309*, 2019a.

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

Wang, Y., Hu, J., Chen, X., and Wang, L. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2020c. URL https://openreview.net/forum?id=SJxZnR4YvB.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Weinberger, M. J. and Ordentlich, E. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

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

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Williams, R. J. and Baird, L. C. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Citeseer, 1993.

Xia, Y., Qin, T., Ma, W., Yu, N., and Liu, T.-Y. Budgeted multi-armed bandits with multiple plays.

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

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

Yang, M., Lyu, L., Zhao, J., Zhu, T., and Lam, K.-Y. Local differential privacy and its applications: A comprehensive survey. *arXiv preprint arXiv:2008.03686*, 2020a.

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

Yao, A. C. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pp. 160–164. IEEE, 1982.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018.

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

Yu, F., Zhang, W., Qin, Z., Xu, Z., Wang, D., Liu, C., Tian, Z., and Chen, X. Heterogeneous federated learning. *arXiv preprint arXiv:2008.06767*, 2020a.

Yu, S., Chen, X., Zhou, Z., Gong, X., and Wu, D. When deep reinforcement learning meets federated learning: Intelligent multi-timescale resource management for multi-access edge computing in 5g ultra dense network. *IEEE Internet of Things Journal*, 2020b.

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

Yu, X., Shao, H., Lyu, M. R., and King, I. Pure exploration of multi-armed bandits with heavy-tailed payoffs. In *Conference on Uncertainty in Artificial Intelligence*, pp. 937–946, 2018.

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

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

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

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

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

Zhou, W., Li, J., Yang, Y., and Shah, F. Leverage side information for top-n recommendation with latent gaussian process. *Concurrency and Computation: Practice and Experience*, pp. e5534, 2019.

Zhuang, V. and Sui, Y. No-regret reinforcement learning with heavy-tailed rewards. *arXiv preprint arXiv:2102.12769*, 2021.

Zhuo, H. H., Feng, W., Xu, Q., Yang, Q., and Lin, Y. Federated reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.

ZI, Y. Schur complements and determinant inequalities.

Zimmert, J. and Seldin, Y. An optimal algorithm for adversarial bandits with arbitrary delays. In *International Conference on Artificial Intelligence and Statistics*, pp. 3285–3294. PMLR, 2020.