DISCRETE AND CONTINUOUS OPTIMIZATION FOR COLLABORATIVE AND MULTI-TASK LEARNING

Arman Adibi

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2023

Supervisor of Dissertation

Hamed Hassani, Assistant Professor of Electrical and Systems Engineering

Graduate Group Chairperson

Troy Olsson, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

George J Pappas (Chair), Professor of Electrical and Systems Engineering, University of Pennsylvania
Amin Karbasi, Associate Professor of Electrical Engineering and Computer Science, Yale University
Sanjay Shakkottai, Professor of Electrical and Computer Engineering, University of Texas at Austin

DISCRETE AND CONTINUOUS OPTIMIZATION FOR COLLABORATIVE AND MULTI-TASK LEARNING

To My Family.

ACKNOWLEDGEMENTS

I consider myself incredibly fortunate to have been given the opportunity to pursue my Ph.D. degree at the esteemed University of Pennsylvania. Above all, I want to express my deep appreciation to Prof. Hamed Hassani, my advisor, whose unwavering guidance and mentorship have been invaluable throughout this remarkable journey. He consistently engaged in thought-provoking discussions and collaborative problem-solving sessions, showcasing not only his brilliance but also his kindness and dedication to his students. His support extended beyond research, as he was always there to assist with any challenges I faced. Prof. Hassani is not only an advisor but also one of my closest and most cherished friends. I am truly grateful for his presence in my academic and personal life.

I would also like to extend my heartfelt thanks to my esteemed committee members, Prof. Amin Karbasi, Prof. Sanjay Shakkottai, and Prof. George J. Pappas, for their valuable insights, constructive criticism, and unwavering encouragement. Their expertise and feedback greatly contributed to the development of my work. Additionally, I am immensely grateful to Prof. Aryan Mokhtari, Prof. George J. Pappas, and Prof. Aritra Mitra, with whom I collaborated extensively on my research projects. Their comments, suggestions, and unwavering support have been crucial to my academic growth. Working closely with Prof. Aritra Mitra during the final two years of my Ph.D. studies was a truly rewarding experience. His enthusiasm, clarity of thought, and academic guidance have been immensely valuable. I would like to express my special appreciation to Aritra Mitra for his invaluable feedback on all sections of this thesis. I am also thankful to Prof. Aryan Mokhtari for our collaboration during the initial years of my Ph.D. studies. Furthermore, I extend my gratitude to Nicolò Del Fabro, Alex Robey, and Mohammad Freydounian for their friendship, invaluable assistance, and insightful suggestions, which played a pivotal role in making this thesis possible.

I wish to acknowledge the dedicated teachers who have influenced my academic journey at various stages, consistently inspiring my growth. In particular, I want to express my gratitude to Prof. Mohammad Mahdi Naghsh, Prof. M. Reza Koushesh, Prof. Reza Rezaeian Farashahi, and Prof. Farzad Parvaresh for their unwavering support and encouragement. Their guidance has shaped my

academic path significantly.

Throughout these transformative years, I have been fortunate to collaborate with and build meaningful relationships with an exceptional group of individuals, including Alex Robey, Vahid Nikkhah, Mohammad Freydounian, Mohamad Hossein Idjadi, Ehsan Nahvi, Juan Cervino, Mehran Ebrahimian, Aryan Mokhtari, Saeed Sharifi-Malvajerdi, Behrad Moniri, Eric Lei, Hesam Nikpey, Raghu Arghal, Nicolò Del Fabro, Mahdi Sabbaghi, Shayan Kiyani, Zebang Shen, Nikolaos Boussias, Sima Noorani, Alena Rodionova, Ignacio Hounie, Aritra Mitra, Juan Elenter, Asma Fallah, Mostafa Akbari, Leila Bahrami, Fariborz Soroush, Burce Lee, Thomas Zhang and many others. Their collaboration and friendship have enriched my academic and personal experiences. I would also like to express my sincere appreciation to my friends and colleagues in the labs of Dr. Hassani, Dr. Ribeiro, and Dr. Pappas, whose presence has made my time at UPenn truly memorable.

Lastly, I owe my deepest gratitude to my wife, Sabiha, whose unwavering love and support have been my anchor throughout this challenging journey. I also want to acknowledge the profound influence of my late father, Behrouz, whose wise advice and tireless work ethic continue to inspire me. I am grateful to my mother, Mina, for her unwavering support and affection, even from a distance. To my brother, Armin, who has been a constant source of friendship and support, I am truly thankful. Completing this transformative journey would have been impossible without the unconditional love and support of my family, and I dedicate this thesis to them.

ABSTRACT

DISCRETE AND CONTINUOUS OPTIMIZATION FOR COLLABORATIVE AND MULTI-TASK LEARNING

Arman Adibi

Hamed Hassani

This thesis is dedicated to addressing the challenges of robust collaborative learning and optimization in both discrete and continuous domains. With the ever-increasing scale of data and the growing demand for effective distributed learning, a multitude of obstacles emerge, including communication limitations, resilience to failures and corrupted data, limited information access, and collaboration in multi-task learning scenarios. The thesis consists of eight chapters, each targeting specific aspects of these challenges.

In the second chapter, novel algorithms are introduced for collaborative linear bandits, offering a comprehensive exploration of the benefits of collaboration in the presence of adversaries through thorough analyses and lower bounds. The third chapter delves into multi-agent min-max learning problems by tackling the presence of Byzantine adversarial agents. Chapter four delves into the effects of delays within stochastic approximation schemes, investigating non-asymptotic convergence rates under Markovian noise.

Moving forward, the fifth chapter focuses on analyzing the performance of standard min-max optimization algorithms with delayed updates. The sixth chapter concentrates on robustness in discrete learning, specifically addressing convex-submodular problems in mixed continuous-discrete domains. The seventh chapter tackles the challenge of limited information access in collaborative problems with distributed constraints, developing optimal algorithms for submodular maximization under distributed partition matroid constraints.

Lastly, the eighth chapter introduces a discrete variant of multi-task learning and meta-learning. In summary, this thesis contributes to the field of robust collaborative learning and decision-making by

providing insights, algorithms, and theoretical guarantees in discrete and continuous optimization. The advancements made across linear bandits, minimax optimization, distributed robust learning, delayed optimization, and submodular maximization pave the way for future developments in collaborative and multi-task learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1

Introduction

This thesis focuses on the challenges of robust collaborative learning and optimization in the context of both discrete and continuous domains. With the increasing scale of data and the need for effective distributed learning, numerous obstacles arise, including communication limitations, resilience to failures and corrupted data, limited information access, and collaboration in multi-task learning scenarios. The thesis comprises eight chapters, each addressing specific aspects of these challenges.

The **second chapter** introduces novel algorithms for collaborative linear bandits, striking a balance between collaboration and adversarial disruptions. Leveraging robust confidence intervals, these algorithms optimize exploration and exploitation, achieving regret bounds that approach optimality. The benefits of collaboration in the presence of adversaries are explored through comprehensive analyses and lower bounds, providing a deep understanding of collaborative linear bandits under adversarial conditions.

In the **third chapter**, the thesis tackles multi-agent min-max learning problems by addressing the presence of Byzantine adversarial agents. A robust distributed variant of the Extra-gradient algorithm is proposed within the domain of distributed robust learning. By leveraging robust statistics, this algorithm achieves convergence rates that are close to optimal, enabling the approximation of saddle points for smooth convex-concave and smooth strongly convex-strongly concave functions. The subsequent chapters delve into the challenges posed by delays in optimization.

The **fourth chapter** investigates the effects of delays within stochastic approximation schemes, examining non-asymptotic convergence rates under Markovian noise. Delay-adaptive schemes are introduced to enhance convergence rates without explicit knowledge of the delay sequence.

In the **fifth chapter**, the thesis analyzes the performance of standard min-max optimization algorithms with delayed updates, specifically addressing stochastic optimization with delayed gradients. The convergence guarantees of Gradient Descent-Ascent (GDA) and Extra-gradient (EG) algorithms

with delayed updates are established for convex-concave and strongly convex-strongly concave settings, shedding light on the convergence slowdown induced by delays.

The **sixth chapter** focuses on robustness in discrete learning, particularly convex-submodular problems in mixed continuous-discrete domains. The thesis designs algorithmic procedures to address these challenges, providing convergence rates and characterizations of computational complexity. By combining tools from discrete and continuous optimization, the proposed algorithms offer effective solutions supported by theoretical guarantees and empirical evaluations.

The **seventh chapter** tackles the issue of limited information access in collaborative problems with distributed constraints. Optimal algorithms for submodular maximization with distributed constraints are developed, addressing the challenges of maximizing a submodular objective function subject to a distributed partition matroid constraint. The thesis introduces the Constraint-Distributed Continuous Greedy (CDCG) algorithm, which achieves a tight approximation factor of the optimum global solution through local computation and communication. Empirical results demonstrate the superiority of CDCG over sequential greedy methods in a multi-agent area coverage problem.

Finally, the **eighth chapter** introduces a discrete variant of multi-task learning and meta-learning. A novel meta-learning framework is proposed in the discrete domain, where each task corresponds to maximizing a set function under a cardinality constraint. The framework leverages prior data to train a suitable initial solution set, facilitating quick adaptation to new tasks at a reduced computational cost. Deterministic and randomized algorithms are presented to solve the challenging discrete optimization problem, with strong theoretical guarantees, even when the training objective may not be submodular. The effectiveness of the framework is demonstrated in real-world problem instances, highlighting the significant reduction in computational complexity for solving new tasks while incurring minimal performance loss.

In summary, this thesis contributes to addressing the challenges of robust collaborative learning and decision-making by focusing on discrete and continuous optimization approaches. The insights, algorithms, and theoretical guarantees provided across linear bandits, minimax optimization, distributed

robust learning, delayed optimization, and submodular maximization advance the field and pave the way for further advancements in collaborative and multi-task learning.

CHAPTER 2

Collaborative Linear Bandits with Adversarial Agents

2.1. Introduction

One of the primary challenges in modern large-scale computing systems is that of *security*. Given that the individual agents in such large systems are often vulnerable to attacks, it is important to understand how the overall system behaves in the face of adversarial corruptions. This observation has spurred a line of research dedicated to the design and analysis of distributed algorithms that are provably robust to a small fraction of adversarial agents; notably, motivated by emerging learning paradigms such as federated learning (Konečnỳ et al., 2016; Bonawitz et al., 2019; McMahan et al., 2017), this body of work has focused primarily on empirical risk minimization/stochastic optimization (Chen et al., 2017b; Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2018b; Alistarh et al., 2018; Xie et al., 2018; Li et al., 2019a; Ghosh et al., 2019, 2020a,b; Karimireddy et al., 2021). However, when it comes to multi-agent sequential decision-making problems under uncertainty (e.g., bandits and reinforcement learning), our understanding of analogous questions is quite limited. Our goal in this paper is to bridge the above gap by studying a collaborative linear bandit (Dani et al., 2008; Abbasi-Yadkori et al., 2011) problem in the presence of adversaries. In our model, M agents interact with the same linear bandit characterized by a d-dimensional unknown parameter θ_*, and a finite set of K arms. These agents can collaborate via a central server to improve performance, as measured by cumulative regret. As examples, consider (i) a team of robots exploring actions (arms) in a common environment and interacting with a central controller; and (ii) a group of people exploring restaurants (arms) and writing reviews for a web-recommendation server. In the absence of adversaries, there is a clear reason to collaborate in either case: by exchanging information, each agent can reduce its uncertainty about the arms faster than it could when it acts alone, and thereby incur lesser regret. The situation becomes murkier and more delicate when certain agents *misbehave*: What if certain robots get attacked or certain people deliberately write spam reviews? More generally, the main question we ask in this paper is the following.

In a multi-agent linear stochastic bandit problem, can we still hope for benefits of collaboration when a fraction α of the agents are adversarial? If so, what are the fundamental limits of such benefits?

As far as we are aware, the answers to these questions have thus far remained elusive, motivating our current study. The main technical hurdle we must overcome is to delicately balance the exploration-exploitation trade-off in the presence of both statistical uncertainties due to the environment, and *worst-case* adversarial behavior. Importantly, the above trade-off - intrinsic to sequential decision-making - is absent in static optimization problems. Thus, the ideas used to guarantee robustness for distributed optimization do not apply to our problem, making our task quite non-trivial.

Our Contributions. In this paper, we contribute to a principled study of several canonical *structured* linear bandit settings with adversarial agents. Our specific contributions are as follows.

• **Robust Collaborative Linear Bandit Algorithm.** We propose `RCLB` - a phased elimination algorithm that relies on distributed exploration, and balances the exploration-exploitation dilemma in the presence of adversaries via carefully constructed *robust confidence intervals*. We prove that `RCLB` guarantees $\tilde{O}\left(\left(\alpha + 1/\sqrt{M}\right)\sqrt{dT}\right)$ regret for each good agent; see Theorem 1. This result is both novel, and significant in that it reveals a clear benefit of collaboration (despite adversaries) for small values of α. In particular, when $\alpha = 0$, the regret bound of `RCLB` is minimax-optimal in all relevant parameters: the model-dimension d, the horizon T, and the number of agents M.

• **Fundamental Limits.** At this stage, one may ask: *Is the additive $\alpha\sqrt{T}$ term in Theorem 1 simply an artifact of our analysis?* In Theorem 2, we establish a fundamental lower bound, revealing that such a term is in fact unavoidable; it is the price one must pay due to adversarial corruptions. A key implication of this result is that *our work is the first to provide tight, near-optimal regret bounds for collaborative linear bandits with adversaries.* The proof of Theorem 2 relies on a novel connection between the information-theoretic arguments in (Bubeck et al., 2013), and ideas from the robust mean estimation literature (Chen et al., 2015; Lai et al., 2016). As such, our proof technique may be relevant for related settings.

In our next set of contributions, we significantly extend our results to more general bandit models.

• **Generalized Linear Bandit Setting.** In Theorem 3, we prove that one can achieve bounds akin to that in Theorem 1 for the generalized linear bandit model (GLM) (Filippi et al., 2010; Li et al., 2017) that accounts for non-linear observation maps. To achieve this result, we propose a variant of `RCLB` that leverages very recently developed tools from high-dimensional robust Gaussian mean estimation (Dalalyan and Minasyan, 2022). Deriving robust confidence intervals for this setting requires some work: we exploit regularity properties of the non-linear observation model along with error bounds from (Dalalyan and Minasyan, 2022) for this purpose. As far as we are aware, Theorem 3 is the first result to establish adversarial-robustness for GLMs, allowing our framework to be applicable to a broad class of problems (e.g., logistic and probit regression models).

• **Contextual Bandit Setting.** Finally, we turn our attention to the contextual bandit setting where the feature vectors of the arms can change over time. This setting is practically quite relevant as web-recommendation systems are often modeled as contextual bandits (Li et al., 2010). The main challenge here arises from the need to simultaneously contend with time-varying optimal arms and adversaries. To handle this scenario, we develop a robust variant of the `SUPLINREL` algorithm (Auer, 2002) that guarantees a near-optimal regret bound identical to that of Theorem 1; see Theorem 4.

Overall, via the proposal of new robust algorithms complemented with tight analyses, our work takes an important step towards multi-agent sequential decision-making in the presence of adversaries.

Related Work. There is a growing strand of literature that studies the effect of reward corruption in stochastic bandits (for a single-agent setting) where the adversary has a fixed corruption budget (Jun et al., 2018; Liu and Shroff, 2019; Lykouris et al., 2018; Gupta et al., 2019; Bogunovic et al., 2020; Garcelon et al., 2020; Bogunovic et al., 2021; He et al., 2022). The techniques in these papers do not apply to our work as our setting is very different: the adversarial agents in our model can act *arbitrarily*, and have no budget constraints. Several papers study multi-agent bandit problems in the absence of adversaries (Liu and Zhao, 2010; Kalathil et al., 2014; Kar et al., 2011; Landgren et al., 2016, 2021; Shahrampour et al., 2017; Buccapatnam et al., 2015; Kolla et al., 2018; Wang et al., 2019; Sankararaman et al., 2019; Martínez-Rubio et al., 2018; Dubey et al., 2020; Dubey and Pentland, 2020b; Lalitha and Goldsmith, 2020; Chawla et al., 2020a,b; Ghosh et al., 2021; Agarwal et al., 2021;

Zhu et al., 2021; Shi et al., 2021). A few very recent ones (Dubey and Pentland, 2020a; Vial et al., 2021, 2022; Mitra et al., 2021a) also look at the effect of attacks, but for the simpler unstructured multi-armed bandit problem. Accounting for adversarial agents in the structured linear bandit setting we consider here requires significantly different techniques that we develop in this paper.

2.2. Problem Formulation

We consider a setting comprising of a central server and M agents; the agents can communicate only via the server. Each agent $i \in [M]$ interacts with the same linear bandit model characterized by an unknown parameter θ_* that belongs to a known compact set $\Theta \subset \mathbb{R}^d$. We assume $\|\theta_*\| \leq 1$.[1] The set of actions $\mathcal{A}$ for each agent is given by K distinct vectors in $\mathbb{R}^d$, i.e., $\mathcal{A} = \{a_1, \ldots, a_K\}$, where K is a finite, positive integer. Based on all the information acquired by an agent i up to time $t-1$, it takes an action $a_{i,t} \in \mathcal{A}$ at time t, and receives a reward $y_{i,t}$ given by the following observation model:

$$y_{i,t} = \langle \theta_*, a_{i,t} \rangle + \eta_{i,t}. \tag{2.1}$$

Here, $\{\eta_{i,t}\}$ is a sequence of independent Gaussian random variables with zero mean and unit variance. Thus far, we have essentially described a distributed/multi-agent linear stochastic bandit model. Departing from this standard model, we focus on a setting where a fraction $\alpha \in [0, 1/2)$ of the agents are adversarial; the adversarial set is denoted by $\mathcal{B}$, where $|\mathcal{B}| = \alpha M$. In particular, we consider a *worst-case* attack model, where each adversarial agent $i \in \mathcal{B}$ is assumed to have complete knowledge of the system, and is allowed to act arbitrarily. Under this attack model, an agent $i \in \mathcal{B}$ can transmit arbitrarily corrupted messages to the central server.

Our performance measure of interest is the following group regret metric R_T defined w.r.t. the non-adversarial agents:

$$R_T = \mathbb{E}\left[\sum_{i \in [M] \setminus \mathcal{B}} \sum_{t=1}^{T} \langle \theta_*, a_* - a_{i,t} \rangle \right], \tag{2.2}$$

where $a_* = \arg\max_{a \in \mathcal{A}} \langle \theta_*, a \rangle$ is the optimal arm, and T is the time horizon.[2] We will work under

[1] We will use $\|\cdot\|$ to represent the Euclidean norm, and a' to denote the transpose of a vector a.

[2] For ease of exposition, we assume that there is an unique optimal arm.

a regime where the horizon T is large, satisfying $T \geq Md$. The goal of the good (non-adversarial) agents is to collaborate via the server and play a sequence of actions that minimize the group regret R_T. Let us now make a few key observations. In principle, each good agent can choose to act independently throughout (i.e., not talk to the server at all), and achieve $\tilde{O}(\sqrt{dT})$ regret by playing a standard bandit algorithm. Clearly, the group regret R_T would scale as $\tilde{O}\left((1-\alpha)M\sqrt{dT}\right)$ in such a case. In the absence of adversaries however, one can achieve a significantly better group regret bound of $\tilde{O}\left(\sqrt{MdT}\right)$ via collaboration, i.e., the regret per good agent can be reduced by a factor of $\sqrt{M}$ relative to the case when it acts independently (see, for instance, (Wang et al., 2019) and (Dubey and Pentland, 2020b)). Our specific interest in this paper is to investigate whether, and to what extent, one can retain the benefits of collaboration despite the worst-case attack model described above. Said differently, we ask: *Can we improve upon the trivial per agent regret bound of* $\tilde{O}(\sqrt{dT})$ *in the presence of adversaries?*

Throughout the rest of the paper, we will answer the above question in the affirmative by deriving novel robust algorithms for several canonical bandit models, and then establishing near-optimal regret bounds for each such model.

2.3. Robust Collaborative Phased Elimination Algorithm for Linear Bandits

In this section, we develop a robust phased elimination algorithm that achieves the near-optimal regret bound of $\tilde{O}\left(\left(\alpha + \sqrt{1/M}\right)\sqrt{dT}\right)$ per good agent. This is non-trivial as we must account for the worst-case attack model described in Section 4.4. To highlight the challenges that we need to overcome, consider the following scenario. During the initial stages of the learning process, when the arms in $\mathcal{A}$ have not been adequately sampled by the agents, even a good agent may have "poor estimates" of the true payoffs associated with each arm, i.e., the variance associated with such estimates may be large. This statistical uncertainty can be exploited by the adversarial agents to their benefit. In particular, we need to devise an approach that can distinguish between benign stochastic perturbations (due to the noise in our model) and deliberate adversarial behavior. In what follows, we describe our proposed algorithm - `Robust Collaborative Phased Elimination for Linear Bandits (RCLB)` - that precisely does so in a principled way.

Algorithm 1 Robust Collaborative Phased Elimination for Linear Bandits (`RCLB`)

Input: Action set $\mathcal{A} = \{a_1, \ldots, a_K\}$, confidence parameter δ, and corruption fraction α.
Initialize: $\ell = 1$ and $\mathcal{A}_1 = \mathcal{A}$.

1: Let $V_\ell(\pi) \triangleq \sum_{a \in \mathcal{A}_\ell} \pi(a) aa'$ and $g_\ell(\pi) \triangleq \max_{a \in \mathcal{A}_\ell} \|a\|^2_{V_\ell(\pi)^{-1}}$. Server solves an approximate G-optimal design problem to compute a distribution π_ℓ over $\mathcal{A}_\ell$ such that $g_\ell(\pi_\ell) \leq 2d$ and $|\text{Supp}(\pi_\ell)| \leq 48 d \log\log d$.

2: For each $a \in \mathcal{A}_\ell$, server computes $m_a^{(\ell)}$ via Eq. (2.5), and broadcasts $\{m_a^{(\ell)}\}_{a \in \mathcal{A}_\ell}$ to all agents.

3: **for** $i \in [M] \setminus \mathcal{B}$ **do**

4: For each arm $a \in \mathcal{A}_\ell$, pull it $m_a^{(\ell)}$ times. Let $r_{i,a}^{(\ell)}$ be the average of the rewards observed by agent i for arm a during phase ℓ.

5: Compute local estimate $\hat{\theta}_i^{(\ell)}$ of θ_* as follows.[3]

$$\hat{\theta}_i^{(\ell)} = \tilde{V}_\ell^{-1} Y_{i,\ell}, \text{ where } \tilde{V}_\ell = \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} aa' \; ; \; Y_{i,\ell} = \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} r_{i,a}^{(\ell)} a. \tag{2.3}$$

6: Transmit $\hat{\theta}_i^{(\ell)}$ to server. Adversarial agents can transmit arbitrary vectors at this stage.

7: **end for**

8: Server computes robust mean pay-offs for each active arm: for each $a \in \mathcal{A}_\ell$, estimate $\mu_a^{(\ell)}$ as

$$\mu_a^{(\ell)} = \texttt{Median}\left(\{ \langle \hat{\theta}_i^{(\ell)}, a \rangle, i \in [M] \} \right).$$

9: Define robust confidence threshold $\gamma_\ell \triangleq \sqrt{2} C \left(1 + \alpha\sqrt{M}\right) \epsilon_\ell$, where C is as in Lemma 1. Server performs phased elimination with the robust means and γ_ℓ to update active arm set:

$$\mathcal{A}_{\ell+1} = \{ a \in \mathcal{A}_\ell : \max_{b \in \mathcal{A}_\ell} \mu_b^{(\ell)} - \mu_a^{(\ell)} \leq 2\gamma_\ell \}. \tag{2.4}$$

10: $\ell = \ell + 1$ and **Goto** line 1.

Description of `RCLB` (Algorithm 1). The `RCLB` algorithm we propose is inspired by the phased elimination algorithm in (Lattimore and Szepesvári, 2020, Chapter 22), but features some key differences due to the distributed and adversarial nature of our problem. The algorithm proceeds in epochs/phases, and in each phase ℓ, the server maintains an active candidate set $\mathcal{A}_\ell$ of potential optimal arms. The exploration of arms in $\mathcal{A}_\ell$ is distributed among the agents. Upon such exploration, each agent i reports back a local estimate $\hat{\theta}_i^{(\ell)}$ of the unknown parameter θ_*; adversarial agents can transmit arbitrarily corrupted messages at this stage. Using the local parameter estimates $\{\hat{\theta}_i^{(\ell)}\}_{i \in [M]}$, the server then constructs (i) a *robust* estimate of the true mean payoff $\langle \theta_*, a \rangle$ for each active arm $a \in \mathcal{A}_\ell$, and (ii) an *inflated* confidence interval that captures both statistical and adversarial uncertainties associated with such robust mean estimates. This step is crucial to our

scheme and requires a lot of care as we explain shortly. With the robust mean payoffs and associated confidence intervals in hand, the server eliminates arms with rewards far away from that of the optimal arm a_*. We now elaborate on the above ideas.

• *Optimal Experimental Design.* To minimize the regret incurred in each phase ℓ, we need to minimize the number of arm-pulls made to arms in $\mathcal{A}_\ell$. To that end, we appeal to a concept from statistics known as optimal experimental design. The concept is as follows. Let $\pi : \mathcal{A} \to [0, 1]$ be a distribution on $\mathcal{A}$ such that $\sum_{a\in\mathcal{A}} \pi(a) = 1$. Now define $V(\pi) \triangleq \sum_{a\in\mathcal{A}} \pi(a)aa'$; $g(\pi) \triangleq \max_{a\in\mathcal{A}} \|a\|^2_{V(\pi)^{-1}}$. The so-called *G-optimal design problem* seeks to find a distribution (design) π_* that minimizes g. In essence, sampling each arm $a \in \mathcal{A}$ in proportion to $\pi_*(a)$ minimizes the number of samples/arm-pulls needed to achieve a desired level of precision in the estimates of the arm means $\langle\theta_*, a\rangle$, $a \in \mathcal{A}$. For our purpose, we only need to solve an approximate G-optimal design problem: using the Frank-Wolfe algorithm and an appropriate initialization, one can *efficiently* find an approximate optimal design π such that $g(\pi) \le 2d$, and $|\text{Supp}(\pi)| \le 48d \log\log d$ (Todd, 2016, Chapter 3). Accordingly, the server computes such an approximate optimal distribution π_ℓ over $\mathcal{A}_\ell$ in each epoch ℓ (line 1 of Algo. 1).

• *Construction of Robust Arm-Payoff Estimates and Confidence Intervals.* In each phase ℓ, the server computes $T_a^{(\ell)}$ and $m_a^{(\ell)}$ for every $a \in \mathcal{A}_\ell$ as follows:

$$T_a^{(\ell)} = \left\lceil \frac{\pi_\ell(a)d}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \right\rceil \text{ and } m_a^{(\ell)} = \left\lceil \frac{T_a^{(\ell)}}{M} \right\rceil, \tag{2.5}$$

where $\pi_\ell(a)$ is obtained from the approximate G-optimal design problem, $\epsilon_\ell = 2^{-\ell}$, $\delta_\ell = \bar{\delta}/(K\ell^2)$, and $\bar{\delta}$ is a design variable to be chosen later. The idea is to explore an arm a $T_a^{(\ell)}$ times to estimate $\langle\theta_*, a\rangle$ up to a precision that scales linearly with ϵ_ℓ; in later phases, we require progressively finer precision (hence, ϵ_ℓ decays exponentially with ℓ). The task of exploration is distributed among the agents, with each agent i being assigned $m_a^{(\ell)}$ arm-pulls for every $a \in \mathcal{A}_\ell$. Using the rewards that it observes during phase ℓ, each good agent $i \in [M] \setminus \mathcal{B}$ computes a local estimate $\hat{\theta}_i^{(\ell)}$ of θ_*, and transmits it to the server (lines 3-7 of Algo. 1). The key question now is as follows: *How should the*

[3]Throughout, we assume that $\tilde{V}_\ell^{-1}$ is invertible in each epoch ℓ; this is the case when $\mathcal{A}_\ell$ spans $\mathbb{R}^d$. If $\mathcal{A}_\ell$ does not span $\mathbb{R}^d$, we can consider the lower dimensional subspace given by span($\mathcal{A}_\ell$).

server use the local estimates $\{\hat{\theta}_i^{(\ell)}\}_{i\in[M]}$? Let us consider two natural strategies.

Candidate Strategies. One option could be to use the local estimates $\{\hat{\theta}_i^{(\ell)}\}_{i\in[M]}$ along with a high-dimensional robust mean estimation algorithm to compute a robust estimate of θ_*. Yet another strategy could be for the server to query the raw observations (i.e., the $y_{i,t}$'s) from the agents, use an univariate robust mean estimation algorithm (e.g., trimmed mean or median) to generate a "clean" version of each observation, and then use these clean observations to compute a robust estimate of θ_*. Although feasible, each of the above strategies can unfortunately lead to an additional $\sqrt{d}$ factor in the regret bound; we discuss this point in detail in the Appendix. The main message we want to convey here is that certain natural candidate solutions can lead to sub-optimal regret bounds.

Main Ideas. Our main insight is the following: to achieve near-optimal optimal regret bounds, one need not go through the route of first computing a robust estimate of θ_*. As our analysis will soon reveal, it suffices to instead compute robust estimates of the arm pay-offs $\{\langle\theta_*, a\rangle\}_{a\in\mathcal{A}_\ell}$ *directly* by employing the estimator in line 8 of Algo. 1. The key statistical property that we exploit here is that for each $a \in \mathcal{A}_\ell$ and $i \in [M] \setminus \mathcal{B}$, the quantity $\langle\hat{\theta}_i^{(\ell)}, a\rangle$ is conditionally-Gaussian with mean $\langle\theta_*, a\rangle$. This observation informs the choice of the median operator in line 8 of Algo. 1. Our next task is to compute appropriate confidence intervals for the robust arm-mean-estimates in order to eliminate sub-optimal arms. This is a delicate task as such confidence intervals need to account for both statistical uncertainties *and* adversarial perturbations. Indeed, if the confidence intervals are too tight, then they can lead to elimination of the optimal arm a_*; if they are too loose, then they can lead to large regret. The confidence threshold γ_ℓ in line 9 of Algo. 1 strikes just the right balance; the choice of such intervals is justified in Lemma 1.

We now state and discuss our main result concerning the performance of `RCLB`.

Theorem 1. ***(Performance of*** `RCLB`***)*** *Suppose* $\alpha \in [0, 0.5)$. *Given any* $\delta \in (0, 1)$, *set* $\bar{\delta} = \delta/(10K)$. *Then,* `RCLB` *guarantees that with probability at least* $1 - \delta$, *the following holds for each good agent* $i \in [M] \setminus \mathcal{B}$*:*

$$\sum_{t=1}^{T}\langle\theta_*, a_* - a_{i,t}\rangle = \tilde{O}\left(\left(\alpha + \sqrt{1/M}\right)\sqrt{dT}\right). \tag{2.6}$$

Main Takeaways. From Theorem 1, we note that RCLB guarantees sublinear regret despite the presence of adversarial agents. More importantly, the regret bound in Eq. (2.6) has optimal dependence on the model-dimension d, the horizon T, and also on the number of agents M when $\alpha = 0$ (i.e., in the absence of adversaries). When α is small, Eq. (2.6) reveals that one can indeed retain the benefits of collaboration, and improve upon the trivial per agent regret of $\tilde{O}(\sqrt{dT})$. Interestingly, this result mirrors that of a similar flavor for distributed stochastic optimization under attacks: given M machines, α fraction of which are corrupt, the authors in (Yin et al., 2018) showed that no algorithm can achieve statistical error lower than $\tilde{\Omega}\left(\alpha/\sqrt{T} + 1/\sqrt{MT}\right)$ for strongly convex loss functions; here, T is the number of samples on each machine. *As far as we are aware, this is the first work to establish an analogous result for collaborative linear bandits.* Inspired by the lower bound in (Yin et al., 2018), one may ask: *Is the additive $\alpha\sqrt{T}$ term in Eq.* (2.6) *unavoidable?* We will provide a definitive answer in Section 2.4. Finally, we note that Theorem 1 immediately implies a bound of $\tilde{O}\left(\left(\alpha M + \sqrt{M}\right)\sqrt{dT}\right)$ on the group regret R_T. We provide a formal proof of this fact in Appendix 2.10.

Proof Outline of Theorem 1. The first main step in our analysis of Theorem 1 is to provide guarantees on the estimates $\{\mu_a^{(\ell)}\}_{a\in\mathcal{A}_\ell}$ computed in line 8 of RCLB. This is achieved as follows.

Lemma 1. *(**Robust Confidence Intervals**) Fix any epoch ℓ. There exists an universal constant $C > 0$ such that for each active arm $b \in \mathcal{A}_\ell$, the following holds with probability at least $1 - \delta_\ell$:*

$$|\mu_b^{(\ell)} - \langle \theta_*, b\rangle| \leq \gamma_\ell, \;\; \textit{where } \gamma_\ell = \sqrt{2}C\left(1 + \alpha\sqrt{M}\right)\epsilon_\ell. \tag{2.7}$$

Equipped with the above result, we argue that with high-probability, (i) the optimal arm a_* is never eliminated by RCLB (Lemma 4 in App. 2.10); and (ii) in each epoch ℓ, an active arm in $\mathcal{A}_\ell$ can contribute to at most $O(\gamma_\ell)$ per-time-step regret (Lemma 5 in App. 2.10). Putting these pieces together in a careful manner yields the desired result; we defer a detailed proof of Theorem 1 to Appendix 2.10.

In the next section, we derive a lower bound that provides fundamental insights into the impact

of the adversarial agents for the multi-agent sequential decision-making problem considered in this paper.

2.4. Lower Bounds

In this section, we assess the optimality of the regret bound obtained in Theorem 1. To do so, we consider a slightly different attack model: we assume that each of the agents is adversarial with probability α, independently of the other agents. Thus, the expected fraction of adversaries is α. Next, to provide a clean argument, we will focus on a class of policies Π where at each time-step t, the server assigns the *same* action to every agent, i.e., $a_{i,t} = a_t, \forall i \in [M]$. We note that all the algorithms developed in this paper adhere to policies in Π. Moreover, policies within the class Π yield the minimax optimal per-agent regret bound of order $\tilde{O}(\sqrt{dT}/\sqrt{M})$ when $\alpha = 0$. Since the standard multi-armed bandit setting (Auer et al., 2002) is a special case of the structured linear bandit setting considered here, a lower bound for the former implies one for the latter. With this in mind, let us denote by $\mathcal{E}_{\mathcal{N}}^{(K)}(1)$ the class of multi-armed bandits with K arms, where the reward distribution of each arm is Gaussian with unit variance. An instance $\nu_\mu \in \mathcal{E}_{\mathcal{N}}^{(K)}(1)$ is characterized by the mean vector $\mu \in \mathbb{R}^K$ associated with the K arms. Finally, let $R_T^{(s)}(\nu_\mu)$ denote the expected cumulative regret of the server (which is the same as that of a good agent $i \in [M] \setminus \mathcal{B}$) when it interacts with the instance ν_μ. We can now state the following result which establishes a fundamental lower bound for our problem.

Theorem 2. ***(Fundamental Lower Bound)*** *Given any policy in* Π*, there exist two distinct instances* $\nu_\mu, \nu_{\mu'} \in \mathcal{E}_{\mathcal{N}}^{(2)}(1)$*, and an universal constant* $c > 1$*, such that*

$$\max\{R_T^{(s)}(\nu_\mu), R_T^{(s)}(\nu_{\mu'})\} \geq c\alpha\sqrt{T}, \tag{2.8}$$

irrespective of the number of agents M*.*

Main Takeaways. Observe from Eq. (2.6) that even when M is arbitrarily large, the additive $\alpha\sqrt{T}$ term due to the adversaries remains unaffected - *Is this term truly unavoidable or just an artifact of our analysis?* Theorem 2 settles this question by revealing a fundamental performance limit:

every policy in Π has to suffer the additive $\alpha\sqrt{T}$ regret, regardless of the number of agents. Thus, taken together, *Theorems 1 and 2 provide the first set of tight, near-optimal regret guarantees for the setting considered in this paper. We consider this to be a significant contribution of our work.*

Proof Idea for Theorem 2. For our setting, the standard techniques to prove lower bounds for non-adversarial bandits do not directly apply. The lower-bound proofs used for reward-corruption models (Kapoor et al., 2019), and attacks with a fixed budget (Bogunovic et al., 2021), are not applicable either. This motivates us to use a new proof technique that combines information-theoretic arguments in (Bubeck et al., 2013) with ideas from the robust mean estimation literature (Chen et al., 2015; Lai et al., 2016). Specifically, for a two-armed bandit setting, we carefully construct two instances and attack strategies such that the joint distribution of rewards seen by the server is *identical* for both instances. Moreover, the instances are constructed such that (i) the optimal arm in one instance is sub-optimal for the other; and (ii) the per-time-step regret for selecting a sub-optimal arm in either instance is $\Omega(\alpha/\sqrt{T})$. For a detailed proof, see Appendix 2.11.

Having established tight bounds for the linear bandit model in Section 4.4, in the sequel, we will show how our algorithmic ideas and results can be significantly extended to more general settings.

2.5. Extension to Generalized Linear Models with Adversaries

In this section, we will show how to achieve a regret bound akin to that in Theorem 1 for the non-linear observation model shown below (Filippi et al., 2010; Li et al., 2017), known as the generalized linear model (GLM):

$$y_{i,t} = \mu\left(\langle \theta_*, a_{i,t} \rangle\right) + \eta_{i,t}, \tag{2.9}$$

where $\mu : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable function typically referred to as the (inverse) *link* function, and $\eta_{i,t} \sim \mathcal{N}(0,1)$ is as before. Our goal is to now control the following notion of regret:

$$R_T^{\mathrm{GLM}} = \mathbb{E}\left[\sum_{i\in[M]\setminus\mathcal{B}} \sum_{t=1}^{T} \left(\mu\left(\langle \theta_*, a_* \rangle\right) - \mu\left(\langle \theta_*, a_{i,t} \rangle\right)\right)\right], \tag{2.10}$$

where $a_* = \arg\max_{a \in \mathcal{A}} \mu(\langle \theta_*, a \rangle)$. The main technical challenge relative to the setting considered in Section 4.4 pertains to the construction of the robust confidence intervals. In particular, the non-linearity of the map $\mu(\cdot)$ makes it hard to apply the technique adopted in line 8 of `RCLB`, necessitating a different approach that we describe next. We start with the following standard assumption (Filippi et al., 2010; Li et al., 2017).

Assumption 1. *The function $\mu : \mathbb{R} \to \mathbb{R}$ is continuously differentiable, Lipschitz with constant $k_2 \geq 1$, and such that*

$$k_1 = \min\{1, \inf_{\theta \in \Theta, a \in \mathcal{A}} \dot{\mu}(\langle \theta, a \rangle)\} > 0.$$

Here, $\dot{\mu}(\cdot)$ is used to represent the derivative of $\mu(\cdot)$.

Next, for any $\theta \in \mathbb{R}^d$, we define $h_\ell(\theta) \triangleq \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \mu(\langle \theta, a \rangle) a$. We now describe a variant of `RCLB` - dubbed `RC-GLM` - for generalized linear models. Notably, unlike the standard bandit algorithms (Filippi et al., 2010; Li et al., 2017) for GLM's that build on `LinUCB`, `RC-GLM` is based on phased elimination.

Description of `RC-GLM`. Our algorithm uses as a sub-routine the recently proposed Iteratively Reweighted Mean Estimator for computing a robust estimate of the mean of high-dimensional Gaussian random variables with adversarial outliers (Dalalyan and Minasyan, 2022). Specifically, suppose we are given M d-dimensional samples $x_1, \ldots, x_M$, such that $(1-\alpha)M$ of these samples are drawn i.i.d. from $\mathcal{N}(v, \Sigma)$, where $v \in \mathbb{R}^d$ is an unknown mean vector, and $\Sigma \in \mathbb{R}^{d \times d}$ is a known covariance matrix. The remaining αM samples are adversarial outliers and can be arbitrary. The estimator in (Dalalyan and Minasyan, 2022) takes as input the M samples, the corruption fraction α, and the covariance matrix Σ. It then outputs an estimate $\hat{v}$ of v such that with high probability, $\|\hat{v} - v\| = \tilde{O}\left(\|\Sigma\|_2^{1/2}\left(\sqrt{d/M} + \alpha\sqrt{\log(1/\alpha)}\right)\right)$. Importantly, the estimator in (Dalalyan and Minasyan, 2022) runs in polynomial-time, and is minimax-rate-optimal. Let us now see how this estimator - described in Appendix 2.12 - can be applied to our setting.

We only describe the key differences of `RC-GLM` relative to `RCLB` here, and defer a detailed description of `RC-GLM` to Appendix 2.12. To get around the difficulty posed by the non-linear link function, our

main idea is to first compute a robust estimate $\hat{\theta}^{(\ell)}$ of θ_* at the server, and then use it to develop a phased elimination strategy. To that end, instead of computing a local estimate $\hat{\theta}_i^{(\ell)}$ as in `RCLB`, each good agent $i \in [M] \setminus \mathcal{B}$ transmits $Y_{i,\ell}$ to the server, and the server computes a vector X_ℓ as follows: $X_\ell = \texttt{ITW}(\{\tilde{V}_\ell^{-1/2} Y_{i,\ell}, i \in [M]\})$. Here, $\tilde{V}_\ell^{-1/2}$ and $Y_{i,\ell}$ are as in Eq. (2.3), and we used $\texttt{ITW}(\cdot)$ to denote the output of the robust estimator in (Dalalyan and Minasyan, 2022). Our key observation here is that for each good agent i, $\tilde{V}_\ell^{-1/2} Y_{i,\ell}$ is a d-dimensional Gaussian random variable with mean $\tilde{V}_\ell^{-1/2} h_\ell(\theta_*)$, and covariance matrix $\Sigma = I_d$, justifying the use of the robust Gaussian mean estimator in (Dalalyan and Minasyan, 2022). The fact that $\Sigma = I_d$ is crucial in our algorithm design as the error-bound in (Dalalyan and Minasyan, 2022) scales with the 2-norm of Σ. Essentially, the above steps enable us to extract a statistic X_ℓ that captures information about the agents' observations during epoch ℓ. Using this statistic, the server next computes an estimate $\hat{\theta}^{(\ell)}$ of θ_* by solving $h_\ell(\hat{\theta}^{(\ell)}) = \tilde{V}_\ell^{1/2} X_\ell$,[4] and employs the following phased elimination strategy:

$$\mathcal{A}_{\ell+1} = \{a \in \mathcal{A}_\ell : \max_{b \in \mathcal{A}_\ell} \mu(\langle \hat{\theta}^{(\ell)}, b \rangle) - \mu(\langle \hat{\theta}^{(\ell)}, a \rangle) \leq 2\bar{\gamma}_\ell\}; \;\; \bar{\gamma}_\ell = \bar{C}(k_2/k_1) \left(\sqrt{d} + \alpha\sqrt{M \log(1/\alpha)}\right) \epsilon_\ell,$$

where $\bar{C}$ is an universal constant known to the server. Deriving an analogue of Lemma 1 to compute the robust confidence threshold $\bar{\gamma}_\ell$ requires some work. This is achieved by exploiting the regularity properties of the link function in tandem with the confidence bounds in (Dalalyan and Minasyan, 2022); see Appendix 2.12 for details and a proof of our main result for `RC-GLM` stated below.

Theorem 3. *(**Performance of Algorithm** `RC-GLM`) Suppose $\alpha < (5 - \sqrt{5})/10$, and $M = \Omega(\log(KT))$. Given any $\delta \in (0, 1)$, `RC-GLM` guarantees that with probability at least $1 - \delta$, the following holds for each good agent $i \in [M] \setminus \mathcal{B}$:*

$$\sum_{t=1}^{T} \left(\mu\left(\langle \theta_*, a_* \rangle\right) - \mu\left(\langle \theta_*, a_{i,t} \rangle\right)\right) = \tilde{O}\left((k_2/k_1)\left(\alpha\sqrt{\log\left(1/\alpha\right)} + \sqrt{d/M}\right)\sqrt{dT}\right). \tag{2.11}$$

Main Takeaways. Theorem 3 significantly generalizes Theorem 1 and shows that even for general non-linear observation maps, one can reap the benefits of collaboration in the presence of adversaries.

[4] We argue in Appendix 2.12 that this equation admits a unique solution.

Algorithm 2 Robust BaseLinUCB (at Server)

Input: Confidence parameter $\bar{\delta}$, corruption fraction α, and index set $\psi_t \subseteq [t-1]$.

$A_t \leftarrow \frac{I_d}{M} + \sum_{\tau \in \psi_t} x_{\tau,a_\tau} x'_{\tau,a_\tau}$.

for $i \in [M]$ **do** ▷ Compute local parameters for each agent

  $b_{i,t} \leftarrow \sum_{\tau \in \psi_t} r_{i,\tau} x_{\tau,a_\tau}$; $\hat{\theta}_{i,t} \leftarrow A_t^{-1} b_{i,t}$.

end for

for $a \in [K]$ **do** ▷ Compute robust estimates for each feature vector

  $\hat{r}_{t,a} \leftarrow \texttt{Median}\left(\{\langle \hat{\theta}_{i,t}, x_{t,a} \rangle, i \in [M]\}\right)$; $w_{t,a} \leftarrow \left(\alpha + 2C\sqrt{\frac{\log(\frac{1}{\bar{\delta}})}{M}}\right) \|x_{t,a}\|_{A_t^{-1}}$, where C is as in Lemma 1.

end for

The additional $\sqrt{d}$ factor in the bound of (2.11) relative to that in (2.6) is inherited from the error-rate guarantees in (Dalalyan and Minasyan, 2022); the task of tightening this bound is left as future work. Nonetheless, Theorem 3 is the only result we are aware of that provides adversarial-robustness guarantees for GLMs.

Remark 1. *(**Communication Complexity of `RCLB` and `RC-GLM`**) It is not hard to see that the number of epochs/phases in `RCLB` and `RC-GLM` is $O(\log(MT))$. Since communication between the server and the agents occurs only once in every epoch, we note that the communication complexity of these algorithms scale logarithmically with the horizon T. Thus, our proposed algorithms not only lead to near-optimal regret bounds in the face of worst-case adversarial attacks, they are also communication-efficient by design. This is an important point to take note of as communication-efficiency is a key consideration in large-scale computing paradigms such as federated learning.*

2.6. Robust Collaborative Contextual Bandits with Adversaries

In this section, we will consider a collaborative contextual bandit setting where at each time-step $t \in [T]$, the server and the agents observe K d-dimensional feature vectors, $\{x_{t,a} | a \in [K]\}$, with $\|x_{t,a}\| \leq 1, \forall a \in [K]$ and $\forall t \in [T]$. We assume that the adversarial agents have no control over the generation of the feature vectors. Associated with each arm $a \in [K]$, the stochastic reward observed by an agent $i \in [M]$ comes from the following observation model:

$$y_{i,t}(a) = \langle \theta_*, x_{t,a} \rangle + \eta_{i,t}(a), \tag{2.12}$$

Algorithm 3 Robust Collaborative SupLinUCB for Contextual Bandits (at Server)

Input: Confidence parameter δ, corruption fraction α, and horizon T.

$S \leftarrow \lceil \ln T \rceil; \psi_1^{(s)} \leftarrow \emptyset, \forall s \in [S]$.

for $t \in [T]$ **do**

  $s \leftarrow 1$ and $\mathcal{A}_1 \leftarrow [K]$.

  repeat

  Use Algorithm 2 with $\bar{\delta} = \delta/(KST)$, and index set $\psi_t^{(s)}$ to compute robust estimates $\{\hat{r}_{t,a}^{(s)}, w_{t,a}^{(s)}\}$ of the means and variances of the payoffs associated with each arm $a \in \mathcal{A}_s$.

  If $w_{t,a}^{(s)} > 2^{-s}/\sqrt{M}$ for some $a \in \mathcal{A}_s$, then choose this arm, i.e., set $a_t = a$. Store the corresponding phase: $\psi_{t+1}^{(s)} = \psi_t^{(s)} \cup \{t\}, \psi_{t+1}^{(\ell)} = \psi_t^{(\ell)}\ \forall \ell \neq s$.

  Else if $w_{t,a}^{(s)} \leq 1/\sqrt{MT}\ \forall a \in \mathcal{A}_s$, then select the arm with the *highest robust upper confidence bound*: $a_t = \arg\max_{a \in \mathcal{A}_s} \left(\hat{r}_{t,a}^{(s)} + w_{t,a}^{(s)}\right)$. Do not store this phase, i.e., $\psi_{t+1}^{(\ell)} = \psi_t^{(\ell)}\ \forall \ell \in [S]$.

  Else if $w_{t,a}^{(s)} \leq 2^{-s}/\sqrt{M}\ \forall a \in \mathcal{A}_s$, then update active arm-set as

$$\mathcal{A}_{s+1} = \{a \in \mathcal{A}_s | \max_{b \in \mathcal{A}_s} \left(\hat{r}_{t,b}^{(s)} + w_{t,b}^{(s)}\right) - \left(\hat{r}_{t,a}^{(s)} + w_{t,a}^{(s)}\right) \leq 2^{(1-s)}/\sqrt{M}\}.$$

  $s \leftarrow s + 1$.

  until an action a_t is chosen.

  Broadcast the chosen action a_t to every agent (i.e., $a_{i,t} = a_t, \forall i \in [M]$), and receive corresponding rewards $\{r_{i,t}\}_{i \in [M]}$. Adversarial agents can transmit arbitrary reward values.

end for

where $\{\eta_{i,t}(a)\}$ are drawn i.i.d. from $\mathcal{N}(0,1)$. At each time step t, a good agent $i \in [M] \setminus \mathcal{B}$ plays an action $a_{i,t} \in [K]$, and receives the corresponding reward $r_{i,t} \triangleq y_{i,t}(a_{i,t})$ based on the observation model in (2.12). The main difference of the setting considered here relative to the one in Section 4.4 is that the feature vectors for each arm can change over time. As a result, the optimal action $a_t^* = \arg\max_{a \in [K]} \langle \theta_*, x_{t,a} \rangle$ can change over time, making it particularly challenging to compete with a time-varying optimal action in the presence of adversaries. This dictates the need for a different algorithmic strategy compared to the one we developed in Section 1. Before we develop such a strategy, let us first formally define the performance metric of interest to us in this setting:

$$R_T^{\text{Context}} = \mathbb{E}\left[\sum_{i \in [M] \setminus \mathcal{B}} \sum_{t=1}^{T} \langle \theta_*, x_{t,a_t^*} - x_{t,a_{i,t}} \rangle\right]. \tag{2.13}$$

The main question we ask is: *For the contextual bandit setting described above, can one continue to hope for benefits of collaboration in the presence of adversaries?* In what follows, we will answer this

question in the affirmative by developing a variant of the SUPLINREL algorithm in (Auer, 2002).

Description of Algorithm 3. At each time-step t, our proposed algorithm, namely Algorithm 3, scans through the set $\mathcal{A}$ of arms to determine a suitable action a_t. This scanning process (lines 4-10 of Algo. 3) is done at the server over S phases. Corresponding to each phase $s \in [S]$, the server maintains a set $\psi_t^{(s)}$; the set $\psi_t^{(s)}$ stores all the time-steps in $[t-1]$ where an action is chosen in phase s of the scanning process based on line 6 on Algo. 3. The scanning process itself relies on Algorithm 2 as a sub-routine. Specifically, in each phase s, the server first invokes Algorithm 2 to obtain a robust estimate $\hat{r}_{t,a}^{(s)}$ of $\langle \theta_*, x_{t,a} \rangle$ for each arm $a \in \mathcal{A}_s$, along with an associated *inflated* confidence width $w_{t,a}^{(s)}$ (line 5 of Algo. 3). If the confidence width is too large for a particular arm (as in line 6), then such an arm requires exploration and is accordingly chosen to be a_t. If, on the other hand, the confidence widths of all arms are sufficiently small (as in line 7), then a_t is chosen to be the arm with the highest upper-confidence bound. Thus, we follow the principle of *optimism in the face of uncertainty here, while exercising caution to account for the presence of adversaries (via the use of inflated confidence intervals).* If the conditions in lines 6 and 7 both fail, then the arms in $\mathcal{A}_s$ require further screening. Accordingly, we move to the next phase $s+1$, retaining only those arms that are sufficiently close to the optimal arm a_t^*; see line 8 of Algo. 3. Our main innovation lies in (i) the construction of the robust arm-estimates in Algo. 2 that account for both statistical and adversarial behavior, and (ii) the careful use of such estimates in lines 6-8 of Algo. 3 to pick the action a_t. The next result reveals that the combination of these ideas yields near-optimal regret bounds.

Theorem 4. ***(Performance of Algo. 3)*** *Suppose $\alpha \in (0, 0.5)$. Given any $\delta \in (0,1)$, Algo. 3 guarantees that with probability at least $1-\delta$, the following holds for each good agent $i \in [M] \setminus \mathcal{B}$:*

$$\sum_{t=1}^{T} \langle \theta_*, x_{t,a_t^*} - x_{t,a_{i,t}} \rangle = \tilde{O}\left(\left(\alpha + \sqrt{1/M} \right) \sqrt{dT} \right). \tag{2.14}$$

Main Takeaways. For the contextual bandit setting considered here, the single-agent minimax optimal regret in the absence of adversaries is $\tilde{O}(\sqrt{dT})$ (Auer, 2002). In light of the lower bound in Theorem 2, we see that Theorem 4 provides a near-optimal regret guarantee, just as Theorem 1.

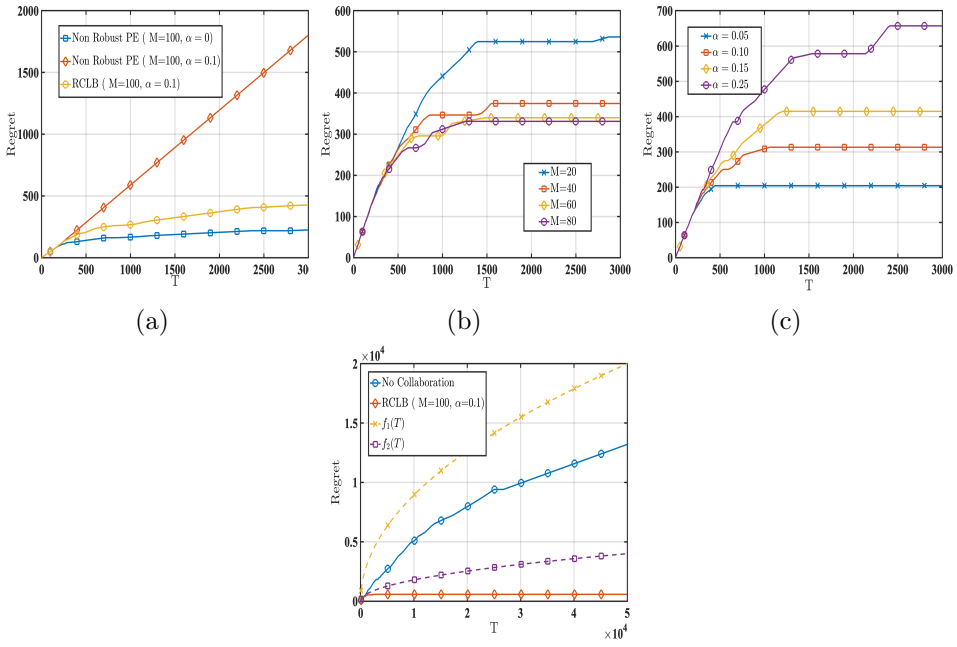

Figure 2.1: Plots of per-agent regret for the linear bandit experiment. (a) Comparison between `RCLB` and a vanilla non-robust phased elimination algorithm. (b) Performance of `RCLB` for varying number of agents M, with $\alpha = 0.1$. (c) Performance of `RCLB` for varying corruption fraction α, with $M = 100$. For (d), we set $\alpha = 0.1$, $M = 100$, and compare `RCLB` to a phased elimination algorithm where the agents do not collaborate. We also plot theoretical upper-bounds: $f_1(T) = 40\sqrt{dT}$ and $f_2(T) = 40(\alpha + \sqrt{(1/M)})\sqrt{dT}$.

Remark 2. *For ease of exposition, we have considered Gaussian noise (in the observation model) throughout the paper. However, both our algorithms and results can be extended with slight modifications to sub-Gaussian noise sequences. We elaborate on this point in Appendix 2.9.*

2.7. Simulation Results

We report simulation results on synthetic data to corroborate our developed theory. Additional simulations on contextual bandits and alternate attack models are presented in Appendix 2.15.

Experimental Setup. We consider a setting with 50 actions in $\mathbb{R}^5$; we describe how these actions and θ^* are generated in Appendix 2.15. The rewards are generated based on the observation model in Eq. (2.1). We now describe the attack model. To manipulate the server into selecting sub-optimal arms, each adversarial agent i employs the simple strategy of reducing the rewards of the good

arms and increasing the rewards of the bad arms. More precisely, in each epoch ℓ, upon pulling an arm a and observing the corresponding reward y_a, an adversarial agent i does the following: if $y_a > p\langle\theta_*, a_*\rangle$, then this reward is corrupted to $\tilde{y}_a = y_a - \beta$; and if $y_a \leq p\langle\theta_*, a_*\rangle$, then the reward is corrupted to $\tilde{y}_a = y_a + \beta$. For this experiment, we fix $p = 0.6$ and $\beta = 5$. Agent $i \in \mathcal{B}$ then uses all the corrupted rewards in epoch ℓ to generate the local model estimate $\hat{\theta}_i^{(\ell)}$ that is transmitted to the server.

Discussion of Simulation Results. Fig. 2.1 summarizes our experimental results. In Fig. 2.1(a), we compare our proposed algorithm `RCLB` to a vanilla distributed phased elimination (PE) algorithm that does not account for adversarial agents. Specifically, the latter is designed by replacing the median operation in line 8 of Algorithm 1 with a mean operation, and setting the threshold γ_ℓ in line 9 to be ϵ_ℓ. Fig. 2.1(a) shows that even a small fraction $\alpha = 0.1$ of adversaries can cause the non-robust PE algorithm to incur linear regret. In contrast, `RCLB` continues to guarantee sub-linear regret bounds despite adversarial corruptions. Furthermore, the regret bound of `RCLB` in the presence of a small fraction of adversarial agents is close to that of the non-robust algorithm in the absence of adversaries. This goes on to establish the robustness of `RCLB`.

Fig. 2.1(b) depicts the performance of `RCLB` for varying values of the number of agents M, at a fixed corruption level $\alpha = 0.1$. We observe that increasing M results in lower regret, indicating a clear benefit of collaboration despite the presence of adversaries. In Fig. 2.1(c), we vary the corruption fraction α, keeping M fixed at 100. As expected, increasing α leads to higher (albeit sub-linear) regret. Importantly, the trends observed in both Fig. 2.1(b) and Fig. 2.1(c) are consistent with the theoretical upper-bound of $O((\alpha + 1/\sqrt{M})\sqrt{dT})$ predicted by Theorem 1.

A trivial way to avoid adversarial corruption is for a good agent to not participate in any collaboration at all, and run a standard single-agent bandit algorithm. This would result in such an agent incurring $O(\sqrt{dT})$ regret. The purpose of Fig. 2.1(d) is to drive home the point that `RCLB` *can lead to significant improvements over a trivial non-collaborative strategy.* To make this point clear, we compare `RCLB` to a standard single-agent phased elimination algorithm that does not involve any collaboration, and observe that despite adversarial corruption, `RCLB` leads to considerably lower regret bounds as

compared to the non-collaborative strategy. This highlights the importance of our approach.

2.8. Detailed Discussion of Related Work

Below, we provide a detailed discussion of relevant work.

• **Reward Corruption Attacks in Stochastic Bandits.** In the single-agent setting, there is a rich body of work that studies the effect of reward-corruption in stochastic bandits, both for the unstructured multi-armed bandit problem (Jun et al., 2018; Liu and Shroff, 2019; Lykouris et al., 2018; Gupta et al., 2019), and also for structured linear bandits (Bogunovic et al., 2020; Garcelon et al., 2020; Bogunovic et al., 2021; He et al., 2022). In these works, an adversary can modify the true stochastic reward/feedback on certain rounds; a corruption budget C captures the total corruption injected by the adversary over the horizon T. The attack model we study is fundamentally different: the adversaries in our setting can inject corruptions of *arbitrary* magnitude in *all* rounds, i.e., there are no budget constraints. As such, the algorithmic techniques in (Jun et al., 2018; Liu and Shroff, 2019; Lykouris et al., 2018; Gupta et al., 2019; Bogunovic et al., 2020; Garcelon et al., 2020; Bogunovic et al., 2021) do not apply to our model.

Continuing with this point, we note that in (Gupta et al., 2019), the authors proved an *algorithm-independent* lower bound of $\Omega(C)$ on the regret. This lower bound suggests that for the reward-corruption attack model, when the attacker's budget C scales linearly with the horizon T, there is no hope for achieving sub-linear regret. In (Kapoor et al., 2019), the authors studied a reward-corruption model closely related to those in (Lykouris et al., 2018; Gupta et al., 2019; Bogunovic et al., 2020), where in each round, with probability η (independently of the other rounds), the attacker can bias the reward seen by the learner. Similar to the lower bound in (Gupta et al., 2019), the authors in (Kapoor et al., 2019) proved a lower bound of $\Omega(\eta T)$ on the regret for their model. In sharp contrast to the fundamental limits established in (Gupta et al., 2019; Kapoor et al., 2019), for our setup, as long as the corruption fraction α is strictly less than half, we prove that with high-probability it is in fact possible to achieve sub-linear regret. The key is that for our setting, the server can leverage "clean" information from the good agents in every round; of course, the identities of such good agents are not known to the server. We finally note that beyond the task of minimizing cumulative regret,

the impact of fixed-budget reward-contamination has also been explored for the problem of best-arm identification in (Zhong et al., 2021).

• **Multi-Agent Bandits.** There is a growing literature that studies multi-agent multi-armed bandit problems in the absence of adversaries, both over peer-to-peer networks, and also for the server-client architecture model (Liu and Zhao, 2010; Kalathil et al., 2014; Kar et al., 2011; Landgren et al., 2016, 2021; Shahrampour et al., 2017; Buccapatnam et al., 2015; Kolla et al., 2018; Wang et al., 2019; Sankararaman et al., 2019; Martínez-Rubio et al., 2018; Dubey et al., 2020; Dubey and Pentland, 2020b; Lalitha and Goldsmith, 2020; Chawla et al., 2020a,b; Ghosh et al., 2021; Agarwal et al., 2021; Zhu et al., 2021; Shi et al., 2021). The main focus in these papers is the design of coordination protocols among the agents that balance communication-efficiency with performance. A few very recent works (Dubey and Pentland, 2020a; Vial et al., 2021, 2022; Mitra et al., 2021a) also look at the effect of attacks, but for the simpler unstructured multi-armed bandit problem (Auer et al., 2002). Accounting for adversarial agents in the structured linear bandit setting we consider here requires significantly different ideas that we develop in this paper. At this point, we should mention that the concurrent work of (Kwon et al., 2022) looks at a contextual bandit model somewhat different from what we study; they identify fundamental performance limits and design efficient robust algorithms.

• **Security in Distributed Optimization and Federated Learning.** As we mentioned earlier, several papers have studied the problem of accounting for adversarial agents in the context of supervised learning (Chen et al., 2017b; Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2018b; Alistarh et al., 2018; Xie et al., 2018; Li et al., 2019a; Sundaram and Gharesifard, 2018; Ghosh et al., 2019, 2020a,b; Su and Vaidya, 2020; Kuwaranancharoen et al., 2020; Gupta et al., 2021; Karimireddy et al., 2021; Adibi et al., 2022a). One of the primary applications of interest here is the emerging paradigm of federated learning (Konečnỳ et al., 2016; Bonawitz et al., 2019; McMahan et al., 2017). Different from the sequential decision-making setting we investigate in our paper, the aforementioned works essentially abstract out the supervised learning task as a *static* distributed optimization problem, and then apply some form of secure aggregation on either gradient vectors or parameter estimates.

• **Robust Statistics.** The algorithms that we develop in this paper borrow tools from the literature on robust statistics, pioneered by Huber (Huber, 1992, 2004). We point the reader to (Chen et al., 2015; Lai et al., 2016; Cheng et al., 2019; Minsker, 2018; Lugosi and Mendelson, 2021; Dalalyan and Minasyan, 2022), and the references therein, to get a sense of some of the main results in this broad area of research. In a nutshell, given multiple samples of a random variable - with a small fraction of samples corrupted by an adversary - the essential goal of this line of work is to come up with statistically optimal and computationally efficient robust estimators of the mean of the random variable. Notably, unlike both the sequential bandit setting and the iterative optimization setting, the robust statistics literature focuses on one-shot estimation. In other words, the adversary gets to corrupt the batch of samples *only once*, and the effect of such corruption does not compound over time or iterations.

2.9. Further Comments on our Algorithms

In what follows, we comment on certain key aspects of our proposed algorithms.

• **On the knowledge of the corruption fraction.** In practice, if we do not know the corruption fraction α, but have access to an upper bound on α, say $\tilde{\alpha}$, we can essentially use $\tilde{\alpha}$ as a proxy for α, both in our algorithms and their analyses. In this case, all our results go through identically by simply replacing α with $\tilde{\alpha}$ in the bounds. However, if we have no idea at all about α, then it is not clear whether one can design a robust algorithm that achieves minimax-optimal rates for the setting we consider in this paper. The main reason is as follows. Note that although computing a median does not require knowledge of α, the robust confidence threshold we use in line 9 of Algorithm 1 critically relies on the knowledge of the corruption fraction α (or an upper-bound on it). Moreover, the correctness of our overall approach, and the fact that it is minimax-optimal, relies heavily on the tightness of the robust confidence thresholds in Lemma 1. Thus, at the moment, we do not know of a way that can achieve tight regret bounds without any knowledge whatsoever of α; investigating this aspect further is an interesting topic of future research.

• **Beyond Gaussian Noise.** In what follows, we explain in detail that *both our algorithms and analysis apply, with slight modifications, to both sub-Gaussian noise, and more generally, noise with*

bounded variance. The reason why we chose Gaussian noise was primarily for ease of exposition. Suppose the noise samples are i.i.d. with zero-mean (a standard assumption) and unit variance (the argument we present next trivially extends to bounded variance σ^2). To deal with such general noise distributions, we need to make two minor changes to Algorithm 1. First, we replace the Median operation in line 8 of Algorithm 1 with the scalar univariate trimmed mean estimator recently proposed in (Lugosi and Mendelson, 2021). The estimator in (Lugosi and Mendelson, 2021) is *easily implementable and minimax-optimal.* Second, we set the robust threshold γ_ℓ in line 9 to be $\gamma_\ell = C(1 + \sqrt{\alpha M})\epsilon_\ell$, where C is a suitably large universal constant; note that the only thing that has changed from before is the replacement of α by $\sqrt{\alpha}$ in γ_ℓ. We will justify the choice of this threshold shortly, but before that, we mention the implications.

Implications. With the two minor modifications to Algorithm 1 described above, we can establish an analogue of Theorem 1 where for each good agent, with probability at least $1 - \delta$, the regret is bounded above by $\tilde{O}\left(\left(\sqrt{\alpha} + \sqrt{1/M}\right)\sqrt{dT}\right)$. Compared to the bound in Theorem 1, we note that our new bound is worsened by the replacement of α with $\sqrt{\alpha}$ - this is the price paid to account for general non-Gaussian distributions. For robust mean estimation with general noise distributions (with bounded fourth moments), the additive $\sqrt{\alpha}$ factor is *unavoidable*; see (Lugosi and Mendelson, 2021), for instance. Thus, we conjecture that the bound on regret we mentioned above is also minimax-optimal.

For sub-Gaussian distributions, we can achieve a tighter regret bound: by setting the robust confidence threshold to be $\gamma_\ell = C(1 + \alpha\sqrt{\log(1/\alpha)M})\epsilon_\ell$, we can achieve a per-agent regret bound of $\tilde{O}\left(\left(\alpha\sqrt{\log(1/\alpha)} + \sqrt{1/M}\right)\sqrt{dT}\right)$. *Hence, with sub-Gaussian noise, we essentially recover the same bounds as with Gaussian noise (up to logarithmic factors).* It is very likely that with sub-Gaussian noise, the median will continue to yield the bound above, i.e., we may not even need the univariate trimmed mean estimator from (Lugosi and Mendelson, 2021). However, we do not have a concrete proof of this fact yet.

Changes in Analysis. We now go over the minor changes that need to be made to the analysis of Theorem 1 in view of replacing the median operator by the trimmed mean estimator in

(Lugosi and Mendelson, 2021). We first need an analog of Lemma 2 in Appendix 2.10. This is supplied by Theorem 1 in (Lugosi and Mendelson, 2021) that provides guarantees on the univariate trimmed mean estimator. Suppose we are given M data samples, where all the good samples are i.i.d. with mean μ and variance σ^2. Moreover, suppose the fraction of bad samples is at most $1/2$. With probability at least $1-\delta$, we then have

$$|\hat{\mu} - \mu| \leq C\left(\sqrt{\alpha} + \sqrt{\frac{\log(1/\delta)}{M}}\right)\sigma,$$

where $\hat{\mu}$ is the output of the univariate trimmed mean estimator. Equipped with this result, one can follow the *exact same steps* as in the proof of Lemma 1 to justify the choice of the robust confidence threshold $\gamma_\ell = C(1+\sqrt{\alpha M})\epsilon_\ell$. When the noise is additionally sub-Gaussian, the above bound can be tightened by replacing $\sqrt{\alpha}$ with $\alpha\sqrt{\log(1/\alpha)}$. The rest of the proof goes through *identically*.

To sum up, we have argued that with minor modifications to both our algorithm and analysis, our overall approach can handle both sub-Gaussian noise, and noise with bounded variance.

• **Is it possible to tolerate a corruption fraction** $\alpha > 0.5$? The key question here is: *Who is the learner? The agent or the server?* As we explain shortly, the answer to this question has significant implications for whether or not we can tolerate the $\alpha > 0.5$ case. Suppose the learners are the agents, in that each good agent is capable of taking their own decisions (actions). If it is known ahead of time that $\alpha > 0.5$, then acting alone is the most natural thing to do. This is because there is nothing to be gained by collaborating: no robust aggregation scheme can provide any guarantees when $\alpha > 0.5$; indeed, $\alpha = 0.5$ is a *fundamental breakdown point* when one considers an *arbitrary* corruption model. On the other hand, when $\alpha < 0.5$, one can significantly improve upon the trivial per-agent regret of $O(\sqrt{dT})$ by using the approach developed in our paper. This is revealed not only by our theory, but the simulations that we report in Section 3.6.

Now suppose the learner is the server, i.e., *the actions are decided by the server*, and the agents interact with an unknown environment and report certain relevant statistics to the server. As a

concrete example, consider a web-advertising example where the server displays ads (actions) to a group of people (agents), assesses their (potentially corrupted) feedback, and then decides upon subsequent ad displays to maximize click-through rates. Crucially, unlike the previous setting (where the learners were the agents), the only way the server can acquire feedback about the environment is by interacting with the agents (some of whom might be adversarial). *In other words, the server does not have at its disposal the trivial option of not interacting with the agents, and acting alone.* Since in this setting, the server is forced to interact with potentially corrupted agents, $\alpha < 0.5$ is the only scenario that can lead to meaningful regret bounds. More precisely, when $\alpha > 0.5$, there is no hope of achieving sub-linear regret, let alone benefiting from collaboration.

2.10. Analysis of RCLB: Proof of Theorem 1

In this section, we will prove Theorem 1. We start with a standard result from robust statistics on the guarantees afforded by the median operator for robust mean estimation of univariate Gaussian random variables; see, for instance, (Lai et al., 2016).

Lemma 2. *Consider a set $\mathcal{S} = \{x_1, \ldots, x_M\}$ of M samples partitioned as $\mathcal{S} = \mathcal{S}_g \cup \mathcal{S}_b$, such that (i) all the samples in $\mathcal{S}_g$ are drawn i.i.d. from $\mathcal{N}(\mu, \sigma^2)$, where $\mu, \sigma^2 \in \mathbb{R}$; (ii) the samples in $\mathcal{S}_b$ are chosen by an adversary, and can be arbitrary; and (iii) $|\mathcal{S}_b| < \alpha|\mathcal{S}|$, where $\alpha < 1/2$. Let $\hat{\mu} = \texttt{Median}(\{x_i\}, i \in [M])$. Given any $\delta \in (0,1)$, we then have that with probability at least $1-\delta$,*

$$|\hat{\mu} - \mu| \leq C\left(\alpha + \sqrt{\frac{\log(\frac{1}{\delta})}{M}}\right)\sigma, \tag{2.15}$$

where C is a suitably large universal constant.

The next key lemma - a restatement of Lemma 1 in the main body of the paper - informs us about the quality of the robust mean payoffs computed in line 8 of Algorithm 1. Before proceeding to prove this result, we define by $\mathcal{F}_\ell$ the σ-algebra generated by all the actions and rewards up to the beginning of epoch ℓ.

Lemma 3. ***(Robust Confidence Intervals)*** *Fix any epoch ℓ. For each active arm $b \in \mathcal{A}_\ell$, the following holds with probability at least $1 - \delta_\ell$:*

$$|\mu_b^{(\ell)} - \langle \theta_*, b \rangle| \leq \gamma_\ell, \;\; \textit{where} \;\; \gamma_\ell = \sqrt{2}C\left(1 + \alpha\sqrt{M}\right)\epsilon_\ell, \tag{2.16}$$

where C is as in Lemma 2.

Proof. Fix an epoch ℓ, an active arm $b \in \mathcal{A}_\ell$, and a good agent $i \in [M] \setminus \mathcal{B}$. We start by analyzing

the statistics of the quantity $\langle \hat{\theta}_i^{(\ell)}, b \rangle$. From the definition of $\hat{\theta}_i^{(\ell)}$ and $\tilde{V}_\ell$ in Eq.(2.3), we have

$$\begin{aligned}
\hat{\theta}_i^{(\ell)} &= \tilde{V}_\ell^{-1} Y_{i,\ell} \\
&= \tilde{V}_\ell^{-1} \left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} r_{i,a}^{(\ell)} a \right) \\
&= \tilde{V}_\ell^{-1} \left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \left(\langle \theta_*, a \rangle + \bar{\eta}_{i,a}^{(\ell)} \right) a \right) \\
&= \theta_* + \tilde{V}_\ell^{-1} \left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} a \right).
\end{aligned} \tag{2.17}$$

For the third equality above, we used the observation model (2.1), and denoted by $\bar{\eta}_{i,a}^{(\ell)}$ the average of the noise terms associated with the rewards observed by agent i during phase ℓ for arm a. From (2.17), we then have

$$\langle \hat{\theta}_i^{(\ell)}, b \rangle = \langle \theta_*, b \rangle + \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} \langle \tilde{V}_\ell^{-1} a, b \rangle. \tag{2.18}$$

Now conditioned on $\mathcal{F}_\ell$, the only randomness in the above equation corresponds to the noise terms $\{\bar{\eta}_{i,a}^{(\ell)}\}_{a \in \text{Supp}(\pi_\ell)}$. Furthermore, based on our noise model, it is clear that $\bar{\eta}_{i,a}^{(\ell)} \sim \mathcal{N}(0, 1/m_a^{(\ell)})$ for each $a \in \text{Supp}(\pi_\ell)$. It then follows that

$$\mathbb{E}\left[\langle \hat{\theta}_i^{(\ell)}, b \rangle | \mathcal{F}_\ell \right] = \langle \theta_*, b \rangle.$$

We also have

$$\begin{aligned}
\mathbb{E}\left[\left(\langle \hat{\theta}_i^{(\ell)} - \theta_*, b \rangle \right)^2 | \mathcal{F}_\ell \right] &= \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \left(\langle \tilde{V}_\ell^{-1} a, b \rangle \right)^2 \\
&= b' \tilde{V}_\ell^{-1} \left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} a a' \right) \tilde{V}_\ell^{-1} b \\
&= \|b\|^2_{\tilde{V}_\ell^{-1}},
\end{aligned} \tag{2.19}$$

where we used the fact that the noise terms are independent across arms. We conclude that conditioned on $\mathcal{F}_\ell$,

$$\langle \hat{\theta}_i^{(\ell)}, b \rangle \sim \mathcal{N}\left(\langle \theta_*, b \rangle, \|b\|^2_{\tilde{V}_\ell^{-1}} \right).$$

In each epoch ℓ, the server has access to a set $\mathcal{S}_b^{(\ell)} = \{\langle \hat{\theta}_i^{(\ell)}, b\rangle\}_{i\in[M]}$, where the samples corresponding to agents in $[M] \setminus \mathcal{B}$ are independent and identically distributed as per the distribution above. Moreover, at most $\alpha \in [0, 1/2)$ fraction of the samples in $\mathcal{S}_b^{(\ell)}$ are corrupted. Recalling that $\mu_b^{(\ell)} = \texttt{Median}\left(\{\langle \hat{\theta}_i^{(\ell)}, b\rangle, i \in [M]\}\right)$, and using Lemma 2, we immediately observe that conditioned on $\mathcal{F}_\ell$, with probability at least $1 - \delta_\ell$,

$$|\mu_b^{(\ell)} - \langle \theta_*, b\rangle| \le C\left(\alpha + \sqrt{\frac{\log(\frac{1}{\delta_\ell})}{M}}\right) \|b\|_{\tilde{V}_\ell^{-1}}. \tag{2.20}$$

We now proceed to bound the term $\|b\|_{\tilde{V}_\ell^{-1}}$. To that end, let us start by noting that

$$\begin{aligned}
\tilde{V}_\ell &= \sum_{a\in \text{Supp}(\pi_\ell)} m_a^{(\ell)} aa' \\
&= \sum_{a\in \text{Supp}(\pi_\ell)} \left\lceil \frac{T_a^{(\ell)}}{M} \right\rceil aa' \\
&\succcurlyeq \frac{1}{M} \sum_{a\in \text{Supp}(\pi_\ell)} T_a^{(\ell)} aa' \\
&\succcurlyeq \frac{d}{M\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \sum_{a\in \text{Supp}(\pi_\ell)} \pi_\ell(a) aa' \\
&= \frac{d}{M\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \sum_{a\in \mathcal{A}_\ell} \pi_\ell(a) aa' \\
&= \frac{d}{M\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) V_\ell(\pi_\ell).
\end{aligned} \tag{2.21}$$

Thus, we have

$$\tilde{V}_\ell^{-1} \preccurlyeq \frac{M\epsilon_\ell^2}{d\log\left(\frac{1}{\delta_\ell}\right)} V_\ell^{-1}(\pi_\ell).$$

Using the above bound, we proceed as follows.

$$\begin{aligned}
\|b\|_{\tilde{V}_\ell^{-1}} &= \sqrt{b'\tilde{V}_\ell^{-1}b} \\
&\leq \epsilon_\ell \sqrt{\frac{M}{d\log\left(\frac{1}{\delta_\ell}\right)}} \sqrt{b'V_\ell^{-1}(\pi_\ell)b} \\
&\leq \epsilon_\ell \sqrt{\frac{M}{d\log\left(\frac{1}{\delta_\ell}\right)}} \sqrt{\max_{a\in\mathcal{A}_\ell} \|a\|^2_{V_\ell^{-1}(\pi_\ell)}} \\
&\overset{(a)}{=} \epsilon_\ell \sqrt{\frac{M}{d\log\left(\frac{1}{\delta_\ell}\right)}} \sqrt{g_\ell(\pi_\ell)} \\
&\overset{(b)}{\leq} \epsilon_\ell \sqrt{\frac{2M}{\log\left(\frac{1}{\delta_\ell}\right)}}.
\end{aligned} \tag{2.22}$$

In the above steps, we used the definition of $g_\ell(\pi_\ell)$ for (a); for (b), we used the fact that based on the approximate G-optimal design problem solved by the server in line 1 of `RCLB`, $g_\ell(\pi_\ell) \leq 2d$. Plugging the bound from (2.22) into (2.20), and using the fact that $\log\left(\frac{1}{\delta_\ell}\right) \geq 1$, we have that

$$\mathbb{P}\left(|\mu_b^{(\ell)} - \langle\theta_*, b\rangle| \geq \gamma_\ell | \mathcal{F}_\ell\right) \leq \delta_\ell. \tag{2.23}$$

Consider the following event $\mathcal{E}_\ell \triangleq \{|\mu_b^{(\ell)} - \langle\theta_*, b\rangle| \geq \gamma_\ell\}$. Now observe that

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_\ell) &= \mathbb{E}[\mathbf{1}_{\mathcal{E}_\ell}] \\
&= \mathbb{E}\left[\mathbb{E}[\mathbf{1}_{\mathcal{E}_\ell}|\mathcal{F}_\ell]\right] \\
&= \mathbb{E}\left[\mathbb{P}(\mathcal{E}_\ell|\mathcal{F}_\ell)\right] \\
&\leq \delta_\ell,
\end{aligned} \tag{2.24}$$

where we used $\mathbf{1}_{\mathcal{E}_\ell}$ to denote an indicator random variable associated with the event $\mathcal{E}_\ell$; also, for the last line, we used (2.23). □

In the following two results, we use the robust confidence intervals from Lemma 3 to construct clean events that hold with high probability on which (i) the optimal arm a_* is never eliminated (Lemma

4); and (ii) any arm retained in epoch ℓ contributes at most $O(\gamma_\ell)$ regret in each time-step within epoch ℓ (Lemma 5). To proceed, for each $a \in \mathcal{A}$, define the arm-gap $\Delta_a = \langle \theta_*, a_* - a \rangle$.

Lemma 4. *Define the event $\mathcal{G}_1 \triangleq \{a_* \in \mathcal{A}_\ell, \forall \ell \in [L]\}$, where L is the total number of epochs. It then holds that $\mathbb{P}(\mathcal{G}_1) \geq 1 - 4\bar{\delta}$.*

Proof. Based on the arm-elimination criterion in line 9 of Algorithm 1, it follows that $\{a_* \in \mathcal{A}_\ell, a_* \notin \mathcal{A}_{\ell+1}\} \implies \{\exists b \in \mathcal{A}_\ell : \mu_b^{(\ell)} - \mu_{a_*}^{(\ell)} > 2\gamma_\ell\}$. Now for any fixed $b \in \mathcal{A}_\ell$, we have

$$\begin{aligned} & \mu_b^{(\ell)} - \mu_{a_*}^{(\ell)} > 2\gamma_\ell \\ & \implies \left(\mu_b^{(\ell)} - \langle \theta_*, b \rangle\right) + \left(\langle \theta_*, a_* \rangle - \mu_{a_*}^{(\ell)}\right) > 2\gamma_\ell + \Delta_b \\ & \implies \left(\mu_b^{(\ell)} - \langle \theta_*, b \rangle\right) + \left(\langle \theta_*, a_* \rangle - \mu_{a_*}^{(\ell)}\right) > 2\gamma_\ell, \end{aligned} \tag{2.25}$$

where for the second step, we used the fact that $\Delta_b \geq 0$. Thus, the event $\{\mu_b^{(\ell)} - \mu_{a_*}^{(\ell)} > 2\gamma_\ell\}$ implies the occurrence of either $\{\mu_b^{(\ell)} - \langle \theta_*, b \rangle > \gamma_\ell\}$ or $\{\langle \theta_*, a_* \rangle - \mu_{a_*}^{(\ell)} > \gamma_\ell\}$. From Lemma 3, we further know that the probability of each of these latter events is at most δ_ℓ. Putting these pieces together, and using an union bound, we have

$$\begin{aligned} \mathbb{P}(\mathcal{G}_1^c) &\leq \sum_{\ell \in [L]} \mathbb{P}(a^* \in \mathcal{A}_\ell, a^* \notin \mathcal{A}_{\ell+1}) \\ &\leq \sum_{\ell \in [L]} \mathbb{P}(\exists b \in \mathcal{A}_\ell : \mu_b^{(\ell)} - \mu_{a_*}^{(\ell)} > 2\gamma_\ell) \\ &\leq 2K \sum_{\ell \in [L]} \delta_\ell \\ &= 2K \sum_{\ell \in [L]} \frac{\bar{\delta}}{K\ell^2} \\ &\leq 2 \sum_{\ell=1}^{\infty} \frac{\bar{\delta}}{\ell^2} \\ &\leq 2\bar{\delta} \int_{x=1}^{\infty} \frac{1}{x^2}\, dx \leq 4\bar{\delta}. \end{aligned} \tag{2.26}$$

This completes the proof. □

In our next result, we work towards bounding the regret incurred from playing each active arm in a given epoch.

Lemma 5. *Consider any arm $a \in \mathcal{A} \setminus \{a_*\}$. Let ℓ_a be defined as $\ell_a \triangleq \min\{\ell : \gamma_\ell < \frac{\Delta_a}{4}\}$. It then holds that $\mathbb{P}(a \in \mathcal{A}_{\ell_a+1}) \leq 6\bar{\delta}$.*

Proof. Let us start by observing that

$$\begin{aligned}
\mathbb{P}(a \in \mathcal{A}_{\ell_a+1}) &= \mathbb{P}(a \in \mathcal{A}_{\ell_a+1}, a^* \in \mathcal{A}_{\ell_a}) + \mathbb{P}(a \in \mathcal{A}_{\ell_a+1}, a^* \notin \mathcal{A}_{\ell_a}) \\
&\leq \mathbb{P}(a \in \mathcal{A}_{\ell_a+1}, a^* \in \mathcal{A}_{\ell_a}) + \mathbb{P}(a^* \notin \mathcal{A}_{\ell_a}) \\
&\leq \mathbb{P}(a \in \mathcal{A}_{\ell_a}, a \in \mathcal{A}_{\ell_a+1}, a^* \in \mathcal{A}_{\ell_a}) + 4\bar{\delta},
\end{aligned} \tag{2.27}$$

where for the last step, we used the fact that $\{a \in \mathcal{A}_{\ell_a+1}\} \implies \{a \in \mathcal{A}_{\ell_a}\}$, and Lemma 4. Now, to bound $\mathbb{P}(a \in \mathcal{A}_{\ell_a}, a \in \mathcal{A}_{\ell_a+1}, a^* \in \mathcal{A}_{\ell_a})$, we note based on line 9 of `RCLB` that

$$\begin{aligned}
\mathbb{P}(a \in \mathcal{A}_{\ell_a}, a \in \mathcal{A}_{\ell_a+1}, a^* \in \mathcal{A}_{\ell_a}) &\leq \mathbb{P}\left(\max_{b \in \mathcal{A}_{\ell_a}} \mu_b^{(\ell_a)} - \mu_a^{(\ell_a)} \leq 2\gamma_{\ell_a}\right) \\
&\leq \mathbb{P}\left(\mu_{a_*}^{(\ell_a)} - \mu_a^{(\ell_a)} \leq 2\gamma_{\ell_a}\right) \\
&\leq \mathbb{P}\left(\left(\mu_a^{(\ell_a)} - \langle \theta_*, a \rangle\right) + \left(\langle \theta_*, a_* \rangle - \mu_{a_*}^{(\ell_a)}\right) > \Delta_a - 2\gamma_{\ell_a}\right) \\
&\leq \mathbb{P}\left(\mu_a^{(\ell_a)} - \langle \theta_*, a \rangle > \gamma_{\ell_a}\right) + \mathbb{P}\left(\langle \theta_*, a_* \rangle - \mu_{a_*}^{(\ell_a)} > \gamma_{\ell_a}\right) \\
&\leq 2\delta_{\ell_a},
\end{aligned} \tag{2.28}$$

where we used $\Delta_a > 4\gamma_{\ell_a}$ for the second last step, and Lemma 3 for the last step. Noting that $\delta_{\ell_a} \leq \bar{\delta}$, and combining the bounds in equations (2.27) and (2.28) leads to the claim of the lemma. □

We are now in place to prove Theorem 1.

Proof. (**Proof of Theorem 1**) We start by constructing an appropriate clean event $\mathcal{E}$ for our subsequent analysis. Accordingly, let us define:

$$\mathcal{E} = \{a_* \in \mathcal{A}_\ell, \forall \ell \in [L]\} \bigcap \{\cap_{a \in \mathcal{A} \setminus \{a_*\}} \{a \notin \mathcal{A}_{\ell_a+1}\}\}. \tag{2.29}$$

Based on Lemmas 4 and 5, we then have

$$\begin{aligned}\mathbb{P}(\mathcal{E}^c) &\leq 4\bar{\delta} + \sum_{a \in \mathcal{A} \setminus \{a_*\}} \mathbb{P}(a \in \mathcal{A}_{\ell_a+1}) \\ &\leq 4\bar{\delta} + 6\bar{\delta}K \\ &\leq 10\bar{\delta}K = \delta,\end{aligned} \tag{2.30}$$

as per the choice of $\bar{\delta}$ in Theorem 1. Thus, $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$. Throughout the rest of the proof, we will condition on the clean event $\mathcal{E}$. Based on the definition of the event $\mathcal{E}$, it is easy to see that for any epoch $\ell \in [L]$, $a \in \mathcal{A}_\ell \implies \Delta_a \leq 8\gamma_\ell$. Using this key fact, we now proceed to bound the regret of any good agent $i \in [M] \setminus \mathcal{B}$.

$$\begin{aligned}\sum_{t=1}^{T} \langle \theta_*, a_* - a_{i,t} \rangle &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \langle \theta_*, a_* - a \rangle \\ &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \left\lceil \frac{T_a^{(\ell)}}{M} \right\rceil \Delta_a \\ &\leq \underbrace{\sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \frac{T_a^{(\ell)}}{M} \Delta_a}_{T_1} + \underbrace{\sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \Delta_a}_{T_2}.\end{aligned} \tag{2.31}$$

We now bound T_1 and T_2 separately. For bounding T_1, we have:

$$\begin{aligned}
T_1 &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \frac{T_a^{(\ell)}}{M} \Delta_a \\
&= \frac{d}{M} \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \sum_{a \in \text{Supp}(\pi_\ell)} \pi_\ell(a) \Delta_a \\
&\leq \frac{8d}{M} \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \gamma_\ell \\
&= \frac{8\sqrt{2}C(1+\alpha\sqrt{M})d}{M} \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell} \log\left(\frac{10K^2\ell^2}{\delta}\right) \\
&\leq \frac{8\sqrt{2}C(1+\alpha\sqrt{M})d}{M} \log\left(\frac{10K^2L^2}{\delta}\right) \sum_{\ell=1}^{L} 2^\ell \\
&= O\left(\frac{(1+\alpha\sqrt{M})d}{M} \log\left(\frac{10K^2L^2}{\delta}\right) 2^L\right).
\end{aligned} \tag{2.32}$$

In the third step above, we used $\sum_{a \in \text{Supp}(\pi_\ell)} \pi_\ell(a) = 1$. We now need an upper-bound on the term 2^L. To that end, notice that the length of the horizon T is bounded below by the length of the last epoch, i.e., the L-th epoch. Moreover, the duration of the L-th epoch corresponds to the number of arm-pulls made by any single good agent during the L-th epoch. We thus have:

$$\begin{aligned}
T &\geq \sum_{a \in \text{Supp}(\pi_L)} \frac{T_a^{(L)}}{M} \\
&\geq \frac{4^L d}{M} \sum_{a \in \text{Supp}(\pi_L)} \pi_L(a) \log\left(\frac{1}{\delta_L}\right) \\
&= \frac{4^L d}{M} \log\left(\frac{10K^2L^2}{\delta}\right) \sum_{a \in \text{Supp}(\pi_L)} \pi_L(a) \\
&= \frac{4^L d}{M} \log\left(\frac{10K^2L^2}{\delta}\right).
\end{aligned} \tag{2.33}$$

Thus, $2^L \leq \sqrt{MT/(d \log(10K^2L^2/\delta))}$. Plugging this bound in (2.32), we obtain

$$T_1 = O\left(\left(\alpha + \frac{1}{\sqrt{M}}\right) \sqrt{\log\left(\frac{KT}{\delta}\right) dT}\right) = \tilde{O}\left(\left(\alpha + \frac{1}{\sqrt{M}}\right) \sqrt{dT}\right).$$

As for the term T_2, we have

$$\begin{aligned} T_2 &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \Delta_a \\ &\leq 8 \sum_{\ell=1}^{L} \gamma_\ell |\text{Supp}(\pi_\ell)| \\ &\overset{(a)}{\leq} 384\sqrt{2} C d \log\log d \left(1 + \alpha\sqrt{M}\right) \sum_{\ell=1}^{L} 2^{-\ell} \\ &= O\left(d \log\log d \left(1 + \alpha\sqrt{M}\right)\right) \\ &\overset{(b)}{=} \tilde{O}\left(\left(\alpha + \frac{1}{\sqrt{M}}\right)\sqrt{dT}\right). \end{aligned} \tag{2.34}$$

In the above steps, for (a), recall from line 1 of `RCLB` that $|\text{Supp}(\pi_\ell)| \leq 48 d \log\log d$ based on the approximate G-optimal design computation. For (b), we used the fact that by assumption, $T \geq Md$. Combining the bounds on T_1 and T_2, and recalling that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$, we have that with probability at least $1 - \delta$,

$$\sum_{t=1}^{T} \langle \theta_*, a_* - a_{i,t} \rangle = O\left(\left(\alpha + \frac{1}{\sqrt{M}}\right)\sqrt{\log\left(\frac{KT}{\delta}\right) dT}\right) = \tilde{O}\left(\left(\alpha + \frac{1}{\sqrt{M}}\right)\sqrt{dT}\right). \tag{2.35}$$

This concludes the proof. □

The following is an immediate corollary of Theorem 1 on the group regret R_T.

Corollary 1. *(**Bound on Group Regret**) Under the conditions of Theorem 1, we have:*

$$R_T = \tilde{O}\left(\left(\alpha M + \sqrt{M}\right)\sqrt{dT}\right). \tag{2.36}$$

Proof. (**Proof of Corollary 1**) Recall from the proof of Theorem 1 that there exists a clean event $\mathcal{E}$ of measure at least $1 - \delta$ on which the regret of every good agent is bounded above as per Eq. (2.35). Let $\tilde{C}$ be an upper bound on the maximum instantaneous regret, i.e.,

$$\max_{a \in \mathcal{A}} \langle \theta_*, a_* - a \rangle \leq \tilde{C}.$$

Now set $\delta = \frac{1}{MT}$ and observe that:

$$\begin{aligned}
R_T &= \mathbb{E}\left[\sum_{i\in[M]\setminus\mathcal{B}}\sum_{t=1}^{T}\langle\theta_*, a_* - a_{i,t}\rangle\right] \\
&= \mathbb{E}\left[\sum_{i\in[M]\setminus\mathcal{B}}\sum_{t=1}^{T}\langle\theta_*, a_* - a_{i,t}\rangle\Big|\mathcal{E}\right]\mathbb{P}(\mathcal{E}) + \mathbb{E}\left[\sum_{i\in[M]\setminus\mathcal{B}}\sum_{t=1}^{T}\langle\theta_*, a_* - a_{i,t}\rangle\Big|\mathcal{E}^c\right]\mathbb{P}(\mathcal{E}^c) \\
&\leq C_1\left(\alpha M + \sqrt{M}\right)\sqrt{\log\left(KMT\right)dT} + \tilde{C}MT \times \frac{1}{MT} \\
&= O\left(\left(\alpha M + \sqrt{M}\right)\sqrt{\log\left(KMT\right)dT}\right) \\
&= \tilde{O}\left(\left(\alpha M + \sqrt{M}\right)\sqrt{dT}\right).
\end{aligned} \tag{2.37}$$

In the above steps, C_1 is a suitably large universal constant. □

2.11. Lower Bound Analysis: Proof of Theorem 2

In this section, we will prove Theorem 2. Before diving into the technical details, we remind the reader that we consider a slightly different adversarial model from the one considered throughout the paper. In this modified model, with probability α, each agent is adversarial independently of the other agents. We will consider a class of policies Π where at each time-step, the same action (decided by the server) is played by every agent. Thus, the regret incurred by any individual agent is the same as the regret incurred by the server. Finally, to prove the lower bound, we will focus on a class of 2-armed bandits where the reward distribution of each arm is Gaussian with unit variance; such a class of bandits is succinctly denoted by $\mathcal{E}_{\mathcal{N}}^{(2)}(1)$.

We will have occasion to use the following result (Lattimore and Szepesvári, 2020, Theorem 14.2).

Lemma 6. *(**Bretagnolle-Huber Inequality**) Let P and Q be two probability measures on the same measurable space $(\Omega, \mathcal{F})$, and let $A \in \mathcal{F}$ be any arbitrary event. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2}\exp\left(-KL(P,Q)\right),$$

where A^c is the complement of the event A, and $KL(P,Q)$ is the Kullback-Leibler distance between

P and Q.

Our proof comprises of two main steps. First, we construct two *distinct* bandit instances within the class $\mathcal{E}_{\mathcal{N}}^{(2)}(1)$ such that the two instances - although different - appear *identical* to the server. Second, we devise an attack strategy and argue that regardless of the policy played by the server, it will end up suffering a regret of $\Omega(\alpha\sqrt{T})$ upon interacting with at least one of the two instances; here, T is the horizon for our problem. We now elaborate on these two steps.

• **Step 1. Construction of the two bandit instances.** We first take a detour and describe an idea that is typically used to prove lower bounds for the robust mean estimation literature (Chen et al., 2015; Lai et al., 2016). It will soon be apparent how such an idea can be exploited to construct the two bandit instances for our problem. We show that there are two univariate Gaussian distributions $P_1 = \mathcal{N}(\mu_1, 1)$, $P_2 = \mathcal{N}(\mu_2, 1)$, and two T-dimensional distributions Q_1, Q_2, such that $\|\mu_1 - \mu_2\|_2 = \Omega(\alpha/\sqrt{T})$, and:

$$(1-\alpha)P_1^T + \alpha Q_1 = (1-\alpha)P_2^T + \alpha Q_2, \tag{2.38}$$

where P_1^T (resp., P_2^T) is the joint distribution of T i.i.d. samples drawn from P_1 (resp., P_2). Clearly, P_1^T (resp., P_2^T) is equivalent to a T-dimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_1, I_T)$ (resp., $\mathcal{N}(\boldsymbol{\mu}_2, I_T)$), where $\boldsymbol{\mu}_1$ (resp., $\boldsymbol{\mu}_2$) is a T-dimensional vector with each entry equal to μ_1 (resp., μ_2). Let ϕ_1 be the p.d.f. of P_1^T and ϕ_2 be the p.d.f. of P_2^T. Next, let μ_1 and μ_2 be chosen such that the total variation distance $\delta(P_1^T, P_2^T)$ between P_1^T and P_2^T is

$$\frac{1}{2}\int \|\phi_1 - \phi_2\|_1 dx = \frac{\alpha}{1-\alpha}.$$

Let Q_1 be the distribution with p.d.f. $\frac{1-\alpha}{\alpha}(\phi_2 - \phi_1)\mathbf{1}_{\phi_2 \geq \phi_1}$, and Q_2 be the distribution with p.d.f. $\frac{1-\alpha}{\alpha}(\phi_1 - \phi_2)\mathbf{1}_{\phi_1 \geq \phi_2}$. With such a construction of Q_1 and Q_2, one can verify that the equality in Eq. (2.38) is satisfied; see, for instance, the arguments in Appendix E of (Chen et al., 2015). Now

from Pinsker's inequality, we know that:

$$\sqrt{\frac{1}{2}\mathrm{KL}(P_1^T, P_2^T)} \geq \delta(P_1^T, P_2^T) = \frac{\alpha}{1-\alpha},$$

where we used $\mathrm{KL}(P_1^T, P_2^T)$ to denote the Kullback-Leibler distance between P_1^T and P_2^T. We conclude that:

$$\frac{1}{2}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 = \mathrm{KL}(P_1^T, P_2^T) \geq 2\left(\frac{\alpha}{1-\alpha}\right)^2.$$

This, in turn, implies

$$\sqrt{T}\,(|\mu_1 - \mu_2|) = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \frac{2\alpha}{1-\alpha} \geq 2\alpha.$$

We have thus shown that there exist $\mu_1, \mu_2 \in \mathbb{R}$, satisfying $|\mu_1 - \mu_2| \geq 2\alpha/\sqrt{T}$, such that with $P_1 = \mathcal{N}(\mu_1, 1)$, $P_2 = \mathcal{N}(\mu_2, 1)$, one can satisfy Eq. (2.38) with appropriately chosen T-dimensional distributions Q_1 and Q_2. With these ideas in place, we now return to our bandit problem.

Without loss of generality, suppose $\mu_2 > \mu_1$, where μ_2 and μ_1 are as in the construction described above. Let us construct two bandit instances ν and ν', each involving two arms, i.e., $\mathcal{A} = \{a_1, a_2\}$. Let $r_{a_k}^{(\nu)}$ and $r_{a_k}^{(\nu')}$ denote the reward distribution of arm $a_k, k \in \{1, 2\}$, in instance ν and instance ν', respectively. These reward distributions are chosen as follows.

$$\begin{aligned}
&\text{Instance } \nu : r_{a_1}^{(\nu)} \sim \mathcal{N}\left(\frac{\mu_1+\mu_2}{2}, 1\right);\ \ r_{a_2}^{(\nu)} \sim \mathcal{N}(\mu_1, 1) \\
&\text{Instance } \nu' : r_{a_1}^{(\nu')} \sim \mathcal{N}\left(\frac{\mu_1+\mu_2}{2}, 1\right);\ \ r_{a_2}^{(\nu')} \sim \mathcal{N}(\mu_2, 1).
\end{aligned} \tag{2.39}$$

Thus, the distribution of the first arm is the same in both instances. However, as $\mu_2 > \mu_1$, the first arm is the best arm in the first instance while the opposite is true for the second instance. The attack strategy for the adversarial agents will be dictated by the distributions Q_1 and Q_2 in a manner to be described shortly.

• **Step 2. The attack strategy and regret analysis.** Inspired by the argument in the proof of (Bubeck et al., 2013, Theorem 5), we consider a full information setting where the server has access to t reward samples from *each* arm from *each* agent at time-step t. Since this full information setting

is simpler than the bandit setting, a lower bound for the former implies one for the latter.

Here is the attack strategy. Suppose an agent is adversarial (which happens with probability α). In either instance, for arm 1, it reports samples from the true distribution for arm 1 corresponding to that instance. In other words, the reward samples for arm 1 are not corrupted by the agent. As for arm 2, in instance ν (resp., ν'), the first t reward samples (where $t \in [T]$) corresponding to a_2 are generated from Q_1^t (resp., Q_2^t) by the adversarial agent. Here, Q_1^t (resp., Q_2^t) is the marginal of the T-dimensional distribution Q_1 (resp., Q_2) corresponding to the first t components. To sum up, in instance ν, the joint distribution $D_{a_1}^{(\nu)}$ of rewards for a_1 over the horizon T, as seen by the server from any given agent, is the T-dimensional Gaussian distribution $\mathcal{N}(\bar{\boldsymbol{\mu}}, I_T)$, where $\bar{\boldsymbol{\mu}}$ is a T-dimensional vector with each component equal to $\frac{\mu_1+\mu_2}{2}$. Based on our discussion above, $D_{a_1}^{(\nu)} = D_{a_1}^{(\nu')}$. Let $D_{a_2}^{(\nu)}$ and $D_{a_2}^{(\nu')}$ have analogous meanings for arm a_2. Then, we have:

$$D_{a_2}^{(\nu)} = (1-\alpha)P_1^T + \alpha Q_1; \;\; D_{a_2}^{(\nu')} = (1-\alpha)P_2^T + \alpha Q_2. \tag{2.40}$$

In light of Eq. (2.38), however, we have $D_{a_2}^{(\nu)} = D_{a_2}^{(\nu')}$. Essentially, what we have established is the following: *the joint distribution of rewards for each arm over the horizon T, as seen by the server from each agent, is identical for both instances.*

In what follows, given any two distributions D_1 and D_2, let $D_1 \otimes D_2$ represent their product distribution. Since the rewards across arms are independent, the joint distribution of rewards for both arms is given by $D^{(\nu)} \triangleq D_{a_1}^{(\nu)} \otimes D_{a_2}^{(\nu)}$ in instance ν, and $D^{(\nu')} \triangleq D_{a_1}^{(\nu')} \otimes D_{a_2}^{(\nu')}$ in instance ν'. Since rewards across agents are independent, the joint distributions of rewards from *all* agents, as seen by the server in each of the two instances, are given by:

$$D_M^{(\nu)} \triangleq \underbrace{D^{(\nu)} \otimes D^{(\nu)} \otimes \cdots \otimes D^{(\nu)}}_{M\text{-fold product distribution}}; \;\; D_M^{(\nu')} \triangleq \underbrace{D^{(\nu')} \otimes D^{(\nu')} \otimes \cdots \otimes D^{(\nu')}}_{M\text{-fold product distribution}}.$$

Let $\mathbb{E}_\nu[\cdot]$ (resp., $\mathbb{E}_{\nu'}[\cdot]$) represent the expectation operation w.r.t. the measure $D_M^{(\nu)}$ (resp., $D_M^{(\nu')}$). Let us use $\mathbb{P}_\nu$ as a shorthand for $D_M^{(\nu)}$, and $\mathbb{P}_{\nu'}$ as a shorthand for $D_M^{(\nu')}$. Furthermore, let $n_k(T)$

be the random variable representing the total number of times arm $a_k, k \in \{1, 2\}$, is chosen by the server over the horizon T. Finally, recall that $R_T^{(s)}(\nu)$ (resp., $R_T^{(s)}(\nu')$) is the regret incurred by the server upon interaction with instance ν (resp., instance ν').

In instance ν, each time arm 2 is chosen by the server, it incurs an instantaneous regret of $(\mu_2 - \mu_1)/2$. We thus have:

$$R_T^{(s)}(\nu) = \left(\frac{\mu_2 - \mu_1}{2}\right)(T - \mathbb{E}_\nu[n_1(T)]) \geq \frac{(\mu_2 - \mu_1)T}{4}\mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right). \tag{2.41}$$

To see why the latter inequality is true, observe:

$$\begin{aligned}
\mathbb{E}_\nu[n_1(T)] &= \mathbb{E}_\nu\left[n_1(T)|n_1(T) \leq \frac{T}{2}\right]\mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right) + \mathbb{E}_\nu\left[n_1(T)|n_1(T) > \frac{T}{2}\right]\mathbb{P}_\nu\left(n_1(T) > \frac{T}{2}\right) \\
&\leq \frac{T}{2}\mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right) + T\left(1 - \mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right)\right) \\
&= T - \frac{T}{2}\mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right).
\end{aligned} \tag{2.42}$$

In instance ν', each time arm 1 is chosen by the server, it incurs an instantaneous regret of $(\mu_2 - \mu_1)/2$. We thus have:

$$R_T^{(s)}(\nu') = \left(\frac{\mu_2 - \mu_1}{2}\right)\mathbb{E}_{\nu'}[n_1(T)] \geq \frac{(\mu_2 - \mu_1)T}{4}\mathbb{P}_{\nu'}\left(n_1(T) > \frac{T}{2}\right). \tag{2.43}$$

Combining Eq. (2.41) and Eq. (2.43) yields:

$$\begin{aligned}
\max\{R_T^{(s)}(\nu), R_T^{(s)}(\nu')\} &\geq \frac{1}{2}\left(R_T^{(s)}(\nu) + R_T^{(s)}(\nu')\right) \\
&\geq \frac{(\mu_2 - \mu_1)T}{8}\left(\mathbb{P}_\nu\left(n_1(T) \leq \frac{T}{2}\right) + \mathbb{P}_{\nu'}\left(n_1(T) > \frac{T}{2}\right)\right) \\
&\overset{(a)}{\geq} \frac{(\mu_2 - \mu_1)T}{16}\exp\left(-\mathrm{KL}\left(\mathbb{P}_\nu, \mathbb{P}_{\nu'}\right)\right) \\
&\overset{(b)}{\geq} \frac{\alpha\sqrt{T}}{8}\exp\left(-\mathrm{KL}\left(\mathbb{P}_\nu, \mathbb{P}_{\nu'}\right)\right) \\
&\overset{(c)}{=} \frac{\alpha\sqrt{T}}{8}.
\end{aligned} \tag{2.44}$$

In the above steps, we used the Bretagnolle-Huber inequality (namely, Lemma 6) for (a); for (b), we used the fact that μ_2 and μ_1 were chosen in Step 1 to satisfy $\mu_2 - \mu_1 \geq 2\alpha/\sqrt{T}$; and for (c), we used $\mathrm{KL}\left(\mathbb{P}_\nu, \mathbb{P}_{\nu'}\right) = 0$. To see why $\mathrm{KL}\left(\mathbb{P}_\nu, \mathbb{P}_{\nu'}\right) = 0$, we use the chain-rule for relative entropies to obtain:

$$\begin{aligned}
\mathrm{KL}\left(\mathbb{P}_\nu, \mathbb{P}_{\nu'}\right) &= \mathrm{KL}\left(D_M^{(\nu)}, D_M^{(\nu')}\right) \\
&= M\left(\mathrm{KL}\left(D^{(\nu)}, D^{(\nu')}\right)\right) \\
&= M\left(\mathrm{KL}\left(D_{a_1}^{(\nu)}, D_{a_1}^{(\nu')}\right) + \mathrm{KL}\left(D_{a_2}^{(\nu)}, D_{a_2}^{(\nu')}\right)\right) \\
&= 0,
\end{aligned} \tag{2.45}$$

where the last step is a consequence of the fact that $D_{a_1}^{(\nu)} = D_{a_1}^{(\nu')}$, and $D_{a_2}^{(\nu)} = D_{a_2}^{(\nu')}$. The claim of Theorem 2 follows from noting that the resulting lower bound in Eq. (2.44) holds regardless of the number of agents M.

■

Remark 3. *The proof of Theorem 2 above reveals a constructive attack strategy for the adversarial agents to follow. As future work, it would be interesting to see if similar ideas can be extended to more general reinforcement learning problems.*

2.12. Algorithms and Analysis for the Generalized Linear Bandit Model

In this section, we first provide a detailed outline of the `RC-GLM` algorithm introduced in Section 2.5; see Algorithm 4. We then proceed to analyze `RC-GLM`, and provide a proof for Theorem 3. Finally, since `RC-GLM` uses the iteratively reweighted mean estimator from (Dalalyan and Minasyan, 2022) as a sub-routine, we also provide a description of this estimator to keep the paper self-contained; this description, however, is deferred to the end of the section. We start by reminding the reader that the non-linear observation model of interest to us in this section is as follows:

$$y_{i,t} = \mu\left(\langle \theta_*, a_{i,t} \rangle\right) + \eta_{i,t}, \tag{2.47}$$

where $\mu : \mathbb{R} \to \mathbb{R}$ is the link function. We also recall the definition of $h_\ell(\theta)$:

$$h_\ell(\theta) \triangleq \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \mu(\langle \theta, a \rangle) a, \forall \theta \in \Theta.$$

From comparing `RC-GLM` (Algorithm 4) to `RCLB` (Algorithm 1), we note that while both algorithms share the same general structure, the key difference between the two stems from the manner in which the robust confidence thresholds are computed. In particular, to tackle the difficulty posed by the non-linearity of the observation map, we first compute a robust estimate $\hat{\theta}^{(\ell)}$ of θ_* at the server - a route that we avoided in `RCLB` - and then use such an estimate to devise a phased elimination rule.

2.12.1. Proof of Theorem 3

In this section, we prove Theorem 3. The crux of the analysis lies in deriving a robust confidence bound akin to that in Lemma 1. To work towards such a result, we need to first go through a few intermediate steps; these are as follows.

Step 1. Prove that conditioned on $\mathcal{F}_\ell$, for each $i \in [M] \setminus \mathcal{B}$, $\tilde{V}_\ell^{-1/2} Y_{i,\ell}$ is a d-dimensional Gaussian random variable with mean $\tilde{V}_\ell^{-1/2} h_\ell(\theta_*)$, and covariance matrix $\Sigma = I_d$.[5]

Step 2. Use the result from Step 1, along with the error-bounds of the iteratively reweighted Gaussian mean estimator from (Dalalyan and Minasyan, 2022), to derive a high-probability error-bound on $\|X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\|$.

Step 3. Exploit regularity properties of the link function in tandem with the bounds from Step 2, to derive high-probability error bounds on $|\mu(\langle \hat{\theta}^{(\ell)}, a \rangle) - \mu(\langle \theta_*, a \rangle)|$, for each $a \in \mathcal{A}_\ell$.

We now proceed to formally establish each of the above steps, starting with step 1.

Lemma 7. *For each epoch ℓ, and each good agent $i \in [M] \setminus \mathcal{B}$, it holds that:*

$$\mathbb{E}\left[\tilde{V}_\ell^{-1/2} Y_{i,\ell} | \mathcal{F}_\ell\right] = \tilde{V}_\ell^{-1/2} h_\ell(\theta_*); \quad \textit{and} \tag{2.48}$$

[5] Recall that $\mathcal{F}_\ell$ is the σ-algebra generated by all the actions and rewards up to the beginning of epoch ℓ.

$$\mathbb{E}\left[\left(\tilde{V}_\ell^{-1/2}Y_{i,\ell} - \tilde{V}_\ell^{-1/2}h_\ell(\theta_*)\right)\left(\tilde{V}_\ell^{-1/2}Y_{i,\ell} - \tilde{V}_\ell^{-1/2}h_\ell(\theta_*)\right)' |\mathcal{F}_\ell\right] = I_d. \tag{2.49}$$

Proof. Fix an epoch ℓ, and a good agent $i \in [M] \setminus \mathcal{B}$. We start by observing that:

$$\begin{aligned} Y_{i,\ell} &= \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} r_{i,a}^{(\ell)} a \\ &= \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \left(\mu\left(\langle \theta_*, a\rangle\right) + \bar{\eta}_{i,a}^{(\ell)}\right) a \\ &= h_\ell(\theta_*) + \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} a, \end{aligned} \tag{2.50}$$

where for the last step, we used the definition of $h_\ell(\theta_*)$. Just as in the proof of Theorem 1, we have used $\bar{\eta}_{i,a}^{(\ell)}$ to denote the average of the noise terms associated with the rewards observed by agent i during phase ℓ for arm a. We thus have:

$$\tilde{V}_\ell^{-1/2}Y_{i,\ell} = \tilde{V}_\ell^{-1/2}h_\ell(\theta_*) + \tilde{V}_\ell^{-1/2}\left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} a\right).$$

Now conditioned on $\mathcal{F}_\ell$, the only randomness in the above equation stems from the noise terms $\{\bar{\eta}_{i,a}^{(\ell)}\}, a \in \text{Supp}(\pi_\ell)$, that are each zero-mean. The claim in Eq. (2.48) thus follows.

Based on Eq. (2.50), we have:

$$\begin{aligned} \mathbb{E}\left[\left(Y_{i,\ell} - h_\ell(\theta_*)\right)\left(Y_{i,\ell} - h_\ell(\theta_*)\right)' |\mathcal{F}_\ell\right] &= \mathbb{E}\left[\left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} a\right)\left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \bar{\eta}_{i,a}^{(\ell)} a\right)' \Big| \mathcal{F}_\ell\right] \\ &\overset{(a)}{=} \sum_{a \in \text{Supp}(\pi_\ell)} \left(m_a^{(\ell)}\right)^2 \mathbb{E}\left[\left(\bar{\eta}_{i,a}^{(\ell)}\right)^2\right] aa' \\ &\overset{(b)}{=} \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} aa' \\ &\overset{(c)}{=} \tilde{V}_\ell. \end{aligned} \tag{2.51}$$

In the above steps, (a) follows by observing that the noise terms are independent across different arms; hence, the expectation of each of the cross-terms vanish. For (b), we used the fact that $\bar{\eta}_{i,a}^{(\ell)}$

is the average of $m_a^{(\ell)}$ independent Gaussian noise terms, each with zero-mean and unit variance; hence, $\mathbb{E}\left[\left(\bar{\eta}_{i,a}^{(\ell)}\right)^2\right] = 1/(m_a^{(\ell)})$. For (c), we simply used the definition of $\tilde{V}_\ell$. In light of Eq. (2.51), it is easy to see why Eq. (2.49) holds. □

We now state - adapted to our notation - one of the main convergence guarantees from (Dalalyan and Minasyan, 2022) for the iteratively reweighted mean estimator.

Lemma 8. *Suppose we are given M d-dimensional samples $x_1, \ldots, x_M$, such that $(1-\alpha)M$ of these samples are drawn i.i.d. from $\mathcal{N}(v, \Sigma)$, where $v \in \mathbb{R}^d$ is an unknown mean vector, and $\Sigma \in \mathbb{R}^{d\times d}$ is a known covariance matrix. The remaining αM samples can be arbitrary. Let the corruption fraction α satisfy $\alpha < (5-\sqrt{5})/10$, and let $\delta \in (16\exp(-M), 1)$ be a given tolerance level. Then, with probability at least $1-\delta$, we have*

$$\|\hat{v} - v\|_2 \leq C_1 \|\Sigma\|_2^{1/2} \left(\sqrt{\frac{d + \log(16/\delta)}{M}} + \alpha \sqrt{\log \frac{1}{\alpha}} \right), \tag{2.52}$$

where C_1 is a suitably large universal constant, and $\hat{v}$ is the output of the Iteratively Reweighted Mean Estimator, namely Algorithm 1 in (Dalalyan and Minasyan, 2022), when it takes as input the M samples, the covariance matrix Σ, and the corruption fraction α.

Let us now see how the above bound can assist in our cause. Fix any epoch ℓ, and recall that $\delta_\ell = \bar{\delta}/(K\ell^2)$, where $\bar{\delta} = \delta/(10K)$, and δ is the given confidence parameter. Suppose we want to derive an error-bound based on Lemma 8 that holds with probability at least $1-\delta_\ell$. For this to happen, we need $\delta_\ell > 16\exp(-M)$. Since $\ell \leq T$, one can verify that the aforementioned condition is satisfied as long as M is large enough in the following sense:

$$M > \log\left(\frac{160K^2T^2}{\delta}\right). \tag{2.53}$$

From now on, we assume that the above condition holds. Next, recall that

$$X_\ell = \texttt{ITW}(\{\tilde{V}_\ell^{-1/2} Y_{i,\ell}, i \in [M]\}).$$

Based on Lemma 7, Lemma 8, and the same line of reasoning as used to arrive at Eq. (2.24), we have that with probability at least $1 - \delta_\ell$:

$$\|X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\| \leq C_1 \left(\sqrt{\frac{d + \log(16/\delta_\ell)}{M}} + \alpha \sqrt{\log \frac{1}{\alpha}} \right). \tag{2.54}$$

We will call upon the above bound later in our analysis. For now, this ends Step 2. As for Step 3, we start with the following result.

Lemma 9. *Consider any $\theta_1, \theta_2 \in \Theta$, and any epoch ℓ. There exists a symmetric positive definite matrix $G_\ell(\theta_1; \theta_2)$ satisfying $k_1 \tilde{V}_\ell \preccurlyeq G_\ell(\theta_1; \theta_2) \preccurlyeq k_2 \tilde{V}_\ell$, such that:*

$$h_\ell(\theta_1) - h_\ell(\theta_2) = G_\ell(\theta_1; \theta_2)(\theta_1 - \theta_2). \tag{2.55}$$

Proof. For any $\theta \in \Theta$, let us denote by $\nabla h_\ell(\theta)$ the Jacobian matrix of $h_\ell(\cdot)$ at θ. Such a matrix exists based on Assumption 1. Now based on the mean value theorem, $\exists \alpha \in (0, 1)$ such that

$$h_\ell(\theta_1) - h_\ell(\theta_2) = \left(\nabla h_\ell \left(\alpha \theta_1 + (1 - \alpha) \theta_2 \right) \right) (\theta_1 - \theta_2).$$

Let $\bar{\theta} = \alpha \theta_1 + (1 - \alpha) \theta_2$, and $G_\ell(\theta_1; \theta_2) = \nabla h_\ell(\bar{\theta})$. To complete the proof, we need to show that the matrix $G_\ell(\theta_1; \theta_2)$ defined above is symmetric, positive definite, and bounded above and below (in the Loewner sense) by scalar multiples of $\tilde{V}_\ell$. To that end, observe that:

$$\begin{aligned} \nabla h_\ell(\bar{\theta}) &\overset{(a)}{=} \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \dot{\mu}(\langle \bar{\theta}, a \rangle) a a' \\ &\overset{(b)}{\succcurlyeq} k_1 \left(\sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} a a' \right) \\ &= k_1 \tilde{V}_\ell. \end{aligned} \tag{2.56}$$

In the above steps, we used the definition of $h_\ell(\cdot)$ for (a); and for (b), we used Assumption 1. From the above steps, it is clear that $G_\ell(\theta_1; \theta_2)$ is symmetric. That it is also positive definite follows from the fact that $\tilde{V}_\ell \succ 0$. Finally, using the fact that $\mu(\cdot)$ is k_2-Lipschitz, and a similar line of reasoning,

one can show that $G_\ell(\theta_1;\theta_2) \preccurlyeq k_2\tilde{V}_\ell$. This concludes the proof. □

We now have all the pieces required to establish an analogue of Lemma 1.

Lemma 10. *(**Robust Confidence Intervals for** `RC-GLM`) Suppose M satisfies the condition in Eq. (2.53), and $\alpha < (5-\sqrt{5})/10$. Fix any epoch ℓ. For each arm $a \in \mathcal{A}_\ell$, with probability at least $1-\delta_\ell$, it holds that:*

$$|\mu(\langle\hat{\theta}^{(\ell)}, a\rangle) - \mu(\langle\theta_*, a\rangle)| \leq \bar{\gamma}_\ell, \quad \textit{where } \bar{\gamma}_\ell = 4C_1\frac{k_2}{k_1}\left(\sqrt{d} + \alpha\sqrt{M\log(1/\alpha)}\right)\epsilon_\ell, \tag{2.57}$$

and C_1 is as in Lemma 8.

Proof. Let us start by conditioning on the event of measure at least $1-\delta_\ell$ on which Eq. (2.54) holds. Invoking Lemma 9, we know that there exists a symmetrix positive definite matrix G_ℓ such that $k_1\tilde{V}_\ell \preccurlyeq G_\ell \preccurlyeq k_2\tilde{V}_\ell$, and:[6]

$$\begin{aligned} G_\ell\left(\hat{\theta}^{(\ell)} - \theta_*\right) &= h_\ell(\hat{\theta}^{(\ell)}) - h_\ell(\theta_*) \\ &= \tilde{V}_\ell^{1/2}X_\ell - h_\ell(\theta_*) \\ &= \tilde{V}_\ell^{1/2}\left(X_\ell - \tilde{V}_\ell^{-1/2}h_\ell(\theta_*)\right). \end{aligned} \tag{2.58}$$

For the second step above, we used the fact that based on line 9 of `RC-GLM`, $h_\ell(\hat{\theta}^{(\ell)}) = \tilde{V}_\ell^{1/2}X_\ell$. Now fix any arm $a \in \mathcal{A}_\ell$, and observe that:

$$\langle\hat{\theta}^{(\ell)} - \theta_*, a\rangle = \left\langle G_\ell^{-1}\tilde{V}_\ell^{1/2}\left(X_\ell - \tilde{V}_\ell^{-1/2}h_\ell(\theta_*)\right), a\right\rangle.$$

This, in turn, implies:

$$|\langle\hat{\theta}^{(\ell)} - \theta_*, a\rangle| \leq \underbrace{|||[||\Big]G_\ell^{-1}\tilde{V}_\ell^{1/2}\left(X_\ell - \tilde{V}_\ell^{-1/2}h_\ell(\theta_*)\right)_{\tilde{V}_\ell}}_{T_1}||[||\,\big]a_{\tilde{V}_\ell^{-1}}. \tag{2.59}$$

[6] Here, we have dropped the dependence of G_ℓ on $\hat{\theta}^{(\ell)}$ and θ_* to lighten the notation.

We bound T_1 as follows.

$$\begin{aligned} T_1 &= \sqrt{\left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)' \tilde{V}_\ell^{1/2} G_\ell^{-1} \tilde{V}_\ell G_\ell^{-1} \tilde{V}_\ell^{1/2} \left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)} \\ &\overset{(a)}{\leq} \frac{1}{\sqrt{k_1}} \sqrt{\left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)' \tilde{V}_\ell^{1/2} G_\ell^{-1} \tilde{V}_\ell^{1/2} \left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)} \\ &\overset{(b)}{\leq} \frac{1}{k_1} \sqrt{\left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)' \left(X_\ell - \tilde{V}_\ell^{-1/2} h_\ell(\theta_*)\right)} \\ &\overset{(c)}{\leq} \frac{C_1}{k_1} \left(\sqrt{\frac{d + \log(16/\delta_\ell)}{M}} + \alpha \sqrt{\log \frac{1}{\alpha}} \right). \end{aligned} \tag{2.60}$$

For both (a) and (b) above, we used $k_1 \tilde{V}_\ell \preccurlyeq G_\ell$; for (c), we invoked the bound in Eq. (2.54). Now recall from Eq. (2.22) that

$$|||[||\,]a_{\tilde{V}_\ell^{-1}} \leq \epsilon_\ell \sqrt{\frac{2M}{\log\left(\frac{1}{\delta_\ell}\right)}}.$$

Combining the above bound with the ones in Eq. (2.59) and Eq. (2.60), and using $\log(1/\delta_\ell) \geq 1$, we obtain

$$|\langle \hat{\theta}^{(\ell)} - \theta_*, a \rangle| \leq \frac{\sqrt{2} C_1}{k_1} \left(\sqrt{d + \frac{\log(16/\delta_\ell)}{\log(1/\delta_\ell)}} + \alpha \sqrt{M \log \frac{1}{\alpha}} \right) \epsilon_\ell.$$

Elementary calculations coupled with the fact that $\log(1/\delta_\ell) \geq 1$ yields:

$$\frac{\log(16/\delta_\ell)}{\log(1/\delta_\ell)} \leq 4.$$

Putting all the pieces together, and simplifying, we arrive at the following bound:

$$|\langle \hat{\theta}^{(\ell)} - \theta_*, a \rangle| \leq \frac{4 C_1}{k_1} \left(\sqrt{d} + \alpha \sqrt{M \log \frac{1}{\alpha}} \right) \epsilon_\ell. \tag{2.61}$$

Using the fact that $\mu(\cdot)$ is k_2-Lipschitz then yields:

$$|\mu(\langle \hat{\theta}^{(\ell)}, a \rangle) - \mu(\langle \theta_*, a \rangle)| \leq k_2 |\langle \hat{\theta}^{(\ell)} - \theta_*, a \rangle| \leq 4 C_1 \frac{k_2}{k_1} \left(\sqrt{d} + \alpha \sqrt{M \log \frac{1}{\alpha}} \right) \epsilon_\ell,$$

which is the desired claim. This completes the proof. □

Having derived the robust confidence bounds for RC-GLM, we can complete the proof of Theorem 3.

Proof. (**Proof of Theorem 3**). Using essentially the same arguments as those used to prove Lemmas 4 and 5, we can prove that there exists a clean event, say $\mathcal{E}$, of measure at least $1-\delta$, such that on $\mathcal{E}$, the following hold: (i) $a_* \in \mathcal{A}_\ell, \forall \ell \in [L]$, where L is the total number of epochs; and (ii) for any epoch $\ell \in [L], a \in \mathcal{A}_\ell \implies \tilde{\Delta}_a \leq 8\bar{\gamma}_\ell$. Here, $\tilde{\Delta}_a = \mu(\langle \theta_*, a_* \rangle) - \mu(\langle \theta_*, a \rangle)$. Let us condition on this clean event $\mathcal{E}$. The remainder of the proof follows the same line of reasoning as that of Theorem 1. For any good agent $i \in [M] \setminus \mathcal{B}$, we can bound the regret as follows.

$$\begin{aligned}
\sum_{t=1}^{T} \left(\mu(\langle \theta_*, a_* \rangle) - \mu(\langle \theta_*, a_{i,t} \rangle)\right) &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} m_a^{(\ell)} \left(\mu(\langle \theta_*, a_* \rangle) - \mu(\langle \theta_*, a \rangle)\right) \\
&= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \left\lceil \frac{T_a^{(\ell)}}{M} \right\rceil \tilde{\Delta}_a \\
&\leq \underbrace{\sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \frac{T_a^{(\ell)}}{M} \tilde{\Delta}_a}_{T_1} + \underbrace{\sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \tilde{\Delta}_a}_{T_2}.
\end{aligned} \tag{2.62}$$

As in the proof of Theorem 1, we bound T_1 and T_2 separately. For bounding T_1, we have:

$$\begin{aligned}
T_1 &= \sum_{\ell=1}^{L} \sum_{a \in \text{Supp}(\pi_\ell)} \frac{T_a^{(\ell)}}{M} \tilde{\Delta}_a \\
&= \frac{d}{M} \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \sum_{a \in \text{Supp}(\pi_\ell)} \pi_\ell(a) \tilde{\Delta}_a \\
&\leq \frac{8d}{M} \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \bar{\gamma}_\ell \\
&= \frac{32 C_1 d \left(\sqrt{d} + \alpha\sqrt{M \log(1/\alpha)}\right)}{M} \left(\frac{k_2}{k_1}\right) \sum_{\ell=1}^{L} \frac{1}{\epsilon_\ell} \log\left(\frac{10K^2\ell^2}{\delta}\right) \\
&\leq \frac{32 C_1 d \left(\sqrt{d} + \alpha\sqrt{M \log(1/\alpha)}\right)}{M} \left(\frac{k_2}{k_1}\right) \log\left(\frac{10K^2L^2}{\delta}\right) \sum_{\ell=1}^{L} 2^\ell \\
&= O\left(\left(\frac{k_2}{k_1}\right) \frac{d \left(\sqrt{d} + \alpha\sqrt{M \log(1/\alpha)}\right)}{M} \log\left(\frac{10K^2L^2}{\delta}\right) 2^L\right).
\end{aligned} \tag{2.63}$$

Recall the following fact that we proved earlier for Theorem 1:

$$2^L \leq \sqrt{\frac{MT}{d \log\left(10K^2L^2/\delta\right)}}.$$

Plugging this bound in Eq. (2.63), we obtain:

$$T_1 = O\left(\left(\frac{k_2}{k_1}\right)\left(\alpha\sqrt{\log(1/\alpha)} + \sqrt{\frac{d}{M}}\right)\sqrt{\log\left(\frac{KT}{\delta}\right)dT}\right).$$

One can upper-bound T_2 using the same bound as above using exactly the same steps as in the proof of Theorem 1. Combining the bounds on T_1 and T_2 leads to the desired claim. □

• **Comments on solving for $\hat{\theta}^{(\ell)}$ in** `RC-GLM`. Recall that line 9 of Algorithm `RC-GLM` requires the server to solve for $\hat{\theta}^{(\ell)}$ based on the following equation:

$$h_\ell(\hat{\theta}^{(\ell)}) = \tilde{V}_\ell^{1/2} X_\ell. \tag{2.64}$$

Based on Lemma 9, we have that for any $\theta_1, \theta_2 \in \Theta$ such that $\theta_1 \neq \theta_2$:

$$(\theta_1 - \theta_2)' \left(h_\ell(\theta_1) - h_\ell(\theta_2)\right) = (\theta_1 - \theta_2)' \, G_\ell(\theta_1; \theta_2) \, (\theta_1 - \theta_2) > 0,$$

since $G_\ell(\theta_1; \theta_2)$ is positive-definite. Thus, the map $h_\ell : \mathbb{R}^d \to \mathbb{R}^d$ is injective, and h_ℓ^{-1} is well-defined. This, in turn, implies that Eq. (2.64) has a unique solution.

2.12.2. The Iteratively Reweighted Mean Estimation Algorithm

In this section, we briefly explain the main idea behind the Iteratively Reweighted Mean Estimation Algorithm in (Dalalyan and Minasyan, 2022). Suppose we are given an α-corrupted set $\mathcal{X}$ of samples, namely, M d-dimensional samples $x_1, \ldots, x_M$, such that $(1-\alpha)M$ of these samples are drawn i.i.d. from $\mathcal{N}(v, \Sigma)$, and the remaining αM samples are arbitrarily corrupted by an adversary. The goal is to recover the unknown mean vector $v \in \mathbb{R}^d$, given knowledge of the samples $\mathcal{X}$, the corruption

fraction α, and the covariance matrix Σ. To see how this is done, let us define a couple of quantities for any pair of vectors $w \in [0,1]^d$ and $\mu \in \mathbb{R}^d$:

$$\bar{x}_w = \sum_{i=1}^{M} w_i x_i; \;\; G(w,\mu) = \lambda_{\max}\left(\sum_{i=1}^{M} w_i \left(x_i - \mu\right)\left(x_i - \mu\right)' - \Sigma\right). \tag{2.65}$$

The basic idea is to find a weight vector $\hat{w}$ within the d-dimensional probability simplex such that the weighted average $\bar{x}_{\hat{w}}$ is close to the true mean v. Intuitively, a "small" value of $G(\hat{w}, \bar{x}_{\hat{w}})$ is an indicator of a good candidate for such a weight vector. This is essentially the strategy pursued in Algorithm 5 where one iteratively updates the weight vectors, and the associated weighted averages, so as to minimize the function $G(\cdot,\cdot)$ defined in Eq. (2.65). At the termination of this algorithm, say after N iterations, the goal is to output a weight vector $\hat{w}_N$ that mimics the ideal weight vector w^* defined by: $w_j^* = \mathbf{1}(j \in \mathcal{I})/|\mathcal{I}|$, where $\mathcal{I}$ is the set of good samples (inliers). The steps of the Iteratively Reweighted Mean Estimator are outlined in Algorithm 5.

Algorithm 5 Iteratively Reweighted Mean Estimator (`ITW`)

Input: α-corrupted set of M samples $\mathcal{X}$, corruption fraction α, and covariance matrix Σ.
Output: Robust estimate of the mean $\hat{v}$.
Initialize: Compute $\hat{v}_0$ as a minimizer of $\arg\min_{\mu} \sum_{i=1}^{M} \|x_i - \mu\|$.
Let $N = 0 \vee \left\lceil \frac{\log(4r_\Sigma) - 2\log(\alpha(1-2\alpha))}{2\log((1-2\alpha)) - \log(\alpha) - \log(1-\alpha)} \right\rceil$. Here, $r_\Sigma = \mathrm{Trace}(\Sigma)/\|\Sigma\|_2$.

for $k = 1$ to N **do**
    Compute current weights:

$$w \in \underset{(M - M\epsilon)\|w\|_\infty \leq 1}{\arg\min} \lambda_{\max}\left(\sum_{i=1}^{M} w_i (x_i - \hat{v}_{k-1})(x_i - \hat{v}_{k-1})' - \Sigma\right) \vee 0.$$

    Update the estimator:

$$\hat{v}_k = \sum_{i=1}^{M} w_i x_i.$$

end for
Return $\hat{v} = \hat{v}_K$.

2.13. Analysis for the Contextual Bandit Setting: Proof of Theorem 4

The proof of Theorem 4 proceeds in multiple steps. We start with an analysis of the Robust BaseLinUCB subroutine, namely Algorithm 2.

Lemma 11. ***(Bounds for Robust BaseLinUCB)*** *Suppose the input index set Ψ_t is constructed so that for fixed $x_{\tau,a_\tau}, \tau \in \Psi_t$, the rewards $\{r_{i,\tau}\}_{\tau\in\Psi_t}$ are independent random variables for each good agent $i \in [M] \setminus \mathcal{B}$. Then, for each $a \in [K]$, with probability at least $1-\bar{\delta}$, it holds that:*

$$|\hat{r}_{t,a} - \langle \theta_*, x_{t,a}\rangle| \leq \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}}\right) \|x_{t,a}\|_{A_t^{-1}}, \tag{2.66}$$

where C is as in Lemma 2.

Proof. Fix any agent $i \in [M] \setminus \mathcal{B}$. Now from the expression for $\hat{\theta}_{i,t}$ in Algorithm 2, observe that:

$$\begin{aligned}
\hat{\theta}_{i,t} &= A_t^{-1} b_{i,t} \\
&= A_t^{-1}\left(\sum_{\tau\in\Psi_t} r_{i,\tau} x_{\tau,a_\tau}\right) \\
&= A_t^{-1}\left(\sum_{\tau\in\Psi_t} \left(\langle \theta_*, x_{\tau,a_\tau}\rangle + \eta_{i,\tau}\right) x_{\tau,a_\tau}\right) \\
&= A_t^{-1}\left(\sum_{\tau\in\Psi_t} x_{\tau,a_\tau} x'_{\tau,a_\tau}\right)\theta_* + A_t^{-1}\left(\sum_{\tau\in\Psi_t} \eta_{i,\tau} x_{\tau,a_\tau}\right) \\
&= \left(I_d - \frac{A_t^{-1}}{M}\right)\theta_* + A_t^{-1}\left(\sum_{\tau\in\Psi_t} \eta_{i,\tau} x_{\tau,a_\tau}\right),
\end{aligned} \tag{2.67}$$

where we used $\eta_{i,\tau}$ as a shorthand for $\eta_{i,\tau}(a_\tau)$. Now consider any $a \in [K]$. Using the above expression, we will now decompose the error in estimation of $\langle \theta_*, x_{t,a}\rangle$ into a bias term and a variance term:

$$\langle \hat{\theta}_{i,t} - \theta_*, x_{t,a}\rangle = \underbrace{-\frac{1}{M}\langle A_t^{-1}\theta_*, x_{t,a}\rangle}_{\text{bias term}} + \underbrace{\sum_{\tau\in\Psi_t} \langle A_t^{-1} x_{\tau,a_\tau}, x_{t,a}\rangle \eta_{i,\tau}}_{\text{variance term}}. \tag{2.68}$$

For a fixed set of feature vectors, under our assumption that $\{r_{i,\tau}\}_{\tau\in\Psi_t}$ are independent random variables, the variance term is a sum of independent zero-mean Gaussian noise variables. Thus,

$$\mathbb{E}\left[\sum_{\tau\in\Psi_t}\langle A_t^{-1}x_{\tau,a_\tau},x_{t,a}\rangle\eta_{i,\tau}\right]=\sum_{\tau\in\Psi_t}\langle A_t^{-1}x_{\tau,a_\tau},x_{t,a}\rangle\mathbb{E}\left[\eta_{i,\tau}\right]=0.$$

Furthermore, we have:

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{\tau\in\Psi_t}\langle A_t^{-1}x_{\tau,a_\tau},x_{t,a}\rangle\eta_{i,\tau}\right)^2\right] &= x_{t,a}'A_t^{-1}\left(\sum_{\tau\in\Psi_t}\mathbb{E}\left[\left(\eta_{i,\tau}^2\right)\right]x_{\tau,a_\tau}x_{\tau,a_\tau}'\right)A_t^{-1}x_{t,a}\\
&= x_{t,a}'A_t^{-1}\left(\sum_{\tau\in\Psi_t}x_{\tau,a_\tau}x_{\tau,a_\tau}'\right)A_t^{-1}x_{t,a}\\
&\le x_{t,a}'A_t^{-1}\underbrace{\left(\frac{I_d}{M}+\sum_{\tau\in\Psi_t}x_{\tau,a_\tau}x_{\tau,a_\tau}'\right)}_{A_t}A_t^{-1}x_{t,a}\\
&= \|x_{t,a}\|_{A_t^{-1}}^2.
\end{aligned}\tag{2.69}$$

From the above arguments, we conclude that for each $i\in[M]\setminus\mathcal{B}$,

$$\langle\hat{\theta}_{i,t},x_{t,a}\rangle\sim\ \mathcal{N}\left(\langle\theta_*-\frac{A_t^{-1}\theta_*}{M},x_{t,a}\rangle,\sigma^2\right),\ \text{ where }\ \sigma^2\le\|x_{t,a}\|_{A_t^{-1}}^2.$$

Since the noise samples are independent across agents, we also know that $\{\langle\hat{\theta}_{i,t},x_{t,a}\rangle\}_{i\in[M]\setminus\mathcal{B}}$ are independent. Recalling that $\hat{r}_{t,a}=\texttt{Median}\left(\{\langle\hat{\theta}_{i,t},x_{t,a}\rangle,i\in[M]\}\right)$, and invoking Lemma 2, we then have that with probability at least $1-\bar{\delta}$,

$$|\hat{r}_{t,a}-\langle\theta_*-\frac{A_t^{-1}\theta_*}{M},x_{t,a}\rangle|\le\left(\alpha+C\sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}}\right)\|x_{t,a}\|_{A_t^{-1}}.\tag{2.70}$$

This immediately implies:

$$|\hat{r}_{t,a}-\langle\theta_*,x_{t,a}\rangle|\le\frac{1}{M}|\langle A_t^{-1}\theta_*,x_{t,a}\rangle|+\left(\alpha+C\sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}}\right)\|x_{t,a}\|_{A_t^{-1}}.\tag{2.71}$$

It remains to bound the first term in the above display. To that end, we proceed as follows:

$$\begin{aligned}
\frac{1}{M}|\langle A_t^{-1}\theta_*, x_{t,a}\rangle| &= \frac{1}{M}|\langle A_t^{-1}x_{t,a}, \theta_*\rangle| \\
&\leq \frac{1}{M}\|A_t^{-1}x_{t,a}\|\|\theta_*\| \\
&\overset{(a)}{\leq} \frac{1}{\sqrt{M}}\sqrt{x'_{t,a}A_t^{-1}\frac{I_d}{M}A_t^{-1}x_{t,a}} \\
&\leq \frac{1}{\sqrt{M}}\sqrt{x'_{t,a}A_t^{-1}\left(\frac{I_d}{M}+\sum_{\tau\in\Psi_t}x_{\tau,a_\tau}x'_{\tau,a_\tau}\right)A_t^{-1}x_{t,a}} \\
&= \frac{1}{\sqrt{M}}\|x_{t,a}\|_{A_t^{-1}} \\
&\overset{(b)}{\leq} C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\|x_{t,a}\|_{A_t^{-1}},
\end{aligned} \tag{2.72}$$

where for (a), we used the fact that $\|\theta_*\| \leq 1$ by assumption; and for (b), we used $C \geq 1, \log\left(\frac{1}{\delta}\right) \geq 1$. Combining the above bound with the one in Eq.(2.71) leads to the desired claim. □

For Lemma 11 to hold, the crucial requirement is for the rewards corresponding to indices in Ψ_t to be independent. Our next result shows that this is indeed the case; the proof of this lemma essentially follows the same arguments as that of (Auer, 2002, Lemma 14), we reproduce these arguments here only for completeness.

Lemma 12. ***(Independence of Samples)*** *Fix any agent $i \in [M] \setminus \mathcal{B}$. For each $s \in [S]$ and each $t \in [T]$, given any fixed sequence of feature vectors $\{x_{\tau,a_\tau}\}_{\tau\in\Psi_t^{(s)}}$, the rewards $\{r_{i,\tau}\}_{\tau\in\Psi_t^{(s)}}$ are independent random variables.*

Proof. Let us start by observing that a time-step t can be added to $\Psi_t^{(s)}$ only in line 6 of Algorithm 3. Thus, the event $\{t \in \Psi_t^{(s)}\}$ only depends on all prior phases $\cup_{\ell<s}\Psi_t^{(\ell)}$, and on the confidence width $w_{t,a}^{(s)}$. From the definition of $w_{t,a}^{(s)}$ in line 7 of Algorithm 2, we note that $w_{t,a}^{(s)}$ depends only on the feature vectors $x_{\tau,a_\tau}, \tau \in \Psi_t^{(s)}$, and $x_{t,a}$. Combining the above observations, it is easy to see that $\{t \in \Psi_t^{(s)}\}$ only depends on the feature vectors. Noting that the feature vector sequence is fixed, and cannot be controlled by the adversarial agents, leads to the claim of the lemma. □

The next result tells us that with high-probability, at each time-step $t \in [T]$, (i) the best arm a_t^* is retained in all stages of the screening process; and (ii) an active arm in phase s can contribute to at most $8/(2^s\sqrt{M})$ instantaneous regret.

Lemma 13. *With probability at least $1 - \bar{\delta}KST$, for any $t \in [T]$ and any $s \in [S]$, the following hold:*

(i) $|\hat{r}_{t,a}^{(s)} - \langle \theta_*, x_{t,a} \rangle| \leq w_{t,a}^{(s)}, \forall a \in \mathcal{A}_s$.

(ii) $a_t^* \in \mathcal{A}_s$.

(iii) $\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a} \rangle \leq 8/(2^s\sqrt{M}), \forall a \in \mathcal{A}_s$.

Proof. Part (i) of the result follows directly from Lemma 11, and an union bound over all time-steps, phases and arms.

For part (ii), let us condition on the clean event, say $\mathcal{E}$, on which part (i) holds. From the rules of Algorithm 3, it holds trivially that $a_t^* \in \mathcal{A}_s$ for $s = 1$. Now suppose there exists some phase $s > 1$ such that $a_t^* \in \mathcal{A}_{s-1}$, but $a_t^* \notin \mathcal{A}_s$. From line 8 in Algorithm 3, we must have $w_{t,a}^{(s-1)} \leq 2^{1-s}/\sqrt{M}, \forall a \in \mathcal{A}_{s-1}$. From the phased elimination strategy in line 8 of Algorithm 3, $a_t^* \notin \mathcal{A}_s$ implies the existence of some arm $a \in \mathcal{A}_s$ such that:

$$\begin{aligned}
&\left(\hat{r}_{t,a}^{(s-1)} + w_{t,a}^{(s-1)}\right) - \left(\hat{r}_{t,a_t^*}^{(s-1)} + w_{t,a_t^*}^{(s-1)}\right) > \frac{2^{2-s}}{\sqrt{M}} \\
\overset{(a)}{\implies} &\langle \theta_*, x_{t,a} \rangle + 2w_{t,a}^{(s-1)} - \left(\hat{r}_{t,a_t^*}^{(s-1)} + w_{t,a_t^*}^{(s-1)}\right) > \frac{2^{2-s}}{\sqrt{M}} \\
\overset{(b)}{\implies} &\langle \theta_*, x_{t,a} - x_{t,a_t^*} \rangle + 2w_{t,a}^{(s-1)} > \frac{2^{2-s}}{\sqrt{M}} \\
\overset{(c)}{\implies} &2w_{t,a}^{(s-1)} > \frac{2^{2-s}}{\sqrt{M}},
\end{aligned} \tag{2.73}$$

which leads to a contradiction as $w_{t,a}^{(s-1)} \leq 2^{1-s}/\sqrt{M}$. In the above steps, both (a) and (b) follow from the defining property of the clean event $\mathcal{E}$; for (c), we used $\langle \theta_*, x_{t,a} - x_{t,a_t^*} \rangle \leq 0$ from the optimality of a_t^*. This completes the proof of part (ii).

For part (iii), let us once again condition on the clean event $\mathcal{E}$ on which part (i) holds. Now

$a \in \mathcal{A}_s \implies a \in \mathcal{A}_{s-1}$. We also know from part (ii) that $a_t^* \in \mathcal{A}_{s-1}$. The retention of arm a in $\mathcal{A}_s$ implies (based on line 8 of Algorithm 3),

$$\begin{aligned}
&\left(\hat{r}_{t,a_t^*}^{(s-1)} + w_{t,a_t^*}^{(s-1)}\right) - \left(\hat{r}_{t,a}^{(s-1)} + w_{t,a}^{(s-1)}\right) \leq \frac{2^{2-s}}{\sqrt{M}} \\
\overset{(a)}{\implies} &\langle \theta_*, x_{t,a_t^*} \rangle - \left(\hat{r}_{t,a}^{(s-1)} + w_{t,a}^{(s-1)}\right) \leq \frac{2^{2-s}}{\sqrt{M}} \\
\overset{(b)}{\implies} &\langle \theta_*, x_{t,a_t^*} - x_{t,a} \rangle \leq \frac{2^{2-s}}{\sqrt{M}} + 2w_{t,a}^{(s-1)} \\
\overset{(c)}{\implies} &\langle \theta_*, x_{t,a_t^*} - x_{t,a} \rangle \leq \frac{8}{2^s\sqrt{M}},
\end{aligned} \tag{2.74}$$

which is the desired claim. Here, for (a) and (b) we used the defining property of $\mathcal{E}$. As for (c), we used the fact that $w_{t,a}^{(s-1)} \leq 2^{1-s}/\sqrt{M}$. □

The final piece needed in the proof of Theorem 4 is a bound on $|\Psi_T^{(s)}|$ for each $s \in [S]$.

To that end, we will make use of the elliptical potential lemma from (Abbasi-Yadkori et al., 2011).

Lemma 14. ***(Elliptical Potential Lemma)*** *Let $\{X_t\}_{t=1}^{\infty}$ be a sequence in $\mathbb{R}^d$, V a $d \times d$ positive definite matrix, and define $\bar{V}_t = V + \sum_{\tau=1}^{t} X_\tau X_\tau'$. If $\|X_t\| \leq L$ for all t, then we have that*

$$\sum_{t=1}^{T} \min\{1, \|X_t\|_{\bar{V}_{t-1}^{-1}}^2\} \leq 2\log\frac{\det(\bar{V}_T)}{\det(V)} \leq 2(d\log((trace(V) + TL^2)/d) - \log(\det V)).$$

We have the following result.

Lemma 15. ***(Bound on $|\Psi_T^{(s)}|$)*** *Fix any $s \in [S]$. The following then holds:*

$$|\Psi_T^{(s)}| \leq 2^s\sqrt{M}\left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\sqrt{2d|\Psi_T^{(s)}|\log(2M|\Psi_T^{(s)}|)}. \tag{2.75}$$

Proof. Fix any phase $s \in [S]$. Now consider any time-step $t \in \Psi_T^{(s)}$. Since $t \in \Psi_T^{(s)}$, based on line 6

of Algorithm 3, it must be that:

$$\frac{1}{2^s\sqrt{M}} < w_{t,a_t}^{(s)} = \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\|x_{t,a_t}\|_{A_{s,t}^{-1}}, \text{ where} \tag{2.76}$$

$$A_{s,t} = \left(\frac{I_d}{M} + \sum_{\tau \in \Psi_t^{(s)}} x_{\tau,a_\tau} x'_{\tau,a_\tau}\right).$$

Also, since $s \geq 1$, we have the following trivial inequality:

$$\frac{1}{2^s\sqrt{M}} < \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right).$$

Combining the above inequality with the one in (2.76), we note that for each $t \in \Psi_T^{(s)}$, it holds that:

$$\frac{1}{2^s\sqrt{M}} < \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\min\{1, \|x_{t,a_t}\|_{A_{s,t}^{-1}}\}.$$

Summing the above display over all indices in $\Psi_T^{(s)}$ yields:

$$\begin{aligned}
\frac{|\Psi_T^{(s)}|}{2^s\sqrt{M}} &< \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\sum_{t \in \Psi_T^{(s)}} \min\{1, \|x_{t,a_t}\|_{A_{s,t}^{-1}}\} \\
&\overset{(a)}{\leq} \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\sqrt{|\Psi_T^{(s)}|\sum_{t \in \Psi_T^{(s)}} \min\{1, \|x_{t,a_t}\|_{A_{s,t}^{-1}}^2\}} \\
&\overset{(b)}{\leq} \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\sqrt{2d|\Psi_T^{(s)}|\log\left(1 + \frac{M}{d}|\Psi_T^{(s)}|\right)} \\
&\leq \left(\alpha + 2C\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{M}}\right)\sqrt{2d|\Psi_T^{(s)}|\log\left(2M|\Psi_T^{(s)}|\right)},
\end{aligned} \tag{2.77}$$

where (a) follows from Jensen's inequality, and (b) follows from an application of Lemma 14. Reorganizing the resulting inequality above leads to the desired claim. □

We are now ready to prove Theorem 4.

Proof. (**Proof of Theorem 4**) Since $S = \lceil \log T \rceil$, we have $1/(2^S\sqrt{M}) \leq 1/(\sqrt{MT})$. Thus, from the rules of Algorithm 3, it is apparent that at every time-step t, an action a_t is *always* chosen, either based on line 6, or on line 7. If we use Ξ_T to store those time steps in $[T]$ where an action is chosen based on line 7 of Algorithm 3, then the above reasoning implies: $[T] = \Xi_T \cup \bigcup_{s\in[S]} \Psi_T^{(s)}$.

Throughout the rest of the proof, we will condition on the clean event on which items (i)-(iii) in Lemma 13 hold. We also recall that this clean event has measure at least $1 - \bar{\delta}KST$. Now fix any good agent $i \in [M] \setminus \mathcal{B}$, and note that the same action a_t is played by every good agent at time t. The cumulative regret for agent i can thus be decomposed as follows:

$$\sum_{t=1}^{T} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_{i,t}} \rangle \right) = \underbrace{\sum_{t \in \Xi_T} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_t} \rangle \right)}_{T_1} + \underbrace{\sum_{s=1}^{S} \sum_{t \in \Psi_T^{(s)}} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_t} \rangle \right)}_{T_2}. \tag{2.78}$$

Let us first bound T_2 as follows.

$$\begin{aligned}
T_2 &= \sum_{s=1}^{S} \sum_{t \in \Psi_T^{(s)}} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_t} \rangle \right) \\
&\overset{(a)}{\leq} \sum_{s=1}^{S} \sum_{t \in \Psi_T^{(s)}} \frac{8}{2^s \sqrt{M}} \\
&= \sum_{s=1}^{S} \frac{8}{2^s \sqrt{M}} |\Psi_T^{(s)}| \\
&\overset{(b)}{\leq} \sum_{s=1}^{S} 8 \left(\alpha + 2C \sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}} \right) \sqrt{2d |\Psi_T^{(s)}| \log(2M |\Psi_T^{(s)}|)} \\
&\overset{(c)}{\leq} 8S \left(\alpha + 2C \sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}} \right) \sqrt{2dT \log(2MT)} \\
&\leq 8(1 + \log(T)) \left(\alpha + 2C \sqrt{\frac{\log\left(\frac{1}{\bar{\delta}}\right)}{M}} \right) \sqrt{2dT \log(2MT)},
\end{aligned} \tag{2.79}$$

where for (a), we used item (iii) of Lemma 13; for (b), we invoked Lemma 15 to bound $|\Psi_T^{(s)}|$; and for (c), we used the trivial bound $|\Psi_T^{(s)}| \leq T$. With $\bar{\delta} = \delta/(KST)$, the bound on T_2 reads as follows:

$$T_2 = O\left(\left(\alpha + 2C \sqrt{\frac{\log\left(\frac{KT}{\delta}\right)}{M}} \right) \log(T) \sqrt{2dT \log(2MT)} \right). \tag{2.80}$$

Now let us turn to bounding T_1. Consider any time-step $t \in \Xi_T$ where the action a_t is chosen based on line 7 of Algorithm 3. Since a_t^* is never eliminated on the clean event (item (ii) of Lemma 13), and since a_t has the highest robust upper confidence bound among all active arms, it must be that:

$$\begin{aligned}
& \hat{r}_{t,a_t^*}^{(s)} + w_{t,a_t^*}^{(s)} \leq \hat{r}_{t,a_t}^{(s)} + w_{t,a_t}^{(s)} \\
\implies & \langle \theta_*, x_{t,a_t^*} - x_{t,a_t} \rangle \leq 2 w_{t,a_t}^{(s)} \\
\implies & \langle \theta_*, x_{t,a_t^*} - x_{t,a_t} \rangle \leq \frac{2}{\sqrt{MT}},
\end{aligned} \tag{2.81}$$

where for the first implication, we used item (i) of Lemma 13; and for the second, we used the fact for an action to be chosen based on line 7, it's confidence width must be bounded above by $1/(\sqrt{MT})$. We conclude that

$$T_1 = \sum_{t \in \Xi_T} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_t} \rangle \right) = O\left(\sqrt{\frac{T}{M}} \right).$$

Combining the above bound on T_1 with that on T_2 in Eq. (2.80), we have that with probability at least $1 - \delta$, the following is true for each good agent $i \in [M] \setminus \mathcal{B}$:

$$\begin{aligned} \sum_{t=1}^{T} \left(\langle \theta_*, x_{t,a_t^*} \rangle - \langle \theta_*, x_{t,a_t} \rangle \right) &= O\left(\left(\alpha + 2C\sqrt{\frac{\log\left(\frac{KT}{\delta}\right)}{M}} \right) \log(T) \sqrt{2dT \log(2MT)} \right) \\ &= \tilde{O}\left(\left(\alpha + \sqrt{1/M} \right) \sqrt{dT} \right). \end{aligned} \tag{2.82}$$

This completes the proof. □

2.14. Alternate Strategies for Robust Collaborative Phased Elimination can lead to Sub-Optimal Regret Bounds.

In Section 2.3, where we introduced `RCLB`, we alluded to the fact that certain natural candidate strategies may lead to sub-optimal regret bounds. In this section, we elaborate on this point. Note that our end goal is to come up with a phased elimination step akin to line 9 of `RCLB`. To achieve this, in every epoch ℓ, we need estimates of $\langle \theta_*, a \rangle$ along with associated confidence intervals for each $a \in \mathcal{A}_\ell$. In what follows, we will consider two natural candidate strategies for the same, and demonstrate that they each lead to confidence bounds that are looser than the ones we derived in Lemma 1. As such, using such bounds in the phased elimination step will lead to regret guarantees that are sub-optimal in their dependence on the model dimension d.

• **Candidate Strategy 1.** Suppose in every epoch ℓ, the server collects the local model estimates $\{\hat{\theta}_i^{(\ell)}\}_{i \in [M]}$, and aims to first construct a robust estimate $\hat{\theta}^{(\ell)}$ of θ_*. Subsequently, it uses $\langle \hat{\theta}^{(\ell)}, a \rangle$ as an estimate of $\langle \theta_*, a \rangle$ for each active arm $a \in \mathcal{A}_\ell$. To extract $\hat{\theta}^{(\ell)}$ from the local estimates

$\{\hat{\theta}_i^{(\ell)}\}_{i\in[M]}$, we need a high-dimensional robust mean estimator. One natural candidate for this is the Iteratively Reweighted Mean Estimator from (Dalalyan and Minasyan, 2022) since it leads to minimax-optimal error bounds. However, there is an immediate obstacle to directly applying the estimator from (Dalalyan and Minasyan, 2022) on the model estimates $\{\hat{\theta}_i^{(\ell)}\}_{i\in[M]}$. This stems from the observation that although $\hat{\theta}_i^{(\ell)}$ is an unbiased estimate of θ_* for each good agent i, the covariance matrix associated with such an estimate may be *ill-conditioned*. In particular, it is not hard to verify that for each $i \in [M] \setminus \mathcal{B}$:

$$\mathbb{E}\left[\left(\hat{\theta}_i^{(\ell)} - \theta_*\right)\left(\hat{\theta}_i^{(\ell)} - \theta_*\right)' \Big| \mathcal{F}_\ell\right] = \tilde{V}_\ell^{-1}.$$

Thus, if we were to construct $\hat{\theta}^{(\ell)}$ as

$$\hat{\theta}^{(\ell)} = \texttt{ITW}(\{\hat{\theta}_i^{(\ell)},\, i \in [M]\}),$$

then based on Lemma 8, the error bound $\|\hat{\theta}^{(\ell)} - \theta_*\|$ would scale with $\|\tilde{V}_\ell^{-1}\|_2^{1/2}$.[7] This is undesirable as $\|\tilde{V}_\ell^{-1}\|_2^{1/2}$ can potentially take on a large value. To bypass this problem, we can use the same trick as we did for `RC-GLM`, and compute $\hat{\theta}^{(\ell)}$ as follows:

$$\hat{\theta}^{(\ell)} = \tilde{V}_\ell^{-1/2}\left(\texttt{ITW}(\{\tilde{V}_\ell^{1/2}\hat{\theta}_i^{(\ell)},\, i \in [M]\})\right).$$

The rationale behind the above approach is that the covariance matrix associated with $\tilde{V}_\ell^{1/2}\hat{\theta}_i^{(\ell)}, i \in [M] \setminus \mathcal{B}$, is I_d. Using Lemma 8, and following similar arguments as used to arrive at Lemmas 1 and 10, one can show that for each $a \in \mathcal{A}_\ell$, with probability at least $1 - \delta_\ell$, it holds that:

$$|\langle \hat{\theta}^{(\ell)}, a\rangle - \langle \theta_*, a\rangle| = O\left(\left(\sqrt{d} + \alpha\sqrt{M\log(1/\alpha)}\right)\epsilon_\ell\right). \tag{2.83}$$

It is instructive to compare the above estimate on $\langle \theta_*, a\rangle$ with the estimate $\mu_a^{(\ell)}$ we used in `RCLB`.

[7] Recall that we use $\texttt{ITW}(\cdot)$ to represent the output of the Iteratively Reweighted Mean Estimator from (Dalalyan and Minasyan, 2022), namely Algorithm 5.

Specifically, recall from Lemma 1 that for each $a \in \mathcal{A}_\ell$, with probability at least $1 - \delta_\ell$, it holds that

$$|\mu_a^{(\ell)} - \langle \theta_*, a \rangle| = O\left(\left(1 + \alpha\sqrt{M}\right)\epsilon_\ell\right).$$

Comparing the above upper bound with the one in Eq. (2.83), we note that while the former is independent of the dimension d, the latter does exhibit a dependence via the $\sqrt{d}$ term. Now suppose we use the upper-bound from Eq. (2.83) to construct a robust confidence threshold - say $\tilde{\gamma}_\ell$ - and use it to devise a phased elimination step as the one in line 9 of `RCLB`. Then, following the reasoning as that used to prove Theorem 1, one can establish a per-agent regret bound of

$$\tilde{O}\left(\left(\alpha\sqrt{\log(1/\alpha)} + \sqrt{\frac{d}{M}}\right)\sqrt{dT}\right),$$

which is unfortunately weaker than the guarantee we have in Theorem 1.

• **Candidate Strategy 2.** The main idea is as follows. In every epoch ℓ, the server queries each agent $i \in [M]$ to report their aggregate observation $r_{i,a}^{(\ell)}$ for each arm $a \in \text{Supp}(\pi_\ell)$. Recall that $r_{i,a}^{(\ell)}$ is the average of the rewards for arm a observed by agent i during epoch ℓ. The server next computes an aggregate "clean" observation $\tilde{r}_a^{(\ell)}$ for each $a \in \text{Supp}(\pi_\ell)$ as follows:

$$\tilde{r}_a^{(\ell)} = \texttt{Median}\left(\{r_{i,a}^{(\ell)}, i \in [M]\}\right).$$

It then uses these clean observations to compute an estimate $\hat{\theta}^{(\ell)}$ of θ_* as follows:

$$\hat{\theta}^{(\ell)} = \bar{V}_\ell^{-1} Y_\ell, \text{ where } \bar{V}_\ell = \sum_{a \in \text{Supp}(\pi_\ell)} T_a^{(\ell)} a a' \; ; \; Y_\ell = \sum_{a \in \text{Supp}(\pi_\ell)} T_a^{(\ell)} \tilde{r}_a^{(\ell)} a,$$

and

$$T_a^{(\ell)} = \left\lceil \frac{\pi_\ell(a) d}{\epsilon_\ell^2} \log\left(\frac{1}{\delta_\ell}\right) \right\rceil.$$

The quantity $\hat{\theta}^{(\ell)}$ obtained above is now used to compute $\langle \hat{\theta}^{(\ell)}, a \rangle$ as an estimate of the true mean payoff $\langle \theta_*, a \rangle$ of each arm $a \in \mathcal{A}_\ell$.[8] As before, our goal is to bound $|\langle \hat{\theta}^{(\ell)}, a \rangle - \langle \theta_*, a \rangle|$ for each $a \in \mathcal{A}_\ell$.

[8]Note that the observations obtained from each agent i, namely $r_{i,a}^{(\ell)}, a \in \text{Supp}(\pi_\ell)$, provide direct information

To that end, we start by noting that for each good agent i, $r_{i,a}^{(\ell)} \sim \mathcal{N}\left(\langle \theta_*, a \rangle, \frac{1}{m_a^{(\ell)}}\right)$. Invoking Lemma 2 then tells us that with probability at least $1 - \delta_\ell$,

$$|\tilde{r}_a^{(\ell)} - \langle \theta_*, a \rangle| \leq C \left(\alpha + \sqrt{\frac{\log(\frac{1}{\delta_\ell})}{M}}\right) \frac{1}{\sqrt{m_a^{(\ell)}}} \leq C \left(\alpha\sqrt{M} + \sqrt{\log\left(\frac{1}{\delta_\ell}\right)}\right) \frac{1}{\sqrt{T_a^{(\ell)}}}. \tag{2.84}$$

For our subsequent discussion, let us condition on the event on which the above bound holds for every arm in $\mathcal{A}_\ell$. On this event, we can say that for each $a \in \mathcal{A}_\ell$, $\tilde{r}_a^{(\ell)} = \langle \theta_*, a \rangle + e_a^{(\ell)}$, where $e_a^{(\ell)}$ is an error term satisfying the bound in Eq. (2.84). Now fix any $b \in \mathcal{A}_\ell$. Simple calculations reveal that:

$$\begin{aligned}
|\langle \hat{\theta}^{(\ell)} - \theta_*, b \rangle| &= \Big| \sum_{a \in \mathrm{Supp}(\pi_\ell)} T_a^{(\ell)} \langle \bar{V}_\ell^{-1} a, b \rangle e_a^{(\ell)} \Big| \\
&\leq \sum_{a \in \mathrm{Supp}(\pi_\ell)} T_a^{(\ell)} \, |\langle \bar{V}_\ell^{-1} a, b \rangle| |e_a^{(\ell)}| \\
&\overset{(a)}{\leq} C \left(\alpha\sqrt{M} + \sqrt{\log\left(\frac{1}{\delta_\ell}\right)}\right) \underbrace{\sum_{a \in \mathrm{Supp}(\pi_\ell)} \sqrt{T_a^{(\ell)}} |\langle \bar{V}_\ell^{-1} a, b \rangle|}_{T_1} \\
&\overset{(b)}{\leq} C \left(\alpha\sqrt{M} + \sqrt{\log\left(\frac{1}{\delta_\ell}\right)}\right) \sqrt{|\mathrm{Supp}(\pi_\ell)| \left(b' \bar{V}_\ell^{-1} \left(\sum_{a \in \mathrm{Supp}(\pi_\ell)} T_a^{(\ell)} a a'\right) \bar{V}_\ell^{-1} b\right)} \\
&\leq C \left(\alpha\sqrt{M} + \sqrt{\log\left(\frac{1}{\delta_\ell}\right)}\right) \sqrt{|\mathrm{Supp}(\pi_\ell)|} \, \|b\|_{\bar{V}_\ell^{-1}} \\
&\overset{(c)}{\leq} C \left(\alpha\sqrt{M} + \sqrt{\log\left(\frac{1}{\delta_\ell}\right)}\right) \sqrt{48 d \log\log d} \, \|b\|_{\bar{V}_\ell^{-1}} \\
&\overset{(d)}{=} \tilde{O}\left(\sqrt{d}\left(1 + \alpha\sqrt{M}\right)\epsilon_\ell\right).
\end{aligned} \tag{2.85}$$

In the above steps, for (a) we used the bound from Eq. (2.84); for (b), we used Jensen's inequality; for (c), we used the fact that $|\mathrm{Supp}(\pi_\ell)| \leq 48 d \log\log d$; and for (d), following a similar argument as

about the mean payoffs of arms *only* in $\mathrm{Supp}(\pi_\ell)$. However, for the phased elimination step, we need estimates of the mean payoffs of *all arms in* $\mathcal{A}_\ell$, not just the ones in $\mathrm{Supp}(\pi_\ell) \subseteq \mathcal{A}_\ell$. This is precisely why we need to go through an intermediate regression step to first compute an estimate of θ_*.

in the proof of Lemma 1, we used that

$$\|b\|_{\bar{V}_\ell^{-1}} = O\left(\frac{\epsilon_\ell}{\sqrt{\log\left(1/\delta_\ell\right)}}\right).$$

Comparing the bound in Eq. (2.85) with the one in Lemma 1, we once again note that while the latter bound is d-independent, the former has a clear dependence on $\sqrt{d}$. At the risk of sounding repetitive, if one were to employ the bound in Eq. (2.85) to construct a confidence threshold for phased elimination, and run through the same arguments as in the proof of Theorem 1, one would end up with a per-agent regret bound of

$$\tilde{O}\left(\left(1+\alpha\sqrt{M}\right)d\sqrt{T}\right).$$

Unlike the near-optimal guarantee we have in Theorem 1, the above bound is clearly off by a factor of $\sqrt{d}$ from the optimal dependence on the model dimension d. The looseness in the bound mainly stems from the following fact: the error terms $\{e_a^{(\ell)}\}_{a\in\text{Supp}(\pi_\ell)}$ are not necessarily sub-Gaussian random variables that are independent across arms. One can contrast this to the analysis in Lemma 1, where the noise terms $\{\bar{\eta}_{i,a}^{(\ell)}\}_{a\in\text{Supp}(\pi_\ell)}$ were in fact Gaussian, and independent across arms. It is precisely the lack of nice statistical properties for the error terms $\{e_a^{(\ell)}\}_{a\in\text{Supp}(\pi_\ell)}$ that compels us to use Jensen's inequality to bound the term T_1 in Eq. (2.85). At the moment, it is unclear to us whether one can come up with a tighter bound for this candidate strategy.

Main Takeaway. The main message from this section is that deriving robust confidence intervals that lead to near-optimal bounds (such as the one in Theorem 1) is non-trivial, and requires a lot of care. In particular, the above discussion serves to highlight the significance of our algorithmic approach.

2.15. Experimental Results

In this section, we will provide various simulation results on synthetic data to corroborate the theory developed in our work. We start by describing the experimental setup for the linear bandit setting considered in Section 3.6.

2.15.1. Experiments for the Linear Bandit Setting

Linear Bandit Experimental Setup. We generate 50 arms $a_1, a_2, \ldots, a_{50} \in \mathbb{R}^d$, where $d = 5$. Each arm $a_j, j \in [50]$, is generated by drawing each of the arm's coordinates i.i.d from the interval $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$. It thus follows that $\|a_j\|_2 \leq 1, \forall j \in [50]$. The model parameter θ_* is chosen to be a 5-dimensional vector with each entry equal to $1/\sqrt{d}$. The rewards are generated based on the observation model in Eq. (2.1). We now describe the attack model for the linear bandit setting.

Attack Model for Linear Bandit Setting. The collective goal of the adversarial agents is to manipulate the server into selecting sub-optimal arms. To that end, each adversarial agent i employs the simple strategy of reducing the rewards of the good arms and increasing the rewards of the bad arms. More precisely, in each epoch ℓ, upon pulling an arm a and observing the corresponding reward y_a, an adversarial agent i does the following: if $y_a > p\langle\theta_*, a_*\rangle$, then this reward is corrupted to $\tilde{y}_a = y_a - \beta$; and if $y_a \leq p\langle\theta_*, a_*\rangle$, then the reward is corrupted to $\tilde{y}_a = y_a + \beta$. For this experiment, we fix $p = 0.6$ and $\beta = 5$. Agent $i \in \mathcal{B}$ then uses all the corrupted rewards in epoch ℓ to generate the local model estimate $\hat{\theta}_i^{(\ell)}$ that is transmitted to the server.

An Alternate Attack Model. To further test the robustness of `RCLB`, we consider an attack model different from the one in Section 3.6. In this attack, in each epoch ℓ, every adversarial agent $i \in \mathcal{B}$ generates and transmits the following corrupted local model estimate to the server:

$$\hat{\theta}_i^{(\ell)} = -\frac{M}{|\mathcal{B}|}\theta_* - \frac{1}{|\mathcal{B}|}\sum_{j \in [M]\setminus\mathcal{B}} \hat{\theta}_j^{(\ell)}. \tag{2.86}$$

The idea behind the above attack is to trick the server into thinking that the true model estimate is $-\theta_*$, as opposed to θ_*, by shifting the average of the agents' local model estimates towards $-\theta_*$.

As we can see from Figure 2.2, the adversarial agents succeed in doing so when one employs a vanilla non-robust distributed phased elimination algorithm. However, our proposed approach `RCLB` continues to remain immune to such attacks, and guarantees sub-linear regret as suggested by our theory.

2.15.2. Experiments for the Contextual Bandit Setting

The goal of this section is to validate our proposed robust collaborative algorithm for the contextual bandit setting, namely Algorithm 3.

Contextual Bandit Experimental Setup. As in the linear bandit experiment, we set the number of arms K to be 50, the model dimension d to be 5, and the true parameter θ_* to be a d-dimensional vector with each entry equal to $1/\sqrt{d}$. At each time-step t, for each $a \in \mathcal{A}$, we generate the feature vector $x_{t,a}$ by drawing each of its entries i.i.d from the interval $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$. The rewards are then generated based on the observation model in Eq. (2.12).

Attack Model for Contextual Bandit Setting. We use an attack strategy similar in spirit to the first attack model for the linear bandit setting. Specifically, at each time-step t, each adversarial agent $i \in \mathcal{B}$ does the following: if $r_{i,t} > p\langle\theta_*, x_{t,a_t^*}\rangle$, then the attacker sets $\tilde{r}_{i,t} = r_{i,t} - \beta$; if $r_{i,t} < p\langle\theta_*, x_{t,a_t^*}\rangle$, then the attacker sets $\tilde{r}_{i,t} = r_{i,t} + \beta$. The corrupted reward $\tilde{r}_{i,t}$ is then sent to the server. In this experiment, we fixed $p = 0.6$ and $\beta = 5$.

Discussion of Simulation Results. Figure 2.3 illustrates the results for the contextual linear bandit experiment. In Figure 2.3(a), we compare our proposed algorithm, namely Algorithm 3, to a naive distributed implementation of Algorithm 3 that does not account for adversarial agents. Similar to what we observed in Figure 2.1(a), while the non-robust algorithm incurs linear regret in the presence of adversaries, Algorithm 3 continues to guarantee sub-linear regret bounds. The plots in Figures 2.3(b)-(d) are analogous to the ones in Figures 2.1(b)-(d). In short, these plots once again indicate a clear benefit of collaboration (for small α) in the presence of adversarial agents, thereby highlighting the importance of Algorithm 3, and validating Theorem 4.

Algorithm 4 Robust Collaborative Phased Elimination for Generalized Linear Bandits (`RC-GLM`)

Input: Action set $\mathcal{A} = \{a_1, \ldots, a_K\}$, confidence parameter δ, and corruption fraction α.
Initialize: $\ell = 1$ and $\mathcal{A}_1 = \mathcal{A}$.

1: Let $V_\ell(\pi) \triangleq \sum_{a \in \mathcal{A}_\ell} \pi(a) aa'$ and $g_\ell(\pi) \triangleq \max_{a \in \mathcal{A}_\ell} \|a\|^2_{V_\ell(\pi)^{-1}}$. Server solves an approximate G-optimal design problem to compute a distribution π_ℓ over $\mathcal{A}_\ell$ such that $g_\ell(\pi_\ell) \leq 2d$ and $|\mathrm{Supp}(\pi_\ell)| \leq 48 d \log\log d$.

2: For each $a \in \mathcal{A}_\ell$, server computes $m_a^{(\ell)}$ via Eq. (2.5), and broadcasts $\{m_a^{(\ell)}\}_{a \in \mathcal{A}_\ell}$ to all agents.

3: **for** $i \in [M] \setminus \mathcal{B}$ **do**

4: For each arm $a \in \mathcal{A}_\ell$, pull it $m_a^{(\ell)}$ times. Let $r_{i,a}^{(\ell)}$ be the average of the rewards observed by agent i for arm a during phase ℓ.

5: Compute $\tilde{V}_\ell$ and $Y_{i,\ell}$ as follows.

$$\tilde{V}_\ell = \sum_{a \in \mathrm{Supp}(\pi_\ell)} m_a^{(\ell)} aa' \; ; \; Y_{i,\ell} = \sum_{a \in \mathrm{Supp}(\pi_\ell)} m_a^{(\ell)} r_{i,a}^{(\ell)} a.$$

6: Transmit $Y_{i,\ell}$ to server. Adversarial agents can transmit arbitrary vectors at this stage.

7: **end for**

8: Server computes a statistic X_ℓ as follows:

$$X_\ell = \texttt{ITW}(\{\tilde{V}_\ell^{-1/2} Y_{i,\ell}, i \in [M]\}),$$

where $\texttt{ITW}(\cdot)$ is the output of the Iteratively Reweighted Mean Estimator from (Dalalyan and Minasyan, 2022).

9: Server computes a robust estimate $\hat{\theta}^{(\ell)}$ of θ_* by solving:

$$h_\ell(\hat{\theta}^{(\ell)}) = \tilde{V}_\ell^{1/2} X_\ell.$$

10: Define robust confidence threshold $\bar{\gamma}_\ell = 4C_1(k_2/k_1)\left(\sqrt{d} + \alpha\sqrt{M \log(1/\alpha)}\right)\epsilon_\ell$, where $\epsilon_\ell = 2^{-\ell}$, and C_1 is as in Lemma 8. Server performs phased elimination with the estimate $\hat{\theta}^{(\ell)}$ and confidence threshold $\bar{\gamma}_\ell$ to update active arm set:

$$\mathcal{A}_{\ell+1} = \{a \in \mathcal{A}_\ell : \max_{b \in \mathcal{A}_\ell} \mu(\langle \hat{\theta}^{(\ell)}, b\rangle) - \mu(\langle \hat{\theta}^{(\ell)}, a\rangle) \leq 2\bar{\gamma}_\ell\}. \tag{2.46}$$

11: $\ell = \ell + 1$ and **Goto** line 1.

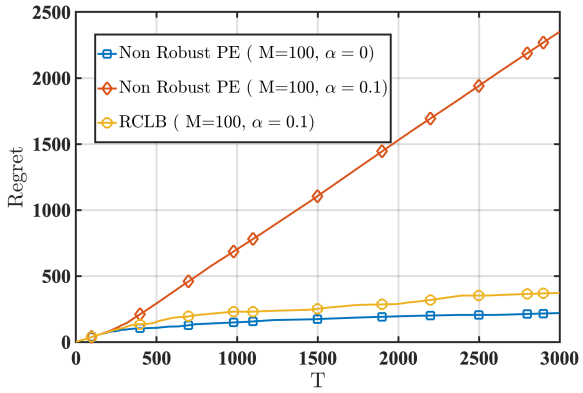

Figure 2.2: Performance of a vanilla non-robust distributed phased elimination algorithm vs `RCLB` for the attack model in Eq. (2.86).

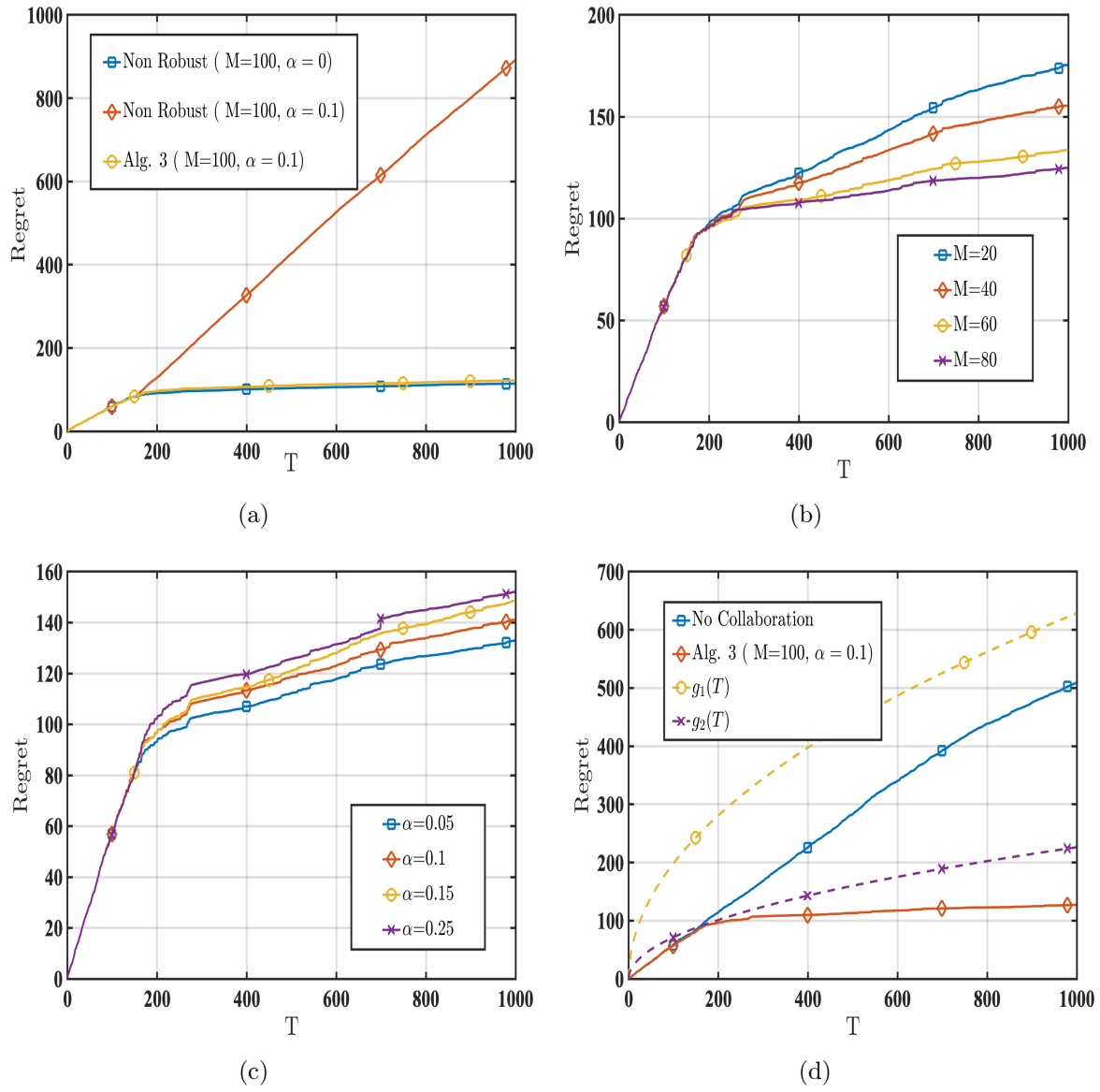

Figure 2.3: Plots of per-agent regret for the contextual bandit experiment. (a) Comparison between our proposed algorithm, namely Algorithm 3, and a vanilla non-robust distributed contextual bandit algorithm. (b) Performance of Algorithm 3 for varying number of agents M, with $\alpha = 0.1$. (c) Performance of Algorithm 3 for varying corruption fraction α, with $M = 100$. (d) Comparison of Algorithm 3 to a non-robust contextual bandit algorithm where the agents do not collaborate; here, $\alpha = 0.1$ and $M = 100$. We also plotted theoretical upper bounds: $g_1(T) = 3\sqrt{dT}$ and $g_2(T) = 17(\alpha + \sqrt{\frac{1}{M}})\sqrt{dT}$.

CHAPTER 3

Distributed Min-Max Learning in the Presence of Byzantine Agents

3.1. Introduction

We consider a min-max learning problem of the form

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} f(x,y) \triangleq \mathbb{E}_{\xi\sim\mathcal{D}}[F(x,y;\xi)]. \tag{3.1}$$

Here, $\mathcal{X}$ and $\mathcal{Y}$ are convex, compact sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively; $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are model parameters; ξ is a random variable representing a data point sampled from the distribution $\mathcal{D}$; and $f(x,y)$ is the population function corresponding to the stochastic function $F(x,y;\xi)$. Throughout this paper, we assume that $f(x,y)$ is continuously differentiable in x and y, and is *convex-concave* over $\mathcal{X}\times\mathcal{Y}$. Specifically, $f(\cdot,y):\mathcal{X}\to\mathbb{R}$ is convex for every $y\in\mathcal{Y}$, and $f(x,\cdot):\mathcal{Y}\to\mathbb{R}$ is concave for every $x\in\mathcal{X}$. Our goal is to find a saddle point (x^*,y^*) of $f(x,y)$ over the set $\mathcal{X}\times\mathcal{Y}$, where a saddle point is defined as a vector pair $(x^*,y^*)\in\mathcal{X}\times\mathcal{Y}$ that satisfies

$$f(x^*,y) \le f(x^*,y^*) \le f(x,y^*), \forall x\in\mathcal{X}, y\in\mathcal{Y}. \tag{3.2}$$

The min-max optimization problem described above features in a variety of applications: from classical developments in game theory (Von Neumann and Morgenstern, 2007) and online learning (Cesa-Bianchi and Lugosi, 2006), to robust optimization (Ben-Tal et al., 2009a) and reinforcement learning (Dai et al., 2017). More recently, in the context of machine learning, min-max problems have found important applications in training generative adversarial networks (GANs) (Goodfellow et al., 2014a), and in robustifying deep neural networks against adversarial attacks (Madry et al., 2017a). Motivated by this recent line of work, we consider a min-max learning problem of the form in Eq. (3.1), where the data samples required for finding a saddle-point are distributed across multiple devices (agents). Specifically, we focus on a large-scale distributed setup comprising of M agents, each of which can access i.i.d. data samples from the distribution $\mathcal{D}$. The agents collaborate under

the orchestration of a central server to compute an approximate saddle point of statistical accuracy higher relative to the setting when they act alone. The intuition here is simple: since all agents receive data samples from the *same* distribution, exchanging information via the server can help reduce the randomness (variance) associated with these samples.[9] An example of the above setup that aligns with the modern federated learning paradigm is one where multiple devices (e.g., cell phones or tablets) collaborate via a server to train a robust statistical model; see, for instance, (Reisizadeh et al., 2020).

To reap the benefits of collaboration in modern distributed computing systems, one needs to contend with the critical challenge of *security*. In particular, this challenge arises from the fact that the individual agents in such systems are easily susceptible to adversarial attacks. In fact, unless appropriately accounted for, even a single malicious agent can severely degrade the overall performance of the system by sending corrupted messages to the central server.

Objective. Thus, given the emerging need for security in large-scale computing, *our objective in this paper is to design an algorithm that achieves near-optimal statistical performance in the context of distributed min-max learning, while being robust to worst-case attacks.* To that end, we consider a setting where a fraction of the agents is Byzantine (Lamport et al., 2019). Each Byzantine agent is assumed to have complete knowledge of the system and learning algorithms; moreover, leveraging such knowledge, the Byzantine agents can send arbitrary messages to the server and collude with each other.

Challenges. Even in the absence of noise or attacks, recent work (Daskalakis et al., 2017a) has shown that algorithms such as gradient descent ascent (`GDA`) can diverge for simple convex-concave functions. We have to contend with both noise (due to our statistical setup) *and* worst-case attacks - this makes the analysis for our setting non-trivial. In particular, the adversarial agents can introduce complex probabilistic dependencies across iterations that need to be carefully accounted for; we do so in this work by making the following contributions.

[9] This intuition will be made precise in Section 3.4.

Contributions. Our contributions are summarized below.

• *Problem.* Given the importance and relevance of security, several recent works have studied distributed optimization/learning in the face of adversarial agents. However, we are unaware of any analogous paper for adverarially-robust distributed *min-max* learning. Our work closes this gap.

• *Algorithm.* In Section 3.3, we develop an algorithm for finding an approximate saddle point to the min-max learning problem in Eq. (3.1), subject to the presence of Byzantine agents. Our proposed algorithm - called Robust Distributed Extra-Gradient (`RDEG`) - brings together two separate algorithmic ideas: (i) the classical extra-gradient algorithm due to Korpelevich (Korpelevich, 1976a) that has gained a lot of popularity due to its empirical performance in training GANs, and (ii) the recently proposed univariate trimmed mean estimator due to Lugosi and Mendelson (Lugosi and Mendelson, 2021).

• *Theoretical Results.* Our main contribution is to provide a rigorous theoretical analysis of the performance of `RDEG` for smooth convex-concave (Theorem 7), and smooth strongly convex-strongly concave (Theorem 8) settings. In each case, we establish that as long as the fraction of corrupted agents is "small", `RDEG` guarantees convergence to approximate saddle points at *near-optimal* statistical rates with high probability. The rates that we derive precisely highlight the benefit of collaboration in effectively reducing the variance of the noise model. At the same time, they indicate the (unavoidable) additive bias introduced by adversarial corruption. Notably, our results in the context of min-max learning complement those of a similar flavor in (Yin et al., 2018) for stochastic optimization under attacks. However, our analysis differs significantly from that in (Yin et al., 2018): unlike the covering argument employed in (Yin et al., 2018), our proofs rely on a simpler, and more direct probabilistic analysis. An immediate benefit of such an analysis is that one can build on it for the more challenging nonconvex-nonconcave setting as future work.

Related Work. In what follows, we discuss connections to relevant strands of literature.

• *Min-Max Optimization.* Convergence guarantees of first-order algorithms for saddle point problems over compact sets were studied in (Nemirovski, 2004a) and (Nedić and Ozdaglar, 2009). More re-

cently, there has been a surge of interest in analyzing the performance of such algorithms from different perspectives: a dynamical systems approach in (Liang and Stokes, 2019; Daskalakis and Panageas, 2018), and a proximal point perspective in (Mokhtari et al., 2020c). We refer to (Lin et al., 2020c) for a detailed survey on this topic.

• *Robust Distributed Optimization and Learning.* Robustness to adversarial agents in distributed optimization has been extensively studied in (Su and Vaidya, 2016; Sundaram and Gharesifard, 2018; Ravi et al., 2019). However, these works consider deterministic settings, and do not provide statistical error rates like we do. In the context of statistical learning over a server-client computing architecture, several works have proposed and analyzed robust algorithms (Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2018c, 2017b; Pillutla et al., 2022). Notably, none of the above works consider the min-max learning problem studied in this paper.

• *Robust Statistics.* Robust mean estimation in the presence of outliers is a classical topic in statistics pioneered by Huber (Huber, 1992), with follow-up work in (Cheng et al., 2019). In our work, we exploit some recent results on this topic from (Lugosi and Mendelson, 2021).

3.2. Problem Formulation

In this section, we formally set up the problem of interest by first introducing some notation. Our setting comprises of M agents, αM of whom are Byzantine; see Fig. 3.1. We denote the adversarial agents by $\mathcal{B} \in [M]$.[10] For any $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$, let $\nabla_x f(\bar{x}, \bar{y})$ and $\nabla_y f(\bar{x}, \bar{y})$ denote the gradient of $f(x, y)$ with respect to x and y, respectively, at $(\bar{x}, \bar{y})$. Upon drawing a sample $\xi \sim \mathcal{D}$ at a point $(\bar{x}, \bar{y})$, each normal agent receives noisy estimates of $\nabla_x f(\bar{x}, \bar{y})$ and $\nabla_y f(\bar{x}, \bar{y})$ denoted by $g_x(\bar{x}, \bar{y}; \xi)$ and $g_y(\bar{x}, \bar{y}; \xi)$, respectively. For each normal agent in $[M] \setminus \mathcal{B}$, these noisy estimates satisfy the following for all $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$:

$$\begin{aligned} \mathbb{E}_{\xi \sim \mathcal{D}}[g_x(\bar{x}, \bar{y}; \xi)] &= \nabla_x f(\bar{x}, \bar{y}) \\ \mathbb{E}_{\xi \sim \mathcal{D}}[g_y(\bar{x}, \bar{y}; \xi)] &= \nabla_y f(\bar{x}, \bar{y}). \end{aligned} \tag{3.3}$$

[10] Given a positive integer N, we use $[N]$ to represent the set $\{1, \ldots, N\}$.

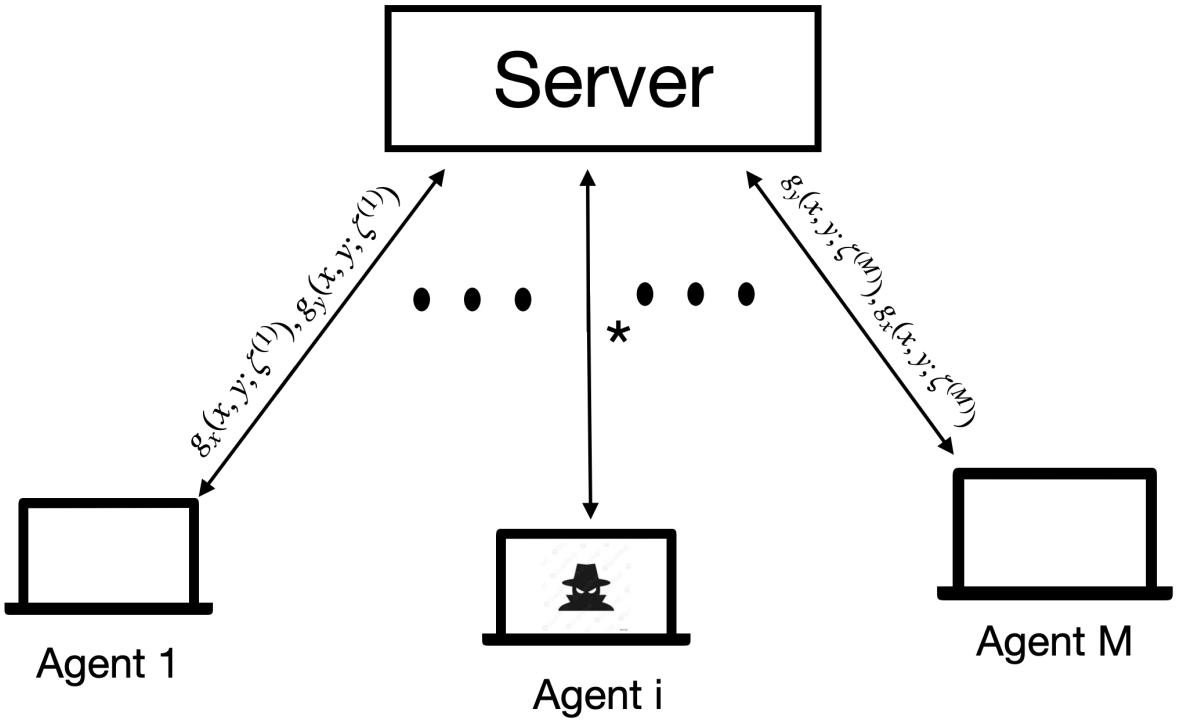

Figure 3.1: A group of M agents collaborate to find a saddle point for the min-max learning problem in Eq. (3.1). A fraction α of the agents is adversarial and upload arbitrarily corrupted messages (denoted by $*$) to the server. All the remaining good agents upload noisy partial gradients of $f(x, y)$.

Furthermore, $\forall j \in [n]$ and $\forall k \in [m]$, we have

$$\begin{aligned} \mathbb{E}_{\xi\sim\mathcal{D}}\left[\|[g_x(\bar{x},\bar{y};\xi)]_j - [\nabla_x f(\bar{x},\bar{y})]_j\|^2\right] &\leq \sigma_x^2(j) \\ \mathbb{E}_{\xi\sim\mathcal{D}}\left[\|[g_y(\bar{x},\bar{y};\xi)]_k - [\nabla_y f(\bar{x},\bar{y})]_k\|^2\right] &\leq \sigma_y^2(k), \end{aligned} \tag{3.4}$$

where we used $[a]_j$ to represent the j-th component of a vector a.[11] In words, each normal agent receives unbiased estimates of the gradients of $f(x, y)$ (w.r.t. x and y) with component-wise bounded variance - essentially, a standard stochastic oracle model. With a slight abuse of notation, we will continue to use $\{g_x(x, y; \xi), g_y(x, y; \xi)\}$ to denote the gradients transmitted by an adversarial agent as well; these could, however, be arbitrary corrupted vectors. Our problem of interest can now be stated as follows.

Problem 5. *Given access to the stochastic oracle model described by equations* (3.3) *and* (3.4),

[11] We use $\|\cdot\|$ to represent the Euclidean norm.

Algorithm 6 Robust Distributed Extra-Gradient (`RDEG`)

Require: Initial vectors $x_1 \in \mathcal{X}$, $y_1 \in \mathcal{Y}$; algorithm parameters: step-size $\eta > 0$ and trimming parameter ϵ.

1: **for** $t = 1, \ldots, T$ **do**

2: Server sends (x_t, y_t) to each agent.

3: Each *normal* agent i draws an i.i.d. sample $\xi_{1,t}^{(i)} \sim \mathcal{D}$, and transmits $g_x(x_t, y_t; \xi_{1,t}^{(i)})$, $g_y(x_t, y_t; \xi_{1,t}^{(i)})$ to server.[12]

4: Server computes robust gradients:

$$\begin{aligned} \tilde{g}_x(x_t, y_t) &\leftarrow \texttt{Trim}_\epsilon\{g_x(x_t, y_t; \xi_{1,t}^{(i)}) : i \in [M]\} \\ \tilde{g}_y(x_t, y_t) &\leftarrow \texttt{Trim}_\epsilon\{g_y(x_t, y_t; \xi_{1,t}^{(i)}) : i \in [M]\}. \end{aligned} \tag{3.5}$$

5: Server computes mid-points $(\hat{x}_t, \hat{y}_t)$ as follows, and transmits them to each agent.

$$\begin{aligned} \hat{x}_t &\leftarrow \Pi_{\mathcal{X}}\left(x_t - \eta \tilde{g}_x(x_t, y_t)\right) \\ \hat{y}_t &\leftarrow \Pi_{\mathcal{Y}}\left(y_t + \eta \tilde{g}_y(x_t, y_t)\right). \end{aligned} \tag{3.6}$$

6: Each *normal* agent i draws an i.i.d. sample $\xi_{2,t}^{(i)} \sim \mathcal{D}$, and transmits $g_x(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)})$, $g_y(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)})$ to server.

7: Server computes robust gradients:

$$\begin{aligned} \tilde{g}_x(\hat{x}_t, \hat{y}_t) &\leftarrow \texttt{Trim}_\epsilon\{g_x(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)}) : i \in [M]\} \\ \tilde{g}_y(\hat{x}_t, \hat{y}_t) &\leftarrow \texttt{Trim}_\epsilon\{g_y(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)}) : i \in [M]\}. \end{aligned} \tag{3.7}$$

8: Server computes new updates x_{t+1} and y_{t+1}:

$$\begin{aligned} x_{t+1} &\leftarrow \Pi_{\mathcal{X}}\left(x_t - \eta \tilde{g}_x(\hat{x}_t, \hat{y}_t)\right) \\ y_{t+1} &\leftarrow \Pi_{\mathcal{Y}}\left(y_t + \eta \tilde{g}_x(\hat{x}_t, \hat{y}_t)\right). \end{aligned} \tag{3.8}$$

9: **end for**

design a distributed algorithm that finds a saddle point (in the sense of Eq. (5.2)*) for the function* $f(x, y)$ *in Eq.* (3.1)*, despite the presence of the Byzantine adversarial set* $\mathcal{B}$*.*

In the next section, we will develop our proposed algorithm to address Problem 5.

3.3. Robust Distributed Extra-Gradient

In this section, we develop the Robust Distributed Extra-Gradient (`RDEG`) algorithm outlined in Algorithm 6. Our algorithm evolves in discrete-time iterations $t \in [T]$, where T is the total number

[12] Recall that $\{g_x(x_t, y_t; \xi_{1,t}^{(i)}), g_y(x_t, y_t; \xi_{1,t}^{(i)})\}$ could be arbitrary vectors for an adversarial agent $i \in \mathcal{B}$.

of iterations. There are two main steps in RDEG. In the first step, the server computes robust gradient estimates $\{\tilde{g}_x(x_t, y_t), \tilde{g}_y(x_t, y_t)\}$ at the current iterate (x_t, y_t) by applying a Trim operator to the gradients collected from all agents (line 4); we will describe this operator shortly. The robust gradient estimates are then used to compute a mid-point $(\hat{x}_t, \hat{y}_t)$ by performing a projected primal-dual update (line 5). In the second step, the server now computes robust gradients at the mid-point (line 7), and performs a projected primal-dual update using these gradients to generate the next iterate (x_{t+1}, y_{t+1}). We now describe the Trim operation.

The Trim operator in equations (3.5) and (3.7) takes as input M vectors, and applies the univariate trimmed mean estimator in (Lugosi and Mendelson, 2021) - described in Algorithm 7 - to each coordinate of these vectors separately. To describe the trimmed mean estimator, suppose the data comprises of M independent copies of a scalar random variable Z with mean μ_Z and variance σ_Z^2. An adversary corrupts at most αM of these copies; the corrupted data-set is then made available to the estimator. The estimator splits the corrupted data set into two equal chunks, denoted by $Z_1, \ldots, Z_{M/2}$, $\tilde{Z}_1, \ldots \tilde{Z}_{M/2}$. One of the chunks is used to compute appropriate quantile levels for truncation (line 2 of Algo. 7). The robust estimate $\hat{\mu}_Z$ of μ_Z is an average of the data points in the other chunk, with those data points falling outside the estimated quantile levels truncated prior to averaging (line 3 of Algo. 7).

Algorithm 7 Univariate Trimmed-Mean Estimator (Lugosi and Mendelson, 2021)

Require: Corrupted data set $Z_1, \ldots, Z_{M/2}$, $\tilde{Z}_1, \ldots \tilde{Z}_{M/2}$, corruption fraction α, and confidence level δ.

1: Set $\epsilon = 8\alpha + 24\frac{\log(4/\delta)}{M}$.

2: Let $Z_1^* \leq Z_2^* \leq \cdots \leq Z_{M/2}^*$ represent a non-decreasing arrangement of $\{Z_i\}_{i\in[M/2]}$. Compute quantiles: $\gamma = Z^*_{\epsilon M/2}$ and $\beta = Z^*_{(1-\epsilon)M/2}$.

3: Compute robust mean estimate $\hat{\mu}_Z$ as follows:

$$\hat{\mu}_Z = \frac{2}{M}\sum_{i=1}^{M/2} \phi_{\gamma,\beta}(\tilde{Z}_i); \phi_{\gamma,\beta}(x) = \begin{cases} \beta & x > \beta \\ x & x \in [\gamma, \beta] \\ \gamma & x < \gamma \end{cases}$$

The following result on the performance of Algorithm 7 will play a key role in our subsequent analysis of RDEG.

Theorem 6. *(Lugosi and Mendelson, 2021, Theorem 1) Consider the trimmed mean estimator in Algorithm 7. Suppose $\alpha \in [0, 1/16)$, and let $\delta \in (0, 1)$ be such that $\delta \geq 4e^{-M/2}$. Then, there exists an universal constant c, such that with probability at least $1 - \delta$,*

$$|\hat{\mu}_Z - \mu_Z| \leq c\sigma_Z \left(\sqrt{\alpha} + \sqrt{\frac{\log(1/\delta)}{M}} \right).$$

In the next section, we will provide rigorous guarantees on the performance of our proposed algorithm `RDEG`.

3.4. Performance Guarantees for `RDEG`

Before stating our main results, we first make a standard smoothness assumption on the function $f(x, y)$.

Assumption 2. *There exists a constant $L > 0$ such that the following holds for all $x_1, x_2 \in \mathcal{X}$, and all $y_1, y_2 \in \mathcal{Y}$:*

$$\|\nabla_x f(x_1, y_1) - \nabla_x f(x_2, y_2)\| \leq L \left(\|x_1 - x_2\| + \|y_1 - y_2\| \right),$$
$$\|\nabla_y f(x_1, y_1) - \nabla_y f(x_2, y_2)\| \leq L \left(\|x_1 - x_2\| + \|y_1 - y_2\| \right).$$

We now define a few key quantities that will show up in our main results. Let $\sigma_x = \sqrt{\sum_{j \in [n]} \sigma_x^2(j)}$, $\sigma_y = \sqrt{\sum_{k \in [m]} \sigma_y^2(k)}$, and $\sigma = \max\{\sigma_x, \sigma_y\}$. Moreover, let $d = \max\{n, m\}$, and $D = \max\{D_x, D_y\}$, where D_x and D_y are the diameters of the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. With the above notations in place, we state our first main result that provides a bound on the primal-dual gap $\phi_T \triangleq \max_{y \in \mathcal{Y}} f(\bar{x}_T, y) - \min_{x \in \mathcal{X}} f(x, \bar{y}_T)$, where

$$\bar{x}_T = (1/T) \sum_{t \in [T]} \hat{x}_t, \text{ and } \bar{y}_T = (1/T) \sum_{t \in [T]} \hat{y}_t.$$

Theorem 7. *Suppose Assumption 11 holds, the fraction α of corrupted devices satisfies $\alpha \in [0, 1/16)$, and the number of agents M is sufficiently large: $M \geq 48 \log(16dT^2)$. Then, with a step-size η satisfying $\eta \leq 1/(2L)$, and the confidence parameter δ in Algorithm 7 set to $\delta = 1/(4dT^2)$, `RDEG`*

guarantees the following with probability at least $1 - 1/T$*:*

$$\phi_T \leq \frac{D^2}{\eta T} + \tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right). \tag{3.9}$$

Discussion. Theorem 7 tells us that with high probability, the primal-dual gap ϕ_T converges to a ball of radius $\tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{1/M}\right)\right)$ at a $O(1/T)$ rate.[13] Notably, the primal-dual gap is zero if and only if $(\bar{x}_T, \bar{y}_T)$ is a saddle point of $f(x, y)$ over the set $\mathcal{X} \times \mathcal{Y}$. Thus, `RDEG` provably generates approximate saddle points. The following result is one of the main implications of Theorem 7.

Corollary 2. *Suppose the conditions in Theorem 7 hold. Then,* `RDEG` *guarantees the following with probability at least* $1 - 1/T$*:*

$$|f(\bar{x}_T, \bar{y}_T) - f(x^*, y^*)| \leq \frac{D^2}{\eta T} + \tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right). \tag{3.10}$$

Corollary 2 tells us that with high probability, the function values $f(\hat{x}_t, \hat{y}_t)$ of the averaged iterates generated by `RDEG` converge to the saddle-point value $f(x^*, y^*)$ up to an error-floor of $\tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{1/M}\right)\right)$, at a $O(1/T)$ rate. There are several key messages from this result. First, in the absence of adversaries (i.e., when $\alpha = 0$), the classical extra-gradient algorithm with a constant step-size would yield convergence to the saddle-point value with an error floor of $\tilde{O}(\sigma(\sqrt{1/M}))$ at a $O(1/T)$ rate. Thus, modulo the biasing effect of the adversaries, the statistical performance of `RDEG` is *near-optimal*. Second, the additive biasing effect due to adversarial corruption shows up even in the context of stochastic minimization (Yin et al., 2018). In fact, the authors in (Yin et al., 2018) argue that an additive biasing effect of order $\tilde{\Omega}(\alpha)$ is unavoidable, albeit for the minimization setting. This is all to say that the dependence of our rate on the corruption level in Eq. (3.10) is only to be expected. Third, when the corruption level is small, the benefit of collaboration is evident from the second term in Eq. (3.10): the variance σ arising from the noise term is effectively reduced by a factor of $\sqrt{M}$ due to the averaging effect of the normal agents. This effect will be aptly demonstrated by the simulations in Section 3.6.

[13] In the statement of our results, we will use the $\tilde{O}(\cdot)$ notation to hide terms that are logarithmic in n, m, and T.

We now turn to the goal of achieving faster convergence rates than those in Theorem 7. To that end, we study the performance of the RDEG algorithm for strongly convex-strongly concave (SC-SC) functions. Accordingly, we first make the following assumption on $f(x, y)$.

Assumption 3. *The function $f(x, y)$ is μ-strongly convex-μ-strongly concave (SC-SC) over $\mathcal{X} \times \mathcal{Y}$, i.e., for all $x_1, x_2 \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, the following holds:*

$$f(x_2, y_1) \geq f(x_1, y_1) + \langle \nabla_x f(x_1, y_1), x_2 - x_1 \rangle + \frac{\mu}{2}\|x_2 - x_1\|^2,$$
$$f(x_1, y_2) \leq f(x_1, y_1) + \langle \nabla_y f(x_1, y_1), y_2 - y_1 \rangle - \frac{\mu}{2}\|y_2 - y_1\|^2.$$

For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, define $z \triangleq [x; y]$. We have the following result for functions satisfying Assumption 12.

Theorem 8. *Suppose Assumptions 11 and 12 hold in conjunction with the assumptions on α and M in Theorem 7. Then, with $\delta = 1/(4dT^2)$ and step-size $\eta \leq 1/(4L)$, RDEG guarantees the following with probability at least $1 - 1/T$:*

$$\|z^* - z_{T+1}\|^2 \leq 2e^{-\frac{T}{4\kappa}} D^2 + \tilde{O}\left(\frac{\sigma D \kappa}{L}\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right), \tag{3.11}$$

where $\kappa = \mu/L$.

Theorem 8 says that for smooth strongly convex-strongly concave functions, the iterates generated by RDEG converge linearly to a ball around the saddle point (x^*, y^*) with high probability. The size of the ball is dictated by the second term in Eq. (3.11).

Remark 4. *(Comments on α) The requirement that the fraction of corruption $\alpha \in [0, 1/16)$ in our results is inherited from the analysis of the trimmed mean estimator in (Lugosi and Mendelson, 2021). One can potentially tolerate a larger fraction of corruption (up to $\alpha < 1/2$) by using the robust estimators in (Yin et al., 2018). However, this would likely come at a price: the authors in (Yin et al., 2018) impose additional statistical assumptions on the partial gradients; we do not make such assumptions.*

Remark 5. *(Comments on M) In our results, we need the number of agents M to scale with* $\log(dT)$. *We note that similar conditions show up in the context of adversarially-robust distributed statistical learning; see, for instance, (Yin et al., 2018) and (Pillutla et al., 2022). In fact, the covering argument in (Yin et al., 2018) requires M to scale linearly with the model dimension d. By avoiding such an argument in our analysis, we can get by with a far milder logarithmic dependence on d. As an example, for $d = 100$, and number of iterations $T = 2^{10}$ (which should suffice for all practical purposes),* $\log(dT) \approx 12$. *This is a very reasonable requirement for large-scale computing systems where the number of devices is of the order of thousands. Furthermore, with $T = 2^{10}$, our guarantees in Theorems 7 and 8 hold with probability* $1 - 1/T \approx 1$.

3.5. Proof Sketch of Theorem 7

In this section, we provide a sketch of the proof of Theorem 7. Due to space constraints, detailed proofs of our main results (including that of Theorem 8) are omitted here, but can be found in (Adibi et al., 2022a). Essentially, our proofs comprise of a perturbation analysis of the extra-gradient algorithm, where the perturbations arise due to adversarial corruption. As the starting point of such an analysis, we establish some simple relations in the following lemma.

Lemma 16. *For the* `RDEG` *algorithm, the following inequalities hold for all $t \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:*

$$\begin{aligned}
2\eta\langle \tilde{g}_x(x_t, y_t), \hat{x}_t - x\rangle &\leq \|x - x_t\|^2 - \|x - \hat{x}_t\|^2 - \|\hat{x}_t - x_t\|^2 \\
-2\eta\langle \tilde{g}_y(x_t, y_t), \hat{y}_t - y\rangle &\leq \|y - y_t\|^2 - \|y - \hat{y}_t\|^2 - \|\hat{y}_t - y_t\|^2 \\
2\eta\langle \tilde{g}_x(\hat{x}_t, \hat{y}_t), x_{t+1} - x\rangle &\leq \|x - x_t\|^2 - \|x - x_{t+1}\|^2 - \|x_{t+1} - x_t\|^2 \\
-2\eta\langle \tilde{g}_y(\hat{x}_t, \hat{y}_t), y_{t+1} - y\rangle &\leq \|y - y_t\|^2 - \|y - y_{t+1}\|^2 - \|y_{t+1} - y_t\|^2.
\end{aligned} \tag{3.12}$$

Using the previous result, our next goal is to track the progress made by the mid-point vector $(\hat{x}_t, \hat{y}_t)$ in each iteration, as a function of the errors introduced by adversarial corruption. To that end, for each $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$, we define the following errors vectors:

$$e_x(\bar{x}, \bar{y}) \triangleq \tilde{g}_x(\bar{x}, \bar{y}) - \nabla_x f(\bar{x}, \bar{y}); e_y(\bar{x}, \bar{y}) \triangleq \tilde{g}_y(\bar{x}, \bar{y}) - \nabla_y f(\bar{x}, \bar{y}). \tag{3.13}$$

We have the following key lemma.

Lemma 17. *Suppose Assumption 11 holds and $\eta \leq 1/(2L)$. For the* `RDEG` *algorithm, the following then holds for all $t \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:*

$$\begin{aligned}
&\eta\langle \nabla_x f(\hat{x}_t, \hat{y}_t), \hat{x}_t - x\rangle - \eta\langle \nabla_y f(\hat{x}_t, \hat{y}_t), \hat{y}_t - y\rangle \\
&\leq \frac{1}{2}\left(\|x - x_t\|^2 - \|x - x_{t+1}\|^2 + \|y - y_t\|^2 - \|y - y_{t+1}\|^2\right) \\
&+ \eta D\left(\|e_x(x_t, y_t)\| + \|e_x(\hat{x}_t, \hat{y}_t)\| + \|e_y(x_t, y_t)\| + \|e_y(\hat{x}_t, \hat{y}_t)\|\right).
\end{aligned} \tag{3.14}$$

In our next result, we establish high-probability bounds on the error vectors by leveraging Theorem 6.

Lemma 18. *Consider the event $\mathcal{H}_t$ defined as follows:*

$$\mathcal{H}_t \triangleq \{\max\{\|e_x(x_t, y_t)\|, \|e_x(\hat{x}_t, \hat{y}_t)\|, \|e_y(x_t, y_t)\|, \|e_y(\hat{x}_t, \hat{y}_t)\|\} \leq \Delta\},$$

where

$$\Delta = c\sigma\left(\sqrt{\alpha} + \sqrt{\frac{\log(4dT^2)}{M}}\right). \tag{3.15}$$

For the `RDEG` *algorithm, we have:*

$$\mathbb{P}\left(\mathcal{H}_t\right) \geq 1 - \frac{1}{T^2}, \text{ for each } t \in [T]. \tag{3.16}$$

Proof. We begin by defining certain "good" events:

$$\begin{aligned}
&\mathcal{G}_{x,t} \triangleq \{\|e_x(x_t, y_t)\| \leq \Delta\}, \mathcal{G}_{y,t} \triangleq \{\|e_y(x_t, y_t)\| \leq \Delta\}, \\
&\bar{\mathcal{G}}_{x,t} \triangleq \{\|e_x(\hat{x}_t, \hat{y}_t)\| \leq \Delta\}, \bar{\mathcal{G}}_{y,t} \triangleq \{\|e_y(\hat{x}_t, \hat{y}_t)\| \leq \Delta\}.
\end{aligned}$$

To analyze the probability of occurrence of the above events, we need to next define an appropriate filtration. Accordingly, let $\mathcal{F}_t$ denote the sigma field generated by $\{x_k, y_k\}_{k\in[t]}$ and $\{\hat{x}_k, \hat{y}_k\}_{k\in[t-1]}$;

and $\bar{\mathcal{F}}_t$ denote the sigma field generated by $\{x_k, y_k\}_{k\in[t]}$ and $\{\hat{x}_k, \hat{y}_k\}_{k\in[t]}$. From definition, we have

$$\mathcal{F}_1 \subset \bar{\mathcal{F}}_1 \subset \mathcal{F}_2 \subset \bar{\mathcal{F}}_2 \subset \cdots \subset \mathcal{F}_T \subset \bar{\mathcal{F}}_T.$$

Clearly, (x_t, y_t) is $\mathcal{F}_t$-measurable. Thus, conditioned on $\mathcal{F}_t$, for each coordinate $j \in [n]$, the data set $\{[g_x(x_t, y_t; \xi_{1,t}^{(i)})]_j : i \in [M]\}$ has the following properties: (i) at most αM of the samples are corrupted; and (ii) the uncorrupted samples are i.i.d. scalar random variables with mean $[\nabla_x f(x_t, y_t)]_j$ and variance bounded above by $\sigma_x^2(j)$. Invoking Theorem 6 for the trimmed mean estimator in Algorithm 7, we conclude that conditioned on $\mathcal{F}_t$, with probability at least $1 - 1/(4dT^2)$,

$$|[\tilde{g}_x(x_t, y_t)]_j - [\nabla_x f(x_t, y_t)]_j| \leq c\sigma_x(j)\left(\sqrt{\alpha} + \sqrt{\frac{\log(4dT^2)}{M}}\right).$$

Now union-bounding over each of the n coordinates, we have that conditioned on $\mathcal{F}_t$, with probability at least $1 - \frac{n}{4dT^2} \geq 1 - \frac{1}{4T^2}$,

$$\|\tilde{g}_x(x_t, y_t) - \nabla_x f(x_t, y_t)\| \leq \Delta.$$

Here, we used the fact that $d = \max\{n, m\} \geq n$, and $\sqrt{\sum_{j\in[n]} \sigma_x^2(j)} = \sigma_x \leq \sigma$. We have thus shown that $\mathbb{P}(\mathcal{G}_{x,t}|\mathcal{F}_t) \geq 1 - 1/(4T^2)$. Using an identical argument, we can establish an analogous result for the event $\mathcal{G}_{y,t}$. An union bound thus yields $\mathbb{P}(\mathcal{G}_t|\mathcal{F}_t) \geq 1 - 1/(2T^2)$, where $\mathcal{G}_t = \mathcal{G}_{x,t} \cap \mathcal{G}_{y,t}$. Noting that $(\hat{x}_t, \hat{y}_t)$ is $\bar{\mathcal{F}}_t$-measurable, we can similarly show that $\mathbb{P}\left(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t\right) \geq 1 - 1/(2T^2)$, where $\bar{\mathcal{G}}_t = \bar{\mathcal{G}}_{x,t} \cap \bar{\mathcal{G}}_{y,t}$. Our next task is to analyze the probability of occurrence of the event $\mathcal{H}_t = \mathcal{G}_t \cap \bar{\mathcal{G}}_t$ by exploiting the nested sigma-field structure: $\mathcal{F}_t \subset \bar{\mathcal{F}}_t$. To that end, observe:

$$\begin{aligned}
\mathbb{P}(\bar{\mathcal{G}}_t|\mathcal{F}_t) &= \mathbb{E}[1_{\bar{\mathcal{G}}_t}|\mathcal{F}_t] \\
&\overset{(a)}{=} \mathbb{E}[\mathbb{E}[1_{\bar{\mathcal{G}}_t}|\bar{\mathcal{F}}_t]|\mathcal{F}_t] \\
&= \mathbb{E}[\mathbb{P}(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t)|\mathcal{F}_t] \\
&\overset{(b)}{\geq} 1 - \frac{1}{2T^2}.
\end{aligned} \tag{3.17}$$

Here, we used $1_{\mathcal{A}}$ to represent the indicator random variable for an event $\mathcal{A}$. For (a), we used the fact that given a random variable X and two sigma-fields $\mathcal{B}_1$ and $\mathcal{B}_2$ with $\mathcal{B}_1 \subset \mathcal{B}_2$, it holds that

$\mathbb{E}[\mathbb{E}[X|\mathcal{B}_2]|\mathcal{B}_1] = \mathbb{E}[X|\mathcal{B}_1]$, i.e., the smaller sigma-field "wins" (Durrett, 2019, Theorem 5.1.6). For (b), we used the previously established fact that $\mathbb{P}\left(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t\right) \geq 1 - 1/(2T^2)$. Using (3.17) and an union bound, we conclude that $\mathbb{P}(\mathcal{H}_t|\mathcal{F}_t) = \mathbb{P}(\mathcal{G}_t \cap \bar{\mathcal{G}}_t|\mathcal{F}_t) \geq 1 - 1/T^2$. To complete the proof, we note that

$$\mathbb{P}(\mathcal{H}_t) = \mathbb{E}[1_{\mathcal{H}_t}] = \mathbb{E}[\mathbb{E}[1_{\mathcal{H}_t}|\mathcal{F}_t]] = \mathbb{E}[\mathbb{P}(\mathcal{H}_t|\mathcal{F}_t)] \geq 1 - \frac{1}{T^2}.$$

□

The proof of Theorem 7 is a fairly simple consequence of Lemma's 43, 44, and 18. In particular, it follows by conditioning on the clean event $\mathcal{H} = \bigcap_{t\in[T]} \mathcal{H}_t$, where $\mathcal{H}_t$ is as defined in Lemma 18, and exploiting the convex-concave property of $f(x, y)$ in a standard way. For details, see (Adibi et al., 2022a).

3.6. Simulations

In this section, we study a specific instance of problem (3.1), namely, a bilinear game of the following form:

$$\min_{\|x\|\leq\rho} \max_{\|y\|\leq\rho} f(x, y) \triangleq \mathbb{E}[x^T A y + 2(b + \zeta)^T x - 2(c + \zeta)^T y].$$

Here, $x, y, b, c \in \mathbb{R}^{10}$, $A \in \mathbb{R}^{10\times10}$, and $\rho = 100$. The parameters A, b, c are fixed, and $\zeta \sim N(0, \sigma^2 I)$. As our measure of performance, we consider the instantaneous primal-dual gap $\phi_t = \max_{y\in\mathcal{Y}} f(\bar{x}_t, y) - \min_{x\in\mathcal{X}} f(x, \bar{y}_t)$. We simulate two algorithms: the vanilla extra-gradient algorithm that does not account for adversaries, and the proposed `RDEG` algorithm. In Fig. 3.2(a), we plot the performance of these algorithms with $\alpha = 0.06$, $M = 100$, and $\sigma^2 = 10$. We observe that even a small number of Byzantine workers can cause the extra-gradient algorithm to diverge from the saddle point. In Fig. 3.2(b), with $M = 100$ and $\sigma^2 = 10$, we explore the impact of varying the corruption fraction α. Complying with Theorem 7, the error floor of `RDEG` increases as a function of α. Next, in Fig. 3.2(c), to demonstrate the benefit of collaboration, we fix $\alpha = 0.06$ and $\sigma^2 = 10$, and plot the performance of `RDEG` as a function of the number of agents M. As expected, by increasing M, `RDEG` converges to a smaller ball around the saddle point, highlighting the benefit of collaboration in reducing the variance of the noise model. Finally, in Fig. 3.2(d), we fix $M = 100$ and $\alpha = 0.06$, and change the

variance of the noise σ^2. We observe that increasing σ^2 leads to a higher error-floor. Importantly, all of the above plots verify the bound in Theorem 7.

3.7. Conclusion

We studied the problem of distributed min-max learning under adversarial agents for the first time. By exploiting recent ideas from robust statistics, we developed a novel robust distributed extra-gradient algorithm. For both smooth convex-concave and smooth strongly convex-strongly concave functions, we showed that with high probability, our proposed approach guarantees convergence to approximate saddle points at near-optimal statistical rates.

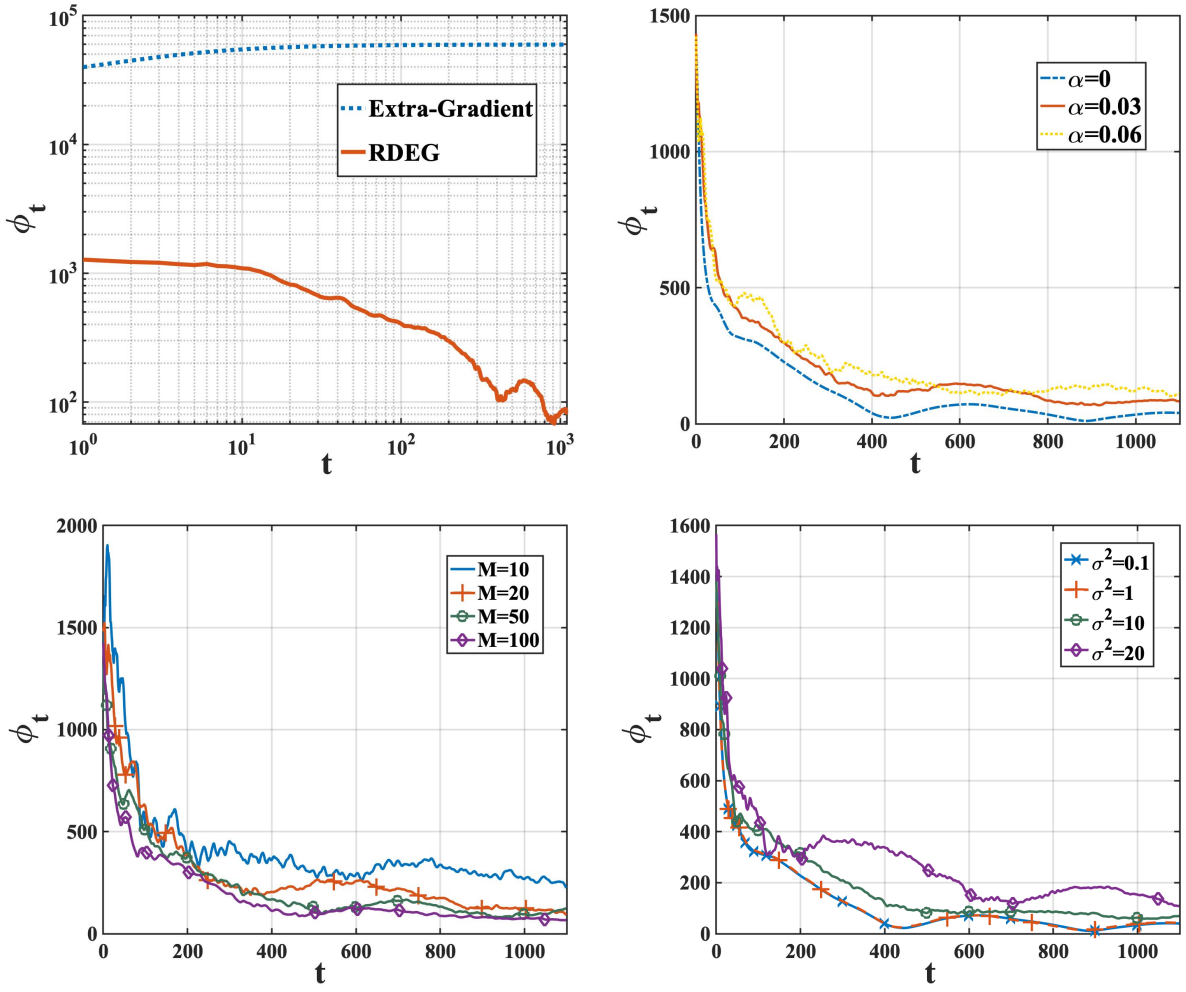

Figure 3.2: **Top Left (a).** Comparison between vanilla extra-gradient and RDEG. **Top Right (b).** Performance of RDEG vs. level of corruption fraction. **Bottom Left (c).** Performance of RDEG vs. number of agents. **Bottom Right (d).** Performance of RDEG vs. level of noise variance.

CHAPTER 4

Stochastic Approximation under Delays

4.1. Summary

We study non-asymptotic convergence rates of general stochastic approximation (SA) under Markovian sampling with delayed updates for the first time. In this setup , iterative updates of SA are based on delayed versions of the SA operator evaluated at stale iterates and samples from the past. We are interested in understanding the finite-time performance of this updating scheme with a focus on characterizing the interplay between the properties of the underlying Markov process and the delay sequence. Our first contribution is to show that, under standard assumptions, the delayed SA update rule guarantees exponentially fast convergence to a ball around the desired fixed point of the operator. In a constant delay setting, we prove that a carefully weighted average of iterates achieves the optimal rate of convergence in which the exponent of convergence gets scaled down by a factor of $\max\{\tau, \tau_{mix}\}$, where τ represents the delay, and τ_{mix} denotes the mixing time of the Markov process. In the case of time-varying delays, we show that the exponent of convergence for the last iterate gets scaled down by a factor of $(\tau_{max} + \tau_{mix})^2$, where τ_{max} represents the maximum delay. To improve this bound, we propose a delay-adaptive SA scheme where updates are made only when the staleness in iterates falls below a carefully chosen threshold. With this simple modification, we prove that the new algorithm guarantees exponentially fast convergence with a rate that now gets scaled down by $\tau_{mix}\tau_{avg}$ for the last iterate, where τ_{avg} is the average of the delays over all iterations. Remarkably, not only does our update rule significantly improve the convergence rate relative to the vanilla scheme, but it also does so without the choice of the step size requiring any knowledge about the delay sequence. Overall, our theoretical findings apply to various algorithms where the finite-time effects of delays were previously unknown, such as TD learning and Q-learning with function approximation, and stochastic gradient descent under Markovian sampling.

4.2. Introduction

Stochastic Approximation (SA) is an iterative technique used to solve root-finding problems in the presence of noisy information. This method finds its application in various fields such as machine learning and reinforcement learning. In this section, we will provide a brief summary of previous works and then highlight our contributions.

4.2.1. Previous Works on Stochastic Approximation

The Stochastic Approximation (SA) framework, was originally introduced in 1951 (Robbins and Monro, 1951) and it has been extensively studied in the literature, with a focus on understanding its convergence behavior and applications in various domains. Several notable works have contributed to the finite-time analysis and other properties of SA algorithms.

(Bhandari et al., 2018b) conducted the first finite analysis of SA in the context of TD learning with a bounded domain. Their work specifically addressed a special case of linear SA with a bounded domain, providing insights into the convergence behavior of the algorithm. Building on this, (Srikant and Ying, 2019) made significant strides by proving a finite-time rate of convergence for linear SA, expanding the understanding of the algorithm's performance in learning tasks.

The finite-time analysis of SA was further extended to a broader context (Chen et al., 2023b) who provided the first finite-time analysis for general SA algorithms. Their study encompassed various applications, including asynchronous Q-learning, off-policy actor-critic, average reward TD learning, and Q-learning. This seminal work laid the groundwork for understanding the convergence guarantees of SA in diverse reinforcement learning settings.

(Zeng et al., 2022) explored decentralized stochastic approximation, a data-driven approach for finding the root of an operator under noisy measurements. Their work addressed a network of agents working cooperatively to find the fixed point of the aggregate operator over a decentralized communication graph. Notably, they provided a finite-time analysis of this decentralized approach, considering data observed at each agent to be sampled from Markov processes. This novel analysis accounted for the bias and potential unboundedness of iterates, providing valuable insights into the

convergence rate in decentralized multi-agent and multi-task learning scenarios.

Further developments in finite-time analysis were achieved by (Chen et al., 2021), who investigated contractive stochastic approximation using smooth convex envelopes. They derived finite-sample error bounds for the algorithm using different step sizes and introduced a smooth Lyapunov function based on the generalized Moreau envelope. This construction resulted in a convergence bound with only a logarithmic dependence on the state-space size, facilitating its application in reinforcement learning settings, including the V-trace algorithm for off-policy TD-learning.

Collectively, the works mentioned above significantly contribute to the understanding of finite-time convergence rates and other essential properties of SA algorithms.

Contributions:

In this chapter, we make significant contributions to the field of stochastic approximation (SA) by studying the non-asymptotic convergence rates of SA under Markovian sampling with delayed updates. Our investigation focuses on understanding the finite-time performance of this updating scheme, considering the interplay between the properties of the underlying Markov process and the delay sequence. These are our contributions:

1. Exploration of Finite-Time Analysis of Delayed Stochastic Approximation: The first major contribution of this work is the exploration of finite-time analysis for delayed Stochastic Approximation (SA). We delve into the effects of delayed updates on the convergence behavior of SA algorithms, which is an area that has not been studied before.

2. Proof of Optimal Bound in the Constant Delay Case: In the setting of constant delay, we present a rigorous proof that the delayed SA update rule guarantees exponentially fast convergence to a ball around the desired fixed point of the operator. Notably, we establish that a carefully weighted average of iterates achieves the optimal rate of convergence, with the exponent of convergence scaled down by a factor of $\max\{\tau, \tau_{mix}\}$, where τ represents the delay, and τ_{mix} denotes the mixing time of the Markov process. This result sheds light on the trade-off

between the delay and mixing time, providing valuable insights into the impact of the delay on convergence performance.

3. Investigation of Convergence Rate with Time-Varying Delay: We extend our analysis to consider time-varying delays, which are common in practical scenarios. In this context, we demonstrate that the exponent of convergence for the last iterate gets scaled down by a factor of $(\tau_{max} + \tau_{mix})^2$, where τ_{max} represents the maximum delay. This finding contributes to a deeper understanding of the convergence behavior under more realistic conditions.

4. Introduction of a Delay-Adaptive Algorithm: To further improve the convergence rate, we propose a novel delay-adaptive SA scheme. In this algorithm, updates are made only when the staleness in iterates falls below a carefully chosen threshold. Remarkably, this simple modification results in a significant improvement in the convergence rate relative to the vanilla scheme. The new algorithm guarantees exponentially fast convergence, and its rate is now scaled down by $\tau_{mix}\tau_{avg}$ for the last iterate, where τ_{avg} represents the average of the delays over all iterations. Importantly, our delay-adaptive scheme achieves this performance enhancement without requiring any knowledge about the delay sequence, making it highly practical for real-world implementations.

5. General Applicability of Theoretical Findings: Finally, our theoretical contributions have broad applicability, extending to various algorithms where the finite-time effects of delays were previously unknown. Notable examples include TD learning and Q-learning with function approximation, as well as stochastic gradient descent under Markovian sampling. By shedding light on the impact of delays in these contexts, our work opens new avenues for research and optimization in these areas.

In summary, our paper provides a comprehensive analysis of the finite-time convergence behavior of delayed Stochastic Approximation. The insights and results presented in this work pave the way for the development of more efficient and adaptive algorithms that can handle delays effectively in a wide range of applications.

4.3. Related Work

In this section, we review the existing literature on delays in optimization, bandits, and reinforcement learning (RL). We categorize the papers into three groups: delays in optimization, delays in bandits and RL, and specific models related to our work on delays in RL. Additionally, we discuss empirical results on asynchronous RL.

4.3.1. Delays in Optimization

The study of delays and asynchrony in optimization has been a topic of interest since the seminal work (Bertsekas and Tsitsiklis, 1989), which investigates convergence rates of asynchronous iterative algorithms in parallel or distributed computing systems. Subsequently, many researchers have explored the effects of delay and asynchrony on various learning and optimization methods. We summarize some of the significant works in this area:

(Agarwal and Duchi, 2011) focuses on distributed delayed stochastic optimization, specifically gradient-based optimization algorithms that rely on delayed stochastic gradient information. They analyze the convergence of such algorithms and propose procedures to overcome communication bottlenecks and synchronization requirements. Their work demonstrates that delays are asymptotically negligible, achieving order-optimal convergence results for smooth stochastic problems in distributed optimization settings.

(Stich and Karimireddy, 2020) introduced the error-feedback framework, which examines stochastic gradient descent (SGD) with delayed updates on smooth quasi-convex and non-convex functions. They derive non-asymptotic convergence rates and show that the delay only linearly slows down the higher-order deterministic term, while the stochastic term remains unaffected. This result illustrates the robustness of SGD to delayed stochastic gradient updates, improving upon previous rates for different forms of delayed gradients. Notably, this work provides the best-known rate for SGD with i.i.d. noise. It is worth mentioning that most existing literature has focused on bounds depending only on the maximum delay. However, recent works (Cohen et al., 2021) and (Koloskova et al., 2022b) have explored convergence rates that depend on the average delay sequence. Nevertheless,

there is still a gap in the literature regarding the finite-time rate of delayed stochastic approximation under Markovian noise.

The aforementioned studies on delays in optimization contribute to understanding the impact of delays and asynchrony in various optimization algorithms. They provide insights into the convergence properties and shed light on the robustness of these methods to different forms of delay. However, there is a need for further investigation into the specific case of delayed stochastic approximation under Markovian noise.

4.3.2. Delays in Bandits

There has been a lot of research on the impact of delays in bandits. Some of the key papers in this area include:

The nonstochastic multiarmed bandits with unrestricted delays were studied in (Thune et al., 2019). The authors prove that the "delayed" Exp3 algorithm achieves the regret bound for variable but bounded delays. They also introduce a new algorithm that handles delays without prior knowledge of the total delay, achieving the same regret bound. The paper provides insights into the regret bounds for bandit problems with delays.

The challenges of stochastic linear bandits with delayed feedback, where the feedback is randomly delayed and delays are only partially observable, were addressed in (Vernade et al., 2020). The authors propose computationally efficient algorithms, OTFLinUCB and OTFLinTS, capable of integrating new information as it becomes available and handling permanently censored feedback. The authors prove optimal regret bounds for the proposed algorithms and validate their findings through experiments on simulated and real data.

Another paper investigates a variant of the stochastic K-armed bandit problem called "bandits with delayed, aggregated anonymous feedback" (Pike-Burke et al., 2018). In this setting, the player observes only the sum of a number of previously generated rewards that arrive in each round, and the information of which arm led to a particular reward is lost. The authors provide an algorithm that achieves the same worst-case regret as in the non-anonymous problem when the delays are

bounded.

These papers demonstrate that it is possible to design algorithms that can achieve good performance in the presence of delays in bandits. However, the problem of delays is still a challenging one, and there is still much research to be done in this area. It's important to note that the previous papers' findings don't directly provide a rate for delayed stochastic approximation.

4.3.3. Delays in RL

Until recently, the field of reinforcement learning had not thoroughly explored the impact of delay. In this summary, we will highlight some key research works in this area.

(Bouteiller et al., 2020) conducted a study on reinforcement learning with random delays, specifically focusing on environments with delays in actions and observations. They introduced the Delay-Correcting Actor-Critic (DCAC) algorithm, which incorporates off-policy multi-step value estimation to accommodate delays. Through theoretical analysis and practical experiments using a delay-augmented version of the MuJoCo continuous control benchmark, the authors demonstrated that DCAC outperforms other algorithms in delayed environments.

(Mnih et al., 2016) introduced asynchronous methods for deep reinforcement learning. They presented a lightweight framework that utilizes asynchronous gradient descent to optimize deep neural network controllers. The authors showed that parallel actor-learners have a stabilizing effect on training and achieve superior performance compared to state-of-the-art methods in domains such as Atari games and continuous motor control problems.

(Chen et al., 2023a) studied the problem of policy learning in environments with delayed observation. They showed that it is possible to learn a near-optimal policy in this setting, even though the agent does not have access to the most recent state of the system. They established near-optimal regret bounds for this case, which means that the agent's performance is close to that of an agent with full observability.

These selected papers lay the groundwork for understanding the challenges posed by delays in

optimization, bandits, and reinforcement learning. They provide valuable theoretical insights, novel algorithms, and empirical evidence that serve as motivation for our own work on delays in Stochastic approximation.

4.4. Stochastic Approximation with Delays: Problem Formulation

The objective of general SA is to solve a root finding problem of the following form:

$$\text{Find } \boldsymbol{\theta}^* \in \mathbb{R}^m \text{ such that } \bar{\mathbf{g}}(\boldsymbol{\theta}^*) = 0, \tag{4.1}$$

where, for a given approximation parameter $\boldsymbol{\theta} \in \mathbb{R}^m$, the deterministic function $\bar{\mathbf{g}}(\boldsymbol{\theta})$ is the expectation of a noisy operator $\mathbf{g}(\boldsymbol{\theta}, o_t)$, and $\{o_t\}$ denotes a stochastic *observation process*. In this chapter, we consider SA under Markovian sampling, i.e., the observations $\{o_t\}$ are temporally correlated and form a Markov chain. We define

$$\bar{\mathbf{g}}(\boldsymbol{\theta}) \triangleq \mathbb{E}_{o_t \sim \pi}[\mathbf{g}(\boldsymbol{\theta}, o_t)], \tag{4.2}$$

where π is the *stationary distribution* of the Markov chain $\{o_t\}$.

SA consists in finding an approximate solution to (4.1) while having access only to *noisy* instances $\mathbf{g}(\boldsymbol{\theta}, o_t)$ of $\bar{\mathbf{g}}(\boldsymbol{\theta})$. The typical iterative SA update rule with a constant step size α is as follows (Srikant and Ying, 2019; Chen et al., 2022),

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t). \tag{4.3}$$

The asymptotic convergence of SA under Markov randomness method has been thoroughly investigated in prior work (Tsitsiklis and Vanroy, 1997). Recently, there is an increased interest in *finite-time convergence guarantees* for SA. Finite-time analysis of SA is important because it provides theoretical guarantees about the algorithm's convergence rate. In particular, it allows us to determine how quickly the algorithm will converge to the optimal solution in a finite amount of time, which is crucial in many real-world applications where learning time is limited.

Finite-time analysis provides insights into how the algorithm behaves over a fixed number of

iterations or time steps. By analyzing the convergence rate and error bounds, we can make informed decisions about the choice of learning rate, step size, and other hyperparameters that can affect the performance of the algorithm. This analysis can help us optimize the learning process and improve the performance of the agent in the given task.

Moreover, finite-time analysis of SA helps better understand the algorithm and identify its limitations and strengths. It also provides a theoretical foundation for the algorithm, which can lead to the development of more efficient and robust algorithms for solving RL problems.

Exemplar Applications. In this part, we provide some examples of stochastic approximation.

TD learning. TD learning with linear function approximation is a stochastic approximation algorithm that can be used to learn the value function of a Markov decision process (MDP). The algorithm works by iteratively updating a linear function approximator of the value function using the following update rule:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha(r_t + \gamma \hat{V}(s_{t+1}; \boldsymbol{\theta}_t) - \hat{V}(s_t; \boldsymbol{\theta}_t))\boldsymbol{\phi}(s_t) \\ \hat{V}(s; \boldsymbol{\theta}) &= \boldsymbol{\theta}^\top \boldsymbol{\phi}(s) \end{aligned} \tag{4.4}$$

where $\boldsymbol{\theta}_t$ is the parameter vector of the linear function approximator at time t, α is the learning rate, r_t is the reward received at time t, s' is the next state, $\hat{V}(s; \boldsymbol{\theta}_t)$ is the estimated value of the state s at time t, and $\boldsymbol{\phi}(s)$ is a feature vector that maps the state to a vector in $\mathbb{R}^m$.

The update rule for TD learning with linear function approximation can be viewed as a stochastic approximation of the Bellman equation for the value function. The Bellman equation for the value function is given by the following equation:

$$V^*(s) = \mathbb{E}[r + \gamma V^*(s')]$$

where $V^*(s)$ is the optimal value of state s, r is the reward received, s' is the next state, and $\mathbb{E}$ is

the expected value operator. The update rule for TD learning with linear function approximation can be seen as a stochastic approximation because it is an approximation of the Bellman equation that is driven by noise. The noise in the update rule is introduced by the random reward r and the random next state s_{t+1}. More specifically, we can look at the update in (4.4) as a stochastic approximation with

$$\mathbf{g}(\boldsymbol{\theta}_t, o) = (r_t + \gamma \hat{V}(s_{t+1}; \boldsymbol{\theta}_t) - \hat{V}(s_t; \boldsymbol{\theta}_t))\boldsymbol{\phi}(s), \tag{4.5}$$

where o is randomness caused by pair (r_t, s_t, s_{t+1}).

Q-learning. Q-learning is a reinforcement learning algorithm that can be used to learn an optimal policy for a Markov decision process (MDP). In Q-learning, the goal is to learn a Q-function, which is a function that maps state-action pairs to their corresponding expected rewards.

Q-learning with linear function approximation can be viewed as a nonlinear stochastic approximation algorithm (Chen et al., 2022). This is because Q-learning uses a linear function approximator to approximate the Q-function, and the update rule for the Q-function is a stochastic approximation of the Bellman equation.

SGD with Markovian noise. Stochastic gradient descent (SGD) is a method for minimizing a noisy or stochastic function $f(\boldsymbol{\theta}, o)$. It works by iteratively updating a parameter vector in the direction of the negative gradient of the function. SGD can be viewed as a stochastic approximation with $\mathbf{g}(\boldsymbol{\theta}, o) = -\nabla f(\boldsymbol{\theta}, o)$. All our results are applicable to the SGD framework with Markovian noise, which was studied for example in (Doan, 2022; Even, 2023).

Several recent works have investigated linear (Srikant and Ying, 2019; Bhandari et al., 2018a) and non-linear (Chen et al., 2022) SA, and provided finite-time convergence bounds under Markovian sampling. Notably, Finite-time convergence analysis for SA under Markovian sampling are significantly more challenging relative to i.i.d. sampling. Indeed, temporal correlation between samples of $\{o_t\}$, which is also inherited by the iterates $\{\boldsymbol{\theta}_t\}$, prevents the use of some techniques commonly used for the finite-time rates study of SA under i.i.d. sampling, triggering the need for a more elaborate analysis.

In many real-world applications, the SA operator $\mathbf{g}(\cdot)$ is only available when computed with delayed iterates and/or observations. The main objective of this chapter is to provide a unified framework to analyse the finite-time convergence of SA under delay. We proceed by formally introducing the setting.

SA with delays. We consider the following stochastic recursion with delayed updates:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \quad \tau_t \leq t, \tag{4.6}$$

where α is a constant step size and τ_t is the delay with which the operator $\mathbf{g}(\cdot)$ is available to be used at iteration t. This specific update rule is motivated by many scenarios of practical interest. For instance, in distributed machine learning and reinforcement learning, it is often the case that the agents' updates are performed in an asynchronous manner, leading naturally to update rules of the form (4.6).

Update rules of the form (4.6) have been recently studied in the context of SA but with i.i.d. observations (see e.g. (Koloskova et al., 2022b; Nguyen et al., 2022) for SGD updates with delays). However, to the best of our knowledge, nothing is known about the finite-time convergence behaviour of such update rules under Markovian observations. Compared with i.i.d. setting, the Markovian setting introduces major technical challenges, including deal with the joint effect of (i) the use of a delayed operator $\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$ and (ii) sequences of correlated observation samples $\{o_t\}$. The interplay of update rules based on delayed SA operator instances and the presence of time correlation in the noise process requires a notably careful analysis, one which we provide as a main contribution of this chapter. The key features and challenges of the analysis are provided with more details in sections 4.6 and 4.7.

We proceed with describing a few assumptions needed for our analysis. First, we make the following natural assumption on the underlying Markov chain $\{o_t\}$ (Bhandari et al., 2018a; Srikant and Ying, 2019; Chen et al., 2022).

Assumption 4. *The Markov chain $\{o_t\}$ is aperiodic and irreducible.*

Next, we state two further assumptions that are common in the analysis of SA algorithms.

Assumption 5. *Problem* (4.1) *admits a solution $\boldsymbol{\theta}^*$, and $\exists\, \mu > 0$ such that for all $\boldsymbol{\theta} \in \mathbb{R}^m$, we have*

$$\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \bar{\mathbf{g}}(\boldsymbol{\theta}) - \bar{\mathbf{g}}(\boldsymbol{\theta}^*) \rangle \leq -\mu \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2. \tag{4.7}$$

Assumption 6. *For every $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ and $o \in \{o_t\}$, we have*

$$\|\mathbf{g}(\boldsymbol{\theta}_1, o) - \mathbf{g}(\boldsymbol{\theta}_2, o)\| \leq L \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|. \tag{4.8}$$

Assumption 7. *For any $\boldsymbol{\theta} \in \mathbb{R}^m$ and $o \in \{o_t\}$, we have*

$$\|\mathbf{g}(\boldsymbol{\theta}, o)\| \leq L \|\boldsymbol{\theta}\| + 2\sigma. \tag{4.9}$$

Finally, we introduce an assumption on the time-varying delay sequence $\{\tau_t\}$.

Assumption 8. *There exists an integer $\tau_{max} \geq 0$ such that $\tau_t \leq \tau_{max},\ \forall t \geq 0$.*

Assumption 5 is a strong monotone property of the map $-\bar{\mathbf{g}}(\boldsymbol{\theta})$ that guarantees that the iterates generated by a "mean-path" version of Eq. (4.1), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \bar{\mathbf{g}}(\boldsymbol{\theta}_t)$, converge exponentially fast to $\boldsymbol{\theta}^*$. Assumption 6 states that $\mathbf{g}(\boldsymbol{\theta}, o_t)$ is globally uniformly (w.r.t. o_t) Lipschitz in the parameter $\boldsymbol{\theta}$. Without loss of generality, we have fixed the Lipschitz constant to be $L = 2$, which is the Lipschitz constant value in the case of TD learning with linear function approximation.

Corollary 3. *Assumption 5 holds for TD learning (Lemma 1 and Lemma 3 in (Bhandari et al., 2018a)), Q-learning (Chen et al., 2022), and SGD for strongly convex functions. Similarly, Assumption 6 holds for TD learning (Bhandari et al., 2018a), and for Q-learning and SGD analysis, it holds up to some constant L (Chen et al., 2022; Doan, 2022). Our proof technique generalizes to this*

Table 4.1: Summary of results.

Algorithm	Variance Bound	Bias Bound
Constant Delay (4.10)	$O(\sigma^2)$	$O\left(T\exp\left(\frac{-\mu^2 T}{\max\{\tau,\tau_{mix}\}}\right)\right)$
Time-Varying Delays (4.30)	$O(\sigma^2)$	$O\left(\exp\left(\frac{-\mu^2 T}{(\tau_{mix}+\tau_{max})^2}\right)\right)$
Time-Varying Delays Delay-Adaptive update (4.189)	$O(\sigma^2)$	$O\left(\exp\left(\frac{-\mu^2 T}{\tau_{mix}\cdot\tau_{avg}}\right)\right)$

setting easily. Furthermore, Assumption 7 holds for TD learning (Bhandari et al., 2018a), and for Q-learning, it holds up to some constant (Chen et al., 2022).

We now introduce the following notion of *mixing time* τ_α, that plays a crucial role in our analysis, as in the analysis of all existing finite-time convergence studies on SA under Markovian sampling.

Definition 1. *Let τ_α be such that*

$$\|\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}, o_t)|o_0\right] - \bar{\mathbf{g}}(\boldsymbol{\theta})\| \leq \alpha\left(\|\boldsymbol{\theta}\| + 1\right), \forall t \geq \tau_\alpha, \forall \boldsymbol{\theta} \in \mathbb{R}^m, \forall o_0.$$

Assumption 4 implies that the Markov chain $\{o_t\}$ mixes at a geometric rate. This, in turn, implies the existence of some $K \geq 1$ such that τ_α in Definition 1 satisfies $\tau_\alpha \leq K\log(\frac{1}{\alpha})$. In words, this means that for a fixed $\boldsymbol{\theta}$, if we want the noisy operator $\mathbf{g}(\boldsymbol{\theta}, o_t)$ to be α-close (relative to $\boldsymbol{\theta}$) to the expected operator $\bar{\mathbf{g}}(\boldsymbol{\theta})$, then the amount of time we need to wait for this to happen scales logarithmically in the precision α.

4.5. Stochastic Approximation with Constant Delays

In this section, we present the first finite-time convergence analysis of SA with constant delay under Markovian sampling. With respect to the SA with delayed updates introduced in (4.6), we fix $\tau_t = \tau$, with τ the constant delay, and define the following SA update rule accordingly:

$$\textbf{\textit{SA with Constant Delay:}} \quad \boldsymbol{\theta}_{t+1} = \begin{cases} \boldsymbol{\theta}_0 & \text{if } \ 0 \leq t < \tau \\ \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau}) & \text{if } \ t \geq \tau \end{cases} \tag{4.10}$$

The following theorem provides a finite-time convergence bound for the update rule in (4.10) and it is the first main contribution of this chapter.

Theorem 9. *For $T \geq 0$, $\alpha \leq \frac{\mu}{C_1 L^2 \bar{\tau}}$, $w_t := (1 - 0.5\alpha\mu)^{-(t+1)}$, and $W_T = \sum_{t=0}^{T} w_t$, the iterates generated by the update rule in* (4.10) *are such that, if $\boldsymbol{\theta}_{out}$ be a randomly chosen iterate from $\{\boldsymbol{\theta}_t\}_{t=0}^{T}$ with probability $\frac{w_t}{W_T}$ and $r_{out} \triangleq \|\boldsymbol{\theta}_{out} - \boldsymbol{\theta}^*\|$ and $\bar{\tau} = \max\{\tau, \tau_{mix}\}$ then*

$$\mathbb{E}\left[r_{out}^2\right] \leq C_2 T \exp\left(-0.5\alpha\mu T\right) r_0^2 + C_3 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu}. \tag{4.11}$$

where $C_1, C_2, C_3 \geq 2$ are universal constants. Setting $\alpha = \frac{\mu}{C_1 L^2 \bar{\tau}}$, we get

$$\mathbb{E}\left[r_{out}^2\right] \leq C_2 T \exp\left(-0.5 \frac{\mu^2}{C_1 L^2 \bar{\tau}} T\right) r_0^2 + \frac{C_3 \sigma^2}{C_1}. \tag{4.12}$$

Main Takeaways: We now outline the key takeaways of the above Theorem. First, we showed exponential convergence of $\mathbb{E}\left[r_{out}^2\right]$ to a ball around the fixed point $\boldsymbol{\theta}^*$. This latter result represents the first finite-time convergence bound for SA with delayed updates under Markovian sampling. Second, the obtained convergence exponent scales inversely with $\bar{\tau} = \max\{\tau, \tau_{mix}\}$. Hence, if $\tau \geq \tau_{mix}$, we get a dependency on the constant delay τ, which is consistent with what is known for SA with delayed updates in the i.i.d. sampling case, specifically in the case of SGD with constant delay (Stich and Karimireddy, 2020). Note that this dependency has been shown to be tight for SGD (Arjevani et al., 2020), and, consequently, our rate is optimal in terms of the obtained dependency on the delay τ. If $\tau_{mix} \geq \tau$, the obtained convergence exponent scales inversely with τ_{mix}, which is consistent with the non-delayed case of SA with Markovian sampling (Srikant and Ying, 2019; Bhandari et al., 2018a), and has been shown to be in fact minimax optimal (Nagaraj et al., 2020). In summary, in the above Theorem we provide the first finite-time convergence bound for SA with

updates subject to constant delays under Markovian sampling, getting a convergence rate that has optimal dependencies on both the delay τ and the mixing time τ_{mix}.

Outline of the Analysis and Challenges. We now provide the main steps of the analysis and underline the key challenges that make each step necessary. First of all, similarly to (Stich and Karimireddy, 2020), we define a sequence of virtual iterates, $\tilde{\boldsymbol{\theta}}_t$, which, at each iteration t, are updated with the actual SA update direction $\mathbf{g}(\boldsymbol{\theta}_t, o_t)$:

$$\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t). \tag{4.13}$$

Accordingly, we define an error term $\mathbf{d}_t$ which is the gap between $\boldsymbol{\theta}_t$ and $\tilde{\boldsymbol{\theta}}_t$ at each iteration t, i.e., $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t + \mathbf{d}_t$. A key step in the analysis that, as it was the case for (Stich and Karimireddy, 2020), *relies on the fact that the delay τ is constant*, is that, for any $t \geq 0$, we can write the error term as follows,

$$\mathbf{d}_t = \alpha \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l). \tag{4.14}$$

In the first part of the proof of Theorem 9, we provide a convergence bound for the virtual iterates sequence $\tilde{\boldsymbol{\theta}}_t$, studying $\mathbb{E}\left[\tilde{r}_t^2\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\|^2\right]$ and providing a bound that is a function of $\|\mathbf{d}_t\|^2$ and $\tilde{r}_l^2, \|\mathbf{d}_l\|^2$, with $l = t - \bar{\tau}, ..., t-1$. To get this bound, we analyze the following recursion

$$\begin{aligned}
\tilde{r}_{t+1}^2 &= \tilde{r}_t^2 + 2\alpha\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\rangle + \alpha^2 \|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \\
&= \tilde{r}_t^2 + 2\alpha\langle \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\rangle + 2\alpha h_t + 2\alpha m_t + \alpha^2 \|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \\
&\leq (1 - 2\alpha\mu)\tilde{r}_t^2 + 2\alpha h_t + 2\alpha m_t + \alpha^2 \|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2,
\end{aligned} \tag{4.15}$$

where the last inequality follows from Assumption 5, and where we have

$$\begin{aligned}
h_t &:= \langle \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\rangle, \\
m_t &:= \langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\rangle.
\end{aligned} \tag{4.16}$$

The term h_t is an error term related to the Markovian sampling. Indeed, if the process o_t were sampled in an i.i.d. fashion, it would be $\mathbb{E}[h_t] = 0$. However, due to the correlated nature of o_t, this

does not hold true, and, consequently, h_t requires careful care in the analysis. On the other hand, the term m_t is an error term related to the delayed nature of the SA algorithm under consideration. In absence of delays, it would be $m_t = 0$. To obtain the convergence bound for $\mathbb{E}\left[\tilde{r}_t^2\right]$, we provide bounds on $\mathbb{E}\left[h_t\right]$, m_t and $\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2$. Obtaining bounds for these terms require some work, that we present in the form of auxiliary Lemmas in the last part of this section. Providing the bound on $\mathbb{E}\left[h_t\right]$ is the most challenging part, which also requires mixing time arguments, and which we provide in Lemma 24. In order to provide a bound that is a function of $\|\mathbf{d}_t\|^2$ and $\tilde{r}_l^2, \|\mathbf{d}_l\|^2$, with $l = t - \bar{\tau}, ..., t-1$, indeed, we need to provide a novel analysis compared to the one used for the non-delayed SA under Markovian sampling in (Srikant and Ying, 2019). Specifically, we need to introduce a new way to bound the terms $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2$ and use the corresponding bounds accordingly when bounding $\mathbb{E}\left[h_t\right]$. The bound obtained thanks to the auxiliary Lemmas has the following form:

$$\begin{aligned}\mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 2\alpha\mu + 48\alpha^2 L^2 \bar{\tau})\mathbb{E}\left[\tilde{r}_t^2\right] + 128\alpha^2 L^2 \bar{\tau}\sigma^2 \\ + 4\alpha^2 L^2 \mathbb{E}\left[\|\mathbf{d}_t\|^2\right] + 20\alpha^2 L^2 \sum_{l=t-\bar{\tau}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] + 2\alpha \bar{B}_t,\end{aligned} \tag{4.17}$$

with

$$\bar{B}_t = \begin{cases} 48L^2\sigma^2 + 216\alpha L^2 \sigma^2 \tau_{mix} & \text{if } \ 0 \leq t < \tau_{mix} \\ 0 & \text{otherwise} \end{cases}. \tag{4.18}$$

Starting from this bound, the use of the weighted average $\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right]$ comes into play. Applying the weighted average to both sides of the bounds, applying some manipulations and with the proper choice of upper bound on the step size α, we are able to get the following inequality

$$\begin{aligned}\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 0.5\alpha\mu) \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] + 150 W_T \alpha^2 L^2 \bar{\tau}\sigma^2 \\ + 2W_{\tau_{mix}-1}\alpha(48L^2\sigma^2 + 216\alpha L^2\sigma^2\tau_{mix}).\end{aligned} \tag{4.19}$$

This last inequality is obtained thanks to Lemma 20, which we state in the next paragraph and which establishes a bound on $\sum_{t=0}^{T} w_t \mathbb{E}\left[\|\mathbf{d}_t\|^2\right]$. With some further manipulations and using the

fact that $\mathbb{E}\left[r_t^2\right] \leq 2\mathbb{E}\left[\tilde{r}_t^2\right] + 2\mathbb{E}\left[\|\mathbf{d}_t\|^2\right]$ together with Lemma 26, we derive the final result.

Auxiliary Lemmas. Here, we present the main Lemmas needed to prove Theorem 9. We start with two bounds on the norms $\|\mathbf{d}_t\|$ and $\|\mathbf{d}_t\|^2$, as follows

Lemma 19. *The two following inequalities hold:*

$$(i) \qquad \|\mathbf{d}_t\| \ \leq \alpha\tau L\sigma + \alpha L \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\| \tag{4.20}$$

$$(ii) \qquad \|\mathbf{d}_t\|^2 \leq 2\alpha^2\tau^2 L^2\sigma^2 + 2\alpha^2\tau L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2. \tag{4.21}$$

Using this Lemma, we provide the following result, that is needed to obtain the bound in (4.19).

Lemma 20. *For $\alpha \leq \frac{1}{4\tau L}$, the following inequality holds:*

$$\sum_{t=0}^{T} w_t \|\mathbf{d}_t\|^2 \leq 4W_T\alpha^2\tau^2 L^2\sigma^2 + 16\alpha^2\tau^2 L^2 \sum_{t=0}^{T} w_t \|\tilde{\boldsymbol{\theta}}_t\|^2. \tag{4.22}$$

In the next Lemma, we provide bounds on the terms $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2$.

Lemma 21. *For any $t \geq \tau_{mix}$, we have*

$$(i) \qquad \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\| \ \leq L\alpha\sigma\tau_{mix} + L\alpha \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\| \tag{4.23}$$

$$(ii) \qquad \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\|^2 \leq 2L^2\alpha^2\tau_{mix}^2\sigma^2 + 2L^2\alpha^2\tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|^2 \tag{4.24}$$

Note that this Lemma is a variation of Lemma 3 in (Srikant and Ying, 2019), which is key to invoke mixing time arguments to get finite-time convergence bounds in existing non-delayed SA analysis. To obtain a bound in the form (4.17), we need to bound $\mathbb{E}\left[h_t\right]$ properly, for which, in turn, we need

Lemma 21. Furthermore, note that, in contrast to (Srikant and Ying, 2019), the bound is obtained for the sequence of *virtual iterates*. In the next three Lemmas, we provide bounds for the key terms of the bound in (4.15), i.e., $\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2$, m_t and $\mathbb{E}[h_t]$, respectively.

Lemma 22. *For all $t \geq 0$, we have*

$$\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \leq 4L^2\|\mathbf{d}_t\|^2 + 8L^2\tilde{r}_t^2 + 10L^2\sigma^2. \tag{4.25}$$

Lemma 23. *For $t \geq 0$, we have*

$$m_t \leq 6\alpha\tau L^2\sigma^2 + 3\alpha\tau L^2\tilde{r}_t^2 + 2\alpha L^2 \sum_{l=t-\tau}^{t-1} \|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2. \tag{4.26}$$

Note that the proof of this last Lemma relies on the bound on $\|\mathbf{d}_t\|$ established in Lemma 19. The next Lemma establishes a bound on $\mathbb{E}[h_t]$ relying on the mixing properties of the Markov chain $\{o_t\}$ and on the bounds on $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2$ established in Lemma 21.

Lemma 24. *For $t \geq \tau_{mix}$ and $\alpha \leq \frac{1}{8\tau_{mix}L}$, we have*

$$\mathbb{E}[h_t] \leq \alpha\tau_{mix}L^2\left(32\sigma^2 + 12\mathbb{E}\left[\tilde{r}_t^2\right]\right) + 8\alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right]. \tag{4.27}$$

Lemma 24 is the key and most challenging part of the proof, which allows us to get to the bound in (4.17). Note that the above Lemma is only valid for $t \geq \tau_{mix}$, which is a necessary condition to invoke mixing time arguments, an issue that does not exist in the i.i.d. analysis. However, to get the desired result, we need to be able to bound $\mathbb{E}[h_t]$ for any $t \geq 0$. Hence, we also need the following result.

Lemma 25. *For $0 \leq t < \tau_{mix}$, we get*

$$h_t \leq 84L^2\sigma^2 + 216\alpha L^2\sigma^2\tau_{mix} \tag{4.28}$$

Using these last four Lemmas, i.e., Lemma 22, 23, 24 and 25 in combination with Lemma 20 we are able to get the bound in (4.19). At this point, the conclusion of the proof is enabled by using $\mathbb{E}\left[r_t^2\right] \leq 2\mathbb{E}\left[\tilde{r}_t^2\right] + 2\mathbb{E}\left[\|\mathbf{d}_t\|^2\right]$ and by the following Lemma which is inspired by Lemma 12 in (Stich and Karimireddy, 2020).

Lemma 26. *For every non-negative sequence $\{u_t\}_{t\geq 0}$ and any parameters $a > 0$, $c \geq 0$, $T \geq 0$, for a constant $\alpha > 0$ and weights $w_t := (1 - a\alpha)^{-(t+1)}$, it holds*

$$\Psi_T := \frac{1}{W_T} \sum_{t=0}^{T} \left(w_t \left(1 - a\alpha\right) u_t - w_t u_{t+1}\right) + c = \frac{w_{-1} u_0}{W_T} - \frac{w_T u_{T+1}}{W_T} + c. \tag{4.29}$$

The proofs of all the Lemmas in this section and the complete proof of Theorem 9 are available in section 4.9.

4.6. Stochastic Approximation with Time-Varying Delays

In this section, we present the first finite-time convergence analysis of the delayed SA update rule that was introduced in (4.6). Note that, by Assumption 8, we can re-write (4.6) as:

$$\textbf{\textit{Delayed SA:}} \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \quad \tau_t \leq \min\{t, \tau_{max}\} \tag{4.30}$$

The following theorem provides a finite-time convergence bound for the update in (4.30) and it is the second main contribution of this chapter.

Theorem 10. *Let $r_t \triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|$. There exists an absolute constant $C_1 \geq 1$ such that for*

$$\alpha \leq \frac{\mu}{C_1 L^2 (\tau_{mix} + \tau_{max})} \;\; \textit{and} \;\; T \geq 2\tau_{mix} + \tau_{max}, \tag{4.31}$$

we have

$$\mathbb{E}\left[r_T^2\right] \leq O\left(\exp\left(\frac{-\alpha\mu T}{(\tau_{mix} + \tau_{max})}\right)\right) + O\left(L^2 \frac{\alpha(\tau_{mix} + \tau_{max})\sigma^2}{\mu}\right). \tag{4.32}$$

In addition, setting $\alpha = \frac{\mu}{C_1 L^2(\tau_{mix}+\tau_{max})}$ *yields to*

$$\mathbb{E}\left[r_T^2\right] \leq O\left(\exp\left(\frac{-\mu^2 T}{L^2(\tau_{mix}+\tau_{max})^2}\right)\right) + O(\sigma^2). \tag{4.33}$$

Main Takeaways:

There are many relevant takeaways from this theorem, which, as far as we are aware, is the first ever finite-time convergence result for SA under Markovian sampling with time-varying delays. We focus on the convergence bound in (4.33), i.e., the case in which the step size matches the upper bound. We note that, (i) with a choice of step size inversely proportional to $\tau_{mix} + \tau_{max}$ the *Delayed SA* update rule (4.30) converges in expectation to a ball around $\boldsymbol{\theta}^*$ whose radius is proportional to the "variance" term σ^2 exponentially fast; (ii) the exponent of convergence gets scaled down by a factor of $(\tau_{max} + \tau_{mix})^2$. Remarkably, the dependence on τ_{max} is precisely consistent with what is known for stochastic optimization (i.i.d. sampling) with time-varying delays (Assran et al., 2020; Feyzmahdavian et al., 2016; Gurbuzbalaban et al., 2017). In particular, for gradient descent on a stronlgy convex smooth cost function, existing results for time-varying delay (see, e.g., (Gurbuzbalaban et al., 2017, Theorem 3.3)), require a step size inversely proportional to τ_{max} and obtain an exponential convergence with convergence exponent that gets scaled down by a factor proportional to τ_{max}^2. ... Existence of tight analysis for const step delay ... existence of better dependence on taumax for delay adaptive algorithms ... and we can also get that with our adaptive alg ...

Outline of the Analysis. We now provide insights on the key steps in the analysis. To analyze the convergence of the update rule in (4.30), we consider the delay as a perturbation to the original update. We define the error at iteration t as follows

$$\mathbf{e}_t \triangleq \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \tag{4.34}$$

which we use to rewrite the update rule in (4.30) as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \alpha \mathbf{e}_t. \tag{4.35}$$

We now examine $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2$ using (4.35), which leads us to

$$\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2 = J_{t,1} + \alpha^2 J_{t,2} - 2\alpha J_{t,3}. \tag{4.36}$$

with

$$\begin{aligned} J_{t,1} &\triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2, \\ J_{t,2} &\triangleq \|\mathbf{e}_t\|^2, \\ J_{t,3} &\triangleq \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle + \alpha \langle \mathbf{e}_t, \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle. \end{aligned} \tag{4.37}$$

Note that the presence of $J_{t,2}$ and $J_{t,3}$ in (4.138) is a consequence of the delay and it would not occur in the case of non-delayed updates. The convergence analysis is built up providing bounds on the three terms of (4.37).

Main challenges. We now comment on some of the main challenges of the analysis. First, the term $J_{t,1}$ cannot be bounded with the methods used for non-delayed SA under Markovian sampling analysis in (Srikant and Ying, 2019). Indeed, Lemma 3 in (Srikant and Ying, 2019), which is key to prove the finite-time linear convergence rate invoking properties of the geometric mixing of the Markov chain, is not valid when using the delayed operator $\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$. To see why this is the case, note that Lemma 3 in (Srikant and Ying, 2019) establishes a bound on $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|$, for any $\tau > 0$, of the following form

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq O(\alpha\tau)(\|\boldsymbol{\theta}_t\| + \sigma), \tag{4.38}$$

which, however, does not hold true when using the delayed operator $\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$. Indeed, the first key step in proving (4.38) is using the fact that $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| \leq O(\alpha)(\|\boldsymbol{\theta}_t\| + \sigma)$, which is not true for the delayed case, where we can only get $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| \leq O(\alpha)(\|\boldsymbol{\theta}_{t-\tau_t}\| + \sigma)$ by using the bound $\|\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})\| \leq O(\alpha)(\|\boldsymbol{\theta}_{t-\tau_t}\| + \sigma)$ on the delayed operator. This fact, that prevents

us from applying the analysis of Lemma ?? in (Srikant and Ying, 2019), forces us to develop a different strategy and to prove a more general result, the statement of which we provide in Lemma **??**. This new Lemma enables us to deal with $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|$ in an functional way with respect to the proof of finite-time rates for the considered delayed SA algorithm. Second, note that bounding the term $\langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle$ is much more challenging compared to the i.i.d. sampling setting considered in the optimization literature with delays (Zhou et al., 2018; Koloskova et al., 2022b; Arjovsky et al., 2017; Cohen et al., 2021). This difficulty arises due to the statistical correlation among the terms in $\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$ and $\boldsymbol{\theta}_t - \boldsymbol{\theta}^*$, which calls for a more involved analysis. Indeed, the fact that in general, for correlated Markovian samples, $\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_t, o_t)\right] \neq \bar{\mathbf{g}}(\boldsymbol{\theta}_t)$, forces us to invoke mixing time arguments to bound this cross term in a way that is instrumental to get the desired finite-time rate, as it is typically done for SA under Markovian sampling. However, the presence of the delay in the operator $\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$ introduces further statistically correlated iterates $\boldsymbol{\theta}_{t-\tau_t}$ and observations $o_{t-\tau_t}$ in the analysis, whose interplay needs to be carefully taken care of. To do so, we provide a novel analysis, whose result is stated in Lemma 28. This analysis is enabled also thanks to the new result stated in Lemma 27 which generalizes Lemma 3 in (Srikant and Ying, 2019) and which we present next.

Auxiliary Lemmas. We now introduce three Lemmas that are fundamental to prove Theorem 10. We start with a Lemma that provides bounds in expectation on quantities of the form $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|$. This result, as mentioned above, represents a generalization of Lemma 3 in (Srikant and Ying, 2019), being suitable to be applied for the analysis of the delayed case. Define

$$\begin{aligned} r_{t,1} &\triangleq \max_{t-\tau_{mix}-2\tau_{max} \leq l \leq t} \mathbb{E}\left[r_l\right], \\ r_{t,2} &\triangleq \max_{t-\tau_{mix}-2\tau_{max} \leq l \leq t} \mathbb{E}\left[r_l^2\right]. \end{aligned} \tag{4.39}$$

Lemma 27. *For any $t \geq \tau_{mix}$, we have*

$$(i) \qquad \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|\right] \leq \alpha\tau_{mix}L(r_{t,1} + 2\sigma),$$

$$(ii) \qquad \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2\right] \leq 2\alpha^2\tau_{mix}^2L^2(2r_{t,2} + 3\sigma^2).$$

Similarly, for any $t \geq 0$ and $\tau_t \leq t$,

$$(iii) \qquad \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|\right] \leq \alpha\tau_{max}L(r_{t,1} + 2\sigma),$$

$$(iv) \qquad \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|^2\right] \leq 2\alpha^2\tau_{max}^2L^2(2r_{t,2} + 3\sigma^2).$$

The above Lemma plays a critical role in bounding the error caused by the delay in the convergence analysis. Specifically, the Lemma establishes a bound that enables us to relate $\boldsymbol{\theta}_t$, $\boldsymbol{\theta}_{t-\tau_t}$, and $\boldsymbol{\theta}_{t-\tau_{mix}}$, which is key to establish the finite-time linear rate of Theorem 9. Accordingly, we can bound the impact of delay on the terms $J_{t,1}, J_{t,2}$ and $J_{t,3}$ in terms of $r_{t,1}$ and $r_{t,2}$. Next, we state Lemma 28, that provides explicit bounds for $J_{t,1}$, $J_{t,2}$, and $J_{t,3}$.

Lemma 28. *Let $t \geq \tau_{mix} + 2\tau_{max}$, then*

1. *Bounding $\mathbb{E}[J_{t,1}]$:*

$$\mathbb{E}[J_{t,1}] \leq (1 - 2\alpha\mu)\mathbb{E}\left[r_t^2\right] + O(\alpha^2\tau_{mix})(r_{t,2} + \sigma^2). \tag{4.40}$$

2. *Bounding $\mathbb{E}[J_{t,2}]$:*

$$\mathbb{E}[J_{t,2}] \leq O(r_{t,2} + \sigma^2). \tag{4.41}$$

3. *Bounding $\mathbb{E}[J_{t,3}]$:*

$$\mathbb{E}[J_{t,3}] \leq O(\alpha)(\tau_{mix} + \tau_{max})(r_{t,2} + \sigma^2). \tag{4.42}$$

Using the above lemma, we can rewrite equation (4.138) as

$$\begin{aligned}
\mathbb{E}\left[r_{t+1}^2\right] &= \mathbb{E}\left[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2\right] \\
&= \mathbb{E}\left[J_{t,1}\right] + \alpha^2 \mathbb{E}\left[J_{t,2}\right] - 2\alpha \mathbb{E}\left[J_{t,3}\right] \\
&\leq (1 - 2\alpha\mu)\mathbb{E}\left[r_t^2\right] + O(\alpha^2 \tau_{mix})(r_{t,2} + \sigma^2) \\
&\quad + O(\alpha^2)(r_{t,2} + \sigma^2) + O(\alpha^2)(\tau_{mix} + \tau_{max})(r_{t,2} + \sigma^2) \\
&\leq (1 - 2\alpha\mu)\mathbb{E}\left[r_t^2\right] + O(\alpha^2)(\tau_{mix} + \tau_{max})(r_{t,2} + \sigma^2).
\end{aligned} \tag{4.43}$$

To employ the above recursion, we require a technique to deal with the maximum in $r_{t,2}$. For this purpose, we utilize the following lemma.

Lemma 29 ((Feyzmahdavian et al., 2014))**.** *Let V_k be non-negative real numbers that satisfy*

$$V_{k+1} \leq pV_k + q \max_{k - d(k) \leq \ell \leq k} V_\ell + \beta,$$

for some non-negative constants p and q. Here, $k \geq 0$ and $0 \leq d(k) \leq d_{\max}$ for some positive constant $d_{\max}$. If $p + q < 1$, then we have

$$V_k \leq r^k V_0 + \epsilon,$$

where $r = (p+q)^{1/(1+d_{\max})}$ and $\epsilon = \frac{\beta}{1-p-q}$.

By applying Lemma 52 and substituting equation (4.54) into it, we can obtain the result stated in Theorem 10.

The proofs of all the Lemmas in this section and the complete proof of Theorem 10 are available in section 4.10.

4.7. Delay-Adaptive Stochastic Approximation

In the previous section, we discussed the vanilla delayed stochastic approximation and derived its convergence rate. However, the convergence rate is dependent on the maximum delay, τ_{max}, and

selecting the proper step size requires a priori knowledge of τ_{max}. Oftentimes, the worst-case delay τ_{max} is too large, leading to slow convergence of (4.30); and also its exact value might be τ_{max} unknown. In this section, we introduce a new update rule whose convergence rate only depends on the *average* delay, and does not require the knowledge of τ_{max}. Specifically, we consider the following update rule

$$\textbf{\textit{Lazy adaptive SA:}} \qquad \boldsymbol{\theta}_{t+1} = \begin{cases} \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}) & \text{if} \quad \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| \leq \epsilon, \\ \boldsymbol{\theta}_t & \text{otherwise.} \end{cases} \tag{4.44}$$

The rationale behind the *Lazy adaptive SA* is to utilize only the informative pseudo-gradients $\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$ that are in proximity to $\boldsymbol{\theta}_t$, instead of using all of them. By doing so, we can control the impact of delay and achieve a convergence rate that solely relies on the average delay:

$$\tau_{avg} = \frac{1}{T} \sum_{t=1}^{T} \tau_t.$$

The main result regarding the *Lazy adaptive SA* update rule is presented in the following theorem.

Theorem 11. *Let $r_t \triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|$. There exists an absolute constant $C_2 \geq 1$ such that for*

$$\alpha \leq \frac{\mu}{C_2 L^2 \tau_{mix}}, \quad T \geq \max\{\tau_{mix}\tau_{avg}, \tau_{mix} + \tau_{max}\}, \tag{4.45}$$

and $\epsilon = \alpha$, the iterates of (4.189) satisfy

$$\mathbb{E}\left[r_T^2\right] \leq O\left(\exp\left(\frac{-\alpha\mu T}{\tau_{avg}}\right)\right) + O\left(\frac{\alpha L^2 \tau_{mix}\sigma^2}{\mu}\right). \tag{4.46}$$

Additionally, if we set $\alpha = \frac{\mu}{C_2\tau_{mix}}$, we obtain

$$\mathbb{E}\left[r_T^2\right] \leq O\left(\exp\left(\frac{-\mu^2 T}{L^2 \tau_{mix}\tau_{avg}}\right)\right) + O(\sigma^2). \tag{4.47}$$

Outline of Analysis.

Let $I_t = 1$, if $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| \leq \epsilon$ and $I_t = 0$ otherwise. Then, we can express $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2$ as

$$\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2 = I_t \left(J_{t,1} + \alpha^2 J_{t,2} - 2\alpha J_{t,3}\right) + (1 - I_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|^2. \tag{4.48}$$

with

$$\begin{aligned} J_{t,1} &\triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2, \\ J_{t,2} &\triangleq \|\mathbf{e}_t\|^2, \\ J_{t,3} &\triangleq \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle. \end{aligned} \tag{4.49}$$

In order to analyze the convergence of the Lazy adaptive SA update rule, we first derive bounds for $J_{t,1}$, $J_{t,2}$, and $J_{t,3}$ for iterations in which $I_t = 1$. Then, we establish a lower bound on the number of iterations where $I_t = 1$. By using these two results, we are able to obtain a finite-time convergence rate, which we present in Theorem 11.

Comments on the analysis. Similarly to the previous section, in which we proved a result for the vanilla delayed update rule introduced in (4.6), to provide a finite-time rate for *Lazy adaptive SA* we need to provide a bound on $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|$, which we do in Lemma 30, and which is essential for analyzing the convergence of the update rule. Note that the bound presented in this Lemma, unlike Lemma 27, only depends on $\boldsymbol{\theta}_t$ and not on previous iterations. This is key to simplify the analysis and to obtain a better convergence rate. We now comment on the main differences in the analysis with respect to the previous section and on how the delay-adaptive update rule enables the theoretical results of this section. One of the main challenges in the proof of the vanilla delayed SA algorithm, for which we provided the *first finite-time convergence analysis* in the previous section, was related to the interplay between the statistical correlation of iterates and observations of different time steps, i.e., $t, t - \tau_t, t - \tau_{mix}$. While the interplay between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t-\tau}$ is well understood for the SA under Markovian sampling (Srikant and Ying, 2019), the presence of the delays makes the finite-time analysis much more involved, as we have illustrated in the previous section. On the other hand, the adaptive strategy considered in this section and presented in the update rule shown in (4.189) introduces a fundamental property: the capacity to control the term $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|$ via the careful choice of the threshold term ϵ enables us to partially get rid of some of the statistically correlated

terms that were making the analysis complex in the previous analysis. Furthermore, thanks to the adaptive strategy, we can derive a generalized version of Lemma 3 in (Srikant and Ying, 2019) that preserves the dependency only to the current iterate $\boldsymbol{\theta}_t$. As a consequence of the above points, we are able to derive the better convergence rate previously presented that, instead of depending on τ_{max}, only depends on τ_{avg}.

Main Intuition. We are able to derive a bound for the "Lazy adaptive SA" because it can be interpreted as a stochastic approximation with minor errors. This is supported by the fact that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| \leq \epsilon$ implies $\|\mathbf{g}(\boldsymbol{\theta}_t, o) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o)\| \leq 2\epsilon$. As a result, every time we update, we move in the correct direction with only a small amount of error.

Auxiliary Lemmas. Similarly to Section 4.6, we provide three Lemmas that are fundamental to prove the main result of this section, i.e., Theorem 11. As in the case of the vanilla delayed SA update, we start by providing a result that provides a bound on the norm of $\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}$.

Lemma 30. *For any $\tau \geq 1$ and $t \geq \tau$, we have*

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq O(\alpha\tau(r_t + \beta)), \tag{4.50}$$

where $\beta = \epsilon + \sigma$.

Using Lemma 30, we can bound $J_{t,1}$, $J_{t,2}$, and $J_{t,3}$ in Lemma 31.

Lemma 31. *For $t \geq \tau_{mix} + \tau_{max}$, if $I_t = 1$, we have*

1. *Bounding $\mathbb{E}[J_{t,1}]$:*

$$\mathbb{E}[J_{t,1}] \leq \left(1 - 2\alpha\mu + O(\alpha^2\tau_{mix})\right)\mathbb{E}\left[r_t^2\right] + O(\alpha^2\tau_{mix})\beta^2 \tag{4.51}$$

2. *Bounding $\mathbb{E}[J_{t,2}]$:*

$$\mathbb{E}[J_{t,2}] \leq O(\mathbb{E}\left[r_t^2\right] + \beta^2) \tag{4.52}$$

3. Bounding $\mathbb{E}[J_{t,3}]$*:*

$$\mathbb{E}[J_{t,3}] \leq O(\alpha\tau_{mix}(\mathbb{E}\left[r_t^2\right] + \beta^2)) \tag{4.53}$$

With the help of the previous lemma, we can rewrite Equation (4.48) as follows, when $I_t = 1$:

$$\begin{aligned}
\mathbb{E}\left[r_{t+1}^2\right] &= \mathbb{E}\left[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2\right] \\
&= \mathbb{E}[J_{t,1}] + \alpha^2\mathbb{E}[J_{t,2}] - 2\alpha\mathbb{E}[J_{t,3}] \\
&\leq \left(1 - 2\alpha\mu + O(\alpha^2\tau_{mix})\right)\mathbb{E}\left[r_t^2\right] + O(\alpha^2\tau_{mix})\beta^2 \\
&\quad + O(\alpha^2(\mathbb{E}\left[r_t^2\right] + \beta^2)) + O(\alpha^2\tau_{mix}(\mathbb{E}\left[r_t^2\right] + \beta^2)) \\
&\leq \left(1 - 2\alpha\mu + O(\alpha^2\tau_{mix})\right)\mathbb{E}\left[r_t^2\right] + O(\alpha^2\tau_{mix})\beta^2.
\end{aligned} \tag{4.54}$$

To finish the proof, we only need to determine the number of times we update the variable $\boldsymbol{\theta}$ in T iterations. This can be achieved through the following lemma, which has been borrowed from (Cohen et al., 2021). By utilizing this lemma, we can complete the proof and obtain the bound presented in Theorem 11.

Lemma 32. *Let* τ_{avg} *be the average of delay,* $\tau_{avg} = \frac{1}{T}\sum_{t=1}^{T}\tau_t$*. Then the number of updates that Lazy adaptive SA makes is at least* $\Omega\left(\frac{T}{\tau_{avg}}\right)$*.*

The proofs of all the Lemmas in this section and the complete proof of Theorem 11 are available in section 4.11.

4.8. Proofs of Lemmas and Theorems

In the following appendices, we provide the proofs for the theoretical results stated in the paper. In particular, we provide the proofs for all the Theorems and Lemmas. We start by recalling the implications of Section 4.4 in the following.

4.8.1. Preliminaries of Proofs

First, recall that from Assumption 5 we have, $\forall\boldsymbol{\theta} \in \mathbb{R}^d$:

$$\langle\boldsymbol{\theta}^* - \boldsymbol{\theta}, \bar{\mathbf{g}}(\boldsymbol{\theta})\rangle \geq \mu\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|^2. \tag{4.55}$$

We will also use the fact that the SA update directions and their steady-state versions are L-Lipschitz, i.e., $\forall o \in \{o_t\}_{t\in\mathbb{N}}$, and $\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, we have:

$$\begin{aligned} &\|\bar{\mathbf{g}}(\boldsymbol{\theta}) - \bar{\mathbf{g}}(\boldsymbol{\theta}')\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|, \text{ and} \\ &\|\mathbf{g}(\boldsymbol{\theta}, o_t) - \mathbf{g}(\boldsymbol{\theta}', o_t)\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|. \end{aligned} \tag{4.56}$$

We further have

$$\|\mathbf{g}(\boldsymbol{\theta}, o_t)\| \leq L(\|\boldsymbol{\theta}\| + \sigma), \forall o \in \{o_t\}_{t\in\mathbb{N}}, \forall \boldsymbol{\theta} \in \mathbb{R}^d. \tag{4.57}$$

Given that $(x+y)^2 \leq 2(x^2+y^2), \forall x, y \in \mathbb{R}$, we will often use the following inequality:

$$\|\mathbf{g}(\boldsymbol{\theta}, o_t)\|^2 \leq L^2(\|\boldsymbol{\theta}\| + \sigma)^2 \leq 2L^2(\|\boldsymbol{\theta}\|^2 + \sigma^2). \tag{4.58}$$

Without loss of generality, we assume that

$$L \geq 1, \quad \sigma \geq \max\{\|\boldsymbol{\theta}_0\|, \|\boldsymbol{\theta}^*\|\}, \quad \mu < 1. \tag{4.59}$$

We will often use the fact that, for any $x, y \in \mathbb{R}$, we have

$$xy \leq \frac{1}{2}(x^2 + y^2). \tag{4.60}$$

In addition, we will often use the fact that, for $t \geq 2$, $a_i \in \mathbb{R}, i = 0, ..., t-1$, it holds

$$\left(\sum_{i=0}^{t-1} a_i\right)^2 \leq t \sum_{i=0}^{t-1} a_i^2 \tag{4.61}$$

4.9. Appendix A: Proof of Theorem 9

First, we recall the definition of the SA recursion with constant delay:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau}) \text{ for } t \geq 0. \tag{4.62}$$

For analysis purposes, we define a virtual iterate, $\tilde{\boldsymbol{\theta}}_t$. This virtual iterate is updated with the SA update direction without delays. We also introduce the related error term $\mathbf{d}_t$, which is the gap between the virtual iterate and the actual iterate.

$$\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t + \mathbf{d}_t, \quad \text{with } \mathbf{d}_0 = \mathbf{0}. \tag{4.63}$$

For both $\tilde{\boldsymbol{\theta}}_t$ and $\mathbf{d}_t$, we can write the following recursions, for $t \geq 0$:

$$\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t) \tag{4.64}$$

$$\mathbf{d}_{t+1} = \mathbf{d}_t + \alpha(\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau})). \tag{4.65}$$

We define $\mathbf{g}(\boldsymbol{\theta}_l, o_l) = \boldsymbol{\theta}_l = \mathbf{d}_l = \mathbf{0}$ for $l < 0$. Also, we define $\tilde{\boldsymbol{\theta}}_l = \boldsymbol{\theta}^*$ for $l < 0$.

4.9.1. Proofs of Auxiliary Lemmas of Section 4.5

We first state and prove the following Lemma, which we will use in the proof of Theorem 9.

Lemma 33. *For $w_t := (1 - 0.5\mu\alpha)^{-(t+1)}$ with $\alpha \leq \frac{\mu}{C\bar{\tau}}$, $C \geq 2$, the following inequality holds for $0 \leq i \leq 2\bar{\tau}$, and for any t,*

$$w_t \leq 2w_{t-i}. \tag{4.66}$$

Proof.

$$
\begin{aligned}
w_t &= w_{t-i}\left(1-\frac{\mu\alpha}{2}\right)^{-i} \\
&\overset{(a)}{\leq} w_{t-i}\left(1-\frac{\mu^2}{2C\bar{\tau}}\right)^{-i} \\
&\overset{(b)}{\leq} w_{t-i}\left(1-\frac{\mu^2}{2C\bar{\tau}}\right)^{-\bar{\tau}} \\
&\overset{(c)}{\leq} w_{t-i}\left(1-\frac{1}{4\bar{\tau}}\right)^{-\bar{\tau}} \\
&\overset{(d)}{\leq} w_{t-i}\left(1+\frac{1}{2\bar{\tau}}\right)^{\bar{\tau}} \\
&\overset{(e)}{\leq} w_{t-i}\exp\left(\frac{1}{2}\right) \\
&\leq 2w_{t-i},
\end{aligned}
\tag{4.67}
$$

in (a), we used the bound on α, in (b), we used the bound on i, in (c), we used $\mu < 1$ and $C \geq 2$, in (d), we used

$$
(1-x)^{-1} \leq (1+2x) \text{ for } \quad 0 \leq x \leq \frac{1}{2}, \tag{4.68}
$$

and for (e) we used $(1+x)^k \leq \exp(xk)$ for $k \geq 0$. □

We defined $\mathbf{g}(\boldsymbol{\theta}_i, o_i) = 0$ for $i < 0$, $\boldsymbol{\theta}_t = \boldsymbol{\theta}_0$ for $t \leq 0$, and $\mathbf{d}_t = 0$ for $t \leq 0$. First, note that, starting from the definition of $\mathbf{d}_t$ in (4.65),

$$
\begin{aligned}
\mathbf{d}_{t+1} &= \mathbf{d}_t + \alpha\left(\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau}\right) \\
&= \mathbf{d}_{t-1} + \alpha\left(\mathbf{g}(\boldsymbol{\theta}_{t-1}, o_{t-1}) - \mathbf{g}(\boldsymbol{\theta}_{t-1-\tau}, o_{t-1-\tau})\right) \\
&\qquad + \alpha\left(\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau})\right) \\
&= \mathbf{d}_\tau \quad + \alpha\sum_{l=\tau}^{t}\left(\mathbf{g}(\boldsymbol{\theta}_l, o_l) - \mathbf{g}(\boldsymbol{\theta}_{l-\tau}, o_{l-\tau})\right) \\
&\overset{(*)}{=} \mathbf{d}_0 \quad + \alpha\sum_{l=0}^{t}\left(\mathbf{g}(\boldsymbol{\theta}_l, o_l) - \mathbf{g}(\boldsymbol{\theta}_{l-\tau}, o_{l-\tau})\right) \\
&\overset{(**)}{=} \mathbf{0} \quad + \alpha\sum_{l=t-\tau+1}^{t}\mathbf{g}(\boldsymbol{\theta}_l, o_l),
\end{aligned}
\tag{4.69}
$$

where $(*)$ follows by iteratively unfolding the definition of $\mathbf{d}_t$ and noting that $\mathbf{d}_\tau = \tilde{\boldsymbol{\theta}}_\tau - \boldsymbol{\theta}_0 =$

$\mathbf{d}_0 + \alpha \sum_{l=0}^{\tau-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l)$, and $(**)$ follows because the overlapping terms in the sum cancel out. So, we obtain, for all $t \geq 0$,

$$\mathbf{d}_t = \alpha \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l). \tag{4.70}$$

We can now prove Lemma 19, which is key to prove Theorem 9.

Proof of Lemma 19. From (4.70), using the triangle inequality and the bound on the update direction (4.57), we get, recalling that $\sigma \geq \|\boldsymbol{\theta}_0\|$,

$$\begin{aligned} \|\mathbf{d}_t\| &= \|\alpha \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l)\| \\ &\leq +\alpha \sum_{l=t-\tau}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_l, o_l)\| \\ &\leq \alpha L \sum_{l=t-\tau}^{t-1} (\|\boldsymbol{\theta}_l\| + \sigma) \\ &\leq \alpha\tau L\sigma + \alpha L \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|, \end{aligned} \tag{4.71}$$

which proves (i). We now prove (ii). Using the triangle inequality and (4.61),

$$\begin{aligned} \|\mathbf{d}_t\|^2 &= \|\alpha \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l)\|^2 \\ &\leq (\alpha\| \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l)\|)^2 \\ &\leq \alpha^2\| \sum_{l=t-\tau}^{t-1} \mathbf{g}(\boldsymbol{\theta}_l, o_l)\|^2 \\ &\overset{(4.61)}{\leq} \alpha^2\tau \sum_{l=t-\tau}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_l, o_l)\|^2. \end{aligned} \tag{4.72}$$

Now, using the upper bound on the squared gradient norm (4.58),

$$\begin{aligned}\|\mathbf{d}_t\|^2 &\leq \alpha^2\tau \sum_{l=t-\tau}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_l, o_l)\|^2 \\ &\leq 2\alpha^2\tau L^2 \sum_{l=t-\tau}^{t-1} (\|\boldsymbol{\theta}_l\|^2 + \sigma^2) \\ &\leq 2\alpha^2\tau^2 L^2\sigma^2 + 2\alpha^2\tau L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2.\end{aligned} \tag{4.73}$$

which concludes the proof. □

Using the above Lemma, we can now prove Lemma 20.

Proof of Lemma 20. First, recall that, from Lemma 19, we have

$$\|\mathbf{d}_t\|^2 \leq 2\alpha^2\tau^2 L^2\sigma^2 + 2\alpha^2\tau L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2. \tag{4.74}$$

Based on Lemma 33, for $0 \leq i \leq 2\bar{\tau}$, $w_t \leq 2w_{t-i}$ (see (4.67)). Using (4.74),

$$\begin{aligned}\sum_{t=0}^{T} w_t\|\mathbf{d}_t\|^2 &\leq \sum_{t=0}^{T} w_t \left(2\alpha^2\tau^2 L^2\sigma^2 + 2\alpha^2\tau L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2\right) \\ &\leq 2W_T\alpha^2\tau^2 L^2\sigma^2 + 2\alpha^2\tau L^2 \sum_{t=0}^{T} w_t \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2 \\ &\overset{(*)}{\leq} 2W_T\alpha^2\tau^2 L^2\sigma^2 + 4\alpha^2\tau L^2 \sum_{t=0}^{T} \sum_{l=t-\tau}^{t-1} w_l\|\boldsymbol{\theta}_l\|^2 \\ &\overset{(**)}{\leq} 2W_T\alpha^2\tau^2 L^2\sigma^2 + 4\alpha^2\tau^2 L^2 \sum_{t=0}^{T} w_t\|\boldsymbol{\theta}_t\|^2 \\ &\leq 2W_T\alpha^2\tau^2 L^2\sigma^2 + 8\alpha^2\tau^2 L^2 \sum_{t=0}^{T} w_t \left(\|\tilde{\boldsymbol{\theta}}_t\|^2 + \|\mathbf{d}_t\|^2\right) \\ &\leq 2W_T\alpha^2\tau^2 L^2\sigma^2 + 8\alpha^2\tau^2 L^2 \sum_{t=0}^{T} w_t\|\tilde{\boldsymbol{\theta}}_t\|^2 + \frac{1}{2}\sum_{t=0}^{T} w_t\|\mathbf{d}_t\|^2,\end{aligned} \tag{4.75}$$

where for $(*)$ we used the fact that $w_t \leq 2w_l$ for $t - 2\bar{\tau} \leq l \leq t-1$, and for $(**)$ we used the fact

that each element $w_l\|\boldsymbol{\theta}_l\|^2$ appears at most τ times in the sum, for $l = 0, ..., T-1$ (note that, by definition, $\boldsymbol{\theta}_l = 0$ for $l < 0$). In the last inequality, we used $\alpha \leq \frac{1}{4\tau L}$. We can conclude getting

$$\sum_{t=0}^{T} w_t \|\mathbf{d}_t\|^2 \leq 4W_T\alpha^2\tau^2L^2\sigma^2 + 16\alpha^2\tau^2L^2 \sum_{t=0}^{T} w_t\|\tilde{\boldsymbol{\theta}}_t\|^2. \tag{4.76}$$

□

We now prove Lemma 21, that provides a bound on the norm of the gap $\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\|$ and its squared version $\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\|^2$.

Proof of Lemma 21. Inequality (i) of the Lemma can be easily proved by applying the definition of the recursion (4.64),

$$\begin{aligned}
\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\| &\leq \sum_{l=t-\tau_{mix}}^{t-1} \|\tilde{\boldsymbol{\theta}}_{l+1} - \tilde{\boldsymbol{\theta}}_l\| \\
&\leq \alpha \sum_{l=t-\tau_{mix}}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_l, o_l)\| \\
&\leq L\alpha \sum_{l=t-\tau_{mix}}^{t-1} (\|\boldsymbol{\theta}_l\| + \sigma) \\
&= L\alpha\sigma\tau_{mix} + L\alpha \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|.
\end{aligned} \tag{4.77}$$

Similarly, for inequality (ii), note that

$$\begin{aligned}
\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\|^2 &\leq \left(\sum_{l=t-\tau_{mix}}^{t-1} \|\tilde{\boldsymbol{\theta}}_{l+1} - \tilde{\boldsymbol{\theta}}_l\|\right)^2 \\
&\overset{(4.61)}{\leq} \tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} \|\tilde{\boldsymbol{\theta}}_{l+1} - \tilde{\boldsymbol{\theta}}_l\|^2 \\
&\leq \alpha^2 \tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_l, o_l)\|^2 \\
&\leq 2L^2\alpha^2 \tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} (\|\boldsymbol{\theta}_l\|^2 + \sigma^2) \\
&\leq 2L^2\alpha^2 \tau_{mix}^2 \sigma^2 + 2L^2\alpha^2 \tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|^2.
\end{aligned} \tag{4.78}$$

□

We now prove Lemmas 22, 23 and 24, which provide bounds for $\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2$, m_t and $\mathbb{E}[h_t]$, respectively.

Proof of Lemma 22. From (4.58), we have $\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \leq 2L^2(\|\boldsymbol{\theta}_t\|^2 + \sigma^2)$, and so

$$\begin{aligned}
\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 &\leq 2L^2(\|\boldsymbol{\theta}_t\|^2 + \sigma^2) \\
&\leq 2L^2\|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t + \tilde{\boldsymbol{\theta}}_t\|^2 + 2L^2\sigma^2 \\
&\leq 4L^2\|\mathbf{d}_t\|^2 + 4L^2\|\tilde{\boldsymbol{\theta}}_t\|^2 + 2L^2\sigma^2 \\
&\leq 4L^2\|\mathbf{d}_t\|^2 + 4L^2\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^* + \boldsymbol{\theta}^*\|^2 + 2L^2\sigma^2 \\
&\leq 4L^2\|\mathbf{d}_t\|^2 + 8L^2\tilde{r}_t^2 + 8L^2\|\boldsymbol{\theta}^*\|^2 + 2L^2\sigma^2 \\
&\leq 4L^2\|\mathbf{d}_t\|^2 + 8L^2\tilde{r}_t^2 + 10L^2\sigma^2
\end{aligned} \tag{4.79}$$

where we used $\|\boldsymbol{\theta}^*\| \leq \sigma$ and from which we can conclude. □

Proof of Lemma 23. By Cauchy-Schwarz inequality, Lipschitz continuity of $\mathbf{g}(\boldsymbol{\theta}, o_t)$ in $\boldsymbol{\theta}$ (see

(4.56)), and from the definition of $\mathbf{d}_t$, we get

$$\begin{aligned} m_t &= \langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star \rangle \\ &\leq \|\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\| \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\| \\ &\leq L\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\| \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\| \\ &= L\|\mathbf{d}_t\|\tilde{r}_t. \end{aligned} \tag{4.80}$$

Applying Lemma 19 to bound $\|\mathbf{d}_t\|$, we get

$$\begin{aligned} m_t &\leq L(3\alpha\tau L\sigma + \alpha L \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|)\tilde{r}_t \\ &= 3\alpha\tau L^2\sigma\tilde{r}_t + \alpha L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|\tilde{r}_t \\ &\overset{(4.60)}{\leq} 2\alpha\tau L^2\sigma^2 + 2\alpha\tau L^2\tilde{r}_t^2 + \alpha L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2 + \tilde{r}_t^2 \\ &= 2\alpha\tau L^2\sigma^2 + 3\alpha\tau L^2\tilde{r}_t^2 + \alpha L^2 \sum_{l=t-\tau}^{t-1} \|\boldsymbol{\theta}_l\|^2 \\ &\overset{(4.61)}{\leq} 2\alpha\tau L^2\sigma^2 + 3\alpha\tau L^2\tilde{r}_t^2 + 2\alpha L^2 \sum_{l=t-\tau}^{t-1} \|\mathbf{d}_l\|^2 + \|\tilde{\boldsymbol{\theta}}_l\|^2 \\ &\leq 6\alpha\tau L^2\sigma^2 + 3\alpha\tau L^2\tilde{r}_t^2 + 2\alpha L^2 \sum_{l=t-\tau}^{t-1} \|\mathbf{d}_l\|^2 + 4\alpha L^2 \sum_{l=t-\tau}^{t-1} \tilde{r}_l^2. \end{aligned} \tag{4.81}$$

□

Next, we provide the proof of Lemma 24, which provides a bound for $\mathbb{E}[h_t]$, which is the term related to the Markovian sampling and whose analysis requires special care and mixing time arguments.

Proof of Lemma 24. Adding and subtracting $\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}$ in the left hand side of the inner product, we have

$$\begin{aligned} h_t &= \langle \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t) \rangle \\ &= \underbrace{\langle \tilde{\boldsymbol{\theta}}_t - \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t) \rangle}_{T_1} + \underbrace{\langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t) \rangle}_{T_2}, \end{aligned} \tag{4.82}$$

where, using (4.57), Cauchy-Schwarz inequality and Lemma 21,

$$
\begin{aligned}
T_1 &\leq \|\tilde{\boldsymbol{\theta}}_t - \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\|(\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\| + \|\bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\|) \\
&\overset{(4.57)}{\leq} \|\tilde{\boldsymbol{\theta}}_t - \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\| 2L(\|\tilde{\boldsymbol{\theta}}_t\| + \sigma) \\
&\leq 2\alpha L^2 \left(\sigma\tau_{mix} + \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|\right)(\|\tilde{\boldsymbol{\theta}}_t\| + \sigma) \\
&\leq 2\alpha L^2 \sigma\tau_{mix}(\|\tilde{\boldsymbol{\theta}}_t\| + \sigma) + 2\alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|(\|\tilde{\boldsymbol{\theta}}_t\| + \sigma) \\
&\overset{(4.60)}{\leq} 2\alpha L^2\sigma^2\tau_{mix} + 2\alpha L^2\tau_{mix}\sigma\|\tilde{\boldsymbol{\theta}}_t\| \\
&+ 2\alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \frac{1}{2}\|\boldsymbol{\theta}_l\|^2 + \frac{1}{2}(\|\tilde{\boldsymbol{\theta}}_t\| + \sigma)^2 \\
&\overset{(4.61)}{\leq} 2\alpha L^2\sigma^2\tau_{mix} + \alpha L^2\tau_{mix}\sigma^2 + \alpha L^2\tau_{mix}\|\tilde{\boldsymbol{\theta}}_t\|^2 \\
&+ 2\alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \left(\frac{1}{2}\|\boldsymbol{\theta}_l\|^2 + \|\tilde{\boldsymbol{\theta}}_t\|^2 + \sigma^2\right) \\
&\leq 11\alpha L^2\sigma^2\tau_{mix} + 6\alpha L^2\tau_{mix}\tilde{r}_t^2 + \alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|^2.
\end{aligned}
\tag{4.83}
$$

So, taking the expectation,

$$
\mathbb{E}\left[T_1\right] \leq 11\alpha L^2\sigma^2\tau_{mix} + 6\alpha L^2\tau_{mix}\mathbb{E}\left[\tilde{r}_t^2\right] + \alpha L^2 \sum_{l=t-\tau_{mix}}^{t-1} \mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right]. \tag{4.84}
$$

Now, we focus on T_2. Note that, adding and subtracting $\mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t)$ and $\bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}})$ to the right hand side of the inner product, we can write

$$
\begin{aligned}
T_2 &= \langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\rangle \\
&= \bar{T}_1 + \bar{T}_2 + \bar{T}_3
\end{aligned}
\tag{4.85}
$$

with

$$\begin{aligned}\bar{T}_1 &= \langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}})\rangle \\ \bar{T}_2 &= \langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t)\rangle \\ \bar{T}_3 &= \langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\rangle.\end{aligned} \tag{4.86}$$

We first bound $\bar{T}_2$ and $\bar{T}_3$. Note that, using the Lipschitz property of the TD update direction (4.56) and Lemma 21,

$$\begin{aligned}\bar{T}_2 &\leq \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t)\| \\ &\leq \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| L \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\| \\ &\leq L\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t + \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\| \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\| \\ &\leq L\tilde{r}_t \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\| + L\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \tilde{\boldsymbol{\theta}}_t\|^2 \\ &\leq L^2\alpha \left(\sigma\tau_{mix} + \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_l\|\right) \tilde{r}_t + L\left(2L^2\alpha^2\tau_{mix}^2\sigma^2 + 2L^2\alpha^2\tau_{mix}\sum_{l=t-\tau_{mix}}^{t-1}\|\boldsymbol{\theta}_l\|^2\right) \\ &= L^2\alpha\tau_{mix}\frac{1}{2}\left(\sigma^2 + \tilde{r}_t^2\right) + \frac{1}{2}L^2\alpha \sum_{l=t-\tau_{mix}}^{t-1}\|\boldsymbol{\theta}_l\|^2 + \tilde{r}_t^2 \\ &+ 2L^3\alpha^2\tau_{mix}^2\sigma^2 + 2L^3\alpha^2\tau_{mix}\sum_{l=t-\tau_{mix}}^{t-1}\|\boldsymbol{\theta}_l\|^2 \\ &\leq \alpha\tau_{mix}L^2\sigma^2 + \alpha\tau_{mix}L^2\tilde{r}_t^2 + \alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\|\boldsymbol{\theta}_l\|^2,\end{aligned} \tag{4.87}$$

where in the last inequality we used $\alpha \leq \frac{1}{8\tau_{mix}L}$. Taking the expectation,

$$\mathbb{E}\left[\bar{T}_2\right] \leq \alpha\tau_{mix}L^2\sigma^2 + \alpha\tau_{mix}L^2\mathbb{E}\left[\tilde{r}_t^2\right] + \alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right]. \tag{4.88}$$

With the same calculations, we can get

$$\mathbb{E}\left[\bar{T}_3\right] \leq \alpha\tau_{mix}L^2\sigma^2 + \alpha\tau_{mix}L^2\mathbb{E}\left[\tilde{r}_t^2\right] + \alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right]. \tag{4.89}$$

We now proceed to bound $\bar{T}_1$.

$$
\begin{aligned}
\mathbb{E}\left[\bar{T}_1\right] &= \mathbb{E}\left[\langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}})\rangle\right] \\
&= \mathbb{E}\left[\langle \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbb{E}\left[\mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t)|o_{t-\tau}, \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}})\rangle\right] \\
&\leq \mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|\|\mathbb{E}\left[\mathbf{g}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}, o_t)|o_{t-\tau_{mix}}, \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}})\|\right] \\
&\leq \alpha\mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\| + \sigma)\right] \\
&\leq \alpha\mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| + 2\sigma)\right] \\
&\leq \alpha\mathbb{E}\left[\frac{1}{2}\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + \frac{1}{2}(\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| + 2\sigma)^2\right] \\
&\leq \alpha\mathbb{E}\left[\frac{1}{2}\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + \|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + 2\sigma^2\right] \\
&\leq 2\alpha\mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + \sigma^2\right] \\
&\leq 2\alpha\mathbb{E}\left[2\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\|^2 + 2\|\tilde{\boldsymbol{\theta}}_t - \tilde{\boldsymbol{\theta}}_{t-\tau_{mix}}\|^2 + \sigma^2\right] \\
&\leq 2\alpha\mathbb{E}\left[2\tilde{r}_t^2 + 2(2L^2\alpha^2\tau_{mix}^2\sigma^2 + 2L^2\alpha^2\tau_{mix}\sum_{l=t-\tau_{mix}}^{t-1}\|\boldsymbol{\theta}_l\|^2) + \sigma^2\right] \\
&\leq 4\alpha\mathbb{E}\left[\tilde{r}_t^2\right] + 3\alpha\sigma^2 + \alpha\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right].
\end{aligned}
\tag{4.90}
$$

So, we get

$$
\begin{aligned}
\mathbb{E}\left[T_2\right] &= \mathbb{E}\left[\bar{T}_1\right] + \mathbb{E}\left[\bar{T}_2\right] + \mathbb{E}\left[\bar{T}_3\right] \\
&\leq 6\alpha\tau_{mix}L^2\mathbb{E}\left[\tilde{r}_t^2\right] + 5\alpha\tau_{mix}L^2\sigma^2 + 3\alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right].
\end{aligned}
\tag{4.91}
$$

Finally, we get

$$\begin{aligned}
\mathbb{E}\left[h_t\right] &= \mathbb{E}\left[T_1\right] + \mathbb{E}\left[T_2\right] \\
&\leq 16\alpha\tau_{mix}L^2\sigma^2 + 12\alpha\tau_{mix}L^2\sigma^2\mathbb{E}\left[\tilde{r}_t^2\right] + 4\alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\boldsymbol{\theta}_l\|^2\right] \\
&\leq 16\alpha\tau_{mix}L^2\sigma^2 + 12\alpha\tau_{mix}L^2\mathbb{E}\left[\tilde{r}_t^2\right] \\
&+ 8\alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\mathbf{d}_l\|^2\right] + \mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_l\|^2\right] \\
&\leq 32\alpha\tau_{mix}L^2\sigma^2 + 12\alpha\tau_{mix}L^2\mathbb{E}\left[\tilde{r}_t^2\right] + 8\alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right].
\end{aligned} \tag{4.92}$$

□

Proof of Lemma 25. Note that, using (4.58),

$$\begin{aligned}
h_t &= \langle\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\rangle \\
&\leq \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\|\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\| \\
&\overset{(4.60)}{\leq} \frac{1}{2}\tilde{r}_t^2 + \frac{1}{2}\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t) - \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\|^2 \\
&\overset{(4.61)}{\leq} \frac{1}{2}\tilde{r}_t^2 + \|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\|^2 + \|\bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t)\|^2 \\
&\overset{(4.58)}{\leq} \frac{\tilde{r}_t^2}{2} + 2L^2\|\tilde{\boldsymbol{\theta}}_t\|^2 + 2L^2\sigma^2 + 2L^2\|\tilde{\boldsymbol{\theta}}_t\|^2 + 2L^2\sigma^2 \\
&\leq \frac{\tilde{r}_t^2}{2} + 8L^2\tilde{r}_t^2 + 12L^2\sigma^2 \\
&\leq 9L^2\tilde{r}_t^2 + 12L^2\sigma^2.
\end{aligned} \tag{4.93}$$

Now note that

$$\begin{aligned}
\tilde{r}_{t+1}^2 &= \tilde{r}_t^2 + 2\alpha\langle\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*, \mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\rangle + \alpha^2\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\|^2 \\
&\overset{(4.60)}{\leq} \tilde{r}_t^2 + \alpha\tilde{r}_t^2 + \alpha\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\|^2 + \alpha^2\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\|^2 \\
&\overset{\alpha<1}{\leq} (1+\alpha)\tilde{r}_t^2 + 2\alpha\|\mathbf{g}(\tilde{\boldsymbol{\theta}}_t, o_t)\|^2 \\
&\overset{(4.58)}{\leq} (1+\alpha)\tilde{r}_t^2 + 2\alpha(4L^2\tilde{r}_t^2 + 6L^2\sigma^2).
\end{aligned} \tag{4.94}$$

From the last inequality, we can then get, iterating the inequality,

$$\begin{aligned}\tilde{r}_{t+1}^2 &\leq (1+9\alpha L^2)\tilde{r}_t^2 + 12\alpha L^2\sigma^2 \\ &\leq (1+9\alpha L^2)^2\tilde{r}_{t-1}^2 + (1+9\alpha L^2)12\alpha L^2\sigma^2 + 12\alpha L^2\sigma^2 \\ &\leq (1+9\alpha L^2)^{t+1}\tilde{r}_0^2 + 12\alpha L^2\sigma^2\sum_{j=0}^{t}(1+9\alpha L^2)^j.\end{aligned} \tag{4.95}$$

So, for $0 \leq t < \tau_{mix}$,

$$\tilde{r}_{t+1}^2 \leq (1+9\alpha L^2)^{\tau_{mix}}\tilde{r}_0^2 + 12\alpha L^2\sigma^2\sum_{j=0}^{\tau_{mix}}(1+9\alpha L^2)^j. \tag{4.96}$$

Now, given that $L \geq 1$, note that, for $\alpha\tau \leq \frac{1}{8L}$ and $j = 0, ..., \tau - 1$, we have $(1+\alpha)^j \leq (1+\alpha)^\tau \leq e^{\alpha\tau} \leq e^{0.25} \leq 2$. Thus, for $9\alpha L^2 \leq \frac{1}{4\tau_{mix}}$, we get $(1+9\alpha L^2)_{mix}^\tau \leq (1+18\alpha L^2\tau_{mix}) \leq 2$. Hence,

$$\begin{aligned}\tilde{r}_{t+1}^2 &\leq (1+18\alpha L^2\tau_{mix})\tilde{r}_0^2 + 12\alpha L^2\sigma^2\sum_{j=0}^{\tau_{mix}} 2 \\ &\leq 2\tilde{r}_0^2 + 24\alpha L^2\sigma^2\tau_{mix}.\end{aligned} \tag{4.97}$$

Using the last inequality, and noting that $\tilde{r}_0^2 \leq 2\|\boldsymbol{\theta}_0\|^2 + 2\|\boldsymbol{\theta}^*\|^2 \leq 4\sigma^2$.

$$\begin{aligned}\mathbb{E}\left[h_t\right] &\leq 9L^2\tilde{r}_t^2 + 12L^2\sigma^2 \\ &\leq 9L^2(8\sigma^2 + 24\alpha L^2\sigma^2\tau_{mix}) + 12L^2\sigma^2 \\ &\leq 84L^2\sigma^2 + 216\alpha L^2\sigma^2\tau_{mix}.\end{aligned} \tag{4.98}$$

□

Now, we provide the proof of the last auxiliary Lemma needed to prove Theorem 9, i.e., Lemma 26.

Proof of Lemma 26. Plugging $w_t = (1 - a\alpha)^{-(t+1)}$ in Ψ_T, we have

$$\begin{aligned}\Psi_T &= \frac{1}{W_T}\sum_{t=0}^{T}\left(w_t\left(1-a\alpha\right)u_t - w_t u_{t+1}\right) + c \\ &= \frac{1}{W_T}\sum_{t=0}^{T}\left(w_{t-1}u_t - w_t u_{t+1}\right) + c \\ &= \frac{w_{-1}u_0}{W_T} - \frac{w_T u_{T+1}}{W_T} + c\end{aligned} \tag{4.99}$$

□

4.9.2. Proof of Theorem 9

First, we have

$$\tilde{r}_{t+1}^2 = \tilde{r}_t^2 + 2\alpha\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\rangle + \alpha^2\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \tag{4.100}$$

Then, using (4.55), i.e., $\langle \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\rangle \leq -\mu\tilde{r}_t^2$,

$$\begin{aligned}\tilde{r}_{t+1}^2 &= \tilde{r}_t^2 + 2\alpha\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\rangle + \alpha^2\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \\ &= \tilde{r}_t^2 + 2\alpha\langle \bar{\mathbf{g}}(\tilde{\boldsymbol{\theta}}_t), \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\rangle + 2\alpha h_t + 2\alpha m_t + \alpha^2\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \\ &\leq (1 - 2\alpha\mu)\tilde{r}_t^2 + 2\alpha h_t + 2\alpha m_t + \alpha^2\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2.\end{aligned} \tag{4.101}$$

We now apply the inequalities obtained in the auxiliary lemmas of the previous section to bound $\mathbb{E}\left[h_t\right]$, m_t and $\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2$. Recall that $\bar{\tau} = \max\{\tau, \tau_{mix}\}$. Note that from Lemma 24 and 25 we can write $\mathbb{E}\left[h_t\right] \leq \bar{h}_t$, defining

$$\bar{h}_t = \begin{cases} B & \text{if } \ 0 \leq t < \tau_{mix} \\ \\ q_t & \text{if } \ t \geq \tau_{mix} \end{cases}, \tag{4.102}$$

with $B = 48L^2\sigma^2 + 216\alpha L^2\sigma^2\tau_{mix}$, and

$$q_t = \alpha\tau_{mix}L^2\left(32\sigma^2 + 12\mathbb{E}\left[\tilde{r}_t^2\right]\right) + 8\alpha L^2\sum_{l=t-\tau_{mix}}^{t-1}\mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right]. \tag{4.103}$$

As a consequence, we can write, for every $t \geq 0$,

$$\mathbb{E}\left[h_t\right] \leq q_t + \bar{B}_t \tag{4.104}$$

where, in turn,

$$\bar{B}_t = \begin{cases} B & \text{if } \ 0 \leq t < \tau_{mix} \\ 0 & \text{otherwise} \end{cases}. \tag{4.105}$$

Also, recall that, from Lemma 22 and 23, we have

$$\begin{aligned} \|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 &\leq 4L^2\|\mathbf{d}_t\|^2 + 8L^2\tilde{r}_t^2 + 10L^2\sigma^2, \\ m_t &\leq 6\alpha\tau L^2\sigma^2 + 3\alpha\tau L^2\tilde{r}_t^2 + 2\alpha L^2 \sum_{l=t-\tau}^{t-1} \|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2 \end{aligned} \tag{4.106}$$

Combining these inequalities together, we have, for $t \geq 0$,

$$\begin{aligned} \mathbb{E}\left[\tilde{r}_{t+1}^2\right] &\leq (1 - 2\alpha\mu)\mathbb{E}\left[\tilde{r}_t^2\right] + 2\alpha\mathbb{E}\left[h_t\right] + 2\alpha\mathbb{E}\left[m_t\right] + \alpha^2\mathbb{E}\left[\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2\right] \\ &\leq (1 - 2\alpha\mu)\mathbb{E}\left[\tilde{r}_t^2\right] + 2\alpha^2\tau_{mix}L^2\left(32\sigma^2 + 12\mathbb{E}\left[\tilde{r}_t^2\right]\right) \\ &+ 16\alpha^2L^2 \sum_{l=t-\tau_{mix}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] \\ &+ 12\alpha^2\tau L^2\sigma^2 + 6\alpha^2\tau L^2\mathbb{E}\left[\tilde{r}_t^2\right] + 4\alpha^2L^2 \sum_{l=t-\tau}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] \\ &+ 4\alpha^2L^2\mathbb{E}\left[\|\mathbf{d}_t\|^2 + 2\tilde{r}_t^2\right] + 10\alpha^2L^2\sigma^2 + 2\alpha\bar{B}_t. \end{aligned} \tag{4.107}$$

Combining terms, we can get

$$\begin{aligned} \mathbb{E}\left[\tilde{r}_{t+1}^2\right] &\leq (1 - 2\alpha\mu + 48\alpha^2L^2\bar{\tau})\mathbb{E}\left[\tilde{r}_t^2\right] + 128\alpha^2L^2\bar{\tau}\sigma^2 \\ &+ 4\alpha^2L^2\mathbb{E}\left[\|\mathbf{d}_t\|^2\right] + 20\alpha^2L^2 \sum_{l=t-\bar{\tau}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] + 2\alpha\bar{B}_t, \end{aligned} \tag{4.108}$$

where we have used $\tau + \tau_{mix} \leq 2\bar{\tau}$. Multiplying both sides by w_t, we have

$$\begin{aligned} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 2\alpha\mu + 48\alpha^2 L^2 \bar{\tau}) w_t \mathbb{E}\left[\tilde{r}_t^2\right] + 128 w_t \alpha^2 L^2 \bar{\tau} \sigma^2 \\ + 4 w_t \alpha^2 L^2 \mathbb{E}\left[\|\mathbf{d}_t\|^2\right] + 20 w_t \alpha^2 L^2 \sum_{l=t-\bar{\tau}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] + 2 w_t \alpha \bar{B}_t, \end{aligned} \tag{4.109}$$

by summing over $t = 0, ..., T$, we get, with $W_T = \sum_{t=0}^{T} w_t$,

$$\begin{aligned} \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 2\alpha\mu + 48\alpha^2 L^2 \bar{\tau}) \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] \\ + 128 W_T \alpha^2 L^2 \bar{\tau} \sigma^2 + 4\alpha^2 L^2 \underbrace{\sum_{t=0}^{T} w_t \mathbb{E}\left[\|\mathbf{d}_t\|^2\right]}_{p_1} \\ + 20\alpha^2 L^2 \underbrace{\sum_{t=0}^{T} w_t \sum_{l=t-\bar{\tau}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right]}_{p_2} + 2 W_{\tau_{mix}-1} \alpha B \end{aligned} \tag{4.110}$$

Note that, from Lemma 20, we have, picking $\alpha \leq \frac{1}{9\tau L}$,

$$\begin{aligned} p_1 = \sum_{t=0}^{T} w_t \mathbb{E}\left[\|\mathbf{d}_t\|^2\right] &\leq 4 W_T \alpha^2 \tau^2 L^2 \sigma^2 + 16 \alpha^2 \tau^2 L^2 \sum_{t=0}^{T} w_t \mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_t\|^2\right] \\ &\leq 36 W_T \alpha^2 \tau^2 L^2 \sigma^2 + 32 \alpha^2 \tau^2 L^2 \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] \\ &\leq \frac{W_T \sigma^2}{2} + \frac{1}{2} \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right]. \end{aligned} \tag{4.111}$$

Furthermore, using the fact that $w_t \leq 2w_l$ for $l = t - \bar{\tau}, ..., t-1$, we can bound p_2 as follows, using also the above bound on p_1, and picking $\alpha \leq \frac{1}{9\tau L}$,

$$\begin{aligned}
p_2 &= \sum_{t=0}^{T} w_t \sum_{l=t-\bar{\tau}}^{t-1} \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] \\
&\overset{(a)}{\leq} 2\sum_{t=0}^{T} \sum_{l=t-\bar{\tau}}^{t-1} w_l \mathbb{E}\left[\|\mathbf{d}_l\|^2 + 2\tilde{r}_l^2\right] \\
&\overset{(b)}{\leq} 2\bar{\tau}\sum_{t=0}^{T} w_t \mathbb{E}\left[\|\mathbf{d}_t\|^2 + 2\tilde{r}_t^2\right]. \\
&\leq 2\bar{\tau}\sum_{t=0}^{T} w_t \mathbb{E}\left[\|\mathbf{d}_t\|^2\right] + 4\bar{\tau}\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] \\
&\overset{(c)}{\leq} 2\bar{\tau}\left(36W_T\alpha^2\tau^2L^2\sigma^2 + 32\alpha^2\tau^2L^2\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right]\right) \\
&+ 4\bar{\tau}\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] \\
&\leq 5\bar{\tau}\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] + W_T\sigma^2\bar{\tau},
\end{aligned} \tag{4.112}$$

where for (a) we used Lemma 33, for (b) we used the fact that each element $w_l\|\boldsymbol{\theta}_l\|^2$ appears at most τ times in the sum, for $l = 0, ..., T-1$ (note that, by definition, $\mathbf{d}_l = \tilde{r}_l = 0$ for $l < 0$) and for (c) we used the bound on p_1. In the last inequality we used $\alpha \leq \frac{1}{9\tau L}$. Plugging the two bounds on p_1 and p_2 in (4.110), we get

$$\begin{aligned}
\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 2\alpha\mu + 150\alpha^2L^2\bar{\tau})\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] \\
+ 150W_T\alpha^2L^2\bar{\tau}\sigma^2 + 2W_{\tau_{mix}-1}\alpha B.
\end{aligned} \tag{4.113}$$

Now, note that for $\alpha \leq \frac{\mu}{100L^2\bar{\tau}}$, which is such that $(1 - 2\alpha\mu + 150\alpha^2L^2\bar{\tau}) \leq (1 - 0.5\alpha\mu)$, we can re-write (4.113) as

$$\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right] \leq (1 - 0.5\alpha\mu) \sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] + 150 W_T \alpha^2 L^2 \bar{\tau}\sigma^2 + 2W_{\tau_{mix}-1}\alpha B. \tag{4.114}$$

Now, dividing by W_T both sides of (4.114) and bringing $\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right]$ to the right hand side of the inequality, we get

$$0 \leq \frac{1}{W_T} \sum_{t=0}^{T} \left(w_t(1 - 0.5\alpha\mu)\mathbb{E}\left[\tilde{r}_t^2\right] - w_t \mathbb{E}\left[\tilde{r}_{t+1}^2\right]\right) + 150\alpha^2 L^2 \bar{\tau}\sigma^2 + \frac{2W_{\tau_{mix}-1}\alpha B}{W_T}, \tag{4.115}$$

and, recalling that $w_t = (1 - 0.5\alpha\mu)^{-(t+1)}$, we can apply Lemma 26, and get, noting that $w_{-1} = 1$,

$$0 \leq \frac{\mathbb{E}\left[\tilde{r}_0^2\right]}{W_T} - \frac{w_T}{W_T}\mathbb{E}\left[\tilde{r}_{T+1}^2\right] + 150\alpha^2 L^2 \bar{\tau}\sigma^2 + \frac{2W_{\tau_{mix}-1}\alpha B}{W_T}, \tag{4.116}$$

from which we can further obtain

$$\frac{w_T}{W_T}\mathbb{E}\left[\tilde{r}_{T+1}^2\right] \leq \frac{\mathbb{E}\left[\tilde{r}_0^2\right]}{W_T} + 150\alpha^2 L^2 \bar{\tau}\sigma^2 + \frac{2W_{\tau_{mix}-1}\alpha B}{W_T}. \tag{4.117}$$

Note that we can re-write the above inequality as follows

$$\begin{aligned} \mathbb{E}\left[\tilde{r}_{T+1}^2\right] &\leq \frac{1}{w_T}\mathbb{E}\left[\tilde{r}_0^2\right] + 150\frac{W_T}{w_T}\alpha^2 L^2 \bar{\tau}\sigma^2 + \frac{2W_{\tau_{mix}-1}\alpha B}{w_T} \\ &\leq (1 - 0.5\alpha\mu)^{T+1}\mathbb{E}\left[\tilde{r}_0^2\right] + 300\frac{\alpha L^2 \bar{\tau}\sigma^2}{\mu} + \frac{2W_{\tau_{mix}-1}\alpha B}{w_T}, \end{aligned} \tag{4.118}$$

where we used the fact that $\frac{W_T}{w_T} = \sum_{t=0}^{T}(1 - 0.5\alpha\mu)^{-(t+1)}(1 - 0.5\alpha\mu)^{T+1} = \sum_{t=0}^{T}(1 - 0.5\alpha\mu)^t \leq \frac{2}{\alpha\mu}$.

Now note that

$$\begin{aligned}\frac{W_{\tau_{mix}-1}}{w_T} &\leq \frac{W_{\tau_{mix}-1}}{w_0} \\ &= \frac{1}{(1-0.5\alpha\mu)^{-1}} \sum_{t=0}^{\tau_{mix}-1} (1-0.5\alpha\mu)^{-(t+1)} \\ &= \sum_{t=0}^{\tau_{mix}-1} (1-0.5\alpha\mu)^{-t} \\ &\leq \sum_{t=0}^{\tau_{mix}-1} (1-0.5\alpha\mu)^{-\tau_{mix}} \\ &= \tau_{mix}(1-0.5\alpha\mu)^{-\tau_{mix}},\end{aligned} \tag{4.119}$$

and note that, using (4.68), $(1-0.5\alpha\mu)^{-\tau_{mix}} \leq (1+\alpha\mu)^{\tau_{mix}} \leq e^{\alpha\mu\tau_{mix}} \leq e^{0.25} \leq 2$, because $\alpha \leq \frac{1}{4\tau_{mix}} \leq \frac{1}{4\mu\tau_{mix}}$. Hence, we get

$$\frac{W_{\tau_{mix}-1}}{w_T} \leq 2\tau_{mix}. \tag{4.120}$$

Consequently, we can write, for all $T \geq 0$,

$$\mathbb{E}\left[\tilde{r}_{T+1}^2\right] \leq (1-0.5\alpha\mu)^{T+1}\mathbb{E}\left[\tilde{r}_0^2\right] + 300\frac{\alpha L^2\bar{\tau}\sigma^2}{\mu} + 4\alpha\tau_{mix}B \tag{4.121}$$

Also, we have

$$\mathbb{E}\left[r_t^2\right] \leq 2\mathbb{E}\left[\tilde{r}_t^2\right] + 2\mathbb{E}\left[\|\mathbf{d}_t\|^2\right], \tag{4.122}$$

from which we can derive, using the bound on p_1 derived in (4.111),

$$\begin{aligned}\sum_{t=0}^{T} w_t\mathbb{E}\left[r_t^2\right] &\leq 2\sum_{t=0}^{T} w_t\mathbb{E}\left[\tilde{r}_t^2\right] + 2\sum_{t=0}^{T} w_t\mathbb{E}\left[\|\mathbf{d}_t\|^2\right] \\ &\leq 2\sum_{t=0}^{T} w_t\mathbb{E}\left[\tilde{r}_t^2\right] + 2\left(36W_T\alpha^2\tau^2L^2\sigma^2 + 32\alpha^2\tau^2L^2\sum_{t=0}^{T} w_t\mathbb{E}\left[\tilde{r}_t^2\right]\right) \\ &\leq 3\sum_{t=0}^{T} w_t\mathbb{E}\left[\tilde{r}_t^2\right] + 72W_T\alpha^2\tau^2L^2\sigma^2\end{aligned} \tag{4.123}$$

Now, we can get, plugging (4.121) into (4.123),

$$\begin{aligned}
\sum_{t=0}^{T} w_t \mathbb{E}\left[r_t^2\right] &\leq 3\sum_{t=0}^{T} w_t \mathbb{E}\left[\tilde{r}_t^2\right] + 72 W_T \alpha^2 \tau^2 L^2 \sigma^2 \\
&\leq 3\sum_{t=0}^{T} (1-0.5\alpha\mu)^{-1} \mathbb{E}\left[\tilde{r}_0^2\right] + 3W_T 300 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu} \\
&+ 3W_T 4\alpha\tau_{mix} B + 72 W_T \alpha^2 \tau^2 L^2 \sigma^2 \\
&= 3T(1-0.5\alpha\mu)^{-1} \mathbb{E}\left[\tilde{r}_0^2\right] + W_T 900 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu} \\
&+ 12 W_T \alpha \tau_{mix} (84 L^2 \sigma^2 + 216 \alpha L^2 \sigma^2 \tau_{mix}) + 72 W_T \alpha^2 \tau^2 L^2 \sigma^2 \\
&\leq 3T(1-0.5\alpha\mu)^{-1} \mathbb{E}\left[\tilde{r}_0^2\right] + 1945 W_T \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu}
\end{aligned} \tag{4.124}$$

where for the last inequality we used the fact that $\alpha \leq \frac{\mu}{100 L^2 \bar{\tau}}$. Dividing both sides by W_T,

$$\begin{aligned}
\frac{1}{W_T}\sum_{t=0}^{T} w_t \mathbb{E}\left[r_t^2\right] &\leq \frac{3T}{W_T}(1-0.5\alpha\mu)^{-1} \mathbb{E}\left[\tilde{r}_0^2\right] + 1945 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu} \\
&\overset{(*)}{\leq} \frac{3T}{w_T}(1-0.5\alpha\mu)^{-1} r_0^2 + 1945 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu} \\
&\leq 3T(1-0.5\alpha\mu)^{T} r_0^2 + 1945 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu} \\
&\leq 3T \exp\left(-0.5\alpha\mu T\right) r_0^2 + 1945 \frac{\alpha L^2 \bar{\tau} \sigma^2}{\mu},
\end{aligned} \tag{4.125}$$

where for $(*)$ we used the fact that $W_T \geq w_T$ and thus $\frac{1}{W_T} \leq \frac{1}{w_T}$. Choosing the maximum step size $\alpha = \frac{\mu}{100 L^2 \bar{\tau}}$, we get

$$\frac{1}{W_T}\sum_{t=0}^{T} w_t \mathbb{E}\left[r_t^2\right] \leq 3T \exp\left(-0.5 \frac{\mu^2}{100 L^2 \bar{\tau}} T\right) r_0^2 + 20\sigma^2. \tag{4.126}$$

Finally,by definition of $\boldsymbol{\theta}_{out}$ in Theorem 9, we have

$$\mathbb{E}\left[\|\boldsymbol{\theta}_{out} - \boldsymbol{\theta}^*\|^2\right] = \frac{1}{W_T}\sum_{t=0}^{T} w_t \mathbb{E}\left[r_t^2\right] \leq 3T \exp\left(-0.5 \frac{\mu^2}{100 L^2 \bar{\tau}} T\right) r_0^2 + 20\sigma^2. \tag{4.127}$$

□

4.10. Appendix B: Proof of Theorem 10

Let $r_t \triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|$. Define $\tau' = 2\tau_{max} + \tau_{mix}$, and recall

$$\begin{aligned} r_{t,1} &\triangleq \max_{t-\tau' \le l \le t} \mathbb{E}\left[r_l\right] \\ r_{t,2} &\triangleq \max_{t-\tau' \le l \le t} \mathbb{E}\left[r_l^2\right] \end{aligned} \tag{4.128}$$

We start by proving the following bounds on terms of the form $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|^2$, for some $\tau \ge 0$, $t \ge \tau$.

Proof of Lemma 27. Using (4.57), we start by proving (i).

$$\begin{aligned} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\| &\le \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\| = \alpha \sum_{l=t-\tau_{mix}}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_{l-\tau_l}, o_{l-\tau_l})\| \\ &\overset{(4.57)}{\le} \alpha L \sum_{l=t-\tau_{mix}}^{t-1} (\|\boldsymbol{\theta}_{l-\tau_l}\| + \sigma) \\ &\le \alpha L \sum_{l=t-\tau_{mix}}^{t-1} (\|\boldsymbol{\theta}_{l-\tau_l} - \boldsymbol{\theta}^*\| + 2\sigma). \end{aligned} \tag{4.129}$$

Taking the expectation on both sides of the inequality, we get

$$\begin{aligned} \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|\right] &\le \alpha L \sum_{l=t-\tau_{mix}}^{t-1} (\mathbb{E}\left[r_{l-\tau_t}\right] + 2\sigma) \\ &\le \alpha L \sum_{l=t-\tau_{mix}}^{t-1} \max_{t-\tau_{mix}-\tau_{max} \le j \le t} \mathbb{E}\left[r_j\right] + 2\alpha\tau_{mix} L\sigma \\ &\le \alpha\tau_{mix} L(r_{t,1} + 2\sigma). \end{aligned} \tag{4.130}$$

Now, to prove (ii), note that, using (4.61), we can get

$$\begin{aligned}
\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2 &\leq \left(\sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\|\right)^2 \\
&\leq \tau_{mix} \sum_{l=t-\tau_{mix}}^{t-1} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\|^2 \\
&= \tau_{mix}\alpha^2 \sum_{l=t-\tau_{mix}}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_{l-\tau_l}, o_{l-\tau_l})\|^2 \\
&\overset{(4.58)}{\leq} 2\alpha^2\tau_{mix}L^2 \sum_{l=t-\tau_{mix}}^{t-1} (\|\boldsymbol{\theta}_{l-\tau_l}\|^2 + \sigma^2) \\
&\leq 2\alpha^2\tau_{mix}L^2 \sum_{l=t-\tau_{mix}}^{t-1} (2r_{l-\tau_l}^2 + 3\sigma^2).
\end{aligned} \tag{4.131}$$

Taking the expectation on both sides of the inequality, we get

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2\right] &\leq 2\alpha^2\tau_{mix}L^2 \sum_{l=t-\tau_{mix}}^{t-1} (2\mathbb{E}\left[r_{l-\tau_l}^2\right] + 3\sigma^2) \\
&\leq 2\alpha^2\tau_{mix}L^2 \sum_{l=t-\tau_{mix}}^{t-1} (2 \max_{t-\tau_{mix}-\tau_{max}\leq j\leq t} \mathbb{E}\left[r_j^2\right] + 3\sigma^2) \\
&\leq 4\tau_{mix}^2\alpha^2L^2r_{t,2} + 6\alpha^2\tau_{mix}^2L^2\sigma^2 \\
&= 2\alpha^2\tau_{mix}^2L^2(2r_{t,2} + 3\sigma^2).
\end{aligned} \tag{4.132}$$

For (iii), note that

$$\begin{aligned}
\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| &\leq \sum_{l=t-\tau_t}^{t-1} \|\boldsymbol{\theta}_{l+1} - \boldsymbol{\theta}_l\| \\
&= \alpha \sum_{l=t-\tau_t}^{t-1} \|\mathbf{g}(\boldsymbol{\theta}_{l-\tau_l}, o_{l-\tau_l})\| \\
&\overset{(4.57)}{\leq} \alpha L \sum_{l=t-\tau_t}^{t-1} (\|\boldsymbol{\theta}_{l-\tau_l}\| + \sigma) \\
&\leq \alpha L \sum_{l=t-\tau_t}^{t-1} (r_{l-\tau_l} + 2\sigma).
\end{aligned} \tag{4.133}$$

Taking the expectation on both sides of the inequality, we get

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|\right] &\leq \alpha L \sum_{l=t-\tau_t}^{t-1} \left(\mathbb{E}\left[r_{l-\tau_l}\right] + 2\sigma\right) \\ &\leq \alpha L \sum_{l=t-\tau_t}^{t-1} \left(\max_{t-2\tau_{max}\leq j\leq t} \mathbb{E}\left[r_j\right] + 2\sigma\right) \\ &\leq \alpha\tau_t L(r_{t,1} + 2\sigma) \\ &\leq \alpha\tau_{max} L(r_{t,1} + 2\sigma).\end{aligned} \tag{4.134}$$

With computations analogous to the above ones use to get the bounds on $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|$ and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|$, we can conclude getting part (iv) of Lemma 27, i.e.,

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|^2 \leq 2\alpha^2\tau_{max}^2 L^2(2r_{t,2} + 3\sigma^2). \tag{4.135}$$

□

Now, recall the definition of $\mathbf{e}_t$,

$$\mathbf{e}_t \triangleq \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau}, o_{t-\tau}). \tag{4.136}$$

As illustrated in the outline of the analysis in Section 4.6, for the purpose of the analysis, we write the update rule as follows,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \alpha\mathbf{e}_t, \tag{4.137}$$

from which we can write

$$\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2 = J_{t,1} + \alpha^2 J_{t,2} - 2\alpha J_{t,3}, \tag{4.138}$$

with

$$\begin{aligned}J_{t,1} &\triangleq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 \\ J_{t,2} &\triangleq \|\mathbf{e}_t\|^2 \\ J_{t,3} &\triangleq \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle\end{aligned} \tag{4.139}$$

Lemma 34 (bounding $\mathbb{E}[J_{t,1}]$)**.** *Let $t \geq \tau_{mix} + 2\tau_{max}$.*

$$\mathbb{E}[J_{t,1}] \leq (1 - 2\alpha\mu)\mathbb{E}\left[r_t^2\right] + 28\alpha^2\tau_{mix}L^2 r_{t,2} + 34\alpha^2\tau_{mix}L^2\sigma^2 \tag{4.140}$$

Proof.

$$\begin{aligned} J_{t,1} = \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 = r_t^2 + 2\alpha \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle}_{J_{t,11}} \\ + \alpha^2 \underbrace{\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2}_{J_{t,12}}. \end{aligned} \tag{4.141}$$

Note that

$$\begin{aligned} \mathbb{E}[J_{t,12}] &= \mathbb{E}\left[\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2\right] \\ &\leq \mathbb{E}\left[2L^2\left(\|\boldsymbol{\theta}_t\|^2 + \sigma^2\right)\right] \\ &\leq 2L^2\left(2\mathbb{E}\left[r_t^2\right] + 3\sigma^2\right) \\ &\leq 2L^2\left(2r_{t,2} + 3\sigma^2\right) \end{aligned} \tag{4.142}$$

Now note that, using (4.55),

$$\begin{aligned} J_{t,11} &= \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle = -\langle \boldsymbol{\theta}^* - \boldsymbol{\theta}_t, \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle \\ &\quad + \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle \\ &\leq -\mu r_t^2 + \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle}_{T_1'}, \end{aligned} \tag{4.143}$$

where we now omit the dependence on the iterate t in the terms we bound, for notation convenience. Now, note that

$$T_1' = \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle}_{T_{11}'} + \underbrace{\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle}_{T_{12}'}, \tag{4.144}$$

where, using Cauchy-Schwarz inequality and triangle inequality,

$$\begin{aligned} T'_{11} &\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|(\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\| + \|\bar{\mathbf{g}}(\boldsymbol{\theta}_t)\|) \\ &\overset{(4.57)}{\leq} 2L(\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|(\|\boldsymbol{\theta}_t\| + \sigma)) \\ &\overset{(*)}{\leq} L\left(\frac{1}{\alpha\tau_{mix}L}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2 + \alpha\tau_{mix}L(\|\boldsymbol{\theta}_t\| + \sigma)^2\right) \\ &\overset{(4.61)}{\leq} L\left(\frac{1}{\alpha\tau_{mix}L}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2 + 2\alpha\tau_{mix}L(\|\boldsymbol{\theta}_t\|^2 + \sigma^2)\right), \end{aligned} \tag{4.145}$$

where for $(*)$ we used the fact that, from (4.60), we have

$$ab = (\frac{1}{\sqrt{c}}a)(\sqrt{c}b) \leq \frac{1}{2c}a^2 + \frac{cb^2}{2}, \tag{4.146}$$

specifically with $c = \alpha\tau_{mix}L$. Taking the expectation on both sides and applying (ii) of Lemma 27, we get

$$\begin{aligned} \mathbb{E}\left[T'_{11}\right] &\leq L\left(\frac{1}{2\alpha\tau_{mix}L}\mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2\right] + \alpha\tau_{mix}L(2\mathbb{E}\left[r_t^2\right] + 3\sigma^2)\right) \\ &\leq L\left(\frac{2\alpha^2\tau_{mix}^2L^2}{2\alpha\tau_{mix}L}(2r_{t,2} + 3\sigma^2) + \alpha\tau_{mix}L(2r_{t,2} + 3\sigma^2)\right) \\ &= 4\alpha\tau_{mix}L^2r_{t,2} + 6\alpha\tau_{mix}L^2\sigma^2. \end{aligned} \tag{4.147}$$

Now, we proceed to bound $\mathbb{E}\left[T'_{12}\right]$. Note that

$$\begin{aligned} T'_{12} &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle \\ &= \bar{T}_1 + \bar{T}_2 + \bar{T}_3 \end{aligned} \tag{4.148}$$

with

$$\begin{aligned} \bar{T}_1 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau})\rangle \\ \bar{T}_2 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)\rangle \\ \bar{T}_3 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle. \end{aligned} \tag{4.149}$$

We first bound $\bar{T}_2$ and $\bar{T}_3$. Note that, using Lipschitz property of the TD update direction (4.56),

and calculations similar to the ones used to bound $\mathbb{E}[T'_{11}]$, we get

$$\begin{aligned}\bar{T}_2 &\leq \|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)\| \\ &\leq L\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}_t\| \\ &\overset{(4.146)}{\leq} \frac{L^2\alpha\tau_{mix}}{2} r^2_{t-\tau_{mix}} + \frac{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\|^2}{2\alpha\tau_{mix}}.\end{aligned} \tag{4.150}$$

Taking the expectation and applying (ii) of Lemma 27, we can get

$$\begin{aligned}\mathbb{E}\left[\bar{T}_2\right] &\leq \frac{\alpha\tau_{mix}L^2 r_{t,2}}{2} + 2\alpha\tau_{mix}L^2 r_{t,2} + 3\alpha\tau_{mix}L^2\sigma^2 \\ &\leq 3\alpha\tau_{mix}L^2(r_{t,2} + \sigma^2)\end{aligned} \tag{4.151}$$

With the same calculations, we can get

$$\mathbb{E}\left[\bar{T}_3\right] \leq 3\alpha\tau_{mix}L^2(r^2_{t,2} + \sigma^2). \tag{4.152}$$

We now proceed to bound $\bar{T}_1$.

$$\begin{aligned}\mathbb{E}\left[\bar{T}_1\right] &= \mathbb{E}\left[\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\rangle\right] \\ &= \mathbb{E}\left[\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|o_{t-\tau}, \boldsymbol{\theta}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\rangle\right] \\ &\leq \mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|o_{t-\tau_{mix}}, \boldsymbol{\theta}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\|\right] \\ &\overset{(*)}{\leq} \alpha\mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\boldsymbol{\theta}_{t-\tau_{mix}}\| + \sigma)\right] \\ &\leq \alpha\mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| + 2\sigma)\right] \\ &\leq \alpha\mathbb{E}\left[\frac{1}{2}\left(r^2_{t-\tau_{mix}} + 2r^2_{t-\tau_{mix}} + 4\sigma^2\right)\right] \\ &\leq 2\alpha(r_{t,2} + \sigma^2),\end{aligned} \tag{4.153}$$

where for $(*)$ we used Definition 1 of mixing time and the fact that $\sigma \geq 1$. So, putting the above bounds together, we get

$$\mathbb{E}\left[T'_{12}\right] = \mathbb{E}\left[\bar{T}_1\right] + \mathbb{E}\left[\bar{T}_2\right] + \mathbb{E}\left[\bar{T}_3\right] \leq 8\alpha\tau_{mix}L^2(r^2_{t,2} + \sigma^2). \tag{4.154}$$

So, we get

$$\begin{aligned}\mathbb{E}\left[T_1'\right] &= \mathbb{E}\left[T_{11}'\right] + \mathbb{E}\left[T_{12}'\right] \\ &\leq 4\alpha\tau_{mix}L^2 r_{t,2} + 6\alpha\tau_{mix}L^2\sigma^2 + 8\alpha\tau_{mix}L^2(r_{t,2}+\sigma^2) \\ &\leq 12\alpha\tau_{mix}L^2 r_{t,2} + 14\alpha\tau_{mix}L^2\sigma^2\end{aligned} \tag{4.155}$$

so,

$$\mathbb{E}\left[J_{t,11}\right] \leq -\mu\mathbb{E}\left[r_t^2\right] + \mathbb{E}\left[T_1'\right]. \tag{4.156}$$

Hence,

$$\begin{aligned}\mathbb{E}\left[J_{t,1}\right] &= \mathbb{E}\left[r_t^2\right] + 2\alpha\mathbb{E}\left[J_{t,11}\right] + \alpha^2\mathbb{E}\left[J_{t,12}\right] \\ &\leq (1-2\alpha\mu)\mathbb{E}\left[r_t^2\right] + 28\alpha^2\tau_{mix}L^2 r_{t,2} + 34\alpha^2\tau_{mix}L^2\sigma^2,\end{aligned} \tag{4.157}$$

which concludes the proof of the Lemma. □

Lemma 35 (bounding $\mathbb{E}\left[J_{t,2}\right]$)**.** *Let $t \geq \tau_{mix} + 2\tau_{max}$.*

$$\mathbb{E}\left[J_{t,2}\right] \leq 8L^2(2r_{t,2} + 3\sigma^2) \tag{4.158}$$

Proof.

$$\begin{aligned}J_{t,2} = \|\mathbf{e}_t\|^2 &= \|\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})\|^2 \\ &\overset{(4.61)}{\leq} 2\left(\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 + \|\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})\|^2\right) \\ &\overset{(4.58)}{\leq} 2\left(2L^2(\|\boldsymbol{\theta}_t\|^2 + \sigma^2) + 2L^2(\|\boldsymbol{\theta}_{t-\tau_t}\|^2 + \sigma^2)\right) \\ &\leq 4L^2(2r_t^2 + 3\sigma^2 + 2r_{t-\tau_t} + 3\sigma^2).\end{aligned} \tag{4.159}$$

Taking the expectation, we conclude getting

$$\mathbb{E}\left[J_{t,2}\right] = \mathbb{E}\left[\|\mathbf{e}_t\|^2\right] \leq 8L^2(2r_{t,2} + 3\sigma^2) \tag{4.160}$$

□

Lemma 36 (bounding $\mathbb{E}[J_{t,3}]$)**.** *Let $t \geq \tau_{mix} + 2\tau_{max}$.*

$$\mathbb{E}[J_{t,3}] \leq O(\alpha)(\tau_{mix} + \tau_{max})(r_{t,3}^2 + \sigma^2) \tag{4.161}$$

Proof. In the following, we drop the dependence on the iteration t in the terms we bound. We write

$$\begin{aligned} J_{t,3} &= \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle \\ &= \underbrace{\langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta} + \underbrace{\alpha \langle \mathbf{e}_t, \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle}_{\bar{\Delta}}. \end{aligned} \tag{4.162}$$

Note that

$$\begin{aligned} \bar{\Delta} &= \alpha \langle \mathbf{e}_t, \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle \leq \alpha \|\mathbf{e}_t\| \|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\| \\ &\leq O(\alpha)(\|\boldsymbol{\theta}_t\| + \|\boldsymbol{\theta}_{t-\tau}\| + \sigma)(\|\boldsymbol{\theta}_t\| + \sigma) \\ &\leq O(\alpha)(\|\boldsymbol{\theta}_t\|^2 + \|\boldsymbol{\theta}_{t-\tau}\|^2 + \sigma^2) \end{aligned} \tag{4.163}$$

so we get

$$\mathbb{E}\left[\bar{\Delta}\right] \leq O(\alpha)(r_{t,3}^2 + \sigma^2). \tag{4.164}$$

We now proceed to bound Δ.

$$\begin{aligned} \Delta &= \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle = \langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle \\ &= \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_1} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_2} \end{aligned} \tag{4.165}$$

Note that, thanks to the Lipschitz property of the TD direction and with calculations analogous to the ones performed to obtain (4.224), we get

$$\mathbb{E}[\Delta_2] \leq \mathbb{E}[O(\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|) r_t] \leq O(\alpha \tau_{max})(r_{t,3}^2 + \sigma^2). \tag{4.166}$$

We now bound Δ_1.

$$\begin{aligned}\Delta_1 = &\underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t), \boldsymbol{\theta}_t, -\boldsymbol{\theta}^* \rangle}_{\Delta_{11}} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_{12}}\end{aligned} \tag{4.167}$$

Thanks to Lipschitz property of the TD direction and with calculations analogous to the ones performed to obtain (4.224), we get

$$\mathbb{E}\left[\Delta_{11}\right] \leq O(\alpha\tau_{mix})(r_{t,3}^2 + \sigma^2). \tag{4.168}$$

We now proceed to bound Δ_{12}.

$$\begin{aligned}\Delta_{12} = &\underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_1'} \\ &+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_2'}\end{aligned} \tag{4.169}$$

We have

$$\begin{aligned}\Delta_1' = &\underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}), \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^* \rangle}_{\Delta_{11}'} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}), \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}} \rangle}_{\Delta_{12}'}\end{aligned} \tag{4.170}$$

Note that, thanks to the Lipschitz property of the TD direction and with calculations analogous to the ones performed to obtain (4.224), we get

$$\mathbb{E}\left[\Delta_{12}'\right] \leq O(\alpha\tau_{mix})(r_{t,3}^2 + \sigma^2). \tag{4.171}$$

Also note that

$$\begin{aligned}
\mathbb{E}\left[\Delta'_{11}\right] &\leq \mathbb{E}\left[\langle \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|\boldsymbol{\theta}_{t-\tau_{mix}}, o_t\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}), \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\rangle\right] \\
&\leq \mathbb{E}\left[\|\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|\boldsymbol{\theta}_{t-\tau_{mix}}, o_t\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\|\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|\right] \\
&\leq \mathbb{E}\left[O(\alpha)(1 + \|\boldsymbol{\theta}_{t-\tau_{mix}}\|)\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|\right] \\
&\leq \mathbb{E}\left[O(\alpha)(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + \sigma^2)\right] \\
&\leq \mathbb{E}\left[O(\alpha)(r_{t,3}^2 + \sigma^2)\right].
\end{aligned} \tag{4.172}$$

Now note that

$$\begin{aligned}
\Delta'_2 &= \langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle \\
&= \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\Delta'_{21}} \\
&+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\Delta'_{22}}
\end{aligned} \tag{4.173}$$

Thanks to the Lipschitz property of the TD direction and with calculations analogous to the ones performed to obtain (4.224), we get

$$\mathbb{E}\left[\Delta'_{22}\right] \leq O(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}_t\|\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|) \leq O(\alpha\tau_{mix})(r_{t,3}^2 + \sigma^2). \tag{4.174}$$

Now, we write

$$\begin{aligned}
\Delta'_{21} &= \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\bar{\Delta}_1} \\
&+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\bar{\Delta}_2}.
\end{aligned} \tag{4.175}$$

We see that, as before, we can bound $\bar{\Delta}_1$ using the Lipschitz property of the TD direction:

$$\mathbb{E}\left[\bar{\Delta}_1\right] \leq \mathbb{E}\left[O(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}\|\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|)\right] \leq O(\alpha\tau_{max})(r_{t,3}^2 + \sigma^2). \tag{4.176}$$

We write

$$\begin{aligned}\bar{\Delta}_2 = &\underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{21}} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{22}}.\end{aligned} \tag{4.177}$$

Now note that, with calculations analogous to the ones performed to obtain (4.224),

$$\begin{aligned}\mathbb{E}\left[\bar{\Delta}_{22}\right] &\leq \mathbb{E}\left[O(\|\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} - \boldsymbol{\theta}_{t-\tau_{mix}}\| \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|)\right] \\ &\leq \mathbb{E}\left[O(\alpha\tau_{max})(r_{t,3}^2 + \sigma^2)\right]\end{aligned} \tag{4.178}$$

Now note that

$$\begin{aligned}\bar{\Delta}_{21} = &\underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{211}} \\ &+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} \rangle}_{\bar{\Delta}_{212}}.\end{aligned} \tag{4.179}$$

With calculations analogous to the ones performed to obtain (4.224), we get

$$\begin{aligned}\mathbb{E}\left[\bar{\Delta}_{212}\right] &\leq \mathbb{E}\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}\| O\left(\|\bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t})\| + \|\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t})\|\right)\right] \\ &\leq O(\alpha)(\tau_{mix} + \tau_{max})(r_{t,3}^2 + \sigma^2).\end{aligned} \tag{4.180}$$

Now,

$$\begin{aligned}\mathbb{E}\left[\bar{\Delta}_{211}\right] &= \mathbb{E}\left[\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}) - \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t})\right], \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} - \boldsymbol{\theta}^* \rangle\right] \\ &\leq \mathbb{E}\left[\|\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t}) | \boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}, o_{t-\tau_t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t})\| \right. \\ &\quad \left. \|\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} - \boldsymbol{\theta}^*\|\right] \\ &\leq O(\alpha)\mathbb{E}\left[O(\|\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t}\| + \sigma)\|\boldsymbol{\theta}_{t-\tau_{mix}-\tau_t} - \boldsymbol{\theta}^*\|\right] \\ &\leq O(\alpha)(r_{t,3}^2 + \sigma^2).\end{aligned} \tag{4.181}$$

So, $\mathbb{E}\left[T_3\right]$ can be upper bounded by a sum of terms that are upper bounded by either $O(\alpha)(r_{t,3}^2 + \sigma^2)$,

$O(\alpha\tau_{max})(r_{t,3}^2+\sigma^2)$, $O(\alpha\tau_{mix})(r_{t,3}^2+\sigma^2)$ or $O(\alpha)(\tau_{mix}+\tau_{max})(r_{t,3}^2+\sigma^2)$. So, we can conclude that

$$\mathbb{E}\left[T_3\right] \leq O(\alpha)(\tau_{mix}+\tau_{max})(r_{t,3}^2+\sigma^2) \tag{4.182}$$

□

We can now prove the following convergence result for asynchronous TD learning:

Theorem 12. *For $\alpha > 0$ small enough and $T \geq 2\tau_{mix}+\tau_{max}$,*

$$\mathbb{E}\left[r_T^2\right] \leq \rho^T + \epsilon, \tag{4.183}$$

with

$$\begin{aligned}\rho &= (1-2\alpha(1-\gamma)\omega + C_1\alpha^2(\tau_{mix}+\tau_{max}))^{\frac{1}{1+\tau_{mix}+\tau_{max}}} \\ \epsilon &= \frac{C_2\alpha^2(\tau_{mix}+\tau_{max})\sigma^2}{2\alpha(1-\gamma)\omega - C_1\alpha^2(\tau_{mix}+\tau_{max})}\end{aligned} \tag{4.184}$$

Proof. Putting together Lemma 40, 41 and 42, we get

$$\begin{aligned}\mathbb{E}\left[r_{t+1}^2\right] &= \underbrace{\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2}_{T_1} + \alpha^2 \underbrace{\|\mathbf{e}_t\|^2}_{T_2} \\ &\quad - 2\alpha \underbrace{\langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha\mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle}_{T_3} \\ &\leq (1-2\alpha(1-\gamma)\omega)\mathbb{E}\left[r_t^2\right] \\ &\quad + O(\alpha^2)(\tau_{mix}+\tau_{max})r_{t,3}^2 + O(\alpha^2)(\tau_{mix}+\tau_{max})\sigma^2 \\ &\leq \underbrace{(1-2\alpha(1-\gamma)\omega)}_{p}\mathbb{E}\left[r_t^2\right] \\ &\quad + \underbrace{O(\alpha^2)(\tau_{mix}+\tau_{max})}_{q} \max_{t-2\tau_{max}-\tau_{mix}\leq l\leq t}\mathbb{E}\left[r_l^2\right] \\ &\quad + \underbrace{O(\alpha^2)(\tau_{mix}+\tau_{max})\sigma^2}_{r}.\end{aligned} \tag{4.185}$$

For α sufficiently small, we have that $p+q<1$ and for $T \geq \tau_{mix}+2\tau_{max}$ we get, for $C_1, C_2 > 0$

absolute constants,

$$\mathbb{E}\left[r_T^2\right] \leq \rho^T r_0^2 + \epsilon, \tag{4.186}$$

with

$$\rho = (1 - 2\alpha(1-\gamma)\omega + C_1\alpha^2(\tau_{mix} + \tau_{max}))^{\frac{1}{1+\tau_{mix}+\tau_{max}}} \tag{4.187}$$

and

$$\epsilon = \frac{C_2\alpha^2(\tau_{mix} + \tau_{max})\sigma^2}{2\alpha(1-\gamma)\omega - C_1\alpha^2(\tau_{mix} + \tau_{max})} \tag{4.188}$$

□

4.11. Appendix C: Proof of Theorem 11

In this section, we consider following update rule

$$\boldsymbol{\theta}_{t+1} = \begin{cases} \boldsymbol{\theta}_t + \alpha\left(\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})\right) & \text{if} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| \leq \frac{\epsilon}{4} \\ \boldsymbol{\theta}_t & \text{otherwise} \end{cases} \tag{4.189}$$

Assumption 9. *For all $\boldsymbol{\theta}, \boldsymbol{\theta}^{'} \in \boldsymbol{\theta}$ and $o, o' \in O$*

$$\|\mathbf{g}(\boldsymbol{\theta}, o)\| \leq L\|\boldsymbol{\theta}\| + \sigma, \tag{4.190}$$

and

$$\|\mathbf{g}(\boldsymbol{\theta}, o) - \mathbf{g}(\boldsymbol{\theta}, o')\| \leq L\|\boldsymbol{\theta}\| + \sigma, \tag{4.191}$$

and

$$\|\mathbf{g}(\boldsymbol{\theta}, o) - \bar{\mathbf{g}}(\boldsymbol{\theta})\| \leq L\|\boldsymbol{\theta}\| + \sigma, \tag{4.192}$$

and

$$\|\mathbf{g}(\boldsymbol{\theta}, o) - \mathbf{g}(\boldsymbol{\theta}^{'}, o)\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta}^{'}\|. \tag{4.193}$$

We also define $\beta = L\epsilon + \sigma$ and assume $\|\boldsymbol{\theta}^*\| \leq \sigma$.

Lemma 37. *Let τ_{avg} be the average of delay. Then if the algorithm fails, then the number of updates that it makes is at least $\frac{T}{4(\tau_{avg}+1)}$.*

Proof. Consider $U_{2\tau_{avg}}$, the number of steps t for which the delay τ_t is at least $2\tau_{avg}$. We must have $U_{2\tau_{avg}} \leq \frac{T}{2}$ (otherwise the total sum of delays exceeds $\tau_{avg}T$, contradicting the definition of τ_{avg}). On the other hand, let k be the number of updates that the algorithm makes. Let $t_1 < t_2 < \ldots < t_k$ be the steps in which an update is made. Denote $t_0 = 0$ and $t_{k+1} = T$. Now, fix i and consider the steps at times $s_n = t_i + n$ for $n \in [1, 2, \ldots, t_{i+1} - t_i - 1]$. In all those steps no update takes place and $\boldsymbol{\theta}_{s_n} = \boldsymbol{\theta}_{t_i}$. We must have $\tau_{s_n} > n$ for all n (otherwise $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-t_{\tau_t}}$ for $t = s_n$ and an update occurs). In particular we have that $\tau_{s_n} \geq 2\tau_{avg}$ in at least $t_{i+1} - t_i - 1 - 2\tau_{avg}$ steps

$$\begin{aligned} &\#\text{ of steps in } [t_i, t_{i+1}] \text{ with delay bigger or equal to } 2\tau_{avg} && (4.194)\\ &\geq \max\{0, t_{i+1} - t_i - 1 - 2\tau_{avg}\} && (4.195)\\ &\geq t_{i+1} - t_i - 1 - 2\tau_{avg}. && (4.196) \end{aligned}$$

Hence,

$$\begin{aligned} U_{2\tau_{avg}} &\geq \sum_{i=0}^{k-1} (t_{i+1} - t_i - 1 - 2\tau_{avg}) \\ &= T - k(1 + 2\tau_{avg}). \end{aligned}$$

Finally, it follows that $T - k(1 + 2\tau_{avg}) \leq \frac{T}{2}$ which implies $k \geq \frac{T}{4(\tau_{avg}+1)}$. □

Lemma 38. *Suppose $\tau_t \leq \tau_{max}$ and for all $t \geq 0$. Then, for any $\tau \geq 1$ and $t \geq \tau$, we have*

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq 4\alpha\tau\|\boldsymbol{\theta}_t\| + \alpha\tau c, \tag{4.197}$$

where $c = (4L + 2)\beta = (4L + 2)(L\epsilon + \sigma)$.

Proof. Let $t^{'} = t - \tau_t$ and $I_t = 1$, then

$$\begin{aligned}
\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| &\leq \alpha I_t \|\mathbf{g}(\boldsymbol{\theta}_{t'}, o_{t'})\| \\
&\leq \alpha I_t (L\|\boldsymbol{\theta}_{t'}\| + \sigma) \\
&\leq \alpha I_t (L\|\boldsymbol{\theta}_t\| + L\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t'}\| + \sigma) \\
&\leq \alpha I_t (L\|\boldsymbol{\theta}_t\| + L\epsilon + \sigma) \\
&\leq \alpha L\|\boldsymbol{\theta}_t\| + \alpha\beta,
\end{aligned} \tag{4.198}$$

from which we get

$$\|\boldsymbol{\theta}_{t+1}\| \leq (1 + \alpha L)\|\boldsymbol{\theta}_t\| + \alpha\beta \tag{4.199}$$

By recursively using the above inequality, we have for all $t \geq \tau$

$$\|\boldsymbol{\theta}_t\| \leq (1 + \alpha L)\|\boldsymbol{\theta}_{t-1}\| + \alpha\beta \tag{4.200}$$

$$\leq (1 + \alpha L)^2\|\boldsymbol{\theta}_{t-2}\| + \alpha\beta(1 + \alpha L) + \alpha\beta \tag{4.201}$$

$$\leq (1 + \alpha L)^\tau\|\boldsymbol{\theta}_{t-\tau}\| + \alpha\beta\sum_{j=0}^{\tau-1}(1 + \alpha L)^j \tag{4.202}$$

$$= (1 + 2\alpha\tau L)\|\boldsymbol{\theta}_{t-\tau}\| + \alpha\beta\left(\frac{1 + 2\alpha L\tau - 1}{\alpha}\right) \tag{4.203}$$

$$\leq 2\|\boldsymbol{\theta}_{t-\tau}\| + 2\alpha L\beta\tau \tag{4.204}$$

where we used $\alpha\tau L \leq \frac{1}{4}$ and the fact that $(1 + x)^\tau \leq 1 + 2x\tau$ for $x\tau \leq \frac{1}{4}$.

Now,

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq \sum_{j=t-\tau}^{t-1}\|\boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j\| \tag{4.205}$$

$$\leq \sum_{j=t-\tau}^{t-1}\alpha L\|\boldsymbol{\theta}_j\| + \alpha\beta \tag{4.206}$$

$$\leq \alpha\tau(2\|\boldsymbol{\theta}_{t-\tau}\| + 2L\alpha\beta\tau) + \alpha\beta\tau \tag{4.207}$$

$$\leq 2\alpha\tau\|\boldsymbol{\theta}_{t-\tau}\| + (2L + 1)\alpha\beta\tau, \tag{4.208}$$

where in the last ineq. we used the fact that $\alpha\tau \leq \frac{1}{4}$. Moreover,

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq 2\alpha\tau\|\boldsymbol{\theta}_{t-\tau}\| + (2L+1)\alpha\beta\tau \tag{4.209}$$

$$\leq 2\alpha\tau\|\boldsymbol{\theta}_t\| + (2L+1)\alpha\beta\tau + 2\alpha\tau\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \tag{4.210}$$

$$\leq 2\alpha\tau\|\boldsymbol{\theta}_\tau\| + (2L+1)\alpha\beta\tau + \frac{1}{2}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\|, \tag{4.211}$$

which results in

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq 4\alpha\tau\|\boldsymbol{\theta}_t\| + (4L+2)\alpha\beta\tau, \tag{4.212}$$

then, let $c = (4L+2)\beta$, then

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau}\| \leq 4\alpha\tau\|\boldsymbol{\theta}_t\| + \alpha c\tau. \tag{4.213}$$

□

Lemma 39. *Let* $\mathbf{e}_t = \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t})$. *Then if* $I_t = 1$,

$$\|\mathbf{e}_t\| \leq L\epsilon + L\|\boldsymbol{\theta}_t\| + \sigma \leq L\|\boldsymbol{\theta}_t\| + \beta. \tag{4.214}$$

Proof. First, we can write

$$\|\mathbf{e}_t\| = \leq \|\mathbf{g}(\boldsymbol{\theta}_{t'}, o_{t'}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t'})\| + \|\mathbf{g}(\boldsymbol{\theta}_t, o_{t'}) - \mathbf{g}(\boldsymbol{\theta}_t, o_t)\| \tag{4.215}$$

$$\leq L\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t'}\| + L\|\boldsymbol{\theta}_t\| + \sigma \tag{4.216}$$

$$\leq L\epsilon + L\|\boldsymbol{\theta}_t\| + \sigma \tag{4.217}$$

□

4.11.1. Main Analysis

If $I_t = 1$, we have:

$$\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|^2 = \underbrace{\|\boldsymbol{\theta}_t + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \boldsymbol{\theta}^*\|^2}_{T_1} + \underbrace{\alpha^2 \|\mathbf{e}_t\|^2}_{T_2} \underbrace{-2\alpha \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle}_{T_3} \tag{4.218}$$

Lemma 40 (bounding $\mathbb{E}[T_1]$)**.** *Let $t \geq \tau_{mix}$.*

$$\mathbb{E}[T_1] \leq (1 - 2\alpha(1-\gamma)\omega)\mathbb{E}\left[r_t^2\right] + O(\alpha^2 \tau_{mix})(r_t^2 + \beta^2) \tag{4.219}$$

Proof.

$$\begin{aligned} T_1 = \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2 = r_t^2 + 2\alpha \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle}_{I_1} \\ + \alpha^2 \underbrace{\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2}_{I_2}. \end{aligned} \tag{4.220}$$

Note that

$$\mathbb{E}[I_2] = \mathbb{E}\left[\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\|^2\right] \leq 2M^2 \mathbb{E}\left[\|\boldsymbol{\theta}_t\|^2\right] + 2\sigma^2. \tag{4.221}$$

Now note that

$$\begin{aligned} I_1 &= \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) \rangle = -\langle \boldsymbol{\theta}^* - \boldsymbol{\theta}_t, \bar{\mathbf{g}}(\boldsymbol{\theta}_t) \rangle \\ &\quad + \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t) \rangle \\ &\leq -\omega(1-\gamma) r_t^2 + \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t) \rangle}_{T_1'}. \end{aligned} \tag{4.222}$$

Now,

$$T_1' = \underbrace{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t) \rangle}_{T_{11}'} + \underbrace{\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t) \rangle}_{T_{12}'}, \tag{4.223}$$

where

$$\begin{aligned} T'_{11} &\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\| \|\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\| \\ &\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_{mix}}\| (2\|\boldsymbol{\theta}_t\| + 2\sigma) \\ &\leq (4\alpha\tau_{mix}\|\boldsymbol{\theta}_t\| + 4\alpha\tau_{mix}\beta)(2\|\boldsymbol{\theta}_t\| + 2\sigma) \\ &\leq 8\alpha\tau_{mix}\|\boldsymbol{\theta}_t\|^2 + 8\alpha\beta\tau_{mix}\sigma + (8\alpha\tau_{mix}\sigma + 8\alpha\tau_{mix}\beta)\|\boldsymbol{\theta}_t\|, \end{aligned} \tag{4.224}$$

so, taking the expectation,

$$\mathbb{E}\left[T'_{11}\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.225}$$

Now,

$$\begin{aligned} T'_{12} &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle \\ &= \bar{T}_1 + \bar{T}_2 + \bar{T}_3 \end{aligned} \tag{4.226}$$

with

$$\begin{aligned} \bar{T}_1 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\rangle \\ \bar{T}_2 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_t, o_t) - g(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)\rangle \\ \bar{T}_3 &= \langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}}) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t)\rangle. \end{aligned} \tag{4.227}$$

We first bound $\bar{T}_2$ and $\bar{T}_3$. Note that, using Lipschitz property of the TD update direction,

$$\begin{aligned} \bar{T}_2 &\leq \|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)\| \\ &\leq \|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| O(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}_t\|). \end{aligned} \tag{4.228}$$

Taking the expectation and applying manipulations analogous to the ones performed to obtain (4.225), we get

$$\mathbb{E}\left[\bar{T}_2\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.229}$$

With the same calculations, we can get

$$\mathbb{E}\left[\bar{T}_3\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.230}$$

We now proceed to bound $\bar{T}_1$.

$$\begin{aligned}
\mathbb{E}\left[\bar{T}_1\right] &= \mathbb{E}\left[\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\rangle\right] \\
&= \mathbb{E}\left[\langle \boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*, \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|o_{t-\tau_{mix}}, \boldsymbol{\theta}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\rangle\right] \\
&\leq \mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| \|\mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t-\tau_{mix}}, o_t)|o_{t-\tau_{mix}}, \boldsymbol{\theta}_{t-\tau_{mix}}\right] - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_{mix}})\|\right] \\
&\leq O(\alpha)\mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\boldsymbol{\theta}_{t-\tau_{mix}}\| + \sigma)\right] \\
&\leq O(\alpha)\mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|(\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\| + \sigma)\right] \\
&\leq O(\alpha)\mathbb{E}\left[\|\boldsymbol{\theta}_{t-\tau_{mix}} - \boldsymbol{\theta}^*\|^2 + \sigma^2\right] \\
&\leq O(\alpha)(\mathbb{E}\left[r_t^2\right] + \beta^2).
\end{aligned} \tag{4.231}$$

So, we get

$$\mathbb{E}\left[T'_{12}\right] = \mathbb{E}\left[\bar{T}_1\right] + \mathbb{E}\left[\bar{T}_2\right] + \mathbb{E}\left[\bar{T}_3\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.232}$$

So, we get

$$\mathbb{E}\left[T'_1\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2), \tag{4.233}$$

so,

$$\mathbb{E}\left[I_1\right] \leq -\omega(1-\gamma)\mathbb{E}\left[r_t^2\right] + \mathbb{E}\left[T'_1\right]. \tag{4.234}$$

Hence,

$$\begin{aligned}
\mathbb{E}\left[T_1\right] &= \mathbb{E}\left[r_t^2\right] + 2\alpha\mathbb{E}\left[I_1\right] + \alpha^2\mathbb{E}\left[I_2\right] \\
&\leq (1 - 2\alpha(1-\gamma)\omega)\mathbb{E}\left[r_t^2\right] + O(\alpha^2\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2)
\end{aligned} \tag{4.235}$$

□

Lemma 41 (bounding $\mathbb{E}\left[T_2\right]$)**.** *Let $t \geq \tau_{mix}$.*

$$\mathbb{E}\left[T_2\right] \leq O(\alpha^2(\mathbb{E}\left[r_t^2\right] + \beta^2)) \tag{4.236}$$

Proof. The lemma follows directly from lemma 3. □

Lemma 42 (bounding $\mathbb{E}[T_3]$)**.** *Let $t \geq \tau_{mix} + \tau_{max}$.*

$$\mathbb{E}[T_3] \leq O(2\alpha(\alpha\tau_{mix}(\mathbb{E}\left[r_t^2\right] + \beta^2)) \tag{4.237}$$

Proof. We write

$$\begin{aligned} T_3 &= \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle \\ &= \underbrace{\langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\Delta} + \underbrace{\alpha\langle \mathbf{e}_t, \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle}_{\bar{\Delta}}. \end{aligned} \tag{4.238}$$

Note that

$$\begin{aligned} \bar{\Delta} &= \alpha\langle \mathbf{e}_t, \mathbf{g}(\boldsymbol{\theta}_t, o_t)\rangle \leq \alpha\|\mathbf{e}_t\|\|\mathbf{g}(\boldsymbol{\theta}_t, o_t)\| \\ &\leq O(\alpha)(2\|\boldsymbol{\theta}_k\| + \beta)(\|\boldsymbol{\theta}_t\| + \sigma) \\ &\leq O(\alpha)(\|\boldsymbol{\theta}_t\|^2 + \beta^2) \end{aligned} \tag{4.239}$$

so we get

$$\mathbb{E}\left[\bar{\Delta}\right] \leq O(\alpha)(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.240}$$

We now proceed to bound Δ.

$$\begin{aligned} \Delta &= \langle \mathbf{e}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle = \langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle \\ &= \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t) + \bar{\mathbf{g}}(\boldsymbol{\theta}_t) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\Delta_1} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^*\rangle}_{\Delta_2} \end{aligned} \tag{4.241}$$

We have

$$\Delta_2 \leq O(\epsilon r_t) \leq O(\alpha(r_t^2 + (\frac{\epsilon}{\alpha})^2)) \tag{4.242}$$

if $\epsilon \leq \alpha$, we have

$$\Delta_2 \leq O(\alpha(r_t^2 + 1)) \tag{4.243}$$

and

$$\Delta_1 \leq \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_t, o_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_t), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_{11}} + \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_t) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_{12}}. \tag{4.244}$$

Note that $\Delta_{11} = T_1'$ above, and therefore

$$\mathbb{E}\left[\Delta_{11}\right] \leq O(\alpha\tau_{mix})(\mathbb{E}\left[r_t^2\right] + \beta^2). \tag{4.245}$$

We now bound $\mathbb{E}\left[\Delta_{12}\right]$.

$$\begin{aligned}
\Delta_{12} &= \langle \bar{\mathbf{g}}(\boldsymbol{\theta}_t) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle \\
&= \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_1'} \\
&+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_t) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_2'}.
\end{aligned} \tag{4.246}$$

Note that, using Lemma 38 and the fact that $\epsilon \leq \alpha$,

$$\begin{aligned}
\Delta_2' &\leq O(\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\|)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\| \\
&\leq O(\epsilon r_t) \\
&\leq O\left(\frac{\epsilon^2}{\alpha} + \alpha r_t^2\right) \\
&\leq O(\alpha)(1 + r_t^2) \\
&\leq O(\alpha)(r_t^2 + \beta^2).
\end{aligned} \tag{4.247}$$

We now bound $\mathbb{E}\left[\Delta_1'\right]$,

$$\begin{aligned}\Delta_1' &= \langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle \\ &= \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_{11}'} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\Delta_{12}'}.\end{aligned} \tag{4.248}$$

Note that, using Lemma 38 and the same calculations used to get (4.247),

$$\begin{aligned}\Delta_{12}' &\leq \|\mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_t, o_{t-\tau_t})\| \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\| \\ &\leq O(\|\boldsymbol{\theta}_{t-\tau_t} - \boldsymbol{\theta}_t\|) \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\| \\ &\leq O(\alpha)(r_t^2 + \beta^2).\end{aligned} \tag{4.249}$$

Note that

$$\begin{aligned}\Delta_{11}' &= \langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle \\ &= \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t}) - \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_1} \\ &+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_2}.\end{aligned} \tag{4.250}$$

Now, using Lemma 38,

$$\begin{aligned}\bar{\Delta}_1 &\leq O\left(\|\boldsymbol{\theta}_{t-\tau_t} - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\|\right) \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\| \\ &\leq O(\alpha\tau_{mix})(\|\boldsymbol{\theta}_{t-\tau_t}\| + \beta) r_t \\ &\leq O(\alpha\tau_{mix})(\|\boldsymbol{\theta}_t\| + \epsilon + \beta) r_t \\ &\leq O(\alpha\tau_{mix})(r_t + \beta) r_t \\ &\leq O(\alpha\tau_{mix})(r_t^2 + \beta^2).\end{aligned} \tag{4.251}$$

Now, note that

$$\begin{aligned}\bar{\Delta}_2 = &\underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{21}} \\ &+ \underbrace{\langle \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}, o_{t-\tau_t}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{22}}.\end{aligned} \tag{4.252}$$

Note that, using the Lipschitz property of the TD update direction and Lemma 38, with calculations

analogous to the ones done to get (4.251),

$$\mathbb{E}\left[\bar{\Delta}_{22}\right] \leq O(\alpha\tau_{mix})(r_t^2 + \beta^2). \tag{4.253}$$

Now, we write

$$\begin{aligned}\bar{\Delta}_{21} = &\underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}} - \boldsymbol{\theta}^* \rangle}_{\bar{\Delta}_{211}} \\ &+ \underbrace{\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}, o_{t-\tau_t}), \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}} \rangle}_{\bar{\Delta}_{212}}.\end{aligned} \tag{4.254}$$

Now, we first bound $\bar{\Delta}_{212}$, using Lemma 38

$$\begin{aligned}\bar{\Delta}_{212} &\leq \|\bar{\mathbf{g}}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}) - \mathbf{g}(\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}, o_{t-\tau_t})\| \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\| \\ &\leq O(\|\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\| + \sigma)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\| \\ &\leq O(\|\boldsymbol{\theta}_t\| + \|\boldsymbol{\theta}_{t-\tau_t-\tau_{mix}} - \boldsymbol{\theta}_t\| + \sigma)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\|\end{aligned} \tag{4.255}$$

and note that, from (4.189), Lemma 38 and the fact that $\epsilon \leq \alpha$

$$\begin{aligned}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\| &\leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-\tau_t}\| + \|\boldsymbol{\theta}_{t-\tau_t} - \boldsymbol{\theta}_{t-\tau_t-\tau_{mix}}\| \\ &\leq \epsilon + O(\alpha\tau_{mix})(\|\boldsymbol{\theta}_{t-\tau_t}\| + \beta) \\ &\leq \epsilon + O(\alpha\tau_{mix})(\|\boldsymbol{\theta}_t\| + \epsilon + \beta) \\ &\leq \epsilon + O(\alpha\tau_{mix})(r_t + \beta) \\ &\leq O(\alpha\tau_{mix})(r_t + \beta).\end{aligned} \tag{4.256}$$

Hence, given that $\sigma \leq \beta$ and $\alpha\tau_{mix} \leq 1$,

$$\begin{aligned}\bar{\Delta}_{212} &\leq O(r_t + O(\alpha\tau_{mix})(r_t + \beta) + \sigma)O(\alpha\tau_{mix})(r_t + \beta) \\ &= O(\alpha^2\tau_{mix}^2)(r_t + \beta)^2 + O(\alpha\tau_{mix})(r_t + \beta)^2 \\ &\leq O(\alpha\tau_{mix})(r_t^2 + \beta^2).\end{aligned} \tag{4.257}$$

To conclude, we bound $\mathbb{E}\left[\bar{\Delta}_{211}\right]$. We use (mixing time), the fact that $\alpha\tau_{mix} \leq 1$ and (4.256). For notation convenience, define $t' \triangleq t - \tau_t - \tau_{mix}$,

$$\begin{aligned}
\mathbb{E}\left[\bar{\Delta}_{211}\right] &= \mathbb{E}\left[\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t'}) - \mathbf{g}(\boldsymbol{\theta}_{t'}, o_{t-\tau_t}), \boldsymbol{\theta}_{t'} - \boldsymbol{\theta}^* \rangle\right] \\
&= \mathbb{E}\left[\langle \bar{\mathbf{g}}(\boldsymbol{\theta}_{t'}) - \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t'}, o_{t-\tau_t}) | \boldsymbol{\theta}_{t'}, o_{t'}\right], \boldsymbol{\theta}_{t'} - \boldsymbol{\theta}^* \rangle\right] \\
&\leq \mathbb{E}\left[\|\bar{\mathbf{g}}(\boldsymbol{\theta}_{t'}) - \mathbb{E}\left[\mathbf{g}(\boldsymbol{\theta}_{t'}, o_{t-\tau_t})\|\right] \|\boldsymbol{\theta}_{t'} - \boldsymbol{\theta}^*\|\right] \\
&\leq \mathbb{E}\left[O(\alpha)(\|\boldsymbol{\theta}_{t'}\| + \sigma)(\|\boldsymbol{\theta}_{t'} - \boldsymbol{\theta}_t\| + \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|)\right] \\
&\leq O(\alpha)\mathbb{E}\left[(r_t + \|\boldsymbol{\theta}_{t'} - \boldsymbol{\theta}_t\| + \sigma)(r_t + \beta)\right] \\
&\leq O(\alpha)(\mathbb{E}\left[r_t^2\right] + \beta^2).
\end{aligned} \tag{4.258}$$

Putting together the above bounds, we can conclude. □

We can now prove the following convergence result for TD learning with the delay adaptive update rule introduced in (4.189):

Theorem 13. *For $\alpha > 0$ small enough and $T \geq \max\{\tau_{mix}\tau_{avg}, \tau_{mix} + \tau_{max}\}$, the iterates generated following the update rule described in* (4.189) *are such that*

$$\mathbb{E}\left[r_T^2\right] \leq \rho^{\frac{T}{\tau_{avg}} - \tau_{mix}} + \gamma, \tag{4.259}$$

with $\rho = (1 - 2\alpha(1-\gamma)\omega + O(\alpha^2)(\tau_{mix}))$ and $\gamma = \frac{\rho^{\frac{T}{\tau_{avg}} - \tau_{mix}} - 1}{\rho}(O(\alpha^2)(\tau_{mix})\beta^2)$.

Proof. Assume $t_1^{'}, t_2^{'}, \ldots t_k^{'} \in [T]$ are iterations that the updates happen. Then, we have $r_j = r_{t_i^{'}}$ for all $j \in [t_i^{'}, t_{i+1}^{'})$. Let $E_{t_1^{'}, t_2^{'}, \ldots t_k^{'}}$ be the event that update happens only in $t_1^{'}, t_2^{'}, \ldots t_k^{'}$. Then we can

write using Lemma 40, 41 and 42, we get for all $j \in [t^{'}_{i+1}, t^{'}_{i+2})$,

$$
\begin{aligned}
\mathbb{E}\left[r_j^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] &= \mathbb{E}\left[r^2_{t^{'}_{i+1}} | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \\
&= \mathbb{E}\left[\|\boldsymbol{\theta}_{t^{'}_{i+1}-1} - \boldsymbol{\theta} + \alpha \mathbf{g}(\boldsymbol{\theta}_{t^{'}_{i+1}-1}, o_{t^{'}_{i+1}-1})\|^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \\
&+ \alpha^2 \mathbb{E}\left[\|\mathbf{e}_{t^{'}_{i+1}-1}\|^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \\
&- 2\alpha \mathbb{E}\left[\langle \mathbf{e}_{t^{'}_{i+1}-1} \boldsymbol{\theta}_{t^{'}_{i+1}-1} - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_{t^{'}_{i+1}-1}, o_{t^{'}_{i+1}-1})\rangle | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \\
&= \underbrace{\mathbb{E}\left[\|\boldsymbol{\theta}_{t^{'}_i} - \boldsymbol{\theta} + \alpha \mathbf{g}(\boldsymbol{\theta}_{t^{'}_i}, o_{t^{'}_i})\|^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right]}_{T_1} + \alpha^2 \underbrace{\mathbb{E}\left[\|\mathbf{e}_{t^{'}_i}\|^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right]}_{T_2} \\
&- 2\alpha \underbrace{\mathbb{E}\left[\langle \mathbf{e}_{t^{'}_i} \boldsymbol{\theta}_{t^{'}_i} - \boldsymbol{\theta}^* + \alpha \mathbf{g}(\boldsymbol{\theta}_{t^{'}_i}, o_{t^{'}_i})\rangle | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right]}_{T_3} \\
&\leq \underbrace{(1 - 2\alpha(1-\gamma)\omega + O(\alpha^2)(\tau_{mix}))}_{\rho} \mathbb{E}\left[r^2_{t^{'}_i} | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \\
&+ O(\alpha^2)(\tau_{mix})\beta^2.
\end{aligned} \tag{4.260}
$$

by recursively using the above equation, we have

$$
\mathbb{E}\left[r_T^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right] \leq \rho^k r_0^2 + \gamma, \tag{4.261}
$$

with $\rho = (1 - 2\alpha(1-\gamma)\omega + O(\alpha^2)(\tau_{mix}))$ and $\gamma = \frac{O(\alpha^2)\tau_{mix}\beta^2}{1-\rho}$. Getting expectation over $E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}$, we have

$$
\mathbb{E}\left[r_T^2\right] = \mathbb{E}\left[\mathbb{E}\left[r_T^2 | E_{t^{'}_1, t^{'}_2, \ldots t^{'}_k}\right]\right] \leq \rho^k r_0^2 + \gamma, \tag{4.262}
$$

and k, number of updates are at least $\frac{T}{\tau_{avg}} - \tau_{mix}$, which means

$$
\mathbb{E}\left[r_T^2\right] \leq \rho^k r_0^2 + \gamma \leq \rho^{\frac{T}{\tau_{avg}} - \tau_{mix}} r_0^2 + \gamma. \tag{4.263}
$$

□

CHAPTER 5

Min-Max Optimization under Delays

5.1. Introduction

Min-max optimization is a fundamental problem with applications in various fields, including game theory (Von Neumann and Morgenstern, 2007), machine learning (Goodfellow et al., 2020), robust optimization (Ben-Tal et al., 2009b), and more recently, adversarial robustness (Madry et al., 2017b). As such, the convergence analysis of various min-max optimization algorithms has received considerable attention over the years (Korpelevich, 1976a; Nedić and Ozdaglar, 2009; Daskalakis et al., 2017a; Mokhtari et al., 2020b). While this has resulted in a rich literature that provides non-asymptotic guarantees for the vanilla versions of these algorithms, not much is known about their *robustness* to different types of perturbations that show up in practice. In particular, for large-scale machine learning problems involving communication between multiple servers and agents, such perturbations get manifested in the form of (unavoidable) delays and asynchrony. Consequently, several works have extensively studied stochastic optimization with delayed gradients; since the literature on this topic is vast, we refer the reader to (Duchi et al., 2015; Doan et al., 2017; Arjevani et al., 2020; Stich and Karimireddy, 2019; Koloskova et al., 2022a) and the references therein. However, to our knowledge, there is no analogous theory for min-max optimization. Motivated by this gap, the goal of our paper is to build an understanding of the effect of delays on the convergence of common min-max optimization algorithms like Gradient Descent-Ascent (`GDA`) and Extra-Gradient (`EG`). Our main contributions in this regard are as follows.

5.1.1. Summary of Main Results

• We start with a result that is perhaps surprising. In Section 5.2.1, we empirically examine the effect of delays on the behavior of the Extra-Gradient algorithm due to Korpelevich (Korpelevich, 1976a). We observe that even with the smallest possible delay, i.e., a unit delay, `EG` diverges on a simple convex-concave function; see Fig. 5.1.[14] Notably, in the absence of delays, `EG` provably

[14]The Gradient Descent-Ascent (`GDA`) algorithm diverges on this instance even in the absence of delays (Daskalakis et al., 2017a).

guarantees convergence to a saddle-point for this function. This observation, although empirical, suggests that *delays can have non-trivial effects on the convergence of popular min-max optimization algorithms.*

• Our empirical study conveys the message that technical assumptions that are typically not required to study vanilla `EG` might, in fact, turn out to be needed to ensure convergence under delays. Accordingly, in Section 5.3, we study `DEG` - a version of `EG` with updates based on delayed gradients - for smooth, convex-concave functions over a *bounded* domain. In Theorem 14, we show that `DEG` guarantees convergence to a saddle-point at a rate $O(\sqrt{\tau_{\max}}/\sqrt{T})$, where T is the number of iterations, and $\tau_{\max}$ is a uniform bound on the delays. Our proof of this result is based on a connection to adversarial perturbations on statistical min-max learning problems in the recent work (Adibi et al., 2022b).

In the absence of delays, the convergence rates of `EG` and Gradient Descent-Ascent (`GDA`) are $O(1/T)$ (Mokhtari et al., 2020c) and $O(1/\sqrt{T})$ (Nedić and Ozdaglar, 2009), respectively. Our empirical divergence result (see Footnote 14) and Theorem 14 collectively suggest *that under delays, the behavior of `EG` is similar to that of `GDA`.*

• To further investigate the above point, we turn our attention to the behavior of `GDA` under delays in Section 5.4; we refer to this delayed version as `DGDA`. For smooth, convex-concave functions with bounded gradients, we prove that `DGDA` exhibits a convergence rate of $O(\sqrt{\tau_{\max}}/\sqrt{T})$ - exactly like `DEG`; see Theorem 15. However, unlike the analysis for `DEG`, we do not assume a bounded domain. Instead, we provide a careful analysis to argue that with suitable step-sizes, the iterates of `DGDA` remain bounded.

• All our results above pertain to scenarios where there is some underlying assumption of boundedness (either on the gradients or on the domain). Thus, one may ask: *Can min-max optimization algorithms under delays converge in the absence of such boundedness assumptions?* In Section 5.5, we answer this question in the affirmative by studying `DGDA` for smooth, strongly convex-strongly concave functions. We prove that `DGDA` guarantees *linear convergence to the saddle point* at a rate of $O(\exp(-T/\tau_{\max}^3))$;

Table 5.1: The table below presents a summary of our findings, outlining the conditions required for each algorithm to achieve the specified convergence rate. In the smooth convex-concave case, the convergence rate corresponds to the number of iterations needed for the duality gap to be less than ϵ. For the smooth strongly convex-strongly concave case (SC-SC), the rate corresponds to the number of iterations needed for the distance to saddle points to be less than ϵ. It is worth noting that in this table, we hide the dependence on G, L, and the strong-convexity parameter in the O notation.

Algorithm	Bounded Gradient	Bounded Domain	SC-SC	Convex-Concave	Rate
`DEG`	✓	✓	✗	✓	$\mathcal{O}(\frac{\tau_{\max}}{\epsilon^2})$
`DGDA`	✓	✗	✗	✓	$O(\frac{\tau_{\max}}{\epsilon^2})$
`DGDA`	✗	✗	✓	✗	$O(\tau_{\max}^3 \log(\frac{1}{\epsilon}))$

see Theorem 16.

As far as we are aware, our results above are novel and provide the first steps toward theoretically understanding the robustness of min-max optimization algorithms to delay-induced perturbations. Our results are summarized in Table 6.1.

5.2. Problem Setting

In this section, we start by describing the basic setup of a min-max optimization problem. Next, we show empirically how `EG` can diverge with even one-step delays. Finally, we conclude the section by outlining some technical assumptions that will be made for the majority of the paper to ensure boundedness and convergence of iterates.

The basic min-max optimization setup. Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty, convex subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.[15] Given a mapping of the form $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we are interested in solving the following optimization problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y). \tag{5.1}$$

[15]While we will assume that $\mathcal{X}$ and $\mathcal{Y}$ are bounded sets in Section 5.3, this assumption will be later relaxed in Sections 5.4 and 5.5.

Throughout the paper, we will assume that $f(x, y)$ is *continuously differentiable* in x and y, and *convex-concave* over $\mathcal{X} \times \mathcal{Y}$. Specifically, $f(\cdot, y) : \mathcal{X} \to \mathbb{R}$ is convex for every $y \in \mathcal{Y}$, and $f(x, \cdot) : \mathcal{Y} \to \mathbb{R}$ is concave for every $x \in \mathcal{X}$. Our goal is to find a saddle point (x^*, y^*) of $f(x, y)$ over the set $\mathcal{X} \times \mathcal{Y}$, where a saddle point is defined as a vector pair $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ that satisfies

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*), \forall x \in \mathcal{X}, y \in \mathcal{Y}. \tag{5.2}$$

For any $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$, let $\nabla_x f(\bar{x}, \bar{y})$ and $\nabla_y f(\bar{x}, \bar{y})$ denote the partial gradients of $f(x, y)$ with respect to x and y, respectively, at $(\bar{x}, \bar{y})$. Typical first-order iterative min-max optimization algorithms such as `GDA`, `EG`, and Optimistic Gradient Descent-Ascent (`OGDA`) aim to solve for (x^*, y^*) based on an oracle that provides partial gradients of $f(x, y)$ evaluated at the most recent iterates of the algorithm.

The delay model. Not much, however, is known about scenarios where the oracle is *imperfect*. To that end, we studied the effect of adversarial perturbations on the partial gradients of $f(x, y)$ in our recent work (Adibi et al., 2022b). In this work, we take a different stance. Instead of considering *arbitrary* adversarial perturbations, we will focus on *structured* perturbations induced by delays. As mentioned earlier in the Introduction, the source of such delays could be communication latencies or system-level computational challenges such as stragglers, both of which are prevalent in distributed systems. In this work, given an iterative min-max optimization algorithm that generates a sequence of iterates $\{(x_k, y_k)\}$, we assume that at iteration k, we only have access to partial gradients of $f(x, y)$ computed at a *stale* iterate $(x_{k-\tau_k}, y_{k-\tau_k})$, i.e., we have access to $\nabla_x f(x_{k-\tau_k}, y_{k-\tau_k})$ and $\nabla_y f(x_{k-\tau_k}, y_{k-\tau_k})$, where τ_k is the delay at iteration k. While we allow the delays to be time-varying, throughout the paper, we will work under the running assumption that all delays are uniformly bounded, i.e., there exists some positive integer $\tau_{\max}$ such that $\tau_k \leq \tau_{\max}, \forall k$.

Our goal is to understand what happens, when for computing the next iterate (x_{k+1}, y_{k+1}), one uses these delayed gradients as opposed to $\nabla_x f(x_k, y_k)$ and $\nabla_y f(x_k, y_k)$. Specifically, we ask:

- Can we hope for convergence to saddle points using delayed versions of algorithms like `GDA`

and EG?

- If so, for different classes of functions, how do the convergence rates get affected by $\tau_{\max}$?

In the next subsection, we demonstrate (empirically) that the answers to such questions are more nuanced than what one might initially expect.

5.2.1. Divergence of Extra-Gradient Algorithm under Delay

Let us start by quickly reviewing how the Extra-gradient (EG) algorithm for finding saddle-points operates in an unconstrained setting. EG first computes a set of mid-points $(\hat{x}_k, \hat{y}_k)$ by using partial gradients evaluated at the current iterate (x_k, y_k):

$$\begin{aligned} \hat{x}_k &\leftarrow x_k - \alpha \nabla_x f(x_k, y_k) \\ \hat{y}_k &\leftarrow y_k + \alpha \nabla_y f(x_k, y_k), \end{aligned} \tag{5.3}$$

where α is a suitable step-size. Next, using gradients evaluated at the mid-points, EG computes the next iterates as

$$\begin{aligned} x_{k+1} &\leftarrow x_k - \alpha \nabla_x f(\hat{x}_k, \hat{y}_k) \\ y_{k+1} &\leftarrow y_k + \alpha \nabla_x f(\hat{x}_k, \hat{y}_k). \end{aligned} \tag{5.4}$$

For smooth, convex-concave functions, the above EG procedure guarantees convergence to a saddle-point at a rate of $O(1/T)$, where T is the number of iterations (Mokhtari et al., 2020c). Moreover, to achieve this convergence, one does not need to make any assumption of a bounded domain or bounded gradients.

Now to illustrate the challenges posed by delays, let us consider solving the following problem

$$\min_x \max_y \langle x, y \rangle, \tag{5.5}$$

using a version of EG where all partial gradients are evaluated at iterates that are delayed by just one time-step.[16] Whereas one might have expected a slow-down in convergence due to delays, Figure

[16] Formally, the delayed EG algorithm we study is outlined in Algorithm 8.

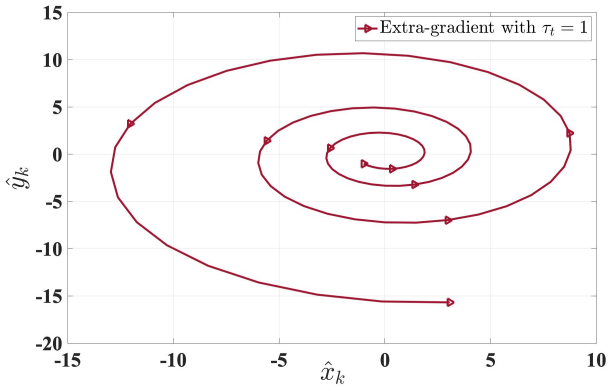

Figure 5.1: The Extra-gradient algorithm fails to converge, even with just one step delay, for the optimization problem $\min_x \max_y \langle x, y \rangle$. In this plot, we used a step size of $\alpha = 0.2$. However, with the same step size and no delay, the Extra-gradient algorithm converges to the origin, which is the saddle-point for this problem.

5.1 shows that in this specific setting, a unit delay causes `EG` to diverge! This demonstrates that delays can lead to non-trivial phenomena for standard min-max algorithms, thereby justifying our current study.

A rough explanation for the above phenomenon is as follows. In (Mokhtari et al., 2020b), the authors argued that `EG` can be studied as an approximate version of the Proximal Point (PP) algorithm, which, in turn, operates as follows:

$$\begin{aligned} x_{k+1} &\leftarrow x_k - \alpha \nabla_x f(x_{k+1}, y_{k+1}) \\ y_{k+1} &\leftarrow y_k - \alpha \nabla_y f(x_{k+1}, y_{k+1}). \end{aligned} \tag{5.6}$$

When the gradients on the right-hand side of the above equations are evaluated at one-step-delayed iterates, the above algorithm reduces to the `GDA` algorithm. Unlike `EG`, however, `GDA` can diverge for smooth, convex-concave problems like the one in Eq. (5.5), *even in the absence of delays*. In particular, some assumption on the boundedness of domain or gradients is needed to ensure the convergence of `GDA` for convex-concave problems. From the above discussion, we conclude that since `EG` with delays tends to behave like `GDA`, we need to impose additional technical assumptions to

ensure convergence to saddle points. As such, we will impose the following assumption of bounded gradients at various points in the paper.

Assumption 10. *There exists a constant $G > 1$ such that the following holds for all $x \in \mathcal{X}$, and all $y \in \mathcal{Y}$: $\|\nabla_x f(x,y)\| \leq G$, and $\|\nabla_y f(x,y)\| \leq G$.*[17]

We will also make the following standard assumption that the partial gradients of $f(x,y)$ are Lipschitz continuous.

Assumption 11. *There exists a constant $L > 1$ such that the following holds for all $x_1, x_2 \in \mathcal{X}$, and all $y_1, y_2 \in \mathcal{Y}$:*

$$\|\nabla_x f(x_1,y_1) - \nabla_x f(x_2,y_2)\| \leq L\left(\|x_1 - x_2\| + \|y_1 - y_2\|\right),$$
$$\|\nabla_y f(x_1,y_1) - \nabla_y f(x_2,y_2)\| \leq L\left(\|x_1 - x_2\| + \|y_1 - y_2\|\right).$$

5.3. Analysis of Delayed Extra-gradient for Convex-Concave functions

In Section 5.2.1, we saw that in the absence of a projection step to ensure the boundedness of iterates, the `EG` algorithm diverges on very simple functions, even with a one-step delay. Based on this empirical observation, in this section, we study delayed extra-gradient (`DEG`) - outlined in Algorithm 8 - under additional assumptions. In particular, throughout this section, we will work under assumptions 10 and 11, i.e., we will assume that the partial gradients of $f(x,y)$ are Lipschitz continuous and uniformly bounded. It is important to note here that the divergence of `DEG`, as illustrated in Figure 5.1, occurs when we do not impose assumption 10. Thus, this assumption will play a crucial role in our analysis.

The update rule for `DEG` (Algorithm 8) involves two steps. In the first step, `DEG` computes a midpoint $(\hat{x}_k, \hat{y}_k)$ based on partial gradients evaluated at a stale iterate $(x_{k-\tau_k}, y_{k-\tau_k})$; see Eq. (5.7). In the second step, `DEG` computes the next iterate (x_{k+1}, y_{k+1}) based on partial gradients evaluated at a stale mid-point $(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k})$; see Eq. (5.8). There are two important things to take note of here.

[17] We will use $\|\cdot\|$ to represent the Euclidean norm.

Algorithm 8 Delayed Extra-Gradient (DEG)

Require: Initial vectors $x_1 \in \mathcal{X}$, $y_1 \in \mathcal{Y}$; algorithm parameters: step-size $\alpha > 0$.

1: **for** $k = 1, \ldots, T$ **do**

2:
$$\begin{aligned} \hat{x}_k &\leftarrow \Pi_{\mathcal{X}} \left(x_k - \alpha \nabla_x f(x_{k-\tau_k}, y_{k-\tau_k}) \right) \\ \hat{y}_k &\leftarrow \Pi_{\mathcal{Y}} \left(y_k + \alpha \nabla_y f(x_{k-\tau_k}, y_{k-\tau_k}) \right). \end{aligned} \tag{5.7}$$

3:
$$\begin{aligned} x_{k+1} &\leftarrow \Pi_{\mathcal{X}} \left(x_k - \alpha \nabla_x f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}) \right) \\ y_{k+1} &\leftarrow \Pi_{\mathcal{Y}} \left(y_k + \alpha \nabla_x f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}) \right). \end{aligned} \tag{5.8}$$

4: **end for**

First, in each of the above updates, we project onto $\mathcal{X} \times \mathcal{Y}$ to ensure the boundedness of iterates. Second, our analysis is general enough to accommodate time-varying delays; furthermore, we allow τ_k and $\hat{\tau}_k$ to also be different. That said, as mentioned before, we will work under the running assumption that all delays are bounded uniformly by $\tau_{\max}$, i.e., $\max\{\tau_k, \hat{\tau}_k\} \leq \tau_{\max}, \forall k$.

Key Insight and Outline of Analysis. The starting point of our analysis for DEG is the observation that the errors induced by delays can be interpreted as *bounded perturbations*. As we shall see in Lemma 45, the boundedness of the delay-induced errors follows as a direct consequence of assumptions 10 and 11, and the uniform boundedness assumption on the delays. This key observation allows us to immediately make a connection to our prior work in (Adibi et al., 2022b), where we studied min-max optimization under adversarial perturbations. Building on this connection, we start with the following result that establishes some basic inequalities for our subsequent analysis; the proof of this result follows the same steps as that of (Adibi et al., 2022b, Lemma 1).

Lemma 43. *For the DEG algorithm, the following inequalities hold for all $k \in [T], x \in \mathcal{X}$, and*

$y \in \mathcal{Y}$.[18]

$$2\alpha\langle \nabla_x f(x_{k-\tau_k}, y_{k-\tau_k}), \hat{x}_k - x\rangle \leq \|x - x_k\|^2 - \|x - \hat{x}_k\|^2 - \|\hat{x}_k - x_k\|^2$$
$$-2\alpha\langle \nabla_y f(x_{k-\tau_k}, y_{k-\tau_k}), \hat{y}_k - y\rangle \leq \|y - y_k\|^2 - \|y - \hat{y}_k\|^2 - \|\hat{y}_k - y_k\|^2$$
$$2\alpha\langle \nabla_x f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}), x_{k+1} - x\rangle \leq \|x - x_k\|^2 - \|x - x_{k+1}\|^2 - \|x_{k+1} - x_k\|^2$$
$$-2\alpha\langle \nabla_y f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}), y_{k+1} - y\rangle \leq \|y - y_k\|^2 - \|y - y_{k+1}\|^2 - \|y_{k+1} - y_k\|^2.$$

Next, to bound the impact of delays, we introduce the following error vectors:

$$e_x(x_k, y_k) \triangleq \nabla_x f(x_{k-\tau_k}, y_{k-\tau_k}) - \nabla_x f(x_k, y_k),$$
$$e_y(x_k, y_k) \triangleq \nabla_y f(x_{k-\tau_k}, y_{k-\tau_k}) - \nabla_y f(x_k, y_k),$$

and

$$e_x(\hat{x}_k, \hat{y}_k) \triangleq \nabla_x f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}) - \nabla_x f(\hat{x}_k, \hat{y}_k),$$
$$e_y(\hat{x}_k, \hat{y}_k) \triangleq \nabla_y f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}) - \nabla_y f(\hat{x}_k, \hat{y}_k).$$

Let $D = \max\{D_x, D_y\}$, where D_x and D_y are the diameters of the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. Leveraging Lemma 43, our next result tracks the progress made by the mid-point sequence $\{(\hat{x}_k, \hat{y}_k)\}$ generated by `DEG`. The proof of this result mirrors that of (Adibi et al., 2022b, Lemma 2).

Lemma 44. *Suppose assumptions 10 and 11 hold. Furthermore, suppose $\alpha \leq 1/(2L)$. Then, for the*

[18] Given a positive integer N, we use $[N]$ to represent the set $\{1, \ldots, N\}$.

DEG algorithm, the following holds for all $k \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:

$$\begin{aligned}
&\alpha\langle \nabla_x f(\hat{x}_k, \hat{y}_k), \hat{x}_k - x\rangle - \alpha\langle \nabla_y f(\hat{x}_k, \hat{y}_k), \hat{y}_k - y\rangle \\
&\leq \frac{1}{2}\left(\|x - x_k\|^2 - \|x - x_{k+1}\|^2 + \|y - y_k\|^2 - \|y - y_{k+1}\|^2\right) \\
&+ \alpha D\left(\|e_x(x_k, y_k)\| + \|e_x(\hat{x}_k, \hat{y}_k)\| + \|e_y(x_k, y_k)\| + \|e_y(\hat{x}_k, \hat{y}_k)\|\right).
\end{aligned}$$

The above result sets things up nicely for a telescoping-sum analysis. However, the missing piece right now is to provide bounds on the delay-induced errors. We derive such bounds in the following result.

Lemma 45. *Suppose assumptions 10 and 11 hold. For the DEG algorithm, the following error-bounds then apply $\forall k \in [T]$:*

$$\max\{\|e_x(x_k, y_k)\|, \|e_x(\hat{x}_k, \hat{y}_k)\|, \|e_y(x_k, y_k)\|, \|e_y(\hat{x}_k, \hat{y}_k)\|\} \leq \Delta_T,$$

where $\Delta_T = 6\alpha G L \tau_{\max}$.

Proof. In what follows, we only show how to bound $\|e_x(x_k, y_k)\|$ and $\|e_x(\hat{x}_k, \hat{y}_k)\|$; bounds for the other two error terms can be derived in an identical manner. We start by bounding $\|e_x(x_k, y_k)\|$. From equation (5.8), we have

$$\begin{aligned}
\|x_k - x_{k-\tau_k}\| &\leq \sum_{j=k-\tau_k}^{k-1} \|x_{j+1} - x_j\| \\
&\overset{(a)}{\leq} \alpha \left(\sum_{j=k-\tau_k}^{k-1} \|\nabla_x f(\hat{x}_{j-\hat{\tau}_j}, \hat{y}_{j-\hat{\tau}_j})\| \right) \\
&\overset{(b)}{\leq} \alpha G \tau_{\max},
\end{aligned} \tag{5.9}$$

where (a) follows from the non-expansive property of the projection operator, and (b) follows from

assumption 10 and the fact that $\tau_k \leq \tau_{\max}$. Using the exact same steps, one can establish the same bound for $\|y_k - y_{k-\tau_k}\|$. Thus, we have

$$\begin{aligned} \|e_x(x_k, y_k)\| &= \|\nabla_x f(x_k, y_k) - \nabla_x f(x_{k-\tau}, y_{k-\tau})\| \\ &\overset{(a)}{\leq} L(\|x_k - x_{k-\tau_k}\| + \|y_k - y_{k-\tau_k}\|) \\ &\overset{(b)}{\leq} 2\alpha G L \tau_{\max}, \end{aligned} \tag{5.10}$$

where (a) follows from smoothness, i.e., assumption 11, and (b) follows from Eq. (5.9). Now to bound $e_x(\hat{x}_k, \hat{y}_k)$, observe

$$\begin{aligned} \|e_x(\hat{x}_k, \hat{y}_k)\| &= \|\nabla_x f(\hat{x}_{k-\hat{\tau}_k}, \hat{y}_{k-\hat{\tau}_k}) - \nabla_x f(\hat{x}_k, \hat{y}_k)\| \\ &\overset{(a)}{\leq} L(\|\hat{x}_k - \hat{x}_{k-\hat{\tau}_k}\| + \|\hat{y}_k - \hat{y}_{k-\hat{\tau}_k}\|) \\ &\leq L(\|\hat{x}_k - x_k\| + \|x_k - x_{k-\hat{\tau}_k}\| + \|x_{k-\hat{\tau}_k} - \hat{x}_{k-\hat{\tau}_k}\| \\ &\quad + \|\hat{y}_k - y_k\| + \|y_k - y_{k-\hat{\tau}_k}\|) + \|y_{k-\hat{\tau}_k} - \hat{y}_{k-\hat{\tau}_k}\|) \\ &\overset{(b)}{\leq} 2\alpha G L \tau_{\max} + 4\alpha G L \\ &\overset{(c)}{\leq} 6\alpha G L \tau_{\max}. \end{aligned}$$

In the above steps, (a) follows from assumption 11, (b) follows from (5.9) and the fact that for any $j \in [T]$, $\|\hat{x}_j - x_j\| \leq \alpha \|\nabla_x f(x_{j-\tau_j}, y_{j-\tau_j})\| \leq \alpha G$, and (c) follows from noting that $\tau_{\max} \geq 1$. This concludes the proof. □

We are now in a position to prove our first main result which establishes complexity bounds for `DEG` for smooth convex-concave functions with bounded gradients.

Theorem 14. *Suppose assumptions 10 and 11 hold. Moreover, suppose the number of iterations T*

is large enough such that $T \geq L$. Then, with

$$\alpha = \sqrt{\frac{1}{24GL\tau_{\max}T}},$$

the iterates generated by `DEG` *satisfy:*

$$\max_{y\in\mathcal{Y}} f(\bar{x}_T, y) - \min_{x\in\mathcal{X}} f(x, \bar{y}_T) \leq 10D^2\sqrt{\frac{GL\tau_{\max}}{T}}, \tag{5.11}$$

where $\bar{x}_T = (1/T)\sum_{k\in[T]}\hat{x}_k$ and $\bar{y}_T = (1/T)\sum_{k\in[T]}\hat{y}_k$.

Proof. Let us start by noting that when $T \geq L$, the choice of step-size above satisfies $\alpha \leq 1/(2L)$. Thus, we can invoke Lemma 44. From the convex-concave property of $f(x, y)$, the following inequalities hold $\forall k \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:

$$\begin{aligned}
\alpha\left(f(\hat{x}_k, \hat{y}_k) - f(x, \hat{y}_k)\right) &\leq \alpha\langle\nabla_x f(\hat{x}_k, \hat{y}_k), \hat{x}_k - x\rangle \\
-\alpha\left(f(\hat{x}_k, \hat{y}_k) - f(\hat{x}_k, y)\right) &\leq -\alpha\langle\nabla_y f(\hat{x}_k, \hat{y}_k), \hat{y}_k - y\rangle.
\end{aligned}$$

Summing the two inequalities above, and using Lemmas 44 and 45, we obtain:

$$\begin{aligned}
\alpha\left(f(\hat{x}_k, y) - f(x, \hat{y}_k)\right) \leq \frac{1}{2}\left(\|x - x_k\|^2 - \|x - x_{k+1}\|^2\right) \\
+\frac{1}{2}\left(\|y - y_k\|^2 - \|y - y_{k+1}\|^2\right) + 4\alpha D\Delta_T,
\end{aligned} \tag{5.12}$$

where Δ_T is as defined in Lemma 45. From the convexity of $f(x, y)$ w.r.t. x and concavity w.r.t. y, note that we have $f(\bar{x}_T, y) \leq (1/T)\sum_{k\in[T]} f(\hat{x}_k, y)$ and $f(x, \bar{y}_T) \geq (1/T)\sum_{k\in[T]} f(x, \hat{y}_k)$, respectively. Combining this with Eq. (5.12), we obtain

$$f(\bar{x}_T, y) - f(x, \bar{y}_T) \leq \frac{D^2}{\alpha T} + 4D\Delta_T. \tag{5.13}$$

The result follows by plugging into the above inequality the choice of α in the statement of the theorem, and by noting that the resulting bound holds for all $x \in \mathcal{X}$ and for all $y \in \mathcal{Y}$. This completes the proof. □

Algorithm 9 Delayed Gradient Descent-Ascent (DGDA)

Require: Initial vector $z_1 = [x_1; y_1] \in \mathbb{R}^{m+n}$; algorithm parameters: step-size $\alpha > 0$.

1: **for** $k = 1, \ldots, T$ **do**

2:
$$z_{k+1} = z_k - \alpha \Phi(z_{k-\tau_k}). \tag{5.14}$$

3: **end for**

Discussion. From Theorem 14, we conclude that for smooth convex-concave functions, DEG guarantees that the primal-dual gap converges to zero at a rate $O(\sqrt{\tau_{\max}}/\sqrt{T})$. The primal-dual gap is zero if and only if $(\bar{x}_T, \bar{y}_T)$ is a saddle point of $f(x, y)$ over the set $\mathcal{X} \times \mathcal{Y}$. Thus, DEG also guarantees convergence to a saddle-point under delays. The important thing to note here is that the $O(1/T)$ convergence rate of EG gets significantly slackened in the presence of delays; whether this is an artifact of our analysis or fundamental is an open question. The $O(1/\sqrt{T})$ rate of DEG mirrors the rate of GDA in the absence of delays. In the following sections, we will further explore this connection.

5.4. Analysis of Delayed Gradient Descent-Ascent for Convex-Concave functions

In this section, we will examine the convergence of a delayed version of the gradient descent ascent algorithm that we refer to as DGDA. As before, we will continue to work under assumptions 10 and 11. However, we will set $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Y} = \mathbb{R}^n$, i.e., as a departure from the previous section, the domains of the variables x and y are no longer assumed to be bounded. As we shall soon see, this makes the analysis more challenging relative to that in Section 5.3.

To proceed, given any $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, we will find it convenient to define a new variable $z = [x; y]$ that resides in $\mathbb{R}^{m+n}$. Next, corresponding to any $z = [x; y]$, let us define the function $\Phi : \mathbb{R}^{m+n} \to \mathbb{R}^{m+n}$ as follows:

$$\Phi(z) = \begin{bmatrix} \nabla_x f(x, y) \\ -\nabla_y f(x, y) \end{bmatrix}, \tag{5.15}$$

With these notations in place, we outline the steps of DGDA in Algorithm 9; the steps are self-explanatory.

Analysis of DGDA. In our analysis, we will make use of the following result from (Nemirovski, 2004a),

stated for our purpose.

Lemma 46. *Let $\Phi(z)$ be as defined in Eq. (5.15), and suppose assumption 11 holds for all $z \in \mathbb{R}^{m+n}$. Then, the following statements are true for any $z_1, z_2 \in \mathbb{R}^{m+n}$:*

1. $\langle \Phi(z_1) - \Phi(z_2), z_1 - z_2 \rangle \geq 0$,

2. *For any saddle-point $z^* = [x^*; y^*]$ of $f(x, y)$, we have $\Phi(z^*) = 0$.*

We start with a simple result that bounds the error $e_k \triangleq \Phi(z_k) - \Phi(z_{k-\tau_k})$ induced by delays as a function of the smoothness parameter L, the uniform bound on the gradients G, and the maximum delay $\tau_{\max}$.

Lemma 47. *Suppose assumptions 10 and 11 hold $\forall z \in \mathbb{R}^{m+n}$. Then, for any $k \in [T]$, the delay-induced error $e_k \triangleq \Phi(z_k) - \Phi(z_{k-\tau_k})$ for `DGDA` satisfies*

$$\|e_k\| \leq 2\alpha L G \tau_{\max}. \tag{5.16}$$

Proof. For any two points $z = [x; y]$ and $\hat{z} = [\hat{x}; \hat{y}]$, we have

$$\begin{aligned} \|\Phi(z) - \Phi(\hat{z})\|^2 &\overset{(a)}{\leq} 2\left(L(\|x - \hat{x}\| + \|y - \hat{y}\|)\right)^2 \\ &\leq 4L^2 \|z - \hat{z}\|^2, \end{aligned} \tag{5.17}$$

where we used assumption 11 for the first inequality. Based on the above inequality, we have

$$\begin{aligned} \|e_k\| &= \|\Phi(z_{k-\tau_k}) - \Phi(z_k)\| \\ &\leq 2L \|z_{k-\tau_k} - z_k\| \\ &\leq 2L \sum_{j=k-\tau_k}^{k-1} \|z_{j+1} - z_j\| \\ &\leq 2\alpha L \sum_{j=k-\tau_k}^{k-1} \|\Phi(z_{j-\tau_j})\| \leq 2\alpha L G \tau_{\max}, \end{aligned} \tag{5.18}$$

where the final step follows from assumption 10. □

Unlike the analysis in Section 5.3 where the boundedness of the domain implied bounded iterates, we need to do more work to establish that the iterates generated by `DGDA` remain bounded. Leveraging Lemma 47, the following result establishes this key fact.

Lemma 48. *Suppose assumptions 10 and 11 hold $\forall z \in \mathbb{R}^{m+n}$. Let $z^* = [x^*; y^*]$, and suppose the step-size α satisfies*

$$\alpha \leq \frac{1}{2\sqrt{LG\tau_{\max}T}}.$$

Then, for the `DGDA` algorithm, the following holds $\forall k \in [T]$:

$$\|z_k - z^*\|^2 \leq 10B, \text{ where } B = \max\{\|z_1 - z^*\|^2, G\}. \tag{5.19}$$

Proof. From Eq. (5.14) and the definition of e_k, we have

$$\begin{aligned}
\|z_{k+1} - z\|^2 =& \|z_k - \alpha\Phi(z_k) - z\|^2 \\
&+ \alpha^2\|e_k\|^2 + 2\alpha\langle e_k, z_k - z - \alpha\Phi(z_k)\rangle \\
=& \|z_k - z\|^2 + \alpha^2\|\Phi(z_k)\|^2 - 2\alpha\langle\Phi(z_k), z_k - z\rangle \\
&+ \alpha^2\|e_k\|^2 + 2\alpha\langle e_k, z_k - z - \alpha\Phi(z_k)\rangle \\
\leq& \|z_k - z\|^2 + 2\alpha^2G^2 - 2\alpha\langle\Phi(z_k), z_k - z\rangle \\
&+ 4\alpha^4G^2L^2\tau_{\max}^2 \underbrace{+2\alpha\langle e_k, z_k - z\rangle}_{T_1} \underbrace{-2\alpha^2\langle e_k, \Phi(z_k)\rangle}_{T_2}.
\end{aligned} \tag{5.20}$$

We now proceed to bound T_1 and T_2. For T_2, we have:

$$\begin{aligned}
T_2 \overset{(a)}{\leq}& \alpha^2\|e_k\|^2 + \alpha^2\|\Phi(z_k)\|^2 \\
\overset{(b)}{\leq}& 4\alpha^4G^2L^2\tau_{\max}^2 + 2\alpha^2G^2,
\end{aligned}$$

where (a) follows from the elementary fact that for any two scalars $c, d \in \mathbb{R}$, it holds that

$$cd \leq \frac{1}{2}c^2 + \frac{1}{2}d^2. \tag{5.21}$$

Moreover, for (b), we used Lemma 47 and assumption 10. For bounding T_1, observe that

$$\begin{aligned} T_1 &= 2\alpha\langle e_k, z_k - z\rangle \\ &\leq 2\alpha\|e_k\|\|z_k - z\| \\ &\overset{(a)}{\leq} 4\alpha^2 GL\tau_{\max}\|z_k - z\| \\ &= \left(2\alpha\sqrt{GL\tau_{\max}}\right)\left(2\alpha\sqrt{GL\tau_{\max}}\|z_k - z\|\right) \\ &\overset{(b)}{\leq} 2\alpha^2 GL\tau_{\max} + 2\alpha^2 GL\tau_{\max}\|z_k - z\|^2, \end{aligned} \tag{5.22}$$

where we again appealed to Lemma 47 for (a). For (b), we used Eq. (5.21). Plugging in the above bounds on T_1 and T_2 into Eq. (5.20), simplifying using $L, G \geq 1$, and rearranging terms, we arrive at the following inequality:

$$\begin{aligned} 2\alpha\langle\Phi(z_k), z_k - z\rangle \leq \left(1 + 2\alpha^2 LG\tau_{\max}\right)\|z_k - z\|^2 \\ - \|z_{k+1} - z\|^2 + A, \end{aligned} \tag{5.23}$$

where $A = 2\alpha^2 GL\tau_{\max}(1 + 2G + 4\alpha^2 GL\tau_{\max})$. Now setting $z = z^*$ in the above inequality, and noting that $\langle\Phi(z_k), z_k - z^*\rangle \geq 0$ based on Lemma 46, we obtain the following recursive inequality that holds for all $k \in [T]$:

$$\|z_{k+1} - z^*\|^2 \leq \left(1 + 2\alpha^2 LG\tau_{\max}\right)\|z_k - z^*\|^2 + A. \tag{5.24}$$

Defining $r_k \triangleq \|z_k - z^*\|$, $\beta \triangleq \left(1 + 2\alpha^2 LG\tau_{\max}\right)$, and iterating the above inequality, we obtain:

$$\begin{aligned} r_k^2 &\leq \beta^{k-1} r_1^2 + \left(\sum_{j=0}^{k-2} \beta^j\right) A \\ &\leq \beta^{k-1} r_1^2 + \frac{\beta^k}{\beta - 1} A \\ &\leq \beta^T r_1^2 + \frac{\beta^T}{\beta - 1} A. \end{aligned} \tag{5.25}$$

We will now bound each of the terms above by using the elementary fact that for any $c \in \mathbb{R}$, it holds that $(1 + c) \leq e^c$. When the step-size α satisfies

$$\alpha \leq \frac{1}{2\sqrt{LG\tau_{\max} T}},$$

we have

$$\beta^T \leq \left(1 + \frac{1}{2T}\right)^T \leq e^{0.5} \leq 2. \tag{5.26}$$

Furthermore, it is easy to see that

$$\frac{A}{\beta - 1} \leq \left(1 + 2G + \frac{1}{T}\right) \leq 4G.$$

Combining the above bounds leads to the claim of the lemma. This concludes the proof. □

Based on the above result, let us introduce a set $\mathcal{H}$ as follows:

$$\mathcal{H} \triangleq \{z | \|z - z^*\|^2 \leq 10B\}, \tag{5.27}$$

where $B = \max\{\|z_1 - z^*\|^2, G\}$. From Lemma 48, we note that as long as the step-size α is chosen appropriately, the iterate sequence $\{z_k\}$ generated by `DGDA` belongs to $\mathcal{H}$. Moreover, $z^* \in \mathcal{H}$ trivially. With these observations in place, we now prove our main convergence result for `DGDA` for smooth convex-concave functions with bounded gradients.

Theorem 15. *Suppose assumptions 10 and 11 hold $\forall x \in \mathbb{R}^m$ and $\forall y \in \mathbb{R}^n$. Let the step-size be*

chosen to satisfy

$$\alpha = \frac{1}{2\sqrt{LG\tau_{\max}T}}.$$

Then, the iterates generated by `DGDA` *satisfy:*

$$\max_{y:(\bar{x}_T,y)\in\mathcal{H}} f(\bar{x}_T, y) - \min_{x:(x,\bar{y}_T)\in\mathcal{H}} f(x, \bar{y}_T) \leq 44B\sqrt{\frac{GL\tau_{\max}}{T}},$$

where $\bar{x}_T = (1/T)\sum_{k\in[T]} \hat{x}_k$, $\bar{y}_T = (1/T)\sum_{k\in[T]} \hat{y}_k$, *and the set* $\mathcal{H}$ *is as defined in Eq.* (5.27).

Proof. Recall the following notation from Lemma 48: $r_k = \|z_k - z^*\|$ and $B = \max\{r_1^2, G\}$. Let us start by noting that the choice of step-size in the statement of the theorem complies with that used to establish Lemma 48. Thus, we can invoke Lemma 48 to conclude that for any $z \in \mathcal{H}$, the following is true:

$$\|z_k - z\|^2 \leq 2r_k^2 + 2\|z - z^*\|^2 \leq 40B, \tag{5.28}$$

where the last inequality follows from the definition of the set $\mathcal{H}$. Using Eq. (5.23) from Lemma 48, we then have for any $z \in \mathcal{H}$:

$$\begin{aligned} 2\alpha\langle\Phi(z_k), z_k - z\rangle &\leq \|z_k - z\|^2 - \|z_{k+1} - z\|^2 + A \\ &\quad + 2\alpha^2 LG\tau_{\max}\|z_k - z\|^2 \\ &\leq \|z_k - z\|^2 - \|z_{k+1} - z\|^2 + \bar{A}, \end{aligned}$$

where $\bar{A} = A + 80\alpha^2 LGB\tau_{\max}$, $A = 2\alpha^2 GL\tau_{\max}(1 + 2G + 4\alpha^2 GL\tau_{\max})$, and we used Eq. (5.28). Now summing the above inequality from $k = 1$ to T, we obtain

$$\sum_{k=1}^{T} 2\alpha\langle\Phi(z_k), z_k - z\rangle \leq \|z_1 - z\|^2 + \bar{A}T. \tag{5.29}$$

Moreover, from Proposition 1 in (Mokhtari et al., 2020c), we have

$$\sum_{k=1}^{T} 2\alpha\langle\Phi(z_k), z_k - z\rangle \geq 2\alpha T(f(\bar{x}_T, y) - f(x, \bar{y}_T)). \tag{5.30}$$

Combining the above display with Eq. (5.29) then yields the following bound $\forall z = [x; y] \in \mathcal{H}$:

$$f(\bar{x}_T, y) - f(x, \bar{y}_T) \leq \frac{\|z_1 - z\|^2}{2\alpha T} + \frac{\bar{A}}{2\alpha}. \tag{5.31}$$

Let us simplify the bound by first noting that for α chosen as in the statement of the theorem, it holds that $\bar{A} \leq 88\alpha^2 GBL\tau_{\max}$. Moreover, since $z \in \mathcal{H}$, we have

$$\|z_1 - z\|^2 \leq 2r_1^2 + 2\|z - z^*\|^2 \leq 22B.$$

Plugging in the above bounds in Eq. (5.31) then gives us:

$$f(\bar{x}_T, y) - f(x, \bar{y}_T) \leq \frac{11B}{\alpha T} + 44\alpha GBL\tau_{\max}. \tag{5.32}$$

The result follows from simply substituting the choice of α in the statement of the theorem. □

Discussion. The main message conveyed by Theorem 15 is that for smooth convex-concave functions with bounded gradients, the convergence rates of `DGDA` and `DEG` are identical in terms of their dependence on $\tau_{\max}$ and T. This complies with the intuition developed earlier in the paper that `EG` under delays behaves like `GDA`.

5.5. Analysis of Delayed Gradient Descent-Ascent for Strongly Convex-Strongly Concave functions

For smooth strongly convex-strongly concave functions, it is known that `GDA` guarantees linear convergence to the saddle point in the absence of delays (Fallah et al., 2020c). In this section, we ask: *Does `DGDA` (Algorithm 9) also guarantee linear convergence to the saddle point for smooth strongly convex-strongly concave functions?* Our analysis in this section will provide an answer to this question in the affirmative. Moreover, we will precisely quantify how the maximum delay $\tau_{\max}$ slackens the exponent of linear convergence relative to when there is no delay. To get started, we now provide a formal definition of strongly convex-strongly concave functions.

Assumption 12. *The function $f(x, y)$ is μ-strongly convex-μ-strongly concave (SC-SC) over $\mathcal{X} \times \mathcal{Y}$,*

i.e., for all $x_1, x_2 \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, the following holds:

$$f(x_2, y_1) \geq f(x_1, y_1) + \langle \nabla_x f(x_1, y_1), x_2 - x_1 \rangle + \frac{\mu}{2}\|x_2 - x_1\|^2,$$
$$f(x_1, y_2) \leq f(x_1, y_1) + \langle \nabla_y f(x_1, y_1), y_2 - y_1 \rangle - \frac{\mu}{2}\|y_2 - y_1\|^2.$$

Throughout this section, we will set $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Y} = \mathbb{R}^n$, i.e., we will make no assumption of bounded domains. Furthermore, unlike prior sections, we will drop the assumption of bounded gradients, i.e., we will no longer work under assumption 10.

Analysis of `DGDA`. To proceed, we start by recalling two results from (Fallah et al., 2020c) that will play a crucial role in our subsequent analysis; at this point, we remind the reader of the definition of $\Phi(\cdot)$ in Eq. (5.15).

Lemma 49 ((Fallah et al., 2020c))**.** *Suppose assumptions 11 and 12 hold. Then, $\forall z, \hat{z} \in \mathbb{R}^{m+n}$, we have*

$$L\|z - \hat{z}\|^2 \geq \langle \Phi(z) - \Phi(\hat{z}), z - \hat{z} \rangle \geq \mu\|z - \hat{z}\|^2. \tag{5.33}$$

Lemma 50 ((Fallah et al., 2020c))**.** *Suppose assumptions 11 and 12 hold. Then, $\forall z, \hat{z} \in \mathbb{R}^{m+n}$, we have*

$$\langle \Phi(z) - \Phi(\hat{z}), z - \hat{z} \rangle \geq \frac{\mu}{4L^2}\|\Phi(z) - \Phi(\hat{z})\|^2. \tag{5.34}$$

Recall the definitions of iterate-suboptimality and delay-induced error: $r_k = \|z_k - z^*\|$ and $e_k = \Phi(z_k) - \Phi(z_{k-\tau_k})$. As before, our starting point will be to establish a bound on $\|e_k\|$. However, to establish a linear convergence rate, we need to provide a finer analysis relative to that in Lemmas 45 and 47. In particular, unlike these results which established uniform convergence bounds on $\|e_k\|$, we will instead seek to bound $\|e_k\|$ as a function of a suitably defined iterate-suboptimality-metric. Our next result formalizes this idea.

Lemma 51. *Suppose assumptions 11 and 12 hold $\forall z \in \mathbb{R}^{m+n}$. Then, for any $k \in [T]$, the delay-induced error $e_k = \Phi(z_k) - \Phi(z_{k-\tau_k})$ for* `DGDA` *satisfies*

$$\|e_k\| \leq 2\alpha M_k, \tag{5.35}$$

where $M_k = L\tau_{\max}(\frac{4L^2}{\mu} + 4L) \max_{k-2\tau_{\max} \leq t \leq k} r_t$.

Proof. For bounding e_k, observe that

$$\begin{aligned}
\|e_k\| =& \|\Phi(z_{k-\tau_k}) - \Phi(z_k)\| \\
&\overset{(a)}{\leq} 2L\|z_{k-\tau_k} - z_k\| \\
&\leq 2L \sum_{j=k-\tau_k}^{k-1} \|z_{j+1} - z_j\| \\
&\leq 2\alpha L \sum_{j=k-\tau_k}^{k-1} \|\Phi(z_{j-\tau_j})\| \\
&\leq 2\alpha L \sum_{j=k-\tau_k}^{k-1} \left(\|\Phi(z_j)\| + \|\Phi(z_{j-\tau_j}) - \Phi(z_j)\|\right) \\
&\overset{(b)}{\leq} 2\alpha L \sum_{j=k-\tau_k}^{k-1} \left(\|\Phi(z_j)\| + 2L\|z_{j-\tau_j} - z_j\|\right) \\
&\leq 2\alpha L \sum_{j=k-\tau_k}^{k-1} \left(\|\Phi(z_j)\| + 2Lr_{j-\tau_j} + 2Lr_j\right).
\end{aligned} \tag{5.36}$$

From Lemma 46, we know that $\Phi(z^*) = 0$. Furthermore, from Lemma 50 and the Cauchy–Schwarz inequality, we obtain

$$\|\Phi(z_k)\|\|z_k - z^*\| \geq \langle \Phi(z_k), z_k - z^* \rangle \geq \frac{\mu}{4L^2}\|\Phi(z_k)\|^2,$$

which means

$$\|\Phi(z_k)\| \leq \frac{4L^2}{\mu}\|z_k - z^*\| = \frac{4L^2}{\mu} r_k,$$

Combining the above display with Eq. (5.36), we obtain

$$\begin{aligned}\|e_k\| &\leq 2\alpha L \sum_{j=k-\tau_{\max}}^{k-1} \left(\frac{4L^2}{\mu} r_j + 2L r_{j-\tau_j} + 2L r_j\right) \\ &\leq 2\alpha L \tau_{\max} \left(\frac{4L^2}{\mu} + 4L\right) \max_{k-2\tau_{\max} \leq t \leq k-1} r_t \\ &\leq 2\alpha M_k.\end{aligned} \tag{5.37}$$

□

We will also make use of the following key result.

Lemma 52 ((Gurbuzbalaban et al., 2017)). *Suppose we have a sequence of non-negative real numbers, V_k, satisfying the inequality*

$$V_{k+1} \leq pV_k + q \max_{k-d(k) \leq \ell \leq k} V_\ell,$$

for some non-negative constants p and q, where $k \geq 0$ and $0 \leq d(k) \leq d_{\max}$ for some positive constant $d_{\max}$. If $p + q < 1$, then we have

$$V_k \leq r^k V_0, \text{ where } r = (p+q)^{1/(1+d_{\max})}.$$

We now prove our main result for `DGDA` for the class of smooth strongly convex-strongly concave (SC-SC) functions.

Theorem 16. *Suppose assumptions 11 and 12 hold $\forall z \in \mathbb{R}^{m+n}$. Let the step-size be chosen to satisfy*

$$\alpha = \frac{\mu^3}{1536 L^6 \tau_{max}^2}.$$

Then, the iterates generated by `DGDA` *satisfy:*

$$r_k \leq \left(1 - \frac{\mu^4}{3072 L^6 \tau_{\max}^2}\right)^{\frac{k-1}{6\tau_{\max}}} r_1, \tag{5.38}$$

where $r_k = \|z_k - z^*\|$.

Proof. From the update rule of the `DGDA` algorithm and Lemma 51, we have

$$\begin{aligned}
&\|z_{k+1} - z^*\|^2 - (1-\alpha\mu)\|z_k - z^*\|^2 = \\
&\alpha\mu\|z_k - z^*\|^2 - 2\alpha\langle \Phi(z_{k-\tau_k}), z_k - z^*\rangle + \alpha^2\|\Phi(z_{k-\tau_k})\|^2 \\
&\leq \alpha\mu\|z_k - z^*\|^2 - 2\alpha\langle \Phi(z_k), z_k - z^*\rangle + 2\alpha^2\|\Phi(z_k)\|^2 \\
&\quad + 2\alpha\langle e_k, z_k - z^*\rangle + 2\alpha^2\|e_k\|^2 \\
&\leq \underbrace{\alpha\mu\|z_k - z^*\|^2 - 2\alpha\langle \Phi(z_k), z_k - z^*\rangle + 2\alpha^2\|\Phi(z_k)\|^2}_{f_k} \\
&\quad + \underbrace{4\alpha^2 M_k r_k + 8\alpha^4 M_k^2}_{p_k}.
\end{aligned} \tag{5.39}$$

From Lemmas 49 and 50, we further know that

$$\langle \Phi(z_k), z_k - z^*\rangle \geq \mu\|z_k - z^*\|^2, \text{ and}$$

$$\langle \Phi(z_k), z_k - z^*\rangle \geq \frac{\mu}{4L^2}\|\Phi(z_k)\|^2.$$

When $\alpha \leq \frac{\mu}{8L^2}$ - a requirement met by the choice of step-size in the statement of the theorem - it is easy to verify that the above equations imply $f_k \leq 0$, where f_k is as in Eq. (5.39). We also have

$$p_k \leq 12\alpha^2 M_k^2 \leq \alpha^2 C \left(\max_{k-2\tau_{\max}\leq t\leq k} r_t^2\right),$$

where $C = 768\frac{L^6}{\mu^2}\tau_{\max}^2$, and we used $L \geq \mu$ for simplifications. From the above discussion, we conclude that

$$r_{k+1}^2 \leq (1-\alpha\mu) r_k^2 + \alpha^2 C \left(\max_{k-2\tau_{\max}\leq t\leq k} r_t^2\right).$$

From the choice of step-size in the statement of the theorem, it is easy to verify that $1 - \alpha\mu + \alpha^2 C = 1 - 0.5\alpha\mu < 1$. Thus, we can immediately apply Lemma 52 to arrive at the desired conclusion. This concludes the proof. $\square$

Discussion. Theorem 16 reveals that for smooth SC-SC functions, `DGDA` guarantees *linear convergence of the iterates to the saddle-point*. The result also clearly demonstrates how the exponent of convergence gets affected by $\tau_{\max}$.

CHAPTER 6

Minimax Optimization: The Case of Convex-Submodular

6.1. INTRODUCTION

The problem of solving a minimax optimization problem, also known as the saddle point problem, appears in many domains such as robust optimization (Ben-Tal et al., 2009b), game theory (Osborne and Rubinstein, 1994), and robust control (Zhou and Doyle, 1998; Hast et al., 2013). It has also recently attracted a lot of attention in the machine learning community due to the rise of generative adversarial networks (GANs) (Goodfellow et al., 2014b) and robust learning (Bertsimas et al., 2011; Lanckriet et al., 2002; Li et al., 2019b). There has been an extensive literature on the design of convergent methods for solving minimax problems for the case that both minimization and maximization variables belong to continuous domains (Tseng, 1995; Nesterov, 2007; Li and Lin, 2015; Ouyang and Xu, 2019; Thekumparampil et al., 2019; Zhao, 2019; Hamedani and Aybat, 2018; Alkousa et al., 2019; Daskalakis et al., 2017b; Ibrahim et al., 2020; Nouiehed et al., 2019; Mokhtari et al., 2020b; Lin et al., 2020a; Murty and Kabadi, 1985). In particular, for the case that the loss function is (strongly) convex with respect to the minimization variable and (strongly) concave with respect to maximization variable several efficient algorithms have been studied (Nesterov, 2007; Li and Lin, 2015; Mokhtari et al., 2020d), including the extragradient method (Korpelevich, 1976b; Nemirovski, 2004b) that is known to be optimal for this setting. However, all these methods suffer from two major limitations: (i) they are provably convergent only in convex-concave settings; (ii) they are designed for the settings that both minimization and maximization variables belong to continuous domains.

There has been some effort to address the first limitation by finding a first-order stationary point or locally stable point for the problems that are not convex-concave (Lin et al., 2020b; Diakonikolas et al., 2021; Yang et al., 2020; Sanjabi et al., 2018). However, these approaches fail to guarantee any global optimality as it is known that finding a saddle point in a general nonconvex-nonconcave setting is NP-hard (Jin et al., 2020). Nonetheless, it might be possible to achieve global approximation guarantees

Table 6.1: Algorithms performance guarantee. Here c_f is the cost of single computation of f, $c_{P_{\mathbf{x}}}$ and $c_{P_{\mathbf{y}}}$ are cost of projection in $\mathcal{X}$ and $\mathcal{Y}$, $c_{\nabla_{\mathbf{x}} f}$ is the cost of computing gradient of f with respect to $\mathbf{x}$, and $c_{\nabla_{\mathbf{x}} F}$ and $c_{\nabla_{\mathbf{y}} F}$ are the cost of computing gradient of multilinear extension F with respect to $\mathbf{x}$ and $\mathbf{y}$, respectively. k is the cardinality constraint ($|S| \leq k$) and n is size of the ground set $|V| = n$.

Alg.	Number of iterations	Approx. ratio	Cost per iteration	Cardinality const.	Matroid const.	Unbounded gradient
GG	$\mathcal{O}(1/\epsilon^2)$	$1-1/e$	$nk.c_f + c_{\nabla_{\mathbf{x}} f + c_{P_{\mathbf{x}}}}$	✓	✗	✗
GG	$\mathcal{O}(1/\epsilon^2)$	$1/2$	$nk.c_f + c_{\nabla_{\mathbf{x}} f + c_{P_{\mathbf{x}}}}$	✗	✓	✗
GRG	$\mathcal{O}(1/\epsilon^2)$	$1/2$	$(n+k)c_f + c_{\nabla_{\mathbf{x}} f + c_{P_{\mathbf{x}}}}$	✓	✗	✗
EGG	$\mathcal{O}(1/\epsilon^2)$	$1-1/e$	$2nk.c_f + 2c_{\nabla_{\mathbf{x}} f + 2c_{P_{\mathbf{x}}}}$	✓	✗	✓
EGG	$\mathcal{O}(1/\epsilon^2)$	$1/2$	$2nk.c_f + 2c_{\nabla_{\mathbf{x}} f + 2c_{P_{\mathbf{x}}}}$	✓	✓	✓
EGRG	$\mathcal{O}(1/\epsilon^2)$	$1/2$	$2(n+k)c_f + 2c_{\nabla_{\mathbf{x}} f + 2c_{P_{\mathbf{x}}}}$	✓	✗	✗
EGCE	$\mathcal{O}(1/\epsilon)$	$1/2$	$2c_{\nabla_{\mathbf{x}} F} + 2c_{\nabla_{\mathbf{y}} F + 2c_{P_{\mathbf{x}}} + 2c_{P_{\mathbf{y}}}}$	✓	✓	✓

for *structured* saddle minimax problems. Addressing the second limitations and developing methods for discrete-continuous domains or fully discrete domains requires exploiting tools from discrete optimization. Several recent works have considered applications involving specific discrete-continuous minimax problems and proposed structure-informed algorithms (Zhou and Bilmes, 2018). However, to our knowledge, there is no work that provides a principled algorithmic or theoretical framework to study minimax problems with mixed discrete-continuous components and it is not even clear if such problems allow for tractable solutions with global guarantees.

In this paper, we tackle these two issues and present iterative methods with theoretical guarantees to solve structured non convex-concave minimax problems, where the minimization variable is from a continuous domain and the maximization variable belongs to a discrete domain. Concretely, for a non-negative function $f : \mathbb{R}^d \times 2^V \to \mathbb{R}_+$, consider the minimax problem

$$\text{OPT} \triangleq \min_{\mathbf{x} \in \mathcal{X}} \max_{S \in \mathcal{I}} f(\mathbf{x}, S), \tag{6.1}$$

where $\mathbf{x}$ belongs to a convex set $\mathcal{X} \subset \mathbb{R}^d$ and S is a subset of the ground set V with n elements that is constrained to be inside a matroid $\mathcal{I}$. Given a fixed S, the function $f(\cdot, S)$ is convex with respect to the continuous (minimization) variable. Further, given a fixed $\mathbf{x}$, the function $f(\mathbf{x}, \cdot)$

is submodular with respect to the discrete (maximization) variable. We refer to this problem as *convex-submodular minimax problem.*

The convex-submodular minimax problem in (7.3) encompasses various applications. In Section 6.4, we describe specific optimization problems, such as convex-facility-location, as well as applications such as designing adversarial attacks on recommender systems. There are various other applications that can be cast into Problem (7.3), in particular, when convex models have to be learned while data points are selected or changed according to notions of summarization, diversity, and deletion. Examples include learning under data deletion (Ginart et al., 2019; Neel et al., 2020; Wu et al., 2020), robust text classification (Lei et al., 2018), minimax curriculum learning (Zhou and Bilmes, 2018; Zhou et al., 2021, 2020b; Soviany et al., 2021), minimax supervised learning (Farnia and Tse, 2016), and minimax active learning (Ebrahimi et al., 2020).

6.1.1. Our Contributions

In this paper, we provide a principled study of the problem defined in (7.3), from both theoretical and algorithmic perspectives, when f is convex in the minimization variable and submodular as well as *monotone* in the maximization variable[19]. We introduce efficient iterative algorithms for solving this problem and develop a theoretical framework for analyzing such algorithms with guarantees on the quality of the resulting solutions according to the notions of optimality that we define.

Notions of (near-)optimality and hardness results. For minimax problems, the strongest notion of optimality is defined through saddle points or their approximate versions. We first provide a negative result that shows finding a saddle point or any approximate version of it (which we term as an (α, ϵ)-saddle point) is NP-hard for general convex-submodular problems (Theorem 17). We thus introduce a slightly weaker notion of optimality that we call (α, ϵ)-approximate minimax solutions for Problem (7.3). Roughly speaking, the quality of the minimax objective at such solutions is at most $\frac{1}{\alpha}(\text{OPT} + \epsilon)$, and hence they are near-optimal when $\alpha < 1$. We show in Theorem 18 that obtaining such solutions for $\alpha > 1 - 1/e$ is NP-hard. This is a non-trivial result that does not readily follow from known hardness results in submodular maximization. Consequently, we focus

[19]For completeness, a function $g : 2^V \to \mathbb{R}$ is called submodular if for any two subsets $S, T \subseteq V$ we have: $g(S \cap T) + g(S \cup T) \leq g(S) + g(T)$. Moreover, g is called monotone if for any $S \subseteq T$ we have $g(S) \leq g(T)$.

on efficiently finding solutions in the regime of $\alpha \leq (1 - 1/e)$. We present several algorithms that achieve this goal and theoretically analyze their complexity and quality of their solution.

Algorithms with guarantees on convergence rate, complexity, and solution quality. Our proposed algorithms are as follows (see also Table 6.1): (i) *Greedy-based methods.* We first present Gradient-Greedy (GG), a method alternating between gradient descent for minimization and greedy for maximization. We further introduce Extra-Gradient-Greedy (EGG) that uses an extra-gradient step instead of gradient step for the minimization variable. We prove that both algorithms achieve a $((1 - 1/e), \epsilon)$-approximate minimax solution after $\mathcal{O}(1/\epsilon^2)$ iterations when $\mathcal{I}$ is a cardinality constraint. Importantly, EGG does not require the bounded gradient norm condition as opposed to GG. Our results for the case that $\mathcal{I}$ is a matroid constraint (see Table 6.1) are provided in the supplementary material. (ii) *Replacement greedy-based methods.* The greedy-based methods require $\mathcal{O}(nk)$ function computations at each iteration. To improve this complexity, we present alternating methods that use replacement greedy for the maximization part to reduce the cost of each iteration to $\mathcal{O}(n)$. The Gradient Replacement-Greedy (GRG) algorithm achieves a $(1/2, \epsilon)$-approximate minimax solution after $\mathcal{O}(1/\epsilon^2)$ iterations and Extra-Gradient Replacement-Greedy (EGRG) achieves a $(1/2, \epsilon)$-approximate minimax solution after $\mathcal{O}(1/\epsilon^2)$, when $\mathcal{I}$ is a cardinality constraint. (iii) *Continuous extension-based methods.* Note that all mentioned methods achieve a convergence rate of $\mathcal{O}(1/\epsilon^2)$. To improve this convergence rate, we further introduce the extra-gradient on continuous extension (EGCE) method that runs extra-gradient update on the continuous extension of the submodular function. We show that EGCE is able to achieve an $(1/2, \epsilon)$-approximate minimax solution after at most $\mathcal{O}(1/\epsilon)$ iterations, when $\mathcal{I}$ is a general matroid constraint.

6.1.2. Related Work

Several recent works have considered specific applications that require solving Problem (7.3) when f is nonconvex-submodular (Zhou and Bilmes, 2018; Lei et al., 2018; Mirzasoleiman et al., 2020). Zhou and Bilmes (2018) consider the problem of minimax curriculum learning which is a special case of minimax strongly convex-submodular optimization, and propose an algorithm similar to gradient-greedy (GG). They provide an upper bound on the distance between their obtained solution and the optimal solution when f is strongly convex in $\mathbf{x}$ and monotone-submodular in S with non-zero

curvature. Moreover, Lei et al. (2018) study designing an adversarial attack in text classification and show that for some specific neural network structures, the task of designing an adversarial attack can be formulated as submodular maximization, leading to a minimax nonconvex-submodular problem. An algorithm similar to gradient-greedy is then proposed by Lei et al. (2018) for designing attacks and it has led to successful experimental results. In contrast, this paper is the first to introduce a principled study of Problem (7.3) for general functions f with newly developed notions of optimality, algorithmic frameworks, and theoretical guarantees.

Another line of work is the literature on differentiable submodular maximization (Tschiatschek et al., 2018; Wilder et al., 2019; Sakaue, 2021) in which the goal is to find a smooth maximization oracle for submodular maximization to compute the gradient of the objective function. Another related work is "Submodular+ Concave"(Mitra et al., 2021b) in which authors studied the problems that can be written as a summation of submodular and concave function. Both of these works consider fundamentally different problems from our setting.

Another relevant line of work is the literature on robust submodular optimization (Krause et al., 2008a; Bogunovic et al., 2017b; Mirzasoleiman et al., 2017; Kazemi et al., 2018; Bogunovic et al., 2018; Iyer, 2021; Orlin et al., 2018; Chen et al., 2017a; Anari et al., 2019; Wilder, 2018; Bogunovic et al., 2017a; Mitrović et al., 2017). This setting corresponds to solving a *max-min* optimization problem which involves only *discrete variables*, and hence, it is different from our setting with fundamentally different methods. For such problems, finding discrete solutions with any approximation factor is NP-hard; and consequently, the literature has mostly focused on obtaining solutions that satisfy a bi-criteria approximation guarantee. Another related work is distributionally robust submodular maximization in (Staib et al., 2019) which is a special case of *max-min* version of Problem (7.3). In this setting, the inner minimization has a special structure that allows for a closed form solution, and hence, the problem can be solved by using appropriate techniques from continuous submodular optimization. We will derive the implication of our results on the max-min version of Problem (7.3) in the supplementary material.

For the minimax problem in (7.3), a natural goal is to find a so-called saddle point. Next, we formally define the notion of saddle point for Problem (7.3).

Definition 1. *A pair* $(\mathbf{x}^*, S^*)$ *is a saddle point of the function* f *if the following condition holds:*

$$\forall \mathbf{x} \in \mathcal{X}, S \in \mathcal{I} : \; f(\mathbf{x}^*, S) \leq f(\mathbf{x}^*, S^*) \leq f(\mathbf{x}, S^*) \tag{6.2}$$

Based on this definition, $(\mathbf{x}^*, S^*)$ is a saddle point of Problem (7.3), if there is no incentive to modify the minimization variable $\mathbf{x}^*$ when the maximization variable is fixed and equal to S^*, and, conversely, there is no incentive to change the maximization variable from S^* when the minimization variable is $\mathbf{x}^*$. In other words, a saddle point can be interpreted as an equilibrium.

There is a rich literature on efficient approaches for finding an ϵ-saddle point for convex-concave minimax optimization, where ϵ is an arbitrary positive constant (Thekumparampil et al., 2019). To define an ϵ-saddle point, we first need to define the duality gap, which is given by $D(\mathbf{x}, S) := \bar{\phi}(\mathbf{x}) - \underline{\phi}(S)$, where

$$\bar{\phi}(\mathbf{x}) := \max_{S \in \mathcal{I}} f(\mathbf{x}, S), \qquad \underline{\phi}(S) := \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, S).$$

Considering these definitions, we call a pair of solution ϵ-saddle point if their duality gap is at most ϵ.

Definition 2. *A pair* $(\bar{\mathbf{x}}, \bar{S})$ *is called an* $\epsilon-$*saddle point of* f *if it satisfies*

$$D(\bar{\mathbf{x}}, \bar{S}) = \bar{\phi}(\bar{\mathbf{x}}) - \underline{\phi}(\bar{S}) \leq \epsilon \tag{6.3}$$

One can verify that if we set $\epsilon = 0$, then Definitions 1 and 2 coincide, i.e., $(\mathbf{x}^*, S^*)$ satisfies (6.3) for $\epsilon = 0$ if and only if $(\mathbf{x}^*, S^*)$ satisfies (6.2). Hence, to derive a finite time analysis we often aim for finding an ϵ-saddle point. For instance, for smooth and convex-concave problems extra-gradient obtains an ϵ-saddle point after $\mathcal{O}(1/\epsilon)$ iterations (which is the optimal complexity).

However, for our convex-submodular setting, one cannot expect to find an ϵ-saddle point efficiently, as the special case of finding an ϵ-accurate solution for submodular maximization is in general NP-hard (Nemhauser and Wolsey, 1978; Wolsey, 1982; Krause and Golovin, 2014). Although solving the problem of maximizing a monotone submodular function subject to a matroid constraint is hard, one can find α-approximate solution of that in polynomial time, i.e., finding a solution that its function value is at least αOPT, where $\alpha \in (0, 1)$. Inspired by this observation, we introduce the notion of (α, ϵ)-saddle point for our convex-submodular setting.

Definition 3. *A pair* $(\hat{\mathbf{x}}, \hat{S})$ *is called an* $(\alpha, \epsilon)-$*saddle point of* f *if it satisfies*

$$\alpha\bar{\phi}(\hat{\mathbf{x}}) - \underline{\phi}(\hat{S}) \leq \epsilon. \tag{6.4}$$

Our first result is a negative result that shows even finding an (α, ϵ)-saddle point is not tractable.

Theorem 17. *Finding* (α, ϵ)*-saddle point for Problem* (7.3) *is NP-hard for any* $\alpha > 0$.

While this result shows intractability of finding (approximate) saddle-points for Problem (7.3), one avenue to provide solutions with guaranteed quality is to see whether we can find solutions that achieve a fraction of OPT. We thus proceed to introduce the notion of *approximate minimax solution*.

Definition 4. *We call a point* $\hat{\mathbf{x}}$ *an* (α, ϵ)*-approximate minimax solution of Problem* (7.3) *if it satisfies*

$$\alpha\bar{\phi}(\hat{\mathbf{x}}) \leq \text{OPT} + \epsilon, \tag{6.5}$$

where $\text{OPT} = \min_{\mathbf{x}\in\mathcal{X}} \bar{\phi}(\mathbf{x}) = \min_{\mathbf{x}\in\mathcal{X}} \max_{S\in\mathcal{I}} f(\mathbf{x}, S)$.

Next, we describe the notion of an (α, ϵ)-approximate minimax solution for Problem (7.3). The minimax problem in (7.3) can be interpreted as a sequential game, where we first select an action $\mathbf{x}$ and then an adversary chooses a set S to maximize our loss $f(\mathbf{x}, S)$. In this case, our goal is to find $\mathbf{x}$ that minimizes the loss obtained by the worst possible action by the adversary, i.e., we aim to minimize the function $\bar{\phi}(\hat{\mathbf{x}}) := \max_{S\in\mathcal{I}} f(\hat{\mathbf{x}}, S)$ over the choice of $\hat{\mathbf{x}}$. Indeed, finding the exact minimizer is also hard and we should seek approximate solutions. Hence, our goal is to find solutions

$\hat{\mathbf{x}}$ whose worst-case loss $\bar{\phi}(\hat{\mathbf{x}})$ is only a factor larger than the best possible loss $\text{OPT} = \min_{\mathbf{x}\in\mathcal{X}} \bar{\phi}(\mathbf{x})$. That said, by finding an (α, ϵ)-approximate minimax solution for Problem (7.3) we obtain a solution whose loss is at most $(\text{OPT} + \epsilon)/\alpha$, where $0 < \alpha \leq 1$ and $\epsilon > 0$.

The task of finding an $\hat{\mathbf{x}}$ that is (α, ϵ)-approximate minimax solution is easier than finding a pair $(\tilde{\mathbf{x}}, \tilde{S})$ that is an (α, ϵ)-saddle point, since if the pair $(\tilde{\mathbf{x}}, \tilde{S})$ satisfies (6.4), then $\tilde{\mathbf{x}}$ satisfies (6.5):

$$\alpha\bar{\phi}(\tilde{\mathbf{x}}) - \underline{\phi}(\tilde{S}) \leq \epsilon \;\Rightarrow\; \alpha\bar{\phi}(\tilde{\mathbf{x}}) - \max_{S\in\mathcal{I}} \underline{\phi}(S) \leq \epsilon$$
$$\Rightarrow\; \alpha\bar{\phi}(\tilde{\mathbf{x}}) - \min_{\mathbf{x}\in\mathcal{X}} \bar{\phi}(\mathbf{x}) \leq \epsilon$$

Hence, the condition in (6.4) is more strict compared to (6.5). In fact, in the next section, we show that unlike the task of finding an (α, ϵ)-saddle point of Problem (7.3) that is NP-hard for any $\alpha \in (0, 1]$, one can find an (α, ϵ)-approximate minimax solution of Problem (7.3) in poly-time for $\alpha \in (0, 1 - 1/e]$. Alas, the problem is still NP-hard for $\alpha \in (1 - 1/e, 1]$ as we show in Theorem 18.

Theorem 18. *Let $\alpha = 1 - 1/e + \gamma$ for a positive constant $\gamma > 0$. If there exists a polynomial time algorithm and a polynomial time oracle that can achieve an (α, ϵ)-approximate solution for any choice of the function $f(\mathbf{x}, S)$ in problem* (7.3)*, then $P = NP$.*

We emphasize that Theorem 18 does not follow directly from that fact that submodular maximization beyond $(1 - 1/e)$-approximation is hard, and hence it is non-trivial. Indeed, one naive way to argue for the proof of this theorem (which is incorrect) is to consider functions $f(\mathbf{x}, S)$ whose output does not depend on the variable $\mathbf{x}$, i.e. $f(\mathbf{x}, S) = f(S)$, and use the hardness results for submodular optimization. But for such functions *any* point $\mathbf{x}$ is an optimal solution (with $\alpha = 1$). Hence, the proof of the theorem (provided in the appendix) requires a novel idea beyond trivial consequences of known results for submodularity.

So far we have shown two results: (i) Finding an approximate (α, ϵ)-saddle point is hard for $\alpha > 0$. (ii) We introduced the notion of (α, ϵ)-approximate solution and showed that for $\alpha > 1-1/e$ finding an (α, ϵ)-approximate solution is hard. The only missing piece is showing whether or not it is possible

to *efficiently* find an (α, ϵ)-approximate minimax solution when $\alpha \leq 1-1/e$. In the rest of the paper, we provide an affirmative answer to this question and present methods achieving this goal.

6.3. ALGORITHMS

In this section, we present a set of algorithms that are able to find an (α, ϵ)-approximate minimax solution of Problem (7.3). To present these algorithms, we first present two subroutines that we use in the implementation of our algorithms[20]: (i) greedy update and (ii) replacement greedy update.

Greedy subroutine. In the greedy update, for a fixed minimization variable $\mathbf{x}$, we select a subset S with k elements in a greedy fashion, i.e., we sequentially pick k elements that maximize the marginal gain. Specifically, if we define $\Delta_e f(\mathbf{x}, S) = f(\mathbf{x}, S \cup \{e\}) - f(\mathbf{x}, S)$ as the marginal gain of element e, in the greedy update, for a given variable $\mathbf{x}$ we perform the update

$$S_{i+1} = S_i \cup \{\arg\max \Delta_e f(\mathbf{x}, S)\}, \tag{6.6}$$

for $i = 0, \ldots, k-1$, where S_0 is the empty set. The output of this process is S_k with k elements. We use the notation $\textsc{Greedy}(f, k, \mathbf{x})$ for the greedy subroutine, which takes function f, cardinality constraint parameter k, and variable $\mathbf{x}$ as inputs, and returns a set S by performing (6.6) for k steps.

Replacement greedy subroutine. In the replacement greedy update (Mitrovic et al., 2018; Schrijver, 2003; Stan et al., 2017a), for a given variable $\mathbf{x}$ and set S, the output is an updated set S^+ whose function value at $\mathbf{x}$ is larger than the one for S, i.e., $f(\mathbf{x}, S) \leq f(\mathbf{x}, S^+)$. The procedure for finding the new set S^+ is relatively simple. If the size of the input set S is less than k, we add one more element to the set S that maximizes the marginal gain and the resulted set would be S^+. In other words, if $|S| < k$,

$$S^+ = S \cup \{\arg\max \Delta_e f(\mathbf{x}, S)\}. \tag{6.7}$$

If the size of the input set S is k, we first remove one element of the set S that leads to minimum decrease in the function value (denoted by e^*) and then replace it with another element of the ground

[20]For better exposition, we consider the case that $\mathcal{I}$ is k-carnality constraint and refer to Appendix for matroids.

set that maximizes the marginal gain. Hence, if $|S| = k$, we have

$$S^+ = (S \setminus \{e^*\}) \cup \{\arg\max \Delta_e f(\mathbf{x}, S \setminus \{e^*\})\}, \tag{6.8}$$

where $e^* = \arg\max_{e \in S}\{f(\mathbf{x}, S \setminus e)\}$.

We use the notation RepGreedy$(f, k, \mathbf{x}, S)$ for the replacement greedy subroutine. Note that replacement greedy is computationally cheaper than greedy, as it requires only one pass over the ground set, while greedy requires k passes.

6.3.1. Greedy-based Algorithms

Next, we present greedy-based methods to find (α, ϵ)-approximate minimax solutions for Problem (7.3).

Gradient Greedy. The first algorithm that we present is Gradient Greedy (GG), which uses a projected gradient descent step to update the minimization iterate $\mathbf{x}_t$ at each iteration, i.e., $\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$, and then uses a greedy procedure to update the maximization variable S_t. This update is performed in an alternating fashion, where we first use $\mathbf{x}_t$ and S_t to find $\mathbf{x}_{t+1}$ and then we use the updated variable $\mathbf{x}_{t+1}$ to compute S_{t+1}. Note that the final output of this process is a weighted average of all variables $\mathbf{x}_t$ that are observed from time $t = 1$ to $t = T$, defined as $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \mathbf{x}_t$. The steps of GG are summarized in Algorithm 10 option I.

Next, we show that GG is able to find a $(1 - 1/e, \epsilon)$-approximate minimax solution after $\mathcal{O}(1/\epsilon^2)$ iterations. To prove this claim we require the following assumptions on the objective function f.

Assumption 13. *The function f is L-smooth with respect to $\mathbf{x}$, i.e., for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, S \in \mathcal{I}$, we have $\|\nabla_{\mathbf{x}} f(\mathbf{x}, S) - \nabla_{\mathbf{x}} f(\mathbf{x}', S)\| \leq L\|\mathbf{x} - \mathbf{x}'\|$.*

Assumption 14. *The gradient of function f with respect to $\mathbf{x}$ is uniformly bounded by a constant M, i.e., for any $\mathbf{x} \in \mathbb{R}^d, S \in 2^V$, we have $\|\nabla_{\mathbf{x}} f(\mathbf{x}, S)\| \leq M$.*

Theorem 19. *Consider Gradient Greedy (GG) in Algorithm 10 option I. If f is convex-submodular*

Algorithm 10
Option I: Gradient Greedy (GG)
Option II: Extra-Gradient Greedy (EGG)

Initialize the set S_1 to $\emptyset$ and variable $\mathbf{x}_1$ to zero.
for $t = 1$ to T **do**
  Option I: $\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$
    $S_{t+1} = \textsc{Greedy}(f, k, \mathbf{x}_{t+1})$
  Option II: $\hat{\mathbf{x}}_t = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$
    $\hat{S}_t = \textsc{Greedy}(f, k, \hat{\mathbf{x}}_t)$
    $\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\hat{\mathbf{x}}_t, \hat{S}_t))$
    $S_{t+1} = \textsc{Greedy}(f, k, \mathbf{x}_{t+1})$
end for
Option I: $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \mathbf{x}_t$
Option II: $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \hat{\mathbf{x}}_t$

and Assumptions 13-14 hold, then the output of this algorithm after $\mathcal{O}(1/\epsilon^2)$ *iterations with step size* $\mathcal{O}(\epsilon)$*, is a* $((1-1/e), \epsilon)$*-approximate minimax solution of Problem* (7.3) .

The smoothness assumption (Assumption 13) is required to guarantee convergence of gradient-based methods at the rate of $1/\epsilon^2$. The bounded gradient assumption (Assumption 14), however, comes from the fact that even in convex-concave problems gradient descent-ascent algorithms only converge when the gradient norm is uniformly bounded. This issue has been addressed in the convex-concave setting by the update of extra-gradient method which converges to a saddle point only under smoothness assumption. However, this improvement is not for free and it requires two gradient computations per update, instead of one. Next, we leverage this technique to present an alternating method that obtains the approximation factor and iteration complexity of GG without requiring Assumption 14.

Extra-gradient greedy. We now present the Extra-Gradient Greedy (EGG) algorithm, which consists of two gradient updates as suggested by extra-gradient and two greedy steps to find the auxiliary set $\hat{S}_t$ and the updated set S_{t+1}. In the extra-gradient method, we take a preliminary step to find a middle/auxiliary point and then compute the next iterate using the gradient information of the middle point. If we consider $\mathbf{x}_t$ and S_t as the current iterates, we first run a gradient step to find the auxiliary minimization variable according to the update $\hat{\mathbf{x}}_t = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$

Algorithm 11
Option I: Gradient Replacement-greedy (GRG)
Option II: Extra-gradient Replacement-greedy (EGRG)

Initialize the set S_1 to $\emptyset$ and variable $\mathbf{x}_1$ to zero.
for $t = 1$ to T **do**
Option I: $\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$
$S_{t+1} = \textsc{RepGreedy}(f, k, \mathbf{x}_{t+1}, S_t)$
Option II: $\hat{\mathbf{x}}_t = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, S_t))$
$\hat{S}_t = \textsc{RepGreedy}(f, k, \mathbf{x}_t, S_t)$
$\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\hat{\mathbf{x}}_t, \hat{S}_t))$
$S_{t+1} = \textsc{RepGreedy}(f, k, \hat{\mathbf{x}}_t, \hat{S}_t)$
end for
Option I: $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \mathbf{x}_t$
Option II: $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \hat{\mathbf{x}}_t$

then we compute the auxiliary set $\hat{S}_t$ by performing a greedy step based on the auxiliary iterate $\hat{\mathbf{x}}_t$, i.e., $\hat{S}_t = \textsc{Greedy}(f, k, \hat{\mathbf{x}}_t)$. Once $\hat{\mathbf{x}}_t$ and $\hat{S}_t$ are computed, we update the minimization variable $\mathbf{x}_{t+1}$ by descending towards a gradient evaluated at $(\hat{\mathbf{x}}_t, \hat{S}_t)$, i.e., $\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla f(\hat{\mathbf{x}}_t, \hat{S}_t))$. Lastly, we compute the new set S_{t+1} by running a greedy update based on the new iterate $\mathbf{x}_{t+1}$, i.e., $S_{t+1} = \textsc{Greedy}(f, k, \mathbf{x}_{t+1})$. Steps of EGG are outlined in Algorithm 10 (option II).

Next we establish our theoretical result for Extra-gradient Greedy and show that only under smoothness assumption it finds an $(1-1/e, \epsilon)$-approximate minimax solution after $\mathcal{O}(1/\epsilon^2)$ iterations.

Theorem 20. *Consider Extra-Gradient Greedy (EGG) outlined in Algorithm 10 option II. If f is convex-submodular and Assumption 13 holds, then the output of this algorithm after $\mathcal{O}(1/\epsilon^2)$ iterations with step size $\mathcal{O}(\epsilon)$, is a $((1-1/e), \epsilon)$-approximate minimax solution of Problem* (7.3).

Remark 6. *Note that as both GG and EGG are greedy based methods, they can also be used for the case of general matroid constraint. However, the approximation guarantee would be* $1/2$ *instead of* $1-1/e$. *The details are provided in the supplementary material.*

6.3.2. Replacement Greedy-based Methods

As we showed earlier, for the cardinality constraint problem GG and EGG achieve the optimal approximation guarantee of $1-1/e$ for the minimax problem in (7.3). However, they both require

running greedy updates at each iteration which makes their per iteration complexity $\mathcal{O}(nk)$. To resolve this issue, we propose the use of replacement-greedy in lieu of greedy update. This modification reduces the complexity of each iteration to $\mathcal{O}(n+k)$ at the cost of lowering the approximation factor.

Gradient replacement-greedy. We first present the Gradient Replacement-Greedy (GRG) algorithm which alternates between a gradient update and a replacement greedy update. As shown in Algorithm 11 option I, the only difference between GRG and GG algorithms is the substitution of greedy update with replacement greedy. Next, we establish the theoretical guarantee of GRG.

Theorem 21. *Consider the Gradient Replacement-Greedy (GRG) algorithm in Algorithm 11 option I. If f is convex-submodular and Assumptions 13-14 hold, then the output of this algorithm after $\mathcal{O}(1/\epsilon^2)$ iterations with step size $\mathcal{O}(\epsilon)$, is a $(1/2,\epsilon)$-approximate minimax solution of Problem* (7.3).

Extra-gradient replacement-greedy. The GRG algorithm requires the bounded gradient assumption similar to GG. To address this issue, a natural idea is to exploiting the extra-gradient approach for updating $\mathbf{x}$ and introducing the Extra-gradient Replacement-greedy (EGRG) algorithm, outlined in Algorithm 11 option II. However, unlike the case of Greedy-based methods, here we can not drop the bounded gradient assumption by exploiting the idea of extra-gradient update. Next, we elaborate on this issue.

Note that to prove that EGRG finds a $(1/2,\epsilon)$-approximate minimax solution we need to find an upper bound on $f(\hat{\mathbf{x}}_t, S) - 2f(\hat{\mathbf{x}}_t, \hat{S}_t)$ for every S. To establish such a bound, we need to relate $f(\hat{\mathbf{x}}_t, \hat{S}_t)$ to $f(\mathbf{x}_t, \hat{S}_t)$ which requires the function f to be Lipschitz with respect to $\mathbf{x}$, which is equivalent to the bounded gradient condition in Assumption 14; see proof of Theorem 2 in the appendix for more details. Note that such argument is not required for the EGG method, as in greedy based method we always have the following inequality $f(\hat{\mathbf{x}}_t, S) - (1-1/e)^{-1} f(\hat{\mathbf{x}}_t, \hat{S}_t) \leq 0$ for every S. As a result, the required conditions for the convergence of GRG and EGRG are similar and we only state EGRG results for completeness.

Theorem 22. *Consider Extra-Gradient Replacement Greedy(EGRG) in Algorithm 11 option II. If f is convex-submodular and Assumptions 13-14 hold, then the output of EGRG after $\mathcal{O}(1/\epsilon^2)$ iterations*

with step size $\mathcal{O}(\epsilon)$, is a $(1/2, \epsilon)$-approximate minimax solution of (7.3).

6.3.3. Extra-gradient on Continuous Extension

So far all proposed algorithms achieve (α, ϵ)-approximate minimax solutions in $\mathcal{O}(1/\epsilon^2)$ iterations. In this section, we investigate the possibility of achieving a faster rate of $\mathcal{O}(1/\epsilon)$. Note that, in the discussed algorithms, the update for the discrete variable is not smooth and the iterates jump from one set to another in consecutive iterations, which results in slowing down the convergence. To overcome this limitation, we introduce the continuous multi-linear extension of Problem (7.3); for introduction to multi-linear extension of submodular maximization problems see (Vondrák, 2007; Calinescu et al., 2011; Badanidiyuru and Vondrák, 2014; Feldman et al., 2011; Hassani et al., 2017; Sadeghi and Fazel, 2020). As we will show, the continuous extension of Problem (7.3) is equivalent to its original version, and by extending the extra-gradient methodology to this setting we achieve a convergence rate of $\mathcal{O}(1/\epsilon)$ for the case that $\mathcal{I}$ is a matroid.

Definition 5. *The* continuous extension *of a function $f : \mathbb{R}^d \times 2^V \to \mathbb{R}_+$ is the function $F : \mathbb{R}^d \times [0,1]^n \to \mathbb{R}_+$ defined as $F(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{S \sim \mathbf{y}}[f(\mathbf{x}, S)]$, where $S \sim \mathbf{y}$ is a random set wherein each element i is included with probability y_i independently.*

We show that for convex-submodular problems we have (see Proposition 1 in Appendix 6.6.8):

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{S \in \mathcal{I}} f(\mathbf{x}, S) = \min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{K}} F(\mathbf{x}, \mathbf{y}), \tag{6.9}$$

where $\mathcal{I}$ is assumed to be a matroid constraint and $\mathcal{K}$ is the corresponding base polytope($\mathcal{K} = \mathbf{conv}\{1_S : S \in \mathcal{I}\}$). We present Extra-Gradient on Continuous Extension (EGCE) in Algorithm 12 which applies the updates of extra-gradient on the continuous extension function $F(\mathbf{x}, \mathbf{y})$.

Theorem 23. *Consider the Extra-Gradient On Continuous Extension (EGCE) algorithm outlined in Algorithm 12. If f is convex-submodular and Assumptions 13-14 hold, then the output of this algorithm after $\mathcal{O}(1/\epsilon)$ iterations with constant step size is a $(1/2, \epsilon)$-approximate minimax solution of Problem* (7.3).

Algorithm 12 Extra-gradient on Continuous Extension

Initialize the variables $\mathbf{y}_1$ and $\mathbf{x}_1$ to zero.
for $t = 1$ to T **do**
$\hat{\mathbf{x}}_t = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla_x F(\mathbf{x}_t, \mathbf{y}_t))$
$\hat{\mathbf{y}}_t = \pi_{\mathcal{K}}(\mathbf{y}_t + \gamma_t \nabla_y F(\mathbf{x}_t, \mathbf{y}_t))$
$\mathbf{x}_{t+1} = \pi_{\mathcal{X}}(\mathbf{x}_t - \gamma_t \nabla_x F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t))$
$\mathbf{y}_{t+1} = \pi_{\mathcal{K}}(\mathbf{y}_t + \gamma_t \nabla_y F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t))$
end for
Return solution $\mathbf{x}_{sol} = (\sum_{t=1}^{T} \gamma_t)^{-1} \sum_{t=1}^{T} \gamma_t \hat{\mathbf{x}}_t$

6.4. EXPERIMENTS

In this section, we study two specific instances of Problem (7.3): (i) convex-facility location functions along with a synthetic experimental setup, and (ii) designing adversarial attacks for item recommendation which is a real world application of our framework.

Convex-facility location functions. Consider the function $f : \mathbb{R}^d \times 2^V \to \mathbb{R}_+$ defined as $f(\mathbf{x}, S) = \sum_{i=1}^{n} \max_{j \in S} f_{i,j}(\mathbf{x}) + g(\mathbf{x})$, where $g : \mathbb{R}^d \to \mathbb{R}$ and $f_{i,j} : \mathbb{R}^d \to \mathbb{R}$ are convex. Indeed, $f(\mathbf{x}, S)$ is convex with respect to $\mathbf{x}$. Also, for a fixed $\mathbf{x}$, we recover the objective of the facility location problem, which is submodular and monotone. To introduce our setup, suppose $\mathbf{x} \in \mathbb{R}_+^d$ can be written as the concatenation of $n = d/m$ vectors $\mathbf{x}_i \in \mathbb{R}_+^m$ of size m, i.e., $\mathbf{x} = [\mathbf{x}_1; \ldots; \mathbf{x}_n]$. In our experiments, we assume that the function $f_{i,j}(\mathbf{x})$ is defined as $f_{i,j}(\mathbf{x}) = \mathbf{x}_i^T Q_{i,j} \mathbf{x}_j$, where $Q_{i,j} \in \mathcal{S}_{++}^m$ is a positive definite matrix and all of its elements are also positive, i.e., $Q_{i,j} > 0$. Moreover, we consider the case that the regularization function g is defined as $g(\mathbf{x}) := \lambda(\sum_{i=1}^{n} \|\mathbf{x}_i\|^2)^{-1}$, and the constraint set for the minimization variable $\mathbf{x}$ is defined as $\mathcal{C} := \{\mathbf{x} = [\mathbf{x}_1; \ldots; \mathbf{x}_n] | \|\mathbf{x}_i\| \leq 1, \text{ for } i = 1, \ldots, n\}$. Considering these definitions the convex-submodular minimax optimization problem that we aim to solve can be written as

$$\min_{\mathbf{x}_i \in \mathcal{C}} \max_{|S| \leq k} \sum_{i=1}^{n} \max_{j \in S} \mathbf{x}_i^T Q_{i,j} \mathbf{x}_j + \lambda \Big(\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 \Big)^{-1} \tag{6.10}$$

where the constraint on the maximization variable S is a cardinality constraint of size k. For our numerical experiments, we tested two cases, in the first case we set the problem parameters as

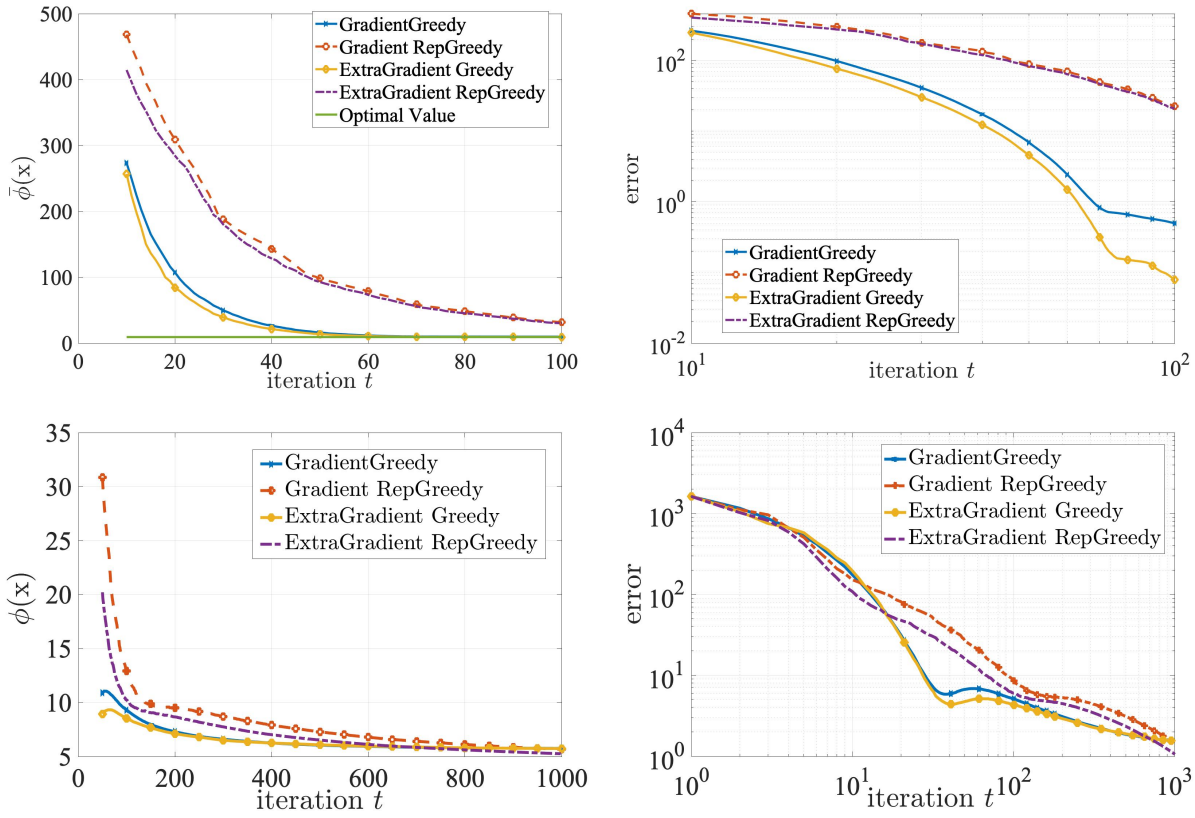

Figure 6.1: Comparison of our proposed methods for convex-facility location functions(case I and Case II)

$m = 10$, $n = 30$, $k = 5$, and $\lambda = 1$ and in the second case we set the problem parameters as $m = 10$, $n = 50$, $k = 10$, and $\lambda = 1$.

Case I ($m = 10$, $n = 30$, $k = 5$, $\lambda = 1$). In this case, we choose m, n to be small so that we can solve the inner max in problem (6.10) and compute $\bar{\phi}(\mathbf{x}) = \max_{|S|\leq k} f(\mathbf{x}, S)$ exactly using search over all the subsets of size k. We report $\bar{\phi}(\mathbf{x}_t)$ as well as optimal value of problem (6.10). Results in Figure 6.1. (first plot) show that the algorithms converge to the optimal minimax value. We also demonstrate the relative error of these algorithms $\text{error}_t := \phi(\mathbf{x}_t) - \text{OPT}$ in second plot. As we observe in Figure 6.1 (second plot), greedy based methods converge faster than replacement greedy based algorithms in terms of iteration complexity.

Case II ($m = 10$, $n = 50$, $k = 10$, $\lambda = 1$). We now investigate the behavior of our proposed methods

for solving (6.10) in the second case when k, n are relatively larger. Note that exact computation of $\bar{\phi}(\mathbf{x}) = \max_{|S|\leq k} f(\mathbf{x}, S)$ is not computationally tractable for this case, since it requires solving a submodular maximization problem. Hence, in third plot in figure 6.1, we report the value of the function $\phi(\mathbf{x}) := f(\mathbf{x}, \textsc{Greedy}(f, k, \mathbf{x}))$ which is an approximation for $\bar{\phi}(\mathbf{x})$. In other words, instead of computing $\bar{\phi}(\mathbf{x})$ which is the maximum of $f(\mathbf{x}, S)$ over the choice of S, we report $\phi(\mathbf{x})$ which is the value of $f(\mathbf{x}, S)$ when S is obtained via the greedy method. The convergence paths of $\phi(\mathbf{x}_t)$ for our proposed methods are reported in the third plot of Figure 6.1. We further show the relative error of these algorithms defined as $\text{error}_t := \phi(\mathbf{x}_t) - \phi(\mathbf{x}_T)$ in the fourth plot to better compare their convergence rates.

Adversarial Attack for Item Recommendation. In this section, we study the application of designing an adversarial attack for a movie recommendation task. Consider a (completed) rating matrix X whose entries $X_{i,j}$ correspond to the estimated rating that user i has given to movie j. Given a rating matrix X, the recommender system chooses k movies via maximizing the utility function $\max_{|S|\leq k} h(X, S) := \frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} \max_{j\in S} X_{u,j}$ where $\mathcal{U}$ is the set of all users. The attacker's goal is to slightly perturb the rating matrix X to a matrix X' such that the utility $\max_{|S|\leq l} h(X', S)$ is minimized. Therefore, the attacker aims at solving the minimax problem

$$\min_{\substack{\|X'-X\|_F\leq\epsilon \\ 0\leq X'_{i,j}\leq 5}} \max_{|S|\leq k} h(X', S), \tag{6.11}$$

where $\|.\|_F$ is the Frobenius norm. Note that $h(X, S)$ is convex-submodular (convexity in $\mathbf{x}$ is clear, and the function $h(\mathbf{x}, S)$ is a facility location function in S). Hence, this problem is an instance of Problem (7.3). To evaluate the performance of our methods, we consider movie recommendation on the Movielens dataset (Harper and Konstan, 2015). We pick 2000 most rated movies with 200 users with highest number of rates for these movies (similar to (Stan et al., 2017b; Adibi et al., 2020)) and we set $k = 10$. The adversary has a power to manipulate up to 0.5% of movies ratings on average (i.e. $\epsilon = 0.5 \times 0.01 \times 200 \times 2000$). We plot $\phi(X_{\text{alg}})$ in each iteration as a measure of effectiveness of our algorithms and compare it to the case that there is no attack. Figure 6.2 shows the comparison of our algorithms. As we can see in Figure 6.2, the facility location based recommendation systems

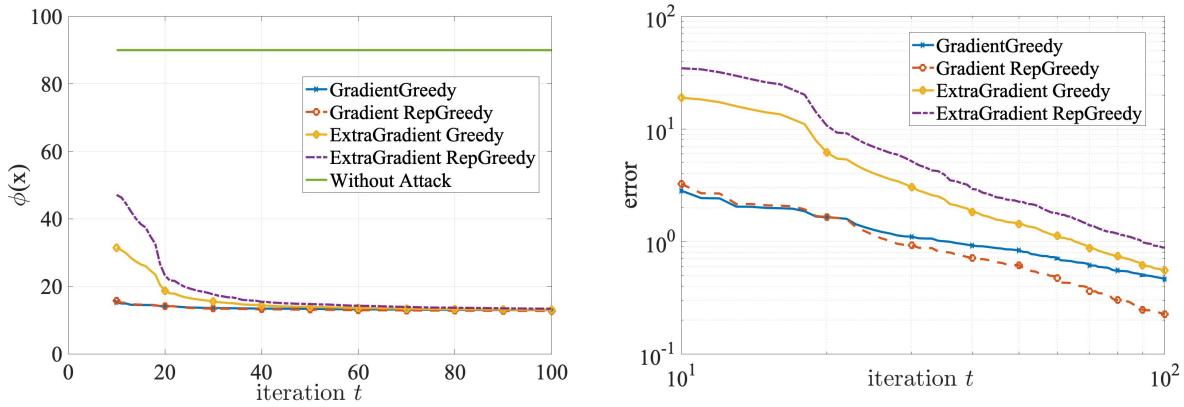

Figure 6.2: Comparison of our proposed methods for for Problem (6.11) (the green line is the performance of the recommender system when there is no adversary)

are extremely vulnerable to adversarial attacks and the performance drops from 90 (when there is no adversary) to around 12 when we have attacks.

6.5. CONCLUSION

In this paper, we introduced and studied the convex-submodular minimax problem in (7.3). We defined multiple notions of (near-) optimality and provided hardness results regarding these notions in various regimes. In particular, one of the notions was (α, ϵ)-approximate minimax solution. We showed that for $\alpha > 1 - 1/e$ finding an (α, ϵ)-approximate minimax solution is hard. For $\alpha \leq 1 - 1/e$, we proposed five algorithms and characterized their theoretical guarantees in different settings. The main take-away message from our algorithmic procedures is that, if the function f has bounded gradient, then one can use the GG Algorithm, or the GRG algorithm which has a better complexity albeit it has a worse approximation factor. If the gradient of f is not uniformly bounded, then one has to resort to the proposed EGG algorithm.

An interesting future direction is to find more computationally efficient algorithms for harder constraints such as matroid constraint with $1 - 1/e$ factor. We believe gradient continuous-greedy will achieve the $1 - 1/e$ factor on matroid constraints; However, it needs to run the continuous greedy algorithm each iteration which is computationally costly.

6.6. Appendix

6.6.1. Proof of Theorem 17

Consider the function $f : \mathbb{R}^d \times 2^V \to \mathbb{R}_+$, where $f(\mathbf{x}, .)$ is submodular for every $\mathbf{x}$ and $f(., S)$ is convex for every S. Then, the maxmin convex-submodular problem is an optimization problem where the maximization is over continuous variable and minimization is over a discrete variable as

$$\text{OPT}_{maxmin} \triangleq \max_{S \in \mathcal{I}} \min_{x \in \mathcal{X}} f(\mathbf{x}, S), \tag{6.12}$$

Let us define the notion of approximate solution for maxmin problem as follows:

Definition 6. *We call a point $\hat{S}$ an (α, ϵ)-approximate maxmin solution of Problem* (6.12) *if it satisfies*

$$\alpha \underline{\phi}(\hat{S}) \geq \text{OPT}_{\text{maxmin}} - \alpha\epsilon, \tag{6.13}$$

We know any $(\alpha, \epsilon)-$saddle point, denoted by $(\bar{\mathbf{x}}, \bar{S})$, has the following properties:

1. $\underline{\phi}(\bar{S}) > \alpha.\text{OPT}_{maxmin} - \epsilon$
2. $\bar{\phi}(\bar{\mathbf{x}}) < \frac{1}{\alpha}.\text{OPT}_{minmax} + \frac{\epsilon}{\alpha}$

This is due to the fact that we have:

1. $\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \bar{S}) \leq \min_{\mathbf{x} \in \mathcal{X}} \max_{S \in \mathcal{I}} f(\mathbf{x}, S) = \text{OPT}_{minmax}$
2. $\text{OPT}_{maxmin} = \max_{S \in \mathcal{I}} \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, S) \leq \max_{S \in \mathcal{I}} f(\bar{\mathbf{x}}, S)$

these two conditions imply that by finding an $(\alpha, \epsilon)-$saddle point we find an $\alpha-$approximate solution for the minimax problem (7.3) and a $\frac{1}{\alpha}-$approximate solution for the max-min problem (6.12). In order to prove finding $(\alpha, \epsilon)-$ saddle point is NP-hard, we will prove that finding approximate solution for maxmin convex-submodular is NP-hard. We do this establishing a connection between this problem and the problem of robust submodular maximization through following result stated

and proved in (Krause et al., 2008a).

Consider monotone-submodular functions $f_1, f_2, \dots f_n$ and the following robust submodular maximization problem:

$$\mathrm{OPT}_1 = \max_{|S|\leq k} \min_{i\in[n]} f_i(S). \tag{6.14}$$

Solving this problem up to approximation factor is NP-hard, i.e. finding a solution S such that $\max_i f_i(S) \geq \alpha \mathrm{OPT}_1$ is an NP-hard task for any $\alpha > 0$.

Now, consider the following problem:

$$\mathrm{OPT}_2 = \max_{|S|\leq k} \min_{\substack{x\in\mathbb{R}^n, x\geq 0 \\ x^T \mathbb{1}=1}} \sum_{i=1}^{n} x_i.f_i(S) \tag{6.15}$$

where $\mathbb{1}$ is vector of all ones and $x = [x_1, x_2, x_3 \dots x_n]^T$. For this problem, it is easy to verify that $\mathrm{OPT}_1 = \mathrm{OPT}_2$ since for every set $S \in V$ we have $\min_{i\in[n]} f_i(S) = \min_{\substack{x\in\mathbb{R}^n, x\geq 0 \\ x^T \mathbb{1}=1}} \sum_{i=1}^{n} x_i.f_i(S)$. Therefore, finding a $\alpha-$approximate solution for problem in (6.15) is NP-hard. Problem (6.15) is max-min convex-submodular optimization which means max-min convex-submodular optimization is NP-hard in general. We show that by finding $(\alpha, \epsilon)-$saddle point we can provide $\frac{1}{\alpha}-$approximate solution for max-min problem; therefore, since we proved finding $\frac{1}{\alpha}-$approximate solution for max-min problem is NP-hard, finding $(\alpha, \epsilon)-$saddle point is NP-hard too.

6.6.2. Proof of Theorem 18

Before stating this proof, let us explain what we mean by "NP-hard" for the considered setting. We note that an algorithm for Problem (7.3) is supposed to search for an approximate solution only in $\mathcal{X}$ (i.e., in terms of the variable $\mathbf{x}$), and for this, it will require some information about the values $f(\mathbf{x}, S)$. However, for every fixed $\mathbf{x}$, there may be restrictions on obtaining some specific values of $f(\mathbf{x}, S)$. For example, finding the exact value of $\bar{\phi}(\mathbf{x})$ can in general be NP-hard (as maximizing a monotone-submodular function beyond $(1 - 1/e)$ approximation is hard). In order to appropriately address these restrictions, we will view our setting as a procedure between the algorithm and an oracle that we now describe below.

Upon receiving an input point $\mathbf{x}_{in} \in \mathcal{X}$, the oracle chooses based on this input a set S_{out} such that $|S_{out}| \leq k$, and returns all the following information: the set S_{out}, the value $f(\mathbf{x}_{in}, S_{out})$, and the gradient of $f(\mathbf{x}_{in}, S_{out})$ with respect to $\mathbf{x}$ at the point $(\mathbf{x}_{in}, S_{out})$. The only restriction on the oracle is that it is a polynomial-time oracle, i.e. the oracle's procedure to find the set S_{out} requires poly-time complexity in terms of the size of the ground-set $|V|$. More precisely, there exists an integer $q > 0$ such that for any ground set V, the oracle uses at most $\mathcal{O}(|V|^q)$ operations to find the output set S_{out} corresponding to an input $\mathbf{x}_{in}$. Note that we do not put any restriction on what the oracle does apart from having poly-time complexity; e.g. it could output a greedy solution, or it could output a random set S_{out}, or it could do any other procedure. We call such an oracle a polynomial-time oracle. Given this choice of the oracle, the algorithm proceeds in m rounds, and in each round $r \in [m]$, it chooses an input point $\mathbf{x}_r \in \mathcal{X}$ to query from the oracle. Importantly, we consider algorithms that require a polynomial number of rounds in terms of the size of the ground set V. More precisely, for any ground set V and α, ϵ, the number of rounds of the algorithm is at most $c(\alpha, \epsilon)|V|^q$ where $q > 0$ is an absolute constant and $c(\alpha, \epsilon)$ is another constant that only depends on α and ϵ. We call such an algorithm a **polynomial-time algorithm**. Next, we show that no polynomial-time algorithm is capable of finding an (α, ϵ)-approximate of (7.3) for $\alpha > 1 - 1/e$.

In the following, we assume for simplicity that $\epsilon = 0$. The proof can be trivially extended to any value of ϵ, as we will explain at the end of the proof. Recall from the statement of the theorem that $\alpha = 1 - 1/e + \gamma$ for a fixed constant $\gamma > 0$.

We know for a fact that monotone-submodular maximization beyond the $(1 - 1/e)$-approximation in NP-hard (Krause and Golovin, 2014). I.e. unless P = NP, for any integer $q > 0$ there exists a monotone submodular function $g_1 : 2^V \to \mathbb{R}_+$ and an integer $k \leq |V|$ such that finding a set S with cardinally k where $g_1(S) \geq (1 - 1/e + \gamma/3) \max_{|S| \leq k} g_1(S)$ requires computing more than $|V|^q$ function values (i.e. complexity is larger than $|V|^q$). Consider such a function g_1 and the choice of k, and define $\text{OPT}_{g_1} = \max_{|S| \leq k} g_1(S)$. We also define another function $g_2 : 2^V \to \mathbb{R}_+$ as follows: $g_2(S) = \min\{g_1(S), (1 - 1/e + \gamma/3)\text{OPT}_{g_1}\}$. It is important to note that finding a set $|S| \leq k$ such that $g_1(S) \neq g_2(S)$ requires complexity larger than $|V|^q$ (also note that the choice of q is arbitrary

here, i.e. for every q there exists a g_1, etc.).

Consider the integer $n = \lceil 4/\gamma + 1 \rceil$ and let $\mathcal{X}$ be the n-dimensional simplex, i.e.

$$\mathcal{X} = \{\mathbf{x} = (x_1, \cdots, x_n) \text{ s.t. } \sum_{i=1}^{n} x_i = 1 \,\&\, x_i \geq 0 \ \forall i = 1, \cdots, n\}.$$

For $j \in \{1, \cdots, n\}$ we define $f_j(\mathbf{x}, S) : \mathcal{X} \times 2^V \to \mathbb{R}_+$ as

$$f_j(\mathbf{x}, S) = \sum_{i \in [n], i \neq j} x_i g_1(S) + x_j g_2(S)$$

We note a few facts about each of the functions $f_j(\mathbf{x}, S)$:

(i) Any $(\alpha, 0)$ approximate solution for the function f_j has the property that $x_j > 1/n + \gamma/4$. This is simply because for any $(\alpha, 0)$-approximate solution $\mathbf{x} = (x_1, \cdots, x_n)$ we have $\alpha\bar{\phi}(\mathbf{x}) \leq \min_{\mathbf{x} \in \mathcal{X}} \max_{|S| \leq k} f_j(\mathbf{x}, S)$, and hence

$$\alpha \left(x_j(1 - 1/e + \gamma/3) + (1 - x_j)\right) \mathrm{OPT}_{g_1} \leq (1 - 1/e + \gamma/3)\mathrm{OPT}_{g_1}$$

From the above inequality (and by noting that $\gamma \in (0, 1]$) we can always deduce that $x_j > \gamma/2$, and thus $x_j > 1/n + \gamma/4$.

(ii) Given a polynomial-time oracle, we can not distinguish between the functions $f_1, \ldots, f_n$ using a query from the oracle. This is because the oracle can not find a set S with carnality at most k for which $g_1(S) \neq g_2(S)$ (as finding that set by the oracle is intractable), and thus, for the set S_{out} that the oracle finds, the outcome of the oracle will be the function value $f(\mathbf{x}_{\text{in}}, S_{\text{out}}) = g_1(S_{\text{out}})$ and $\nabla_{\mathbf{x}} f(\mathbf{x}_{\text{in}}, S_{\text{out}}) = g_1(S_{\text{out}})\mathbf{1}_n$ where $\mathbf{1}_n$ is the all-ones vector of dimension n. These outputs bear absolutely no information about the index j.

Given the above facts, we are now ready to finalize the proof. Consider the scenario where the index

j is chosen uniformly at random inside the set $[n]$, and the algorithm aims at finding an approximate solution of the function f_j. Note that the choice of j is hidden to the algorithm. Now, given fact (ii) above, if both the algorithm and oracle are polynomial-time, then in all the rounds and queries, there will be absolutely no information revealed about the index j. As a result, the mutual information between the outcome of the queries and the index j will be zero.

On the other hand, from fact (i) above, if an algorithm can find an $(\alpha, 0)$-approximate solution, we claim that the solution is informative about the index j. More precisely, given the solution that the algorithm has found, we can infer the hidden index j using the following procedure: the algorithm's solution $\mathbf{x} = (x_1, \cdots, x_n)$ can be viewed as a probability distribution over the set $\{1, \cdots, n\}$. As a result, if we use this probability distribution to draw an integer $\hat{j}$ from the set $\{1, \cdots, n\}$, then we have $\Pr\{\hat{j} = j\} = x_j >= 1/n + \gamma/4$. Thus, we can decode the index j with a probability that is strictly larger than a random guess. This means that the mutual information of the solution found by the algorithm and the index j is strictly lower-bounded by a positive constant (which only depends on γ). This contradicts the result of the previous paragraph.

Note that in the above we have assumed that $\epsilon = 0$. For general ϵ, we note that we can always choose the function g_1 such that OPT_{g_1} is sufficiently large. As a result, we can write $\epsilon = \epsilon' \times \mathrm{OPT}_{g_1}$ where ϵ' can be made arbitrarily small. Hence, proving hardness for obtaining an (α, ϵ)-approximate becomes equivalent to proving harness for obtaining an $(\alpha/(1+\epsilon'), 0)$ approximate solution. The conclusion is now immediate since out proof above works for any $\alpha = 1 - 1/e + \gamma$ and ϵ' can be made arbitrarily small by making OPT_{g_1} sufficiently large.

6.6.3. Proof of Theorem 19: Gradient Greedy(GG) Convergence

Proof. Let us pick $\gamma_t = \alpha$.Then, we can write the following based on update $x_{t+1} = \pi_{\mathcal{X}}(x_t - \gamma_t \nabla f(x_t, S_t))$ in algorithm 10 and assumption 14.

$$\|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \leq \|\mathbf{x}_t - \alpha\nabla f(\mathbf{x}_t, S_t) - \mathbf{x}\|^2 = \|\mathbf{x}_t - \mathbf{x}\|^2 + \|\alpha\nabla f(\mathbf{x}_t, S_t)\|^2 - 2\langle\alpha\nabla f(\mathbf{x}_t, S_t), \mathbf{x}_t - \mathbf{x}\rangle \tag{6.16}$$

$$\leq \|\mathbf{x}_t - \mathbf{x}\|^2 + \alpha^2 M^2 - 2\langle\alpha\nabla f(\mathbf{x}_t, S_t), \mathbf{x}_t - \mathbf{x}\rangle \tag{6.17}$$

which results in

$$2\alpha(f(\mathbf{x}_t, S_t) - f(\mathbf{x}, S_t)) \leq 2\langle\alpha\nabla f(\mathbf{x}_t, S_t), \mathbf{x}_t - \mathbf{x}\rangle \leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + \alpha^2 M^2 \tag{6.18}$$

and finally

$$f(\mathbf{x}_t, S_t) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{1}{2\alpha}(-\|\mathbf{x}_t - \tilde{\mathbf{x}}\|^2 + \|\mathbf{x}_{t-1} - \tilde{\mathbf{x}}\|^2) + \frac{1}{2}\alpha M^2 \tag{6.19}$$

summing up over t we have:

$$\sum_{t=1}^{T} f(\mathbf{x}_t, S_t) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{1}{2\alpha}(\|\mathbf{x}_0 - \tilde{\mathbf{x}}\|^2) + \frac{1}{2}\alpha T M^2 \tag{6.20}$$

our set of continuous variable is bounded which means $\|\mathbf{x}\|^2 \leq H$; this results:

$$\sum_{t=1}^{T} f(\mathbf{x}_t, S_t) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{H}{2\alpha} + \frac{1}{2}\alpha T M^2 \tag{6.21}$$

Also, from greedy update we have for every S(check (Krause and Golovin, 2014)):

$$f(\mathbf{x}_{t-1}, S) - \frac{f(\mathbf{x}_{t-1}, S_t)}{1 - \frac{1}{e}} \leq 0 \tag{6.22}$$

Now, using the Lipschitz condition (consequence of Assumption 14):

$$|f(\mathbf{x}_t, S_t) - f(\mathbf{x}_{t-1}, S_t)| \leq M\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq M\alpha\|\nabla f(\mathbf{x}_{t-1}, S_t)\| \leq M^2\alpha \tag{6.23}$$

Putting (6.22) and (6.23) together:

$$(1 - \frac{1}{e})f(\mathbf{x}_t, S) - f(\mathbf{x}_t, S_t) \leq 2M^2\alpha \tag{6.24}$$

and summing over t we have:

$$\sum_{t=1}^{T}(1 - \frac{1}{e})f(\mathbf{x}_t, S) - f(\mathbf{x}_t, S_t) \leq 2M^2\alpha T \tag{6.25}$$

From (6.21) and (6.25) we can then obtain the following:

$$\sum_{t=1}^{T}(1 - \frac{1}{e})f(\mathbf{x}_t, S) - f(\tilde{\mathbf{x}}, S_t) \leq 2M^2\alpha T + \frac{H}{2\alpha} + \frac{1}{2}\alpha T M^2 \tag{6.26}$$

and finally:

$$\frac{\sum_{t=1}^{T}\alpha((1 - \frac{1}{e})f(\mathbf{x}_t, S) - f(\tilde{\mathbf{x}}, S_t))}{\sum_{t=1}^{T}\alpha} \leq \frac{3M^2\alpha T + \frac{H}{2\alpha}}{T} \tag{6.27}$$

From convexity we have:

$$f(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t, S) \leq \frac{\sum_{t=1}^{T} f(\mathbf{x}_{t-1}, S)}{T} \tag{6.28}$$

which results in:

$$(1-\frac{1}{e})f(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t,S)-\frac{\sum_{t=1}^{T}\alpha(f(\tilde{\mathbf{x}},S_t))}{\sum_{t=1}^{T}\alpha}\leq\frac{3M^2\alpha T+\frac{H}{2\alpha}}{T} \tag{6.29}$$

Defining $\mathbf{x}^*=\arg\min\max f(\mathbf{x},S)$, we know that $\min_{\mathbf{x}}\max_S f(\mathbf{x},S)=\max f(\mathbf{x}^*,S)\geq f(\mathbf{x}^*,S_t)$. Now in (6.29) we let $\tilde{\mathbf{x}}=\mathbf{x}^*$ and write:

$$(1-\frac{1}{e})\max_S f(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t,S)-\min_x\max_S f(\mathbf{x},S)\leq\frac{3M^2\alpha T+\frac{H}{2\alpha}}{T} \tag{6.30}$$

Letting $\alpha=\frac{1}{\sqrt{T}}$ we obtain:

$$(1-\frac{1}{e})\max_S f(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t,S)-\min_{\mathbf{x}}\max_S f(\mathbf{x},S)\leq\frac{3M^2+\frac{H}{2}}{\sqrt{T}} \tag{6.31}$$

Finally, if we define $K=3M^2+\frac{H}{2}$ and let $T=\frac{K^2}{\epsilon^2}$; then $\mathbf{x}_{sol}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$ is a $(1-1/e,\epsilon)$-approximate minimax solution.

6.6.4. Proof of Theorem 21: Gradient Replacement-greedy(GRG) Convergence

Proof. Let g be a monotone-submodular function, and consider sets $B,S\subseteq V$ with size k. Define $e^*=\arg\max_{e\in S}g(S\setminus e)-g(S)$, and $v^*=\arg\max_{v\in V}g(S\cup v\setminus e^*)-g(S\cup v\setminus e^*)$. We have:

$$\begin{aligned}g(S\cup v^*\setminus e^*)-g(S)&\geq\frac{1}{k}\sum_{v\in B}g(S\cup v\setminus e^*)-g(S)\\&=\frac{1}{k}\sum_{v\in B}(g(S\cup v\setminus e^*)-g(S\cup v)+g(S\cup v)-g(S))\end{aligned} \tag{6.32}$$

where the first inequality comes from the definition of v^*. We know that for a monotone-submodular function g we have $g(B\cup S)-g(S)\leq\sum_{v\in B}(g(S\cup v)-g(S))$ for any choice of B,S (Stan et al., 2017b); which results in:

$$\frac{1}{k}\sum_{v\in B}(g(S\cup v)-g(S))\geq\frac{1}{k}(g(B\cup S)-g(S))\geq\frac{1}{k}(g(B)-g(S)) \tag{6.33}$$

Here, the first inequality is due to submodularity and the second inequalities is due to monotonicity. Also, we have:

$$\begin{aligned}\frac{1}{k}\sum_{v\in B}(g(S\cup v)-g(S\cup v\setminus e^*))&\leq\frac{1}{k}\sum_{v\in B}(g(S)-g(S\setminus e^*))=g(S)-g(S\setminus e^*)\\&\leq\frac{1}{k}\sum_{e\in S}(g(S)-g(S\setminus e))\leq\frac{1}{k}g(S)\end{aligned} \tag{6.34}$$

where the first and second inequality comes from submodularity. Combining (6.32),(6.33), and (6.34) we have that for every set B of size k:

$$g(S\cup v^*\setminus e^*)-g(S)\geq\frac{1}{k}(g(B)-2g(S)) \tag{6.35}$$

If we apply (6.35) for the replacement greedy update in Gradient Replacement-greedy(GRG) algorithm, we obtain:

$$f(\mathbf{x}_t,S_t)-f(\mathbf{x}_t,S_{t-1})\geq\frac{1}{k}\left(f(\mathbf{x}_t,S)-2f(\mathbf{x}_t,S_{t-1})\right) \tag{6.36}$$

and hence

$$f(\mathbf{x}_t,S)-2f(\mathbf{x}_t,S_t)\leq(1-\frac{2}{k})(f(\mathbf{x}_t,S)-2f(\mathbf{x}_t,S_{t-1})) \tag{6.37}$$

Note that as f is M-Lipschitz we have for every S (consequence of Assumption 14):

$$|f(\mathbf{x}_t,S)-f(\mathbf{x}_{t-1},S)|\leq M\|\mathbf{x}_t-\mathbf{x}_{t-1}\|\leq M\alpha\|\nabla f(\mathbf{x}_{t-1},S_t)\|\leq M^2\alpha \tag{6.38}$$

Combining (6.37) and (6.38) we obtain that

$$f(\mathbf{x}_t, S) - 2f(\mathbf{x}_t, S_t) \leq (1 - \frac{2}{k})(f(\mathbf{x}_{t-1}, S) - 2f(\mathbf{x}_{t-1}, S_{t-1}) + 3M^2\alpha) \tag{6.39}$$

Using a recursive argument we can show that

$$f(\mathbf{x}_t, S) - 2f(\mathbf{x}_t, S_t) \leq (1 - \frac{2}{k})^t(f(\mathbf{x}_0, S) - 2f(\mathbf{x}_0, S_0)) + \sum_{m=1}^{t}(1 - \frac{2}{k})^m 3M^2\alpha \tag{6.40}$$

Now since $f(\mathbf{x}_0, S_0)$ is non-negative, we can eliminate $-f(\mathbf{x}_0, S_0)$ from the right hand side. Using this observation and by simplifying the geometric sum we obtain that

$$f(\mathbf{x}_t, S) - 2f(\mathbf{x}_t, S_t) \leq (1 - \frac{2}{k})^t f(\mathbf{x}_0, S) + 3M^2\alpha\frac{k}{2} \tag{6.41}$$

Now, note that $(1 - \frac{2}{k})^t$ is bounded above by $e^{-\frac{2t}{k}}$ and therefore we have

$$f(\mathbf{x}_t, S) - 2f(\mathbf{x}_t, S_t) \leq Ae^{-\frac{2t}{k}} + 3M^2\alpha\frac{k}{2}, \tag{6.42}$$

where A is an upper bound for function value at point zero, $f(0, S) \leq A$. Now, from the analysis of gradient descent similar to (6.21), we have:

$$\sum_{t=1}^{T} f(\mathbf{x}_t, S_t) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{H}{2\alpha} + \frac{1}{2}\alpha T M^2 \tag{6.43}$$

Combining this inequality with (6.42) we have:

$$\sum_{t=1}^{T}\frac{1}{2}f(\mathbf{x}_t, S) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{H}{2\alpha} + \frac{\sum_{t=1}^{T} Ae^{-\frac{2t}{k}}}{2} + \frac{5TM^2\alpha k}{4} \leq \frac{K}{2\alpha} + \frac{Ae^{-\frac{2}{k}}}{2(1 - e^{-\frac{2}{k}})} + \frac{5TM^2\alpha k}{4} \tag{6.44}$$

Thus, choosing the parameters $\alpha = \frac{1}{\sqrt{T}}$ will lead to

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}f(\mathbf{x}_t, S) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{H}{2\sqrt{T}} + \frac{Ae^{-\frac{2}{k}}}{2T(1 - e^{-\frac{2}{k}})} + \frac{5M^2 k}{4\sqrt{T}} \leq \frac{K}{\sqrt{T}} \tag{6.45}$$

where K is some constant.

In summary, we have obtained the following relation that will be used to drive the guarantee for the minimax problem:

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}f(\mathbf{x}_t, S) - f(\tilde{\mathbf{x}}, S_t) \leq \frac{K}{\sqrt{T}} \tag{6.46}$$

We know because of convexity we have:

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{t-1}, S\right) \leq \frac{\sum_{t=1}^{T} f(\mathbf{x}_{t-1}, S)}{T} \tag{6.47}$$

Now combining (6.46) and (6.47) we have:

$$\frac{1}{2}f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t, S\right) - \frac{1}{T}\sum_{t=1}^{T} f(\tilde{\mathbf{x}}, S_t) \leq \frac{K}{\sqrt{T}} \tag{6.48}$$

Also for $\mathbf{x}^* = \arg\min_x \max_S f(\mathbf{x}, S)$ we have $\min_{\mathbf{x}} \max_S f(\mathbf{x}, S) = \max_S f(\mathbf{x}^*, S) \geq f(\mathbf{x}^*, S_t)$. By using $\tilde{\mathbf{x}} = \mathbf{x}^*$ we can write:

$$\frac{1}{2}f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t, S\right) - \min_{\mathbf{x}}\max_{S} f(\mathbf{x}, S) \leq \frac{1}{2}f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t, S\right) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^*, S_t) \leq \frac{K}{\sqrt{T}} \tag{6.49}$$

Let $T = \frac{K^2}{\epsilon^2}$; then $\mathbf{x}_{sol} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$ is a $(1/2, \epsilon)$- approximate minimax solution.

6.6.5. Proof of Theorem 20: Extra-gradient Greedy(EGG) Convergence

Consider the Extra-gradient Greedy method, we can write the following equations to find the bound on convergence of $\mathbf{x}$:

$$\begin{aligned}
&\|\hat{\mathbf{x}}_t - \mathbf{x}\|^2 \\
&\leq \|\mathbf{x}_t - \mathbf{x} - \gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\mathbf{x}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_t) \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + 2(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_t) \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 - 2\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2
\end{aligned} \tag{6.50}$$

Hence, we have

$$2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) \leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \tag{6.51}$$

Similarly we can show that

$$\begin{aligned}
&\|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \\
&\leq \|\mathbf{x}_t - \mathbf{x} - \gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_t - \mathbf{x}) + \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) + \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 + 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 + 2(\mathbf{x}_t - \mathbf{x}_{t+1})^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) + \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 - 2\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2
\end{aligned} \tag{6.52}$$

Hence, we have

$$2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) \leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \tag{6.53}$$

Now note that we can write $2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x})$ as

$$\begin{aligned}
& 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) && (6.54)\\
= \; & 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) + 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) && (6.55)\\
= \; & 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) + 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) && (6.56)\\
& + 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) && (6.57)\\
= \; & 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) + 2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}) && (6.58)\\
& + 2\gamma_t \left(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t) \right)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) && (6.59)\\
\leq \; & \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 && (6.60)\\
& + \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 && (6.61)\\
& + 2\gamma_t \left(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) \right) && (6.62)\\
= \; & -\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 && (6.63)\\
& + 2\gamma_t \left(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t) \right)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}), && (6.64)
\end{aligned}$$

where the inequality follows from the results in (6.51) and (6.53).

Next we derive an upper bound for the inner product $\left(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t) \right)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1})$

using the smoothness of the function f, i.e.,

$$\begin{aligned}
&\Big(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)\Big)^{\top} (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) \\
&\leq \|\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)\| \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| \\
&\leq \Big(\|\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, S_t)\| + \|\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, S_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)\|\Big) \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| \\
&\leq \Big(L_{\mathbf{x},S} \|\hat{S}_t - S_t\| + L_{\mathbf{x},\mathbf{x}} \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|\Big) \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|
\end{aligned}$$

Now to complete our upper bound we need to bound $\|\hat{S}_t - S_t\|$ which can be done as

$$\|\hat{S}_t - S_t\| \leq \phi \|\hat{\mathbf{x}}_t - \mathbf{x}_t\| + \sigma \tag{6.65}$$

The above relation holds because for every two feasible set we have $\|A - B\| \leq 2k$;therefore, if we let $\sigma = 2k$ and $\phi = 1$ the above condition is always true. Considering this result we obtain that

$$\begin{aligned}
\Big(\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t) - \nabla_{\mathbf{x}} f(\mathbf{x}_t, S_t)\Big)^{\top} (\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}) \leq (L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}) \|\hat{\mathbf{x}}_t - \mathbf{x}_t\| \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| \\
+ L_{\mathbf{x},S}\sigma \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|
\end{aligned}$$

Applying this upper bound into (6.54) implies that

$$
\begin{aligned}
&2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) \\
&\leq \; -\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \\
&\qquad + 2\gamma_t \left(L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}\right) \|\hat{\mathbf{x}}_t - \mathbf{x}_t\| \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| + 2\gamma_t L_{\mathbf{x},S}\sigma \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| \\
&\leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \\
&\qquad + \left[-\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + 2\gamma_t \left(L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}\right) \|\hat{\mathbf{x}}_t - \mathbf{x}_t\| \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|\right] \\
&\qquad + 2\gamma_t L_{\mathbf{x},S}\sigma \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\| \\
&\leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \\
&\qquad + \gamma_t \left(L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}\right) \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + \gamma_t \left(L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}\right) \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 + 4\gamma_t^2 L_{\mathbf{x},S}^2 \sigma^2 \\
&\qquad + \frac{1}{4}\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 \\
&\leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + 4\gamma_t^2 L_{\mathbf{x},S}^2 \sigma^2
\end{aligned}
$$

where the third inequality holds because of the fact that $2ab \leq a^2 + b^2$, and the last inequality holds since we assume $\gamma_t(L_{\mathbf{x},S}\phi + L_{\mathbf{x},\mathbf{x}}) \leq 3/4$.

Using this result we have that

$$
2\gamma_t \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) \leq \|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 + 4\gamma_t^2 L_{\mathbf{x},S}^2 \sigma^2
$$

Now by convexity of f with respect to $\mathbf{x}$ we have

$$
\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t, \hat{S}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) \geq f(\hat{\mathbf{x}}_t, \hat{S}_t) - f(\mathbf{x}, \hat{S}_t)
$$

and therefore

$$
f(\hat{\mathbf{x}}_t, \hat{S}_t) - f(\mathbf{x}, \hat{S}_t) \leq \frac{1}{2\gamma_t} \left(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2\right) + 2\gamma_t L_{\mathbf{x},S}^2 \sigma^2
$$

Moreover we know that

$$f(\hat{\mathbf{x}}_t, S) - \frac{1}{1-1/e} f(\hat{\mathbf{x}}_t, \hat{S}_t) \leq 0$$

Hence,

$$\begin{aligned} f(\hat{\mathbf{x}}_t, \hat{S}_t) - f(\mathbf{x}, \hat{S}_t) + (1-1/e) f(\hat{\mathbf{x}}_t, S) - f(\hat{\mathbf{x}}_t, \hat{S}_t) \\ \leq \frac{1}{2\gamma_t} \left(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \right) + 2\gamma_t L_{\mathbf{x},S}^2 \sigma^2 \end{aligned} \tag{6.66}$$

Let $\gamma_t = \frac{1}{\sqrt{T}}$. Then, since $\|\mathbf{x}\|^2 \leq H$, we have

$$\frac{1}{T} \sum_{t=1}^{T} -f(\mathbf{x}, \hat{S}_t) + (1-1/e) f(\hat{\mathbf{x}}_t, S) \leq \frac{H}{2\sqrt{T}} + \frac{2L_{\mathbf{x},S}^2 \sigma^2}{\sqrt{T}} \tag{6.67}$$

Let $K = 2L_{\mathbf{x},S}^2 \sigma^2 + K/2$ then we have

$$\frac{1}{T} \sum_{t=1}^{T} -f(\mathbf{x}, \hat{S}_t) + (1-1/e) f(\hat{\mathbf{x}}_t, S) \leq \frac{K}{\sqrt{T}} \tag{6.68}$$

We know because of convexity

$$f(\frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{x}}_{t-1}, S) \leq \frac{\sum_{t=1}^{T} f(\hat{\mathbf{x}}_{t-1}, S)}{T} \tag{6.69}$$

Now combining (6.68) and (6.69) we have

$$(1-1/e) f(\frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{x}}_t, S) - \frac{1}{T} \sum_{t=1}^{T} f(\tilde{\mathbf{x}}, \hat{S}_t) \leq \frac{K}{\sqrt{T}} \tag{6.70}$$

Also for $\mathbf{x}^* = \arg\min_{\mathbf{x}} \max_S f(\mathbf{x}, S)$, we have $\min_{\mathbf{x}} \max_S f(\mathbf{x}, S) = \max_S f(\mathbf{x}^*, S) \geq f(\mathbf{x}^*, S_t)$. We

let $\tilde{\mathbf{x}} = \mathbf{x}^*$ and write

$$(1 - 1/e) f(\frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t, S) - \min_{\mathbf{x}} \max_{S} f(\mathbf{x}, S) \tag{6.71}$$

$$\leq (1 - 1/e) f(\frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t, S) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^*, \hat{S}_t) \leq \frac{K}{\sqrt{T}} \tag{6.72}$$

Let $T = \frac{K^2}{\epsilon^2}$; then, $\mathbf{x}_{sol} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t$ is an $((1 - 1/e), \epsilon)$ approximate minimax solution.

6.6.6. Extra-gradient Replacement-greedy(EGRG) Convergence

For the analysis with respect to $\mathbf{x}$, we can show that

$$f(\hat{\mathbf{x}}_t, \hat{S}_t) - f(\mathbf{x}, \hat{S}_t) \leq \frac{1}{2\gamma_t}\left(\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2\right) + 2\gamma_t L_{\mathbf{x},S}^2 \sigma^2$$

therefore:

$$\frac{\sum_{t=1}^{T} \gamma_t \left[f(\hat{\mathbf{x}}_t, \hat{S}_t) - f(\mathbf{x}, \hat{S}_t)\right]}{\sum_{t=1}^{T} \gamma_t} \leq \frac{\|\mathbf{x} - \mathbf{x}_1\|^2}{2\sum_{t=1}^{T} \gamma_t} + \frac{\sum_{t=1}^{T} 2\gamma_t^2 L_{x,S}^2 \sigma^2}{\sum_{t=1}^{T} \gamma_t} \leq \frac{K_1}{\sqrt{T}} \tag{6.73}$$

It remains to derive an upper bound for $f(\hat{\mathbf{x}}_t, S) - 2f(\hat{\mathbf{x}}_t, \hat{S}_t)$. According to the update of replacement-greedy method, we can write the following inequalities:

$$f(\mathbf{x}_t, \hat{S}_t) - f(\mathbf{x}_t, S_t) \geq \frac{1}{k}\left(f(\mathbf{x}_t, S) - 2f(\mathbf{x}_t, S_t)\right) \tag{6.74}$$

and

$$f(\hat{\mathbf{x}}_t, S_{t+1}) - f(\hat{\mathbf{x}}_t, \hat{S}_t) \geq \frac{1}{k}\left(f(\hat{\mathbf{x}}_t, S) - 2f(\hat{\mathbf{x}}_t, \hat{S}_t)\right) \tag{6.75}$$

Using the second expression, we can write

$$\frac{1}{k}\Big(f(\hat{\mathbf{x}}_{t-1}, S) - 2f(\hat{\mathbf{x}}_{t-1}, \hat{S}_{t-1})\Big) \leq f(\hat{\mathbf{x}}_{t-1}, S_t) - f(\hat{\mathbf{x}}_{t-1}, \hat{S}_{t-1}) \tag{6.76}$$

Let $\bar{\phi}(\mathbf{x}_t) = \max_{|S|\leq k} f(\mathbf{x}_t, S)$; if we assume for every $\mathbf{x}, S$, $\|\nabla_{\mathbf{x}} f(\mathbf{x}, S)\| \leq G$ then we have:

$$|\bar{\phi}(\mathbf{x}) - \bar{\phi}(\mathbf{y})| \leq G||\mathbf{x} - \mathbf{y}|| \tag{6.77}$$

hence

$$\bar{\phi}(\hat{\mathbf{x}}_{t-1}) - 2f(\hat{\mathbf{x}}_{t-1}, S_t) \leq (1 - \frac{2}{k})(\bar{\phi}(\hat{\mathbf{x}}_{t-1}) - 2f(\hat{\mathbf{x}}_{t-1}, \hat{S}_{t-1})) \tag{6.78}$$

Note that

$$\begin{aligned} f(\hat{\mathbf{x}}_{t-1}, S_t) &\leq f(\mathbf{x}_t, S_t) + L_x||\mathbf{x}_t - \hat{\mathbf{x}}_{t-1}|| \\ &\leq f(\mathbf{x}_t, \hat{S}_t) + \gamma_t G^2 \qquad (6.79) \\ &\leq f(\hat{\mathbf{x}}_t, \hat{S}_t) + 2\gamma_t G^2, \qquad (6.80) \end{aligned}$$

therefore,

$$\bar{\phi}(\hat{\mathbf{x}}_{t-1}) - 2f(\hat{\mathbf{x}}_t, \hat{S}_t) \leq (1 - \frac{2}{k})(\bar{\phi}(\hat{\mathbf{x}}_{t-1}) - 2f(\hat{\mathbf{x}}_{t-1}, \hat{S}_{t-1})) + 4\gamma_t G^2 \tag{6.81}$$

Also, note that

$$|\bar{\phi}(\hat{\mathbf{x}}_t) - \bar{\phi}(\hat{\mathbf{x}}_{t-1})| \leq G^2\gamma_t. \tag{6.82}$$

Putting (6.81), (6.80) and (6.82) together, we obtain:

$$\bar{\phi}(\hat{\mathbf{x}}_t) - 2f(\hat{\mathbf{x}}_t, \hat{S}_t) \leq (1 - \frac{2}{k})(\bar{\phi}(\hat{\mathbf{x}}_{t-1}) - 2f(\hat{\mathbf{x}}_{t-1}, \hat{S}_{t-1})) + 5\gamma_t G^2 \tag{6.83}$$

let $\gamma_t = \frac{1}{\sqrt{T}}$ and $\bar{\phi}(\hat{\mathbf{x}}_0) - 2f(\hat{\mathbf{x}}_0, \hat{S}_0) = A_0$, then

$$\sum_{t=1}^{T}\gamma_t(\bar{\phi}(\hat{\mathbf{x}}_t)-2f(\hat{\mathbf{x}}_t,\hat{S}_t))\leq\sum_{t=1}^{T}\frac{1}{\sqrt{T}}(\sum_{t=0}^{t-1}\frac{5G^2}{\sqrt{T}}(1-\frac{2}{k})^t+(1-\frac{2}{k})^tA_0)$$

$$\leq\sum_{t=1}^{T}\frac{1}{\sqrt{T}}(\frac{5kG^2}{2\sqrt{T}}+(1-\frac{2}{k})^tA_0) \tag{6.84}$$

$$\leq k\beta \tag{6.85}$$

where $\beta=\frac{5kG^2}{2}+\frac{kA_0}{2\sqrt{T}}$ and finally for update of S we get:

$$\frac{\sum_{t=1}^{T}\gamma_t(\bar{\phi}(\hat{\mathbf{x}}_t)-2f(\hat{\mathbf{x}}_t,\hat{S}_t))}{\sum_{t=1}^{T}\gamma_t}\leq\frac{k\beta}{\sqrt{T}} \tag{6.86}$$

Adding up (6.86) and (6.73) we have:

$$\frac{\sum_{t=1}^{T}\gamma_t\left[0.5\bar{\phi}(\hat{\mathbf{x}}_t)-f(\mathbf{x},\hat{S}_t)\right]}{\sum_{t=1}^{T}\gamma_t}\leq\frac{K_1}{\sqrt{T}}+\frac{k\beta}{\sqrt{T}}\leq\frac{K}{\sqrt{T}} \tag{6.87}$$

from this for every S

$$\frac{\sum_{t=1}^{T}\gamma_t\left[0.5f(\hat{\mathbf{x}}_t,S)-f(\mathbf{x},\hat{S}_t)\right]}{\sum_{t=1}^{T}\gamma_t}\leq\frac{K}{\sqrt{T}} \tag{6.88}$$

Similar to (6.69) and (6.70), (6.87) results $\mathbf{x}_{sol}=\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{x}}_t$, to be $(1/2,\epsilon)$-approximate minimax solution.

6.6.7. Maxmin Result

In this section, we introduce maxmin convex-submodular problem and discuss how we can exploit the algorithms described in the previous sections for the maxmin problem. Formally, consider the function $f:\mathbb{R}^d\times 2^V\rightarrow\mathbb{R}_+$, where $f(\mathbf{x},.)$ is submodular for every $\mathbf{x}$ and $f(.,S)$ is convex for every S.

Then, the maxmin convex-submodular problem is an optimization problem where the maximization is over continuous variable and minimization is over a discrete variable as

$$\text{OPT}_{maxmin} \triangleq \max_{S\in\mathcal{I}} \min_{x\in\mathcal{X}} f(\mathbf{x}, S), \tag{6.89}$$

Due to hardness of the max-min problem as we stated in Theorem 17 and Appendix 6.6.1, we cannot drive the same result for the maxmin problem as we did for minimax problem. In general, finding an approximation solution for problem (6.89) is NP-hard. Our result as stated in theorem 24 proves that $\cup_{t=1}^{T} S_t$ is an approximate solution for (6.89) which has a larger cardinality than our cardinality constraint (at most Tk elements). Although, the set $\cup_{t=1}^{T} S_t$ is not feasible solution, our algorithm converges quickly, and we can use the small number of steps to solve such a problem which means even for small T the set $\cup_{t=1}^{T} S_t$ can solve maxmin problem approximately. This result is similar to the bi-criterion solutions for robust submodular maximization studied in (Krause et al., 2008a), where the authors propose an approach that finds a set that violates the cardinality constraint, but it is within logarithmic factor of the constraint.

Theorem 24. *Consider all algorithms stated in Algorithms section, if the functions f is convex monotone submodular, and Assumption 13 holds (and Assumption 14 holds for Gradient Greedy(GG), and Gradient Replacement-greedy(GRG)), then the set $\cup_{t=1}^{T} S_t$ is $(\alpha, \epsilon)-$approximate solution for maxmin convex-submodular problem with cardinality constraint after $\mathcal{O}(1/\epsilon^2)$ iterations. Note that parameter α is $\alpha = (1-1/e)^{-1}$ for Gradient and Extra-gradient Greedy, $\alpha = 2$ for Gradient Replacement-greedy, $\alpha = 2 + \frac{k}{k-1}$ for Extra-gradient Replacement-greedy.*

Proof of Theorem 24 for Gradient Greedy(GG)

If we let $S^* = \arg\max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$ we know that for every t we have $f(\mathbf{x}_{t-1}, S^*) \geq \min_{\mathbf{x}} f(\mathbf{x}, S^*) = \max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$. Therefore, if we let $S = S^*$ in (6.26) we have:

$$(1-\frac{1}{e})\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \frac{\sum_{t=1}^{T}\alpha(f(\tilde{\mathbf{x}},S_t))}{\sum_{t=1}^{T}\alpha} \leq \frac{3M^2\alpha T + \frac{H}{2\alpha}}{T} \tag{6.90}$$

Also, if we let $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}, \cup_t S_t)$ and put $\tilde{\mathbf{x}} = \hat{\mathbf{x}}$ in (6.90) then because $f(\hat{\mathbf{x}}, S_t) \leq f(\hat{\mathbf{x}}, \cup_t S_t)$ we have:

$$(1-\frac{1}{e})\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \min_{\mathbf{x}} f(\mathbf{x},\cup_t S_t) \leq \frac{3M^2\alpha T + \frac{H}{2\alpha}}{T} \tag{6.91}$$

and by using $\alpha = \frac{1}{\sqrt{T}}$:

$$(1-\frac{1}{e})\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \min_{\mathbf{x}} f(\mathbf{x},\cup_t S_t) \leq \frac{3M^2\alpha T + \frac{H}{2\alpha}}{\sqrt{T}} \tag{6.92}$$

Now, using specific choices $K = 2M^2 + \frac{H}{2}$ and let $T = \frac{K^2}{\epsilon^2}$; we obtain that $S_{sol} = \cup_t S_t$ is a $((1-1/e)^{-1}, \epsilon)$-approximate maxmin solution.

Proof of Theorem 24 for Gradient Replacement-greedy(GRG)

If we let $S^* = \arg\max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$ we know that for every t we have $f(\mathbf{x}_{t-1}, S^*) \geq min_{\mathbf{x}} f(\mathbf{x}, S^*) = \max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$. Therefore, if we let $S = S^*$ in (6.45) we have :

$$\frac{1}{2}\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \frac{1}{T}\sum_{t=1}^{T} f(\tilde{\mathbf{x}},S_t) \leq \frac{K}{\sqrt{T}} \tag{6.93}$$

Let $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}, \cup_t S_t)$ and put $\tilde{\mathbf{x}} = \hat{\mathbf{x}}$ in (6.93) then because $f(\hat{\mathbf{x}}, S_t) \leq f(\hat{\mathbf{x}}, \cup_t S_t)$ we have:

$$\frac{1}{2}\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \min_{\mathbf{x}} f(\mathbf{x},\cup_t S_t) \leq \frac{1}{2}\max_{S}\min_{\mathbf{x}} f(\mathbf{x},S) - \frac{1}{T}\sum_{t=1}^{T} f(\hat{\mathbf{x}},S_t) \leq \frac{K}{\sqrt{T}} \tag{6.94}$$

let $T = \frac{K^2}{\epsilon^2}$; then $S_{sol} = \cup_t S_t$ is a $(2, \epsilon)$-approximate maxmin solution.

Proof of Theorem 24 for Extra-gradient Greedy(EGG)

If we let $S^* = \arg\max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$ we know that for every t we have $f(\hat{\mathbf{x}}_{t-1}, S^*) \geq min_{\mathbf{x}} f(\mathbf{x}, S^*) = \max_S \min_{\mathbf{x}} f(\mathbf{x}, S)$. Therefore, if we let $S = S^*$ in (6.66) we have :

$$(1 - 1/e) \max_S \min_{\mathbf{x}} f(\mathbf{x}, S) - \frac{1}{T} \sum_{t=1}^{T} f(\tilde{\mathbf{x}}, S_t) \leq \frac{K}{\sqrt{T}} \tag{6.95}$$

Let $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}, \cup_t S_t)$ and put $\tilde{\mathbf{x}} = \hat{\mathbf{x}}$ in (6.93) then because $f(\hat{\mathbf{x}}, S_t) \leq f(\hat{\mathbf{x}}, \cup_t S_t)$ we have:

$$(1 - 1/e) \max_S \min_{\mathbf{x}} f(\mathbf{x}, S) - \min_{\mathbf{x}} f(\mathbf{x}, \cup_t S_t) \tag{6.96}$$

$$\leq (1 - 1/e) \max_S \min_{\mathbf{x}} f(\mathbf{x}, S) - \frac{1}{T} \sum_{t=1}^{T} f(\hat{\mathbf{x}}, S_t) \leq \frac{K}{\sqrt{T}} \tag{6.97}$$

Let $T = \frac{H^2}{\epsilon^2}$; then, $S_{sol} = \cup_t S_t$ is $((1 - 1/e)^{-1}, \epsilon)$ approximate maxmin solution.

Proof of Theorem 24 for Extra-gradient Replacement-greedy(EGRG)

Similar to (6.95), and (6.96), (6.87) results $S_{sol} = \cup_t S_t$ to be $((2 + \frac{k}{k-1}), \epsilon)$-approximate maxmin solution.

6.6.8. Proof of Theorem 23: Extra Gradient on Continuous Extension Convergence

In this section, we will focus on convergence analysis of Extra Gradient on continuous extension. We first provide two propositions and matroid definition that will help us in the proof.

Definition 7. *Let $\mathcal{I}$ be a nonempty family of allowable subsets of the ground set V, then the tuple $(V, \mathcal{I})$ is a matroid if and only if the following conditions hold:*

1. *For any $A \subset B \subset V$, if $B \in \mathcal{I}$, then $A \in \mathcal{I}$*

2. *For all $A, B \in \mathcal{I}$, if $|A| < |B|$, then there is an $e \in B \backslash A$ such that $A \cup \{e\} \in \mathcal{I}$.*

Proposition 1. *we have that*

$$\text{OPT} \triangleq \min_{\mathbf{x}\in\mathcal{C}} \max_{S\in\mathcal{I}} f(\mathbf{x}, S) = \min_{\mathbf{x}\in\mathcal{C}} \max_{\mathbf{y}\in\mathcal{K}} F(\mathbf{x}, \mathbf{y}). \tag{6.98}$$

Furthermore, the function F has the following properties (assuming differentibility):

Proposition 2. *we have for function F((Hassani et al., 2017)):*

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d : F(\mathbf{x}_1, \mathbf{y}) - F(\mathbf{x}_2, \mathbf{y}) \leq \langle \nabla_{\mathbf{x}} F(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_1 - \mathbf{x}_2 \rangle,$$
$$\forall \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^d : F(\mathbf{x}, \mathbf{y}_2) - 2F(\mathbf{x}, \mathbf{y}_1) \leq \langle \nabla_{\mathbf{y}} F(\mathbf{x}, \mathbf{y}_1), \mathbf{y}_2 - \mathbf{y}_1 \rangle.$$

using same procedure as Extra-gradient Greedy we drive following equations similar to (6.53):

$$\begin{aligned}
&\|\hat{\mathbf{x}}_t - \mathbf{x}\|^2 \\
&\leq \|\mathbf{x}_t - \mathbf{x} - \gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t)\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, y_t)^\top (\mathbf{x}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_t) \\
&\leq \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + 2(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_t) \\
&= \|\mathbf{x}_t - \mathbf{x}\|^2 - 2\gamma_t \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}) + \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 - 2\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2
\end{aligned}$$

and similarly we have:

$$2\langle -\gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y} \rangle \leq \|\mathbf{y} - \mathbf{y}_t\|^2 - \|\mathbf{y} - \hat{\mathbf{y}}_t\|^2 - \|\hat{\mathbf{y}}_t - \mathbf{y}_t\|^2$$

$$2\langle \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x} \rangle \leq \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2$$

$$2\langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{y}_{t+1} - \mathbf{y} \rangle \leq \|\mathbf{y} - \mathbf{y}_t\|^2 - \|\mathbf{y} - \mathbf{y}_{t+1}\|^2 - \|\mathbf{y}_{t+1} - \mathbf{y}_t\|^2$$

$$2\langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{x}_{t+1} - \mathbf{x} \rangle \leq \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$$

$$2\langle -\gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle \leq \|\mathbf{y}_{t+1} - \mathbf{y}_t\|^2 - \|\mathbf{y}_{t+1} - \hat{\mathbf{y}}_t\|^2 - \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2$$

$$2\langle \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle \leq \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 - \|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$$

combing the above equations we have:

$$\begin{aligned}
\langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{y}}_t - \mathbf{y} \rangle &= \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle + \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{y}_{t+1} - \mathbf{y} \rangle \\
&= \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) + \gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle \\
&\quad + \langle -\gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle \\
&\quad + \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{y}_{t+1} - \mathbf{y} \rangle \\
&\leq \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) + \gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle \\
&\quad + 0.5(-\|\hat{\mathbf{y}}_t - \mathbf{y}_{t+1}\|^2 - \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2 \\
&\quad + \|\mathbf{y} - \mathbf{y}_t\|^2 - \|\mathbf{y} - \mathbf{y}_{t+1}\|^2)
\end{aligned}$$

and

$$\begin{aligned}
\langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{x}}_t - \mathbf{x} \rangle &= \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle + \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{x}_{t+1} - \mathbf{x} \rangle \\
&= \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) - \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle \\
&\quad + \langle \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle + \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \mathbf{x}_{t+1} - \mathbf{x} \rangle \\
&\leq \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) - \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle + 0.5(-\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 \\
&\quad + \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|^2)
\end{aligned}$$

let

$$\sigma_t^{\mathbf{x}} = \langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) - \gamma_t \nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{x}}_t - \mathbf{x}_{t+1} \rangle + 0.5(-\|\hat{\mathbf{x}}_t - \mathbf{x}_{t+1}\|^2 - \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2)$$

and

$$\sigma_t^{\mathbf{y}} = \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) + \gamma_t \nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t), \hat{\mathbf{y}}_t - \mathbf{y}_{t+1} \rangle + 0.5(-\|\hat{\mathbf{y}}_t - \mathbf{y}_{t+1}\|^2 - \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2)$$

then $2\sigma_t^{\mathbf{x}} + \sigma_t^{\mathbf{y}} \leq 0$ if $\gamma_t \leq \frac{1}{3\max\{L_{\mathbf{x}}, L_{\mathbf{y}}\}}$(check (Nemirovski, 2004b) for more details) ; which results in:

$$2\langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{y}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{y}}_t - \mathbf{y} \rangle \leq \|\mathbf{y} - \mathbf{y}_t\|^2 - \|\mathbf{y} - \mathbf{y}_{t+1}\|^2 \tag{6.99}$$

$$2\langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{x}}_t - \mathbf{x} \rangle \leq \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|^2 \tag{6.100}$$

combing above equations with proposition 2 we have:

$$2\gamma_t F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) - 2\gamma_t F(\mathbf{x}, \hat{\mathbf{y}}_t) \leq 2\langle \gamma_t \nabla_{\mathbf{x}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{x}}_t - x \rangle \leq \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|^2 \tag{6.101}$$

$$-2\gamma_t F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t) + \gamma_t F(\hat{\mathbf{x}}_t, \mathbf{y}) \leq \langle -\gamma_t \nabla_{\mathbf{y}} F(\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t), \hat{\mathbf{y}}_t - \mathbf{y} \rangle \leq 0.5\|\mathbf{y} - \mathbf{y}_t\|^2 - 0.5\|\mathbf{y} - \mathbf{y}_{t+1}\|^2 \tag{6.102}$$

$$-2\gamma_t F(\mathbf{x}, \hat{\mathbf{y}}_t) + \gamma_t F(\hat{\mathbf{x}}_t, \mathbf{y}) \leq 0.5\|\mathbf{y} - \mathbf{y}_t\|^2 - 0.5\|\mathbf{y} - \mathbf{y}_{t+1}\|^2 + \|\mathbf{x} - \mathbf{x}_t\|^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|^2 \tag{6.103}$$

summing over t in (6.103) and divide both side by $\sum_{t=1}^{T} \gamma_t$ (set of variable $\mathbf{x}$ and $\mathbf{y}$ is bounded i.e.

$\|\mathbf{y}\|^2 \leq H, \|\mathbf{x}\|^2 \leq H$):

$$\frac{\sum_{t=1}^{T} \gamma_t \left[-2F(\mathbf{x}, \hat{\mathbf{y}}_t) + F(\hat{\mathbf{x}}_t, \mathbf{y})\right]}{\sum_{t=1}^{T} \gamma_t} \leq \frac{0.5\|\mathbf{y} - \mathbf{y}_1\|^2 + \|\mathbf{x} - \mathbf{x}_1\|^2}{\sum_{t=1}^{T} \gamma_t} \leq \frac{1.5H}{\gamma T} \tag{6.104}$$

which means same as before let $T = \frac{\sqrt{1.5H}}{\gamma\epsilon}$ and constant step size $\gamma_t = \gamma$, and $\mathbf{x}^* = \arg\min\max f(\mathbf{x}, \mathbf{y})$ we have:

$$\frac{1}{2} f(\frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t, \mathbf{y}) - \min_{\mathbf{x}} \max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y}) \leq \frac{1}{2} F(\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t, \mathbf{y}) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{x}^*, \mathbf{y}^t) \leq \epsilon \tag{6.105}$$

then using proposition 1, $\mathbf{x}_{sol} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t$ is $(0.5, \epsilon)$-approximate minimax solution.

CHAPTER 7

Submodular Maximization with Distributed Constraints

7.1. Introduction

Recently, the need has arisen to design algorithms that distribute decision making among a collection of agents or computing devices. This need has been motivated by problems from statistics, machine learning and robotics. More specifically, these problems include:

- *(Density estimation)* What is the best way to estimate a non-parametric density function from a distributed dataset? (Hu et al., 2007)
- *(Non-parametric models)* How should we summarize very large datasets in a distributed manner to facilitate Gaussian process regression? (Mirzasoleiman et al., 2016a)
- *(Information acquisition)* How should a team of mobile robots acquire information about an environmental process or reduce uncertainty in a mapping task? (Schlotfeldt et al., 2018)

Research toward solving the problems posed in these applications has resulted in a large body of work on topics such as sensing and coverage (Zhong and Cassandras, 2011; Singh et al., 2009), natural language processing (Wei et al., 2013), and learning and statistics (Golovin and Krause, 2011; Djolonga et al., 2016). Indeed, inherent to each of these applications is an underlying optimization problem that can be expressed as

$$\text{maximize} \quad f(\mathcal{S}) \tag{7.1a}$$

$$\text{subject to} \quad \mathcal{S} \subseteq \mathcal{Y},\, \mathcal{S} \in \mathcal{I} \tag{7.1b}$$

where f is a submodular set function (i.e. it has a diminishing-returns property), $\mathcal{Y}$ is a finite set of all decision variables, and $\mathcal{I}$ is a family of allowable subsets of $\mathcal{Y}$. In words, the goal of (7.1) is to pick a set $\mathcal{S}$ from the family of allowable subsets $\mathcal{I}$ that maximizes the submodular set function f. A wide class of relevant objective functions such as mutual information and weighted coverage are submodular; this has motivated a growing body of work surrounding submodular optimization

problems (Mokhtari et al., 2018; Mirzasoleiman et al., 2013; Zhou et al., 2020a; Du et al., 2020; Adibi et al., 2020; Chen et al., 2020; Xie et al., 2019).

Intuitively, it is useful to think of the problem in (7.1) as a distributed n-player game. In this game, each player or agent has a distinct local strategy set of actions. The goal of the game is for each agent to choose at most one action from its own strategy set to maximize a problem-specific notion of reward. Therefore, the problem is *distributed* in the sense that agents can only form a control policy with the actions from their local, distinct strategy sets. To maximize reward, agents are allowed to communicate with their direct neighbors in a bidirectional communication graph. In this way, we might think of these agents as robots that collectively aim to solve a coverage problem in an unknown environment by communicating their sensing actions to their nearest neighbors. Throughout this work, we will refer to this multi-agent game example to elucidate our results.

In this paper, our aim is to study problem (1) in a *distributed* setting, which we will formally introduce in Section 8.2; this setting differs considerably from the *centralized* setting, which has been studied thoroughly in past work (see Calinescu et al. (2011)). Notably, the distributed setting admits a more challenging problem because agents can only communicate locally with respect to a communication graph. Therefore designing an efficient communication scheme among agents is a concomitant requirement for the distributed setting, whereas in the centralized setting, there is no such desideratum.

Contributions. In this paper, we formulate the general case of maximizing a submodular set function subject to a distributed partition matroid constraint in Problem 25. We then formulate the continuous relaxation of this problem via the multilinear extension in Problem 26. Both of these problems are formally defined in Section 8.2. To this end, we study the special case of this optimization problem in which each agent can compute the global objective function and the gradient of the objective function; however we assume that each agent only has access to a local, distinct set of actions. Considering these constraints, we develop Constraint-Distributed Continuous Greedy (`CDCG`), a novel algorithm for solving the continuous relaxation of the *distributed* submodular optimization problem that achieves a tight $(1 - 1/e)$ approximation of the optimal solution, which is known to be

the best possible approximation unless $\mathbf{P} = \mathbf{NP}$. We offer an analysis of the proposed algorithm and prove that it achieves the tight $(1 - 1/e)$ approximation and that its error term vanishes at a linear rate.

Previous work on the distributed version of this problem can approximate the optimal solution to within a multiplicative factor of 1/2 via sequential greedy algorithms (Gharesifard and Smith, 2017; Corah and Michael, 2018; Calinescu et al., 2011). Algorithms for different settings, such as the setting of (Mokhtari et al., 2018) in which each node has access to a local objective function which is averaged to form a global objective function, can also achieve the $(1 - 1/e)$ approximation. Similarly, (Calinescu et al., 2011) shows that it is possible to achieve the optimal $(1 - 1/e)$ approximation in the centralized setting. However, to the best of our knowledge the `CDCG` algorithm presented in this paper is the first algorithm that is guaranteed to achieve the $(1 - 1/e)$ approximation of the optimal solution in this distributed setting.

7.2. Related work

The optimization problem in (7.1) has previously been studied in settings that differ significantly from the setting studied in this paper. In particular, (Calinescu et al., 2011) addresses this problem in a centralized setting and shows that a centralized algorithm can obtain the tight $(1 - 1/e)$ approximation of the optimal solution. In this way, (Calinescu et al., 2011) is perhaps the closest to this paper in that both manuscripts introduce algorithms that obtain the tight $(1 - 1/e)$ guarantee for solving the optimization problem in (7.1) with respect to a particular setting. However, the setting of (Calinescu et al., 2011) is inherently centralized, whereas our setting is *distributed*.

Another similar line of work concerns the so-called "master-worker" model. In this framework, agents solve a distributed optimization problem such as (7.1) by exchanging local information with a centralized master node. However, this setting also differs from the setting studied in this work in that our results assume an entirely distributed setting with no centralized node (Mirzasoleiman et al., 2013; Barbosa et al., 2015).

Fundamentally, the optimization problem posed in (7.1) is NP-hard. However, near-optimal solutions

to (7.1) can be approximated by greedy algorithms (Nemhauser et al., 1978; Nemhauser and Wolsey, 1978). In the distributed context, the sequential greedy algorithm (SGA) has been rigorously studied in (Gharesifard and Smith, 2017). This work poses (7.1) as a communication problem among agents distributed in an directed acyclic graph (DAG) working to optimize a global objective function. The authors of (Gharesifard and Smith, 2017) offer upper and lower bounds on the performance of SGA based on the clique number of the underlying DAG. Building on this, (Corah and Michael, 2018) analyzes the communication redundancy in such an approach and proposes a distributed planning technique that randomly partitions the agents in the DAG. On the other hand, (Grimsman et al., 2018) extends the work of (Gharesifard and Smith, 2017) to a sequential setting in which agents have limited access to the prior decisions of other agents. Extensions of SGA such as the distributed SGA (DSGA) have also been proposed. In particular, (Corah and Michael, 2017, 2019) pose (7.1) as a multi-robot exploration problem and uses DSGA to quantify the suboptimality incurred by redundant sensing information.

Others have proposed novel algorithms with the goal of avoiding the communication overhead incurred by deploying SGA for a large number of agents. Instead of explicitly solving (7.1), many of these algorithms seek to solve a continuous relaxation of this problem (Hassani et al., 2017; Mokhtari et al., 2020a). This continualization of the problem in (7.1) was originally introduced in (Calinescu et al., 2011). In particular, (Mokhtari et al., 2018) proposes several gradient ascent-style algorithms for solving a problem akin to (7.1) in which each agent has access to a local objective function. Similarly, novel algorithms have been developed for solving problems such as unconstrained submodular maximization (Buchbinder et al., 2015) and submodular maximization with matroid constraints (Calinescu et al., 2011; Buchbinder et al., 2014) by first lifting these problems to the continuous domain.

Another notable direction in solving problem (7.1) has been to define an auxiliary or surrogate function in place of the original submodular objective. For instance, (Clark et al., 2015) introduces a distributed algorithm for maximizing a submodular auxiliary function subject to matroid constraints that obtains the $(1 - 1/e)$ optimal approximation. This approach of defining surrogate functions in

place of the submodular objective differs significantly from our approach.

7.3. Preliminaries

In this section, we review the notation used throughout this paper and state definitions that are necessary for the problem formulations in Section 8.2.

Notation. Throughout this paper, lowercase bold-face (e.g. $\mathbf{v}$) will denote a vector, while uppercase bold-face (e.g. $\mathbf{W}$) will denote a matrix. The i^{th} component of a vector $\mathbf{v}$ will be denoted v_i; the element in the i^{th} row of the j^{th} column of a matrix $\mathbf{W}$ will be denoted by w_{ij}. The inner product between two vectors $\mathbf{x}$ and $\mathbf{y}$ will be denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$ and the Euclidean norm of a vector $\mathbf{v}$ will be denoted by $||\mathbf{v}||$. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, we define $\mathbf{x} \vee \mathbf{y} = \max(\mathbf{x}, \mathbf{y})$ as the (vector-valued) component-wise maximum between $\mathbf{x}$ and $\mathbf{y}$; similarly, $\mathbf{x} \wedge \mathbf{y} = \min(\mathbf{x}, \mathbf{y})$ will denote the component-wise minimum between $\mathbf{x}$ and $\mathbf{y}$. We will use the notation $\mathbf{0}_n$ to denote an n-dimensional vector in which each component is zero; similarly $\mathbf{1}_n$ will denote an n-dimensional vector in which each component is one. Calligraphic fonts will denote sets (e.g. $\mathcal{Y}$). Given a set $\mathcal{Y}$, $|\mathcal{Y}|$ will denote the cardinality of $\mathcal{Y}$, while $2^{\mathcal{Y}}$ will denote the power set of $\mathcal{Y}$. $1_{\mathcal{Y}} : \mathcal{Y} \mapsto \{0, 1\}$ will represent the indicator function for the set $\mathcal{Y}$. That is, $1_{\mathcal{Y}}$ is the function that takes value one if its argument is an element of $\mathcal{Y}$ and takes value zero otherwise. Finally, $\varnothing$ will denote the null set.

Background and relevant definitions. Let $\mathcal{Y}$ be a finite set and let $f : 2^{\mathcal{Y}} \mapsto \mathbb{R}_+$ be a set function mapping subsets of $\mathcal{Y}$ to the nonnegative real line. In this setting, $\mathcal{Y}$ is commonly referred to as the *ground set*. The function f is called *submodular* if for every $\mathcal{A}, \mathcal{B} \subseteq \mathcal{Y}$,

$$f(\mathcal{A} \cap \mathcal{B}) + f(\mathcal{A} \cup \mathcal{B}) \leq f(\mathcal{A}) + f(\mathcal{B}).$$

In essence, submodularity amounts to f having a so-called diminishing-returns property, meaning that the incremental value of adding a single element to the argument of f is no less than that of adding the same element to a superset of the argument. To illustrate this, we will slightly overburden our notation by defining

$$f(\mathbf{x}|\mathcal{A}) := f(\mathcal{A} \cup \{\mathbf{x}\}) - f(\mathcal{A})$$

as the *marginal reward* of x given $\mathcal{A}$. This gives rise to an equivalent definition of submodularity. In particular, f is said to be submodular if for every $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{Y}$ and $\forall \mathbf{x} \in \mathcal{Y} \backslash \mathcal{B}$,

$$f(\mathbf{x}|\mathcal{B}) \leq f(\mathbf{x}|\mathcal{A}).$$

Throughout this paper, we will consider submodular functions that are also *monotone*, meaning that for every $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{Y}$, $f(\mathcal{A}) \leq f(\mathcal{B})$, and *normalized*, meaning that $f(\varnothing) = 0$.

In practice, one often encounters a constraint on the allowable subsets of the ground set $\mathcal{Y}$ when maximizing a submodular objective function. Concretely, if $\mathcal{I}$ is a nonempty family of allowable subsets of the ground set $\mathcal{Y}$, then the tuple $(\mathcal{Y}, \mathcal{I})$ is a *matroid* if the following criteria are satisfied:

(1) (*Heredity*) For any $\mathcal{A} \subset \mathcal{B} \subset \mathcal{Y}$, if $\mathcal{B} \in \mathcal{I}$, then $\mathcal{A} \in \mathcal{I}$.

(2) (*Augmentation*) For any $\mathcal{A}, \mathcal{B} \in \mathcal{I}$, if $|\mathcal{A}| < |\mathcal{B}|$, then $\exists\ \mathbf{x} \in \mathcal{B} \backslash \mathcal{A}$ such that $\mathcal{A} \cup \{\mathbf{x}\} \in \mathcal{I}$.

Furthermore, if $\mathcal{Y}$ is partitioned into n disjoint sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_n$, then the tuple $(\mathcal{Y}, \mathcal{I})$ is a *partition matroid* if there exists positive integers $\alpha_1, \ldots, \alpha_n$ such that

$$\mathcal{I} \equiv \{\mathcal{A} \ :\ \mathcal{A} \subseteq \mathcal{Y}, |\mathcal{A} \cap \mathcal{Y}_i| \leq \alpha_i \text{ for each } i = 1, \ldots, n\}.$$

Partition matroids are particularly useful when defining the constraints of a distributed optimization problem because they can be used to describe a setting in which a ground set $\mathcal{Y}$ of all possible actions is written as the product of disjoint local action spaces $\mathcal{Y}_i$.

The notion of submodularity can be extended to the continuous domain (Wolsey, 1982). Consider a set $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$, where $\mathcal{X}_i$ is a compact subset of $\mathbb{R}_+$ for each index $i \in \{1, \ldots, n\}$. We call a continuous function $F : \mathcal{X} \to \mathbb{R}_+$ *submodular* if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$,

$$F(\mathbf{x} \vee \mathbf{y}) + F(\mathbf{x} \wedge \mathbf{y}) \leq F(\mathbf{x}) + F(\mathbf{y}).$$

As in the discrete case, we say that a continuous function F is monotone if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\mathbf{x} \preceq \mathbf{y}$

implies that $F(\mathbf{x}) \leq F(\mathbf{y})$. Furthermore, if F is differentiable, we say that F is DR-submodular, where DR stands for "diminishing-returns," if the gradients are *antitone*. That is, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$, F is DR-submodular if $\mathbf{x} \preceq \mathbf{y}$ implies that $\nabla F(\mathbf{x}) \succeq F(\mathbf{y})$.

7.4. Problem Statement

In this section, we formulate the main problem of this paper: maximizing submodular set functions subject to distributed partition matroid constraints.

Problem 25 (**Submodular Maximization Subject to a Distributed Partition Matroid Constraint**). *Consider a collection of n agents that form the set $\mathcal{N} = \{1, \ldots, n\}$. Let $f : 2^{\mathcal{Y}} \mapsto \mathbb{R}_+$ be a normalized and monotone submodular set function and let $\mathcal{Y}_1, \ldots, \mathcal{Y}_n$ be a pairwise disjoint partition of a finite ground set $\mathcal{Y}$, wherein each agent $i \in \mathcal{N}$ can only choose actions from its local strategy set $\mathcal{Y}_i$. Furthermore, consider the partition matroid $(\mathcal{Y}, \mathcal{I})$, where*

$$\mathcal{I} := \{\mathcal{S} \subseteq \mathcal{Y} \ : \ |\mathcal{Y}_i \cap \mathcal{S}| \leq 1 \text{ for } i = 1, \ldots, n\}. \tag{7.2}$$

The problem of submodular maximization subject to a distributed partition matroid constraint is to maximize f by selecting a set $\mathcal{S} \subseteq \mathcal{Y}$ from the family of allowable subsets so that $\mathcal{S} \in \mathcal{I}$. Formally:

$$\text{maximize} \quad f(\mathcal{S}) \tag{7.3a}$$

$$\text{subject to} \quad \mathcal{S} \in \mathcal{I} \tag{7.3b}$$

In effect, the distributed partition matroid constraint in Problem 25 enforces that each agent $i \in \mathcal{N}$ can choose at most one action from its local strategy set $\mathcal{Y}_i$. Note that in this setting, each agent can only choose actions from its own local strategy set. Therefore, this problem is distributed in the sense that agents can only determine the actions taken by other agents by directly communicating with one another.

7.4.1. Sequential greedy algorithm

It is well known that the sequential greedy algorithm (SGA), in which each agent $i \in \mathcal{N}$ chooses an action sequentially based on

$$\mathbf{y}_i = \arg\max_{\mathbf{y} \in \mathcal{Y}_i} f(\mathbf{y}|\mathcal{S}_{i-1}) \tag{7.4}$$

where $\mathcal{S}_{i-1} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}\}$, approximates the optimal solution to within a multiplicative factor of $1/2$ (Gharesifard and Smith, 2017). The drawbacks of this algorithm are twofold. Firstly, as we will show, our algorithm achieves the tight $(1 - 1/e)$ approximation of the optimal solution, which is known to be the best possible approximation unless $\mathbf{P} = \mathbf{NP}$. Secondly, as its name suggests, SGA is sequential in nature and therefore it scales very poorly in the number of agents. That is, each agent must wait for each of the previous agents to compute their contribution to the optimal set $\mathcal{S}^*$. Notably, our algorithm does not suffer from this sequential dependence.

7.4.2. Continuous Extension of Problem 25

Sequential algorithms such as SGA can only achieve a $1/2$ approximation of the optimal solution. To achieve the best possible $(1 - 1/e)$ approximation of the optimal solution, it is necessary to extend Problem 25 to the continuous domain via the so-called *multilinear extension* of the submodular objective function f (Nemhauser et al., 1978). Thus, the method we use in this work to achieve the tight $(1 - 1/e)$ approximation relies on the continualization of Problem 25. Importantly, it has been shown that Problem 25 and the optimization problem engendered by lifting Problem 1 to the continuous domain via this multilinear extension yield the same solution (Calinescu et al., 2011). Furthermore, by applying proper rounding techniques, such as those described in Section 5.1 of (Mokhtari et al., 2018) and in (Calinescu et al., 2011) and (Chekuri et al., 2014) to the continuous relaxation of Problem 25, one can obtain the tight $(1-1/e)$ approximation for Problem 25. Therefore, our approach in this paper will be to lift Problem 25 to the continuous domain. We formulate this problem in the following way:

Problem 26 (**Continuous Extension of Problem 25**). *Consider the conditions of Problem 25.*

Define the DR-submodular continuous multilinear extension $F : \mathcal{X} \mapsto \mathbb{R}_+$ of the objective function f in Problem 25 by

$$F(\mathbf{y}) := \sum_{\mathcal{S} \subseteq \mathcal{Y}} f(\mathcal{S}) \prod_{i \in \mathcal{S}} y_i \prod_{j \notin \mathcal{S}} (1 - y_j) \tag{7.5}$$

and let $\mathcal{P} \subseteq \mathcal{X}$ be the matroid polytope $\mathcal{P} := \mathbf{conv}\{1_{\mathcal{S}} \ : \ \mathcal{S} \in \mathcal{I}\}$ where $\mathcal{I}$ is the family of sets defined in (7.2). The continuous relaxation of Problem 25 is formally defined by

$$\text{maximize} \quad F(\mathbf{y}) \tag{7.6a}$$

$$\text{subject to} \quad \mathbf{y} \in \mathcal{P} \tag{7.6b}$$

Observe that Problem 26 is *distributed* in the sense that each agent $i \in \mathcal{N}$ is associated with its own distinct continuous strategy space $\mathcal{P}_i$. Formally, the set $\mathcal{P}_i$ is defined as

$$\mathcal{P}_i := \mathbf{conv}\{1_{\mathcal{S}} \ : \ S \subseteq \mathcal{I}_i\} \tag{7.7}$$

where $\mathcal{I}_i := \{\mathcal{S} \subseteq \mathcal{Y} \ : \ |\mathcal{Y}_i \cap \mathcal{S}| \leq 1\}$. In this way, $\mathcal{P} = \cap_{i=1}^{n} \mathcal{P}_i$. In this way, the sets $\mathcal{P}_i$ play similar roles in Problem 26 as the sets $\mathcal{Y}_i$ do in Problem 25.

Note that Problem 26 is nonconvex, and therefore cannot be solved by classical convex solvers or algorithms. Further, we assume that each agent $i \in \mathcal{N}$ can compute the multilinear extension F of the submodular objective function f in (7.3a) and the gradient of F.

7.5. Constraint-Distributed Continuous Greedy

In this section, we present Constraint-Distributed Continuous Greedy (`CDCG`), a decentralized algorithm for solving Problem 26. The pseudo-code of `CDCG` is described in Algorithm 13. At a high level, this algorithm involves updating each agent's local decision variable based on the aggregated belief of a small group of other agents about the best control policy. In essence, inter-agent communication within small groups of agents facilitates local decision making.

For clarity, we introduce a simple framework for the inter-agent communication structure. In CDCG, agents $i \in \mathcal{N} = \{1, \ldots, n\}$ share their decision variables $\mathbf{y}_i$ with a small subset of *local* agents in $\mathcal{N}$. To encode the notion of locality, suppose that each agent $i \in \mathcal{N}$ is a node in a bidirectional *communication graph* $\mathcal{G} = (\mathcal{N}, \mathbb{E})$ in which $\mathbb{E}$ denotes the set of edges. Given this structure, we assume that each agent $i \in \mathcal{N}$ can only communicate its decision variable $\mathbf{y}_i$ with its direct neighbors in $\mathcal{G}$. Let us denote the neighbor set of agent $i \in \mathcal{N}$ by $\mathcal{N}_i$. Then the set of edges $\mathbb{E}$ can be written $\{(i, j) \ : \ j \in \mathcal{N}_i\}$. We adopt this notation for the remainder of this paper.

7.5.1. Intuition for the CDCG algorithm

The goal of CDCG at a given node $i \in \mathcal{N}$ is to learn the local decision variable $\mathbf{y}_i$. CDCG is run at each node in $i \in \mathcal{N}$ to assemble the collection $\{\mathbf{y}_1^T, \ldots, \mathbf{y}_n^T\}$ where T is a given positive integer; this collection represents an approximate solution to Problem 26 and guarantees that each agent contributes at most one element to the solution. Then, by applying proper rounding techniques to each element of the collection such as those discussed in (Mokhtari et al., 2018; Calinescu et al., 2011; Chekuri et al., 2014), we obtain a solution to Problem 25. In the proceeding sections, we show that this solution achieves the tight $(1 - 1/e)$ approximation of the optimal solution.

In the analysis of CDCG, we add the superscript t to the vectors $\mathbf{v}_i^t$ and $\mathbf{y}_i^t$ defined in Algorithm 13. This superscript denotes the iteration number so that $\mathbf{y}_i^t$ and $\mathbf{v}_i^t$ represent the values of the local variables $\mathbf{y}_i$ and $\mathbf{v}_i$ at iteration $t \in \{1, \ldots, T\}$ respectively.

7.5.2. Description of the steps for CDCG (Algorithm 13)

From the perspective of node $i \in \mathcal{N}$, CDCG takes two arguments: nonnegative weights w_{ij} for each $j \in \mathcal{N}_i \cup \{i\}$ and a positive integer T. The weights w_{ij} correspond to the i^{th} row in a doubly-stochastic weight matrix $\mathbf{W}$ and T is the number of iterations for which the algorithm will run. The weight matrix $\mathbf{W}$ is a design parameter of the problem and must fulfill a number of technical requirements that are fully described in Appendix A. Before any computation, the local decision variable $\mathbf{y}_i$ is initialized to the zero vector.

Computation proceeds in T rounds. In each round, the first step is to calculate the gradient of the

multilinear extension function F evaluated at the local decision variable $\mathbf{y}_i^{t-1}$ from the previous iteration. Thus, in line 3 of Algorithm 13, we calculate the ascent direction $\mathbf{v}_i^t$ at iteration t in the following way:

$$\mathbf{v}_i^t = \underset{\mathbf{x} \in \mathcal{P}_i \cap \mathcal{C}_i}{\arg\max} \left\langle \nabla F(\mathbf{y}_i^{t-1}), \mathbf{x} \right\rangle .$$

Intuitively, one can think of $\mathbf{v}_i^t$ as the vector from the set $\mathcal{P}_i \cap \mathcal{C}_i$ that is most aligned with $\nabla F(\mathbf{y}_i^{t-1})$. To define the set $\mathcal{C}_i$, first define the set $\mathcal{J}_i$ as the set of indices of the elements in $\mathcal{Y}$ that correspond to elements in $\mathcal{Y}_i$. Then

$$\mathcal{C}_i := \left\{ \mathbf{x} \in \mathbb{R}_+^{|\mathcal{Y}|} \ : \ x_j = 0 \quad \forall j \notin \mathcal{J}_i \right\}. \tag{7.8}$$

Using this notation, we can equivalently define $\mathcal{P}_i = \{\mathbf{x} \in \mathbb{R}_+^{|\mathcal{Y}|} \ : \ \sum_{j \in \mathcal{J}_i} x_j \leq 1\}$. Next, in line 4 of Algorithm 13, $\mathbf{y}_i$ is updated by setting

$$\mathbf{y}_i^t = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{y}_j^{t-1} + \frac{n}{T} \mathbf{v}_i^t.$$

In this way, the governing principle is to collaboratively accumulate the local belief about the optimal decision $\mathbf{y}_i^{t-1}$ and to then move in the approximate direction of steepest ascent from this point.

After T rounds of computation at each node $i \in \mathcal{N}$, we obtain a local decision variable $\mathbf{y}_i^T$ at each node. By applying proper rounding techniques, we obtain a decision variable for each agent $i \in \mathcal{N}$. Rounding in a decentralized manner is discussed in Section 5.1 of (Mokhtari et al., 2018). The rounding techniques of (Mokhtari et al., 2018) build on "pipage rounding" (Calinescu et al., 2011) and "swap rounding" (Chekuri et al., 2014), which are both centralized rounding techniques. The collection of these decision variables form the set $\mathcal{S}^*$, which represents our solution to Problem 25.

7.6. Convergence Analysis

The main result in this paper is to show that in the distributed setting of Problem 26, `CDCG` achieves a tight $(1 - 1/e)$ multiplicative approximation of the optimal solution. The following theorem

Algorithm 13 Constraint-Distributed Continuous Greedy (`CDCG`) at node i

Require: Weights w_{ij} for each neighbor $j \in \mathcal{N}_i \cup \{i\}$ and number of rounds $T \in \mathbb{Z}_{++}$
Returns: Local solution $\mathbf{y}_i^\star$ for node $i \in \mathcal{N}$ to Problem 25

1: Initialize local vectors $\mathbf{y}_i^0 = \mathbf{0}_{|\mathcal{Y}|}$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: • Calculate an ascent direction for the multilinear extension function F via:

$$\mathbf{v}_i^t \leftarrow \arg\max_{\mathbf{x} \in \mathcal{P}_i \cap \mathcal{C}_i} \left\langle \nabla F\left(\mathbf{y}_i^{t-1}\right), \mathbf{x} \right\rangle$$

4: • Update the local variable $\mathbf{y}_i^t$ using the ascent direction $\mathbf{v}_i^t$ via:

$$\mathbf{y}_i^t \leftarrow \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{y}_j^{t-1} + \frac{n}{T} \mathbf{v}_i^t$$

5: **end for**
6: $\mathbf{y}_i^\star \leftarrow \texttt{Round}\left(\mathbf{y}_i^T\right)$

summarizes this result.

Theorem 27. *Consider the `CDCG` algorithm described in Algorithm 13. Let $\mathbf{y}^*$ denote the global maximizer of the optimization problem defined in Problem 26, and assume that a positive integer T and a doubly-stochastic weight matrix $\mathbf{W}$ are given. Then provided that the assumptions outlined in Appendix A hold, for all nodes $i \in \mathcal{N}$, the local variables $\mathbf{y}_i^T$ obtained after T iterations satisfy*

$$F(\mathbf{y}_i^T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{y}^*) - \left[\frac{LD^2}{2T} + \frac{LD^2(n^2 + n^{5/2}) + n^{5/2}DG}{T(1-\beta)}\right] \tag{7.9}$$

where D, G, L, and β are problem-dependent constants that are formally defined in Appendices A and B.

Succinctly, Theorem 27 means that the sequence of local iterates generated by `CDCG` achieves the optimal approximation ratio $(1 - 1/e)$ and that the error term vanishes at a linear rate of $\mathcal{O}(1/T)$. That is,

$$F(\mathbf{y}_i^T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{y}^*) - \mathcal{O}\left(\frac{1}{T}\right),$$

which implies that each agent reaches an objective value larger than $(1 - 1/e - \epsilon)\mathbf{y}^*$ after $\mathcal{O}(1/\epsilon)$

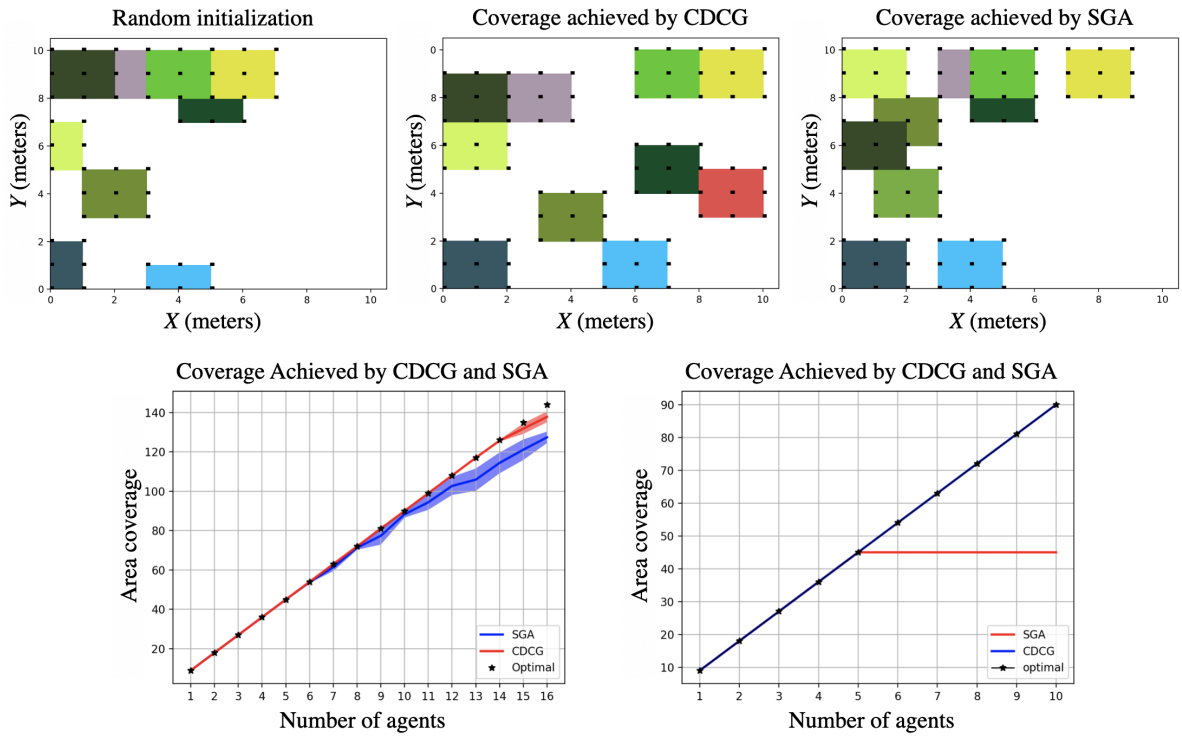

Figure 7.1: **Area coverage simulation results for `CDCG` and `SGA`.** (Top left) Random initialization of $n = 10$ agents in a 10×10 grid. (Top middle & right) Coverage achieved by `CDCG` (top middle) and `SGA` (top right) from the random initialization shown in the top left panel. (Bottom left) Comparison of the mean coverage achieved by `CDCG` and `SGA` averaged over 10 random initializations. (Bottom right) Comparison of the coverage achieved by `CDCG` and `SGA` for a setting in which each agent's starting point is the center of the grid.

rounds of communication. Previous work can only guarantee an objective value of $(1/2)\mathbf{y}^*$ (Gharesifard and Smith, 2017). We provide the proof of this theorem and supporting lemmas in Appendices B and C.

7.7. Simulation Results

To evaluate the proposed algorithm, we consider a multi-agent area coverage problem. In this setting, each agent $i \in \mathcal{N}$ is constrained to move in a two-dimensional grid. We assume that each agent has a finite radius r so that it can observe those grid points that lie with a square with sidelength $2r + 1$. The objective is for the agents to collectively maximize the cardinality of the union of their observation sets of grid points. In other words, given an initial configuration, the problem is to choose an action for each agent that maximizes the overall coverage of the grid. The top three panels

of Figure 7.1 show various configurations of agents in this two-dimensional grid.

Consider an initial configuration of n agents in states $\mathbf{y}_i \in \mathbb{Z}^2$ for $i \in \{1, \ldots, n\}$ with the dynamic constraint $\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \mathbf{u}_i^t$, where $\mathbf{u}_i^t$ is a control input from a discrete set

$$\mathcal{U} = \{(0,1), (0,-1), (-1,0), (1,0), (0,0)\}.$$

Elements from this set represent the admissible actions for each agent in the two-dimensional grid.

In our simulation, we compared the performance of `SGA` against `CDCG` on the coverage task posed above for a variable number of agents. For simplicity, we assumed that the underlying communication graph $\mathcal{G}$ used in `CDCG` was fully connected and that each value in the weight matrix $\mathbf{W}$ was $1/n$. A random initialization for each agent's position and the coverage achieved by `CDCG` and `SGA` are shown in the top three panels of Figure 7.1 respectively. We compared the performance of these algorithms across ten random initializations of starting locations for the agents; the mean performance of each algorithm and the respective standard deviations are shown in the bottom left panel of Figure 7.1. In each trial, we ran both algorithms 50 times, each of which produced a control input $\mathbf{u}_i$ for each agent. For each initialization, we ran `CDCG` for $T = 100$ iterations. Note that as the number of agents increases, `CDCG` is optimal or near optimal in each case; however for larger than eight agents, the performance of `SGA` begins to fall away from the optimal.

We also compared the coverages achieved by `CDCG` and `SGA` for a setting in which each agent's starting position is the center of the grid. The results of this experiment are shown in the bottom right panel of Figure 7.1. In this plot, we averaged the performance over 15 independent trials; in each trial, we ran `CDCG` for $T = 100$ iterations. Interestingly, `SGA` converges to a local maximum in this problem, whereas `CDCG` achieves the optimal value.

7.8. Conclusion

In this work, we described an approach for achieving the optimal approximation to a class of submodular optimization problems subject to a *distributed* partition matroid constraint. The algorithm we proposed outperforms the sequential greedy algorithm in two senses: (1) `CDCG` achieves

the tight $(1 - 1/e)$ approximation for the optimal solution whereas SGA can only achieve a $1/2$ approximation; and (2) `CDCG` imposes a limited communication structure on this problem, which allows for significant gains via parallelization. We showed empirically via an area coverage simulation with multiple agents that `CDCG` outperforms the greedy algorithm.

7.9. Appendix A: Assumptions for Theorem 27

Consider the continuous relaxation of Problem 25 that was described in Section 7.4.2. We assume that the Euclidean distance between elements of the convex set $\mathcal{P}$ are uniformly bounded, i.e. that

$$||\mathbf{x} - \mathbf{y}|| \leq D \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{P}. \tag{7.10}$$

This is a trivial consequence of the multilinear extension F, since $\mathcal{P}$ is contained in the unit cube. Furthermore, we assume that the gradient of the multilinear extension F of the objective function f in Problem 25 is L-Lipschitz continuous, i.e. that

$$||\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})|| \leq L\,||\mathbf{x} - \mathbf{y}|| \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{P} \tag{7.11}$$

so that $||\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})|| \leq LD \ \forall \mathbf{x}, \mathbf{y} \in \mathcal{P}$ by (7.10). Again, this is not a limiting assumption, because the domain of F is compact, which implies the Lipschitzness of F. Also, we assume that the norm of the gradient of F is bounded over $\mathcal{P}$, i.e. that

$$||\nabla F(\mathbf{x})|| \leq G \qquad \forall \mathbf{x} \in \mathcal{P}, \tag{7.12}$$

which again follows from the compactness of the domain of F. It is then easy to show that (7.12) and the multivariable mean value theorem imply that F is G-Lipschitz continuous over $\mathcal{P}$. Note that in this case, since F is the multilinear extension of f, assumptions (7.10), (7.11), and (7.12) all hold. Moreover, the constants L, D, and G all depend on the maximum singleton value of f. For further justification, see (Hassani et al., 2017; Mokhtari et al., 2018). Finally, it will be prudent to mention that for the multilinear extension F of any monotone and submodular function f, it holds

that $F(\mathbf{0}) \geq 0$ and

$$\left\langle \nabla F(\bar{\mathbf{y}}^t), \mathbf{y}^* \right\rangle \geq F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^t) \tag{7.13}$$

For justification, see (Calinescu et al., 2011).

Now consider the communication framework described in Section 7.5 and the weight matrix $\mathbf{W}$. This matrix is a parameter that is designed to match the criteria and setting of a given application. We assume that the weights used in `CDCG` are nonnegative so that $w_{ij} \geq 0 \; \forall i, j \in \mathcal{N}$; furthermore, if node $j \notin \mathcal{N}_i$, then $w_{ij} = 0$. Also, we assume that the weight matrix $\mathbf{W}$ is doubly stochastic and symmetric, and that $(\mathbf{I} - \mathbf{W}) = \text{span}(\mathbf{1}_n)$. The assumptions made about $\mathbf{W}$ are similar to those described in (Mokhtari et al., 2018).

Lastly, consider that past work has studied the case in which the objective function is distributed (Mokhtari et al., 2018). However, our setting is one in which the problem is distributed in the constraints rather than the objective. Therefore, we assume that each agent has access to an oracle for computing the objective submodular function f.

7.10. Appendix B: Preliminary Lemmas

In this appendix, we offer proofs of lemmas that support the proof of Theorem 27. We note that the proofs for Lemmas 7.10 and 7.10 are similar to those that originally appeared in (Mokhtari et al., 2018), and where relevant, pieces of these arguments have been reproduced for completeness.

In general, the goal of Lemma 7.10 is to show that the local decision variable $\mathbf{y}_i$ for each agent $i \in \mathcal{N}$ converges to the mean $\bar{\mathbf{y}} = \frac{1}{n} \sum_{i \in \mathcal{N}} \mathbf{y}_i$. Then, in Lemma 7.10, we show that these means are Cauchy, meaning that for a sufficiently large number of iterations T, the distance between $\bar{\mathbf{y}}^t$ and $\bar{\mathbf{y}}^{t+1}$ becomes arbitrarily small. Together, Lemma 7.10 and Lemma 7.10 establish that for a sufficiently large number of iterations, the set of nodes come to a consensus for the optimal decision. Lemmas 7.10 and Lemma 7.10 are technical results used in the proof of Theorem 27.

Lemma 53. *For any iteration $t \leq T$ where $T \in \mathbb{Z}_{++}$, it follows that the Euclidean distance between*

the local variable $\mathbf{y}_i^t$ *at node* $i \in \mathcal{N}$ *and the mean of the local variables* $\bar{\mathbf{y}}^t$ *can be bounded by*

$$\left|\left|\mathbf{y}_i^t - \bar{\mathbf{y}}^t\right|\right| \leq \frac{n^{3/2} D}{T(1-\beta)}$$

where β *is the magnitude of the eigenvalue of* $\mathbf{W}$ *that among all eigenvalues in* $\sigma(\mathbf{W})$ *has the second largest magnitude.*

Proof. Define $\mathbf{y}_{\text{con}} := \begin{bmatrix}\mathbf{y}_1; \ldots; \mathbf{y}_n\end{bmatrix} \in \mathbb{R}^{np}$ and $\mathbf{v}_{\text{con}} := \begin{bmatrix}\mathbf{v}_1; \ldots; \mathbf{v}_n\end{bmatrix} \in \mathbb{R}^{np}$ as the concatenations of the local variables $\mathbf{y}_i^t$ and descent directions $\mathbf{v}_i$ in `CDCG`. The update rule in step 2 in Algorithm 13 leads to the expression

$$\mathbf{y}_{\text{con}}^t = \frac{n}{T} \sum_{s=0}^{t-1} (\mathbf{W} \otimes \mathbf{I})^{t-1-s} \, \mathbf{v}_{\text{con}}^s \tag{7.14}$$

Next, if we premultiply both sides of (7.14) by the matrix $(\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \otimes \mathbf{I})$, which is the Kronecker product of the matrices $\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \in \mathbb{R}^{n \times n}$ and $\mathbf{I} \in \mathbb{R}^{p \times p}$, we obtain

$$\left(\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \otimes \mathbf{I}\right) \mathbf{y}_{\text{con}}^t = \frac{n}{T} \sum_{s=0}^{t-1} \left[\left(\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \mathbf{W}^{t-1-s}\right) \otimes \mathbf{I}\right] \mathbf{v}_{\text{con}}^s. \tag{7.15}$$

The left hand side of (7.15) can be simplified to

$$\left(\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \otimes \mathbf{I}\right) \mathbf{y}_{\text{con}}^t = \bar{\mathbf{y}}_{\text{con}}^t \tag{7.16}$$

where $\mathbf{y}_{\text{con}}^t = \begin{bmatrix}\bar{\mathbf{y}}^t; \ldots; \bar{\mathbf{y}}^t\end{bmatrix}$. Combining (7.16) and the equality $\mathbf{1}_n \mathbf{1}_n^\dagger \mathbf{W} = \mathbf{1}_n \mathbf{1}_n^\dagger$, we can write (7.15) as

$$\bar{\mathbf{y}}_{\text{con}}^t = \frac{n}{T} \sum_{s=0}^{t-1} \left(\frac{\mathbf{1}_n \mathbf{1}_n^\dagger}{n} \otimes \mathbf{I}\right) \mathbf{v}_{\text{con}}^s. \tag{7.17}$$

Using the expressions in (7.14) and (7.17), we can derive an upper bound on the difference

$\left|\left|\mathbf{y}_{\text{con}}^t - \bar{\mathbf{y}}_{\text{con}}^t\right|\right|$ by

$$\begin{aligned}\left|\left|\mathbf{y}_{\text{con}}^t - \bar{\mathbf{y}}_{\text{con}}^t\right|\right| &= \frac{n}{T}\left|\left|\sum_{s=0}^{t-1}\left[\left(\mathbf{W}^{t-1-s} - \frac{\mathbf{1}_n\mathbf{1}_n^\dagger}{n}\right)\otimes\mathbf{I}\right]\mathbf{v}_{\text{con}}^s\right|\right| \\ &\leq \frac{n}{T}\sum_{s=0}^{t-1}\left|\left|\mathbf{W}^{t-1-s} - \frac{\mathbf{1}_n\mathbf{1}_n^\dagger}{n}\right|\right|\cdot||\mathbf{v}_{\text{con}}^s|| \\ &\leq \frac{nD}{T}\left|\left|\mathbf{W}^{t-1-s} - \frac{\mathbf{1}_n\mathbf{1}_n^\dagger}{n}\right|\right|, \end{aligned} \tag{7.18}$$

where the first inequality follows from the Cauchy-Schwartz inequality and the fact that the norm of a matrix does not change if we Kronecker it by the identity matrix. The second inequality holds because $\left|\left|\mathbf{v}_{\text{con}}^t\right|\right| \leq D$. Note that the eigenvectors of the matrices $\mathbf{W}$ and $\mathbf{W}^{t-1-s}$ are the same for all $s = 0, \ldots, t-1$. Therefore, the largest eigenvalue of $\mathbf{W}^{t-1-s}$ is 1 with eigenvector $\mathbf{1}_n$ and the second largest magnitude of the eigenvalues is β^{t-1-s}, where β is the second largest magnitude of the eigenvalues of $\mathbf{W}$. Also note that because $\mathbf{1}_n$ is an eigenvector of $\mathbf{W}^{t-1-s}$, it follows that all of the other eigenvectors of $\mathbf{W}^{t-1-s}$ are orthogonal to $\mathbf{1}_n$ since $\mathbf{W}$ is symmetric. Hence we can bound the norm $\left|\left|\mathbf{W}^{t-1-s} - (\mathbf{1}_n\mathbf{1}_n^\dagger)/n\right|\right|$ by β^{t-1-s}. Applying this substitution to the right hand side of (7.18) yields

$$\left|\left|\mathbf{y}_{\text{con}}^t - \bar{\mathbf{y}}_{\text{con}}^t\right|\right| \leq \frac{nD}{T}\sum_{s=0}^{t-1}\beta^{t-1-s} \leq \frac{nD}{T(1-\beta)}. \tag{7.19}$$

Since $\left|\left|\mathbf{y}_{\text{con}}^t - \bar{\mathbf{y}}_{\text{con}}^t\right|\right|^2 = \sum_{i=1}^n \left|\left|\mathbf{y}_i^t - \bar{\mathbf{y}}^t\right|\right|^2$, we find that

$$\left|\left|\mathbf{y}_j^t - \bar{\mathbf{y}}^t\right|\right| \leq \sum_{i=1}^n \left|\left|\mathbf{y}_i^t - \bar{\mathbf{y}}^t\right|\right| \leq \sqrt{n}\left(\sum_{i=1}^n \left|\left|\mathbf{y}_i^t - \bar{\mathbf{y}}^t\right|\right|^2\right)^{1/2} \leq \frac{n^{3/2}D}{T(1-\beta)} \tag{7.20}$$

where inequality (7.20) follows from (7.19). □

Lemma 54. *For any iteration $t \leq T$ for $T \in \mathbb{Z}_{++}$, the Euclidean distance between the means $\bar{\mathbf{y}}^t$ and $\bar{\mathbf{y}}^{t-1}$ of the local variables $\mathbf{y}_i^t$ and $\mathbf{y}_i^{t-1}$ respectively for $i \in \mathcal{N}$ at consecutive iterations t and $t-1$*

can be bounded by

$$\left|\left|\bar{\mathbf{y}}^t - \bar{\mathbf{y}}^{t-1}\right|\right|_2 \leq \frac{D}{T}. \tag{7.21}$$

Proof. Averaging both sides of the update rule for $\mathbf{y}_i^t$ of Algorithm 13 across the set of agents $i \in \mathcal{N}$ yields the following expression for $\bar{\mathbf{y}}^t$:

$$\bar{\mathbf{y}}^t = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}\mathbf{y}_j^{t-1} + \frac{1}{T}\sum_{i=1}^{n}\mathbf{v}_i^t. \tag{7.22}$$

Since $w_{ij} = 0$ if $j \notin \mathcal{N}_i \cup \{i\}$, we can rewrite the RHS of (7.22) in the following way:

$$\begin{aligned}
\bar{\mathbf{y}}^t &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\mathbf{y}_j^{t-1} + \frac{1}{T}\sum_{i=1}^{n}\mathbf{v}_i^t \\
&= \frac{1}{n}\sum_{j=1}^{n}\mathbf{y}_j^{t-1}\sum_{i=1}^{n} w_{ij} + \frac{1}{T}\sum_{i=1}^{n}\mathbf{v}_i^t \\
&= \frac{1}{n}\sum_{j=1}^{n}\mathbf{y}_j^{t-1} + \frac{1}{T}\sum_{i=1}^{n}\mathbf{v}_i^t
\end{aligned} \tag{7.23}$$

where (7.23) follows since $\mathbf{W}^T\mathbf{1} = \mathbf{1}$. Rearranging (7.23), it follows that

$$\left|\left|\bar{\mathbf{y}}^t - \bar{\mathbf{y}}^{t-1}\right|\right| = \frac{1}{T}\left|\left|\sum_{i=1}^{n}\mathbf{v}_i^t\right|\right| \leq \frac{D}{T}$$

Note that because the Euclidean distance between points of the polytope P are assumed to be bounded, $\left|\left|\sum_{i=1}^{n}\mathbf{v}_i^t\right|\right| \leq D$. The expression in (7.21) follows. □

Corollary 28. *Let $T \in \mathbb{Z}_{++}$. Then the vector $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i$ is in the constraint set $\mathcal{P}$ $\forall t \leq T$.*

Proof. In Lemma 1 we proved that $\mathbf{y}_i^t$ converges to $\bar{\mathbf{y}}^t$. We show that $\bar{\mathbf{y}}^t \in \mathcal{P}$ by induction. Because we assign $\mathbf{y}_i^0 = \mathbf{0}$, it is clear that $\bar{\mathbf{y}}^0 \in \mathcal{P}$. Now as inductive hypothesis, we assume that $\bar{\mathbf{y}}^{t-1}$ is in $\mathcal{P}$. Observe that we can write $\bar{\mathbf{y}}^t = \bar{\mathbf{y}}^{t-1} + (1/T)\sum_{i=1}^{n}\mathbf{v}_i^t$. Thus by the inductive hypothesis and the

fact that $\sum_{i=1}^{n} \mathbf{v}_i^t \in \mathcal{P}\ \forall t \leq T$, it follows that $\bar{\mathbf{y}}^t$ is a convex combination of elements of $\mathcal{P}$. That is, we can write $\bar{\mathbf{y}}^t = (1/T)\sum_{k=1}^{t}\sum_{i=1}^{n} \mathbf{v}_i^k + (1-t/T)\mathbf{0}$. Therefore $\bar{\mathbf{y}}^t \in \mathcal{P}$, and so $\mathbf{y}_i^t$ converges to a point in $\mathcal{P}$. □

Lemma 55. *Let F be the multilinear extension of a monotone submodular function $f : 2^{\mathcal{Y}} \mapsto \mathbb{R}$ where $\mathcal{Y}$ is a discrete ground set. Then*

$$\max_{\mathbf{v}\in P_i \cap C_i} \langle \nabla F(\mathbf{y}_i), \mathbf{v}\rangle = \max_{\mathbf{x}\in P_i} \langle [\nabla F(\mathbf{y}_i)]_{c_i}, \mathbf{x}\rangle \tag{7.24}$$

where $[\nabla F(\bar{\mathbf{y}}_i)]_{c_i}$ denotes the projection of $\nabla F(\mathbf{y}_i)$ onto the set C_i.

Proof. Consider the definitions of $\mathcal{P}_i$ and $\mathcal{C}_i$ in (7.7) and (7.8) respectively. Maximizing $\langle \nabla F(\mathbf{y}_i), \mathbf{v}\rangle$ over $\mathbf{v} \in \mathcal{P}_i \cap \mathcal{C}_i$ results in the same value as maximizing the inner product of the projection of $\nabla F(\mathbf{y}_i^{t-1})$ onto the set $\mathcal{C}_i$ over $\mathbf{x} \in \mathcal{P}_i$. □

Lemma 56. *Let F be the multilinear extension of a monotone submodular function $f : 2^{\mathcal{Y}} \mapsto \mathbb{R}$ where $\mathcal{Y}$ is a discrete ground set. Then*

$$\left|\left|\nabla F(\bar{\mathbf{y}}^t) - \sum_{i=1}^{n} \left[\nabla F(\mathbf{y}_i^t)\right]_{C_i}\right|\right| \leq \frac{n^{3/2}DL}{T(1-\beta)} \tag{7.25}$$

Proof. Observe that

$$\begin{aligned}
\left|\left|\nabla F(\bar{\mathbf{y}}^t) - \sum_{i=1}^{n} \left[\nabla F(\mathbf{y}_i^t)\right]_{C_i}\right|\right| &\leq \left|\left|\sum_{i=1}^{n} \left(\left[\nabla F(\bar{\mathbf{y}}^t)\right]_{C_i} - \left[\nabla F(\mathbf{y}_i^t)\right]_{C_i}\right)\right|\right| \\
&\leq \sum_{i=1}^{n} \left|\left|\left[\nabla F(\bar{\mathbf{y}}^t)\right]_{C_i} - \left[\nabla F(\mathbf{y}_i^t)\right]_{C_i}\right|\right| && (7.26) \\
&\leq \sum_{i=1}^{n} \left|\left|\nabla F(\bar{\mathbf{y}}^t) - \nabla F(\mathbf{y}_i^t)\right|\right| && (7.27) \\
&\leq \frac{n^{3/2}DL}{T(1-\beta)} && (7.28)
\end{aligned}$$

where (7.26) follows from the triangle inequality, (7.27) follows by the definition of the set $\mathcal{C}_i$, and (7.28) follows from the assumption that ∇F is L-Lipschitz continuous and from Lemma 7.10. $\square$

7.11. Appendix C: Proof of Theorem 27

This Appendix establishes the main result of this paper, which is restated here for convenience.

Theorem 29. *Consider the* `CDCG` *algorithm described in Algorithm 13. Let* $\mathbf{y}^*$ *denote the global maximizer of the optimization problem defined in Problem 26, and assume that a positive integer* T *and a doubly-stochastic weight matrix* $\mathbf{W}$ *are given. Then provided that the assumptions outlined in Appendix A hold, for all nodes* $i \in \mathcal{N}$*, the local variables* $\mathbf{y}_i^T$ *obtained after* T *iterations satisfy*

$$F(\mathbf{y}_i^T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{y}^*) - \left[\frac{LD^2}{2T} + \frac{LD^2(n^2 + n^{5/2}) + n^{5/2}DG}{T(1-\beta)}\right] \tag{7.29}$$

where D*,* G*,* L*, and* β *are problem-dependent constants that are formally defined in Appendices A and B.*

Proof. Due to the assumption that ∇F is L-Lipschitz,

$$\begin{aligned} &F\left(\bar{\mathbf{y}}^{t+1}\right) - F\left(\bar{\mathbf{y}}^{t}\right) \\ &\quad \geq \left\langle \nabla F\left(\bar{\mathbf{y}}^{t}\right), \bar{\mathbf{y}}^{t+1} - \bar{\mathbf{y}}^{t} \right\rangle - \frac{L}{2}\left|\left|\bar{\mathbf{y}}^{t+1} - \bar{\mathbf{y}}^{t}\right|\right|^2 \\ &\quad \geq \left\langle \nabla F\left(\bar{\mathbf{y}}^{t}\right), \bar{\mathbf{y}}^{t+1} - \bar{\mathbf{y}}^{t} \right\rangle - \frac{LD^2}{2T^2} \end{aligned} \tag{7.30}$$

where (7.30) follows from Lemma 7.10. Now consider that the inner-product term on the RHS of (7.30) can be written in the following way:

$$\begin{aligned} \left\langle \nabla F\left(\bar{\mathbf{y}}^{t}\right), \bar{\mathbf{y}}^{t+1} - \bar{\mathbf{y}}^{t} \right\rangle &= \left\langle \nabla F\left(\bar{\mathbf{y}}^{t}\right), \frac{1}{T}\sum_{i=1}^{n} \mathbf{v}_i^{t+1} \right\rangle \\ &= \frac{1}{T}\sum_{i=1}^{n}\Big[\left\langle \nabla F(\bar{\mathbf{y}}^t) - \nabla F(\mathbf{y}_i^t), \mathbf{v}_i^{t+1}\right\rangle + \left\langle \nabla F(\mathbf{y}_i^t), \mathbf{v}_i^{t+1}\right\rangle\Big]. \end{aligned} \tag{7.31}$$

Here (7.31) follows from the linearity of inner products and then from adding and subtracting $\nabla F(\mathbf{y}_i^t)$.

Our immediate goal is to bound (7.31) from below. To do so, consider that by the Cauchy-Schwartz inequality,

$$\begin{aligned}
&\left\langle \nabla F\left(\bar{\mathbf{y}}^t\right) - \nabla F(\mathbf{y}_i^t), \mathbf{v}_i^{t+1} \right\rangle \\
&\qquad \leq \left|\left| \nabla F(\bar{\mathbf{y}}^t) - \nabla F(\mathbf{y}_i^t) \right|\right| \cdot \left|\left| \mathbf{v}_i^{t+1} \right|\right| \\
&\qquad \leq LD \left|\left| \bar{\mathbf{y}}^t - \mathbf{y}_i^t \right|\right| \qquad (7.32) \\
&\qquad \leq \frac{n^{3/2} LD^2}{T(1-\beta)} \qquad (7.33)
\end{aligned}$$

where (7.32) is due to the assumption that ∇F is L-Lipschitz continuous and (7.33) follows from Lemma 7.10. Next, because $\mathbf{v}_i^{t+1}$ is defined as the argmax between $\nabla F(\mathbf{y}_i^t)$ and vectors $\mathbf{x} \in \mathcal{P}_i \cap \mathcal{C}_i$ in the Step 3 of Algorithm 13 and by Lemma 7.10 we have

$$\langle \nabla F(\mathbf{y}_i^t), \mathbf{v}_i^{t+1} \rangle \geq \langle [\nabla F(\mathbf{y}_i^t)]_{C_i}, \mathbf{y}^* \rangle. \qquad (7.34)$$

By Lemma 7.10, if we let $\epsilon = \frac{n^{3/2} DL}{T(1-\beta)}$, we can conclude that

$$-\epsilon \mathbf{1} + \nabla F(\bar{\mathbf{y}}^t) \leq \sum_{i=1}^{n} \left[\nabla F(\mathbf{y}_i^t)\right]_{\mathcal{C}_i} \leq \nabla F(\bar{\mathbf{y}}^t) + \epsilon \mathbf{1}. \qquad (7.35)$$

By construction, $\mathbf{y}^* \succeq 0$ since $\mathbf{y}^* \in \mathcal{P}$. Then we can infer from (7.35) that

$$\left\langle \sum_{i=1}^{n} \left[\nabla F(\mathbf{y}_i^t)\right]_{\mathcal{C}_i}, \mathbf{y}^* \right\rangle \geq \langle -\epsilon \mathbf{1}, \mathbf{y}^* \rangle + \left\langle \nabla F(\bar{\mathbf{y}}^t), \mathbf{y}^* \right\rangle. \qquad (7.36)$$

Our goal is to bound (7.36). To do this, consider that $||\mathbf{y}^*|| \leq D$ by (7.10) and $\langle \mathbf{1}, \mathbf{y}^* \rangle = ||\mathbf{y}^*||_1$ since $\mathbf{y}^* \succeq 0$. Since $||\mathbf{y}^*||_1 \leq \sqrt{n} \, ||\mathbf{y}^*||_2$, we have $\langle \epsilon \mathbf{1}, \mathbf{y}^* \rangle \leq D\epsilon\sqrt{n}$. Thus by replacing $\epsilon = \frac{n^{3/2} DL}{T(1-\beta)}$, we conclude that

$$\begin{aligned}
\left\langle \sum_{i=1}^{n} \left[\nabla F(\mathbf{y}_i^t)\right]_{\mathcal{C}_i}, \mathbf{y}^* \right\rangle &\geq \langle \nabla F(\bar{\mathbf{y}}^t), \mathbf{y}^* \rangle - \frac{n^2 LD^2}{T(1-\beta)} \\
&\geq F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^t) - \frac{n^2 LD^2}{T(1-\beta)}. \qquad (7.37)
\end{aligned}$$

Altogether, we have shown via (7.33), (7.34), and (7.37) that (7.31) can be bounded by

$$\langle \nabla F(\bar{\mathbf{y}}^t), \bar{\mathbf{y}}^{t+1} - \bar{\mathbf{y}}^t \rangle \geq \frac{1}{T} \left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^t) - \frac{LD^2(n^2 + n^{5/2})}{T(1-\beta)} \right]. \tag{7.38}$$

Furthermore, (7.38) and (7.30) imply that

$$F\left(\bar{\mathbf{y}}^{t+1}\right) - F\left(\bar{\mathbf{y}}^t\right) \geq \frac{1}{T}\left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^t)\right] - \frac{LD^2(n^2 + n^{5/2})}{T^2(1-\beta)} - \frac{LD^2}{2T^2} \tag{7.39}$$

Rearranging (7.39), we obtain

$$F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^{t+1}) \leq \left(1 - \frac{1}{T}\right)\left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^t)\right] + \frac{LD^2(n^2 + n^{5/2})}{T^2(1-\beta)} + \frac{LD^2}{2T^2}. \tag{7.40}$$

By applying the inequality in (7.40) for $t = 0, 1, \ldots, T-1$, we find

$$\begin{aligned}
&F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^T) \\
&\leq \left(1 - \frac{1}{T}\right)^T \left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^0)\right] + \sum_{i=0}^{T-1} \left(1 - \frac{1}{T}\right)^i \left[\frac{LD^2(n^2 + n^{5/2})}{T^2(1-\beta)} + \frac{LD^2}{2T^2}\right] \\
&= \left(1 - \frac{1}{T}\right)^T \left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^0)\right] + \left(T - T\left(1 - \frac{1}{T}\right)^T\right) \left[\frac{LD^2(n^2 + n^{5/2})}{T^2(1-\beta)} + \frac{LD^2}{2T^2}\right] \\
&\leq \frac{1}{e}\left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^0)\right] + \left(1 - \frac{1}{e}\right) \left[\frac{LD^2(n^2 + n^{5/2})}{T(1-\beta)} + \frac{LD^2}{2T}\right] \\
&\leq \frac{1}{e}\left[F(\mathbf{y}^*) - F(\bar{\mathbf{y}}^0)\right] + \left[\frac{LD^2(n^2 + n^{5/2})}{T(1-\beta)} + \frac{LD^2}{2T}\right]
\end{aligned} \tag{7.41}$$

where to derive (7.41) we used $(1 - 1/T)^T \leq 1/e$. Now recall that we set $\mathbf{y}_i^0 = \mathbf{0}$. Then from equation (7.5), we have $F(\mathbf{0}) \geq 0 \ \forall i \in \mathcal{N}$. Thus follows that

$$F(\bar{\mathbf{y}}^T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{y}^*) - \left[\frac{LD^2(n^2 + n^{5/2})}{T(1-\beta)} + \frac{LD^2}{2T}\right]. \tag{7.42}$$

Now by the assumption made in (7.12), F is G-Lipschitz continuous and therefore

$$\left|F(\bar{\mathbf{y}}^T) - F(\mathbf{y}_i^T)\right| \leq G\left|\left|\bar{\mathbf{y}}^T - \mathbf{y}_i^T\right|\right| \leq \frac{n^{3/2}DG}{T(1-\beta)} \tag{7.43}$$

where (7.43) follows from Lemma 7.10. Thus by combining the results in (7.42) and (7.43) we find that $\forall i \in \mathcal{N}$,

$$F(\mathbf{y}_i^T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{y}^*) - \left[\frac{LD^2}{2T} + \frac{LD^2(n^2 + n^{5/2}) + n^{5/2}DG}{T(1-\beta)}\right]$$

and the claim in (7.9) follows. □

CHAPTER 8

Submodular Meta-Learning

8.1. Introduction

Many applications in artificial intelligence necessitate exploiting prior data and experience to enhance quality and efficiency on new tasks. This is often manifested through a set of tasks given in the training phase from which we can learn a model or representation that can be later used for new unseen tasks in the test phase. In this regard, meta-learning aims at exploiting the data from the available tasks to learn model parameters or representation that can be later used to perform well on new unseen tasks, in particular, when we have access to limited data and computational power at the test time (Thrun and Pratt, 2012; Schmidhuber, 1992; Bengio et al.; Vilalta and Drissi, 2002). By now, there are several formulations for meta-learning, but perhaps one of the most successful ones is the Model-Agnostic Meta-Learning (MAML) framework proposed in (Finn et al., 2017). In MAML, we aim to train the model parameters such that applying a few steps of gradient-based updates with a small number of samples from a new task would perform well on that task. MAML can also be viewed as a way to provide a proper initialization, from which performance on a new task can be optimized after a few gradient-based updates. Alas, this scheme only applies to settings in which the decision variable belongs to a *continuous* domain and can be adjusted using gradient-based methods at the test time.

Our goal is to extend the methodology of MAML to the *discrete* setting. We consider a setting in which our decision variable is a discrete set, and our goal is to come up with a good initial set that can be quickly adjusted to perform well over a wide range of new tasks. In particular, we focus on submodular maximization to represent the tasks which is an essential class of discrete optimization.

There are numerous applications where the submodular meta-learning framework can be applied to find a personalized solution for each task while significantly reducing the computation load. In general, most recommendation tasks can be cast as an instance of this setting(Gabillon et al., 2013;

El-Arini et al., 2009; Yue and Guestrin, 2011). Consider the task of recommending a set of items, e.g., products, locations, ads, to a set of users. One approach for solving such a problem is to find the subset of items that have the highest score over all the previously-visited users and recommend that subset to a new user. Indeed, this approach leads to a reasonable performance at test time; however, it does not provide a user-specific solution for a new user. Another approach is to find the whole subset at the test time when the new user arrives. In contrast to the previous approach, this scheme leads to a user-specific solution, but at the cost of running a computationally expensive algorithm to select all the elements at the test time.

In our meta-learning framework, the process of selecting set items to be recommended to a new user is done in two parts: In the first part, a set of items is selected offline according to prior experience. These items are the most popular items to the previously-visited users (depending on the context). In the second part, which happens at the test time, a set of items that is *personalized* to the coming user is selected. These are items that are computed specifically according to the features of the coming user. In this manner, the computation for each coming user would be reduced to the selection of the second part, which typically constitutes a small portion of the final set of recommended items. The first part can be done offline with a lower frequency. For instance, in a real recommender system, the first part can be computed once every hour, and the second part can be computed specifically for each coming user (or for a class of similar users). While we have mentioned recommendation (or more generally facility location) as a specific example, it is easy to see that this framework can be easily used to reduce computation in other notable applications of submodular optimization.

Contributions. Our contributions are threefold:

- We propose a novel discrete meta-learning framework where each task is equivalent to maximizing a set function under some cardinality constraint. Our framework aims at using prior data, i.e., previously visited tasks, to train a proper initial solution set that can be quickly adapted to a new task at a low computational cost to obtain a task-specific solution.

- We present computationally efficient deterministic and randomized meta-greedy algorithms to

solve the resulting meta-learning problem. When the tasks are monotone and submodular, we prove that the solution obtained by the deterministic algorithm is at least 0.53-optimal, and the solution of the randomized algorithm is $(1 - 1/e - o(1))$-optimal in expectation, where the $o(1)$ term vanishes by the size of the solution. These guarantees are obtained by introducing new techniques, despite that the meta-learning objective is *not* submodular.

- We study the performance of our proposed meta-learning framework and algorithms for movie recommendation and ride-sharing problems. Our experiments illustrate that the solution of our proposed meta-learning scheme, which chooses a large portion of the solution in the training phase and a small portion adaptively at test time, is very close to the solution obtained by choosing the entire solution at the test time when a new task is revealed.

8.1.1. Related work

Continuous Meta-Learning. Meta-learning has gained considerable attention recently mainly due to its success in few shot learning (Vinyals et al., 2016; Ravi and Larochelle, 2017; Snell et al., 2017; Wang and Yao, 2019) as well as reinforcement learning (Duan et al., 2016; Wang et al., 2016; Song et al., 2020; Fallah et al., 2020b). One of the most successful forms of meta-learning is the gradient-based *Model Agnostic Meta-learning* (MAML) approach(Finn et al., 2017). MAML aims at learning an initialization that can be adapted to a new task after performing one (or a few) gradient-based update(s); see, e.g., (Fallah et al., 2020a). This problem can be written as

$$\min_{w \in W} \mathbb{E}_{a \sim P}[f_a(w - \nabla f_a(w))], \tag{8.1}$$

where $W \subseteq \mathbb{R}^d$ is the feasible set and P is the probability distribution over tasks. The previous works on MAML including (Nichol et al., 2018; Finn et al., 2018; Grant et al., 2018; Yoon et al., 2018; Antoniou et al., 2019; Rajeswaran et al., 2019; Fallah et al., 2020a; Collins et al., 2020) consider the case where W is a continuous space. In fact none of these works can be applied to the case where the feasible parameter space is discrete. In this paper, we aim to close this gap and extend the terminology of MAML to discrete settings.

Submodular Maximization. Submodular functions have become key concepts in numerous applications such as data summarization (Lin and Bilmes, 2011; Wei et al., 2013; Kirchhoff and Bilmes, 2014; Mirzasoleiman et al., 2016a), viral marketing (Kempe et al., 2003), sensor placement (Krause et al., 2008b), dictionary learning (Das and Kempe, 2011), and influence maximization (Kempe et al., 2003). It is well-known that for maximizing a monotone and submodular function under the cardinality constraint, the greedy algorithm provides a $(1 - 1/e)$-optimal solution (Krause and Golovin, 2014; Nemhauser and Wolsey, 1978; Wolsey, 1982). There has been significant effort to improve the scalability and efficiency of the greedy algorithm using lazy, stochastic, and distributed methods (Mirzasoleiman et al., 2015; Karimi et al., 2017; Barbosa et al., 2015; Mirrokni and Zadimoghaddam, 2015; Kumar et al., 2015; Hassani et al., 2017; Mokhtari et al., 2020a; Hassani et al., 2019; Balkanski et al., 2019). However, our framework is fundamentally different and complementary to these approaches as it proposes a new approach to use data at training time to improve performance at new tasks. Indeed, all the aforementioned techniques can be readily used to further speed-up our algorithms. Optimization of related submodular tasks has been a well-studied problem with works on structured prediction (Lin and Bilmes, 2012), submodular bandits (Yue and Guestrin, 2011; Zhang et al., 2019), online submodular optimization (Jegelka and Bilmes, 2011; Streeter and Golovin, 2009; Golovin et al., 2014; Chen et al., 2018a), and public-private data summarization (Mirzasoleiman et al., 2016b). However, unlike our work, these approaches are not concerned with train-test phases for optimization. Another recently-developed methodology to reduce computation is the two-stage submodular optimization framework (Balkanski et al., 2016; Mitrovic et al., 2018; Stan et al., 2017b), which aims at summarizing the ground set to a reasonably small set that can be used at test time. The main difference of our framework with the two-stage approaches is that we allow for *personalization*: A small subset of items that can be found at test time specific to the task at hand. This leads to a completely new problem formulation, and consequently, new algorithms.

8.2. Problem Statement: Discrete Meta-Learning

Setup. We consider a family of tasks $\mathcal{T} = \{\mathcal{T}_i\}_{i \in \mathcal{I}}$, where the set $\mathcal{I}$ could be of infinite size. Each task $\mathcal{T}_i$ is represented via a set function $f_i : 2^V \to \mathbb{R}_+$ that measures the reward of a set $S \subseteq V$ for the i-th task, and performing the task $\mathcal{T}_i$ would mean to maximize the function f_i subject to

a given constraint. For instance, in a recommender system where we aim to recommend a subset of the items to the users, the set $\mathcal{I}$ denotes the set of all the possible users and selecting which items to recommend to a user $i \in \mathcal{I}$ is viewed as the task $\mathcal{T}_i$. Moreover, the function f_i encodes the users satisfaction, i.e., $f_i(S)$ quantifies how suitable the set of items S is for user i. Taking a statistical perspective, we assume that the tasks $\mathcal{T}_i$ occur according to a possibly unknown probability distribution $i \sim p$.

In this paper, we focus on the case where the functions f_i are monotone and submodular set functions and each task $\mathcal{T}_i$ amounts to maximizing f_i under the k-cardinality constraint. That is, the task $\mathcal{T}_i$ is to select a subset $S \subseteq V$ of size k such that the value of $f_i(S)$ is maximized. Submodularity of f_i means that for any $A, B \subseteq V$ the following inequality holds $f_i(A) + f_i(B) \geq f_i(A \cup B) + f_i(A \cap B)$. Furthermore, f_i is called monotone if for any $A \subseteq B$ we have $f_i(A) \leq f_i(B)$.

Training and test tasks. We assume access to a collection of *training* tasks $\{\mathcal{T}_i\}_{i=1}^m$. These are the tasks that we have already experienced, i.e., they correspond to the users that we have already seen. Formally, this means that for each training task $\mathcal{T}_i$, we assume knowledge of the corresponding function f_i. In our formulation, each of the training tasks is assumed to be generated i.i.d. according to the distribution p. Indeed, eventually we aim to optimize performance at *test* time, i.e., obtain the best performance for new and unseen tasks generated independently from the distribution p. For instance, in our recommendation setting, test tasks correspond to new users that will arrive in the future. Our goal is to use the training tasks to reduce the computation load at test time.

Two extremes of computation. Let us use $\mathcal{T}_{\text{test}}$ (and f_{test}) to denote the task (and its corresponding set function) that we aim to learn at test time. Ideally, if we have sufficient computational power, then we should directly optimize f_{test} by solving the following problem

$$\max_{S \in V, |S| \leq k} \; f_{\text{test}}(S). \tag{8.2}$$

We denote the optimal solution of (8.2) by S^*_{test}. For instance, we can use the greedy procedure to

solve (8.2) which leads to a $(1-1/e)$-optimal solution using $\mathcal{O}(kn)$ evaluations of f_{test}, and through k passes over the ground set. However, the available computational power and time in the test phase is often limited, either because we need to make quick decisions to respond to new users or since we need to save energy. For instance, in real-world advertising or recommendation systems, both these requirements are crucial: many users arrive within each hour which means fast optimization is crucial (especially if n, k are large), and also, reducing computation load would lead to huge energy savings in the long run. In such cases, Problem (8.2) should be solved approximately with less computation.

An alternative to reduce computation at test time is to solve the problem associated with the expected reward over all possible tasks in the training phase (when we have sufficient computation time), i.e.,

$$\max_{S \in V, |S| \leq k} \ \mathbb{E}_{i \sim p} \left[f_i(S) \right]. \tag{8.3}$$

We denote the optimal solution of (8.3) by S^*_{exp}. The rationale behind this approach is that the optimal solution to this problem would generalize well over an unseen task if the new task is also drawn according to the probability distribution p. In other words, the solution of (8.3) should perform well for the problem in (8.2) that we aim to solve at the test time, assuming that f_{test} is sampled according to p. In this way, we do not need any extra computation at the test time. However, in this case, the solution that we obtain would not be the best possible solution for the task that we observe at the test time, i.e., S^*_{test} is not equal to S^*_{exp}. Note that we often do not have access to the underlying probability distribution p, and we only have access to a large set of realizations of tasks in the training phase. As a result, instead of solving (8.3), we settle for maximizing the sample average function

$$\max_{S \in V, |S| \leq k} \ \frac{1}{m} \sum_{i=1}^{m} f_i(S), \tag{8.4}$$

where m is the number of available tasks in the training phase.

Problems (8.2) and (8.4) can be considered as two different extreme cases. In the first option, by solving (8.2), we avoid any pre-processing in the training phase, and we obtain the best possible guarantee for the new task, but at the cost of performing computationally expensive operations

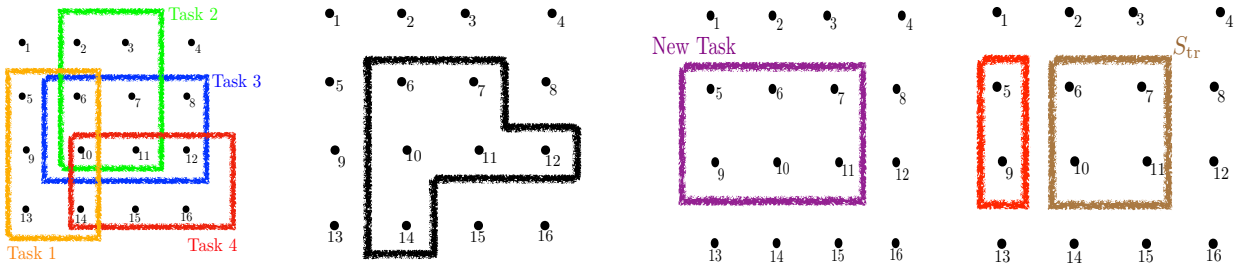

Figure 8.1: (a) Optimal sets for each of the training tasks ($k = 6$); (b) the set obtained by solving the average problem in (8.4); (c) the optimal set for a new task revealed at test time, i.e. solving the problem in (8.2); (d) the optimal set for the new task is also obtained by solving the meta-learning problem in (8.7) with $l = 4$ (brown set) and adding the task-specific elements at test time (red set).

(e.g., full greedy) at the test time. In the second approach, by solving (8.4) in the training phase, we obtain a solution that possibly performs reasonably without any computation at the test phase, but the quality of the solution may not be as good as the first option. In summary, there exists a trade-off between the required computational cost at the test time and the performance guarantee on the unseen task. Hence, a fundamental question that arises is what would be the best scheme at the training phase assuming that at test time we have some limited computational power. For instance, in the monotone submodular case, assume that instead of running the greedy algorithm for k rounds, which has a complexity of $\mathcal{O}(kn)$, we can only afford to run αk rounds of greedy at test time, which has a complexity of $\mathcal{O}(\alpha nk)$, where $\alpha \in (0, 1)$ is small. In this case, a natural solution would be to find an appropriate set of $(1-\alpha)k$ elements in the training phase, and add the remaining αk elements at test time when a new task arrives. This discussion also applies to any other greedy method (e.g., lazy or stochastic greedy). We now formally state this problem.

Discrete Meta-Learning. As we discussed so far, when computational power is limited at test time, it makes sense to divide the process of choosing the best decision between training and test phases. To be more specific, in the training phase, we choose a subset of elements from the ground set that would perform over the training tasks, and then select (or optimize) the remaining elements at the test time *specifically* with respect to the task at hand. To state this problem, consider $S_{\text{tr}} \subseteq V$ with cardinality $|S_{\text{tr}}| = l$, where $l < k$, as the initial set that we aim to find at the training phase,

and the set S_i that we add to the initial set S_{tr} at test time (See Figure 8.1 for an illustration). Hence, the problem of interest can be written as

$$\max_{S_{\text{tr}} \in V, |S_{\text{tr}}| \leq l} \; \mathbb{E}_{i \sim p} \Big[\max_{S_i \in V, |S_i| \leq k-l} f_i(S_{\text{tr}} \cup S_i) \Big], \tag{8.5}$$

Note that the critical decision variable that we need to find is S_{tr} which is the best initial subset of size l overall all possible choices of task when a best subset of size $k-l$ is added to that. In fact, if we define $f_i'(S_{\text{tr}}) := \max_{S_i \in V, |S_i| \leq k-l} f_i(S_{\text{tr}} \cup S_i)$, then we can rewrite the problem in (8.5) as

$$\max_{S_{\text{tr}} \in V, |S_{\text{tr}}| \leq l} \; \mathbb{E}_{i \sim p} \; \left[f_i'(S_{\text{tr}}) \right]. \tag{8.6}$$

As described previously, we often do not have access to the underlying probability distribution p of the tasks, and we instead have access to a large number of sampled tasked that are drawn independently according to p. Hence, instead of solving (8.5), we solve its sample average approximation given by

$$\max_{S_{\text{tr}} \in V, |S_{\text{tr}}| \leq l} \; \frac{1}{m} \sum_{i=1}^{m} \; \left[\max_{S_i \in V, |S_i| \leq k-l} f_i(S_{\text{tr}} \cup S_i) \right] \; = \; \max_{S_{\text{tr}} \in V, |S_{\text{tr}}| \leq l} \; \frac{1}{m} \sum_{i=1}^{m} \; \left[f_i'(S_{\text{tr}}) \right], \tag{8.7}$$

where m is the number of tasks in the training set which are sampled according to p. Even though the functions f_i are submodular, f_i' *is not submodular* or k-submodular (Ohsaka and Yoshida, 2015) (see Appendix 8.11 for specific counter examples). Hence, Problem (8.7) is not a submodular maximization problem. In the next section, we present algorithms for solving Problem (8.7) with provable guarantees.

We finally note that Problem (8.7) will be solved at *training* time to find the solution S_{tr} of size l. This solution is then *completed at test time*, by, e.g., running $k-l$ further rounds of greedy on the new task, to obtain a task-specific solution of size k.

8.3. Algorithms for Discrete Submodular Meta-Learning

Solving Problem (8.7) requires finding a set S_{tr} for the outer maximization and sets $\{S_i\}_{i=1}^m$ for the inner maximization. In this section, we describe our proposed greedy-type algorithms to select the elements S_{tr} and $\{S_i\}_{i=1}^m$. As we deal with $m+1$ sets, the order in which the sets S_{tr} and $\{S_i\}_{i=1}^m$ are updated becomes crucial, i.e., it is not clear which of the sets S_{tr} or S_i's should be preferably updated in each round and how can the functions f_i be incorporated in finding the right order, which is the main challenge in designing greedy methods to solve (8.7). We design greedy procedures with both deterministic and randomized orders and provide strong guarantees for their solutions.

8.3.1. Deterministic Algorithms

In this section, we first describe Algorithms 14 and 15 which use specific orderings to solve Problem (8.7). Based on these two, we then design Algorithm 16 as our *main deterministic* algorithm. Throughout this section, we use $\Delta_i(e|S) = f_i(S \cup \{e\}) - f_i(S)$ to denote the marginal gain of adding an element e to set S for function f_i. In brief, Algorithm 14 first fills S_{tr} greedily up to completion and then it constructs each of the S_i's greedily on the top of S_{tr}. Specifically, starting from the empty set initialization for S_{tr} and S_i's, Algorithm 14 constructs in its first phase the set S_{tr} in l rounds, by adding one element per round, where the next element in each round is chosen according to $e^* = \arg\max_{e \in V} \sum_{i=1}^m f_i(S_{\text{tr}} \cup \{e\}) - f_i(S_{\text{tr}})$. Once S_{tr} is completed, in the second phase, each of the sets S_i is constructed in parallel by running the greedy algorithm on f_i. That is, each S_i is updated in $k-l$ rounds where in each round an element with maximum marginal on f_i is added to S_i based on $e_i^* = \arg\max_{e \in V} f_i(S_{\text{tr}} \cup S_i \cup \{e\}) - f_i(S_{\text{tr}} \cup S_i)$.

Algorithm 15 uses the opposite ordering of Algorithm 14. Initializing with all sets to be empty, in the first phase it constructs the sets S_i using the greedy procedure on f_i, i.e., each S_i is updated in parallel in $k-l$ rounds, where in each round the element e_i^* defined as $e_i^* = \arg\max_{e \in V} f_i(S_{\text{tr}} \cup S_i \cup \{e\}) - f_i(S_{\text{tr}} \cup S_i)$ is added to S_i. In the second phase, the set S_{tr} is formed greedily in l rounds, and in each round the element e^* defined as $e^* = \arg\max_{e \in V} \sum_{i=1}^m f_i(S_{\text{tr}} \cup \{e\} \cup S_i) - f_i(S_{\text{tr}} \cup S_i)$ is added.

Algorithm 14

Initialize $S_{\text{tr}} = \{S_i\}_{i=1}^m = \emptyset$
/* Phase 1: */
for $t = 1, 2, \ldots, l$ **do**
  Find $e^* = \arg\max_{e \in V} \sum_{i=1}^m \Delta_i(e|S_{\text{tr}})$
  $S_{\text{tr}} \leftarrow S_{\text{tr}} \cup \{e^*\}$
end for
end for
/* Phase 2: */
for $t = 1, 2, \ldots, k-l$ **do**
  for $i = 1, 2, \ldots, m$ **do**
    Find $e_i^* = \arg\max_{e \in V} \Delta_i(e|S_{\text{tr}} \cup S_i)$
    $S_i \leftarrow S_i \cup \{e_i^*\}$
  end for
  end for
end for
end for
Return S_{tr} and S_i

Algorithm 15

Initialize $S_{\text{tr}} = \{S_i\}_{i=1}^m = \emptyset$
/* Phase 1: */
for $i = 1, 2, \ldots, m$ **do**
  for $t = 1, 2, \ldots, k-l$ **do**
    Find $e_i^* = \arg\max_{e \in V} \Delta_i(e|S_i)$
    $S_i \leftarrow S_i \cup \{e_i^*\}$
  end for
  end for
end for
end for
/* Phase 2: */
for $t = 1, 2, \ldots, l$ **do**
  Find $e^* = \arg\max_{e \in V} \sum_{i=1}^m \Delta_i(e|S_{\text{tr}} \cup S_i)$
  $S_{\text{tr}} \leftarrow S_{\text{tr}} \cup \{e^*\}$
end for
end for
Return S_{tr} and S_i

While the solutions obtained by Algorithms 14 and 15 are guaranteed to be near-optimal, it turns out that they can be complementary with respect to each other. Our *main* deterministic algorithm, called `Meta-Greedy`, runs both Algorithms 14 and 15 and chooses as output the solution, among the two, that leads to a higher objective value in (8.7). To be more specific, if we consider $S_{\text{tr}}^{(1)}, \{S_i^{(1)}\}_{i=1}^m$ as the outputs of Algorithm 14 and $S_{\text{tr}}^{(2)}, \{S_i^{(2)}\}_{i=1}^m$ as the outputs of Algorithm 15, then `Meta-Greedy` compares the values of $\sum_{i=1}^m f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)})$ and $\sum_{i=1}^m f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)})$ and chooses the solution set that has the higher objective function value. Note that as we described earlier, the main output of this procedure should be the set S_{tr} of size l. Hence, the output of `Meta-Greedy` is either $S_{\text{tr}}^{(1)}$ or $S_{\text{tr}}^{(2)}$ and the sets $\{S_i^{(1)}\}_{i=1}^m$ and $\{S_i^{(2)}\}_{i=1}^m$ are only evaluated for the purpose of comparing objective function values.

Next, we explain why our `Meta-Greedy` method can outperform both Algorithms 14 and 15. This will be done by providing the theoretical guarantees for these methods and consequently explaining why Algorithms 14 and 15 are complementary.

Algorithm 16 `Meta-Greedy`

1: Run Algorithms 14 and 15 and obtain respective solution sets $S_{\text{tr}}^{(1)}, \{S_i^{(1)}\}_{i=1}^m$ and $S_{\text{tr}}^{(2)}, \{S_i^{(2)}\}_{i=1}^m$.
2: Compute the objective value $\sum_{i=1}^m f_i(S_{\text{tr}} \cup S_i)$ for both solution sets.
3: Return S_{tr} and S_i of the solution set that has a higher objective value.

Theoretical guarantees. We begin with the analysis of Algorithm 14. The following proposition relates the overall performance of Algorithm 14 to its performance after phase 1 and shows that the output of the algorithm is at least 1/2-optimal. We use OPT for the optimal value of Problem (8.7).

Proposition 3. *Let $S_{\text{tr}}^{(1)}, \{S_i^{(1)}\}_{i=1}^m$ be the output of Algorithm 14, and define β as $\beta := \frac{1}{m}\sum_{i=1}^m f_i(S_{\text{tr}}^{(1)})$. If the set functions f_i are monotone and submodular, then*

$$\frac{1}{m}\sum_{i=1}^m f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)}) \geq \max\Big\{\beta\,,\, (1-1/e)(\text{OPT}-2\beta)+\beta\Big\}.$$

Consequently, the solution obtained by Algorithm 14 is at least 1/2-optimal for any value of β.

Proof. Check Appendix 8.7. □

The proof of this proposition is relegated to the appendix. The key step in the proof is to relate the progress made in phase 1 to the gap to OPT. This is indeed challenging as phase 1 only involves updates on the outer maximization of (8.7). In this regard, we prove a novel technical lemma that can be generally applicable to any mini-max submodular problem. The guarantee given in Proposition 3 is minimized when $\beta = \text{OPT}/2$. If β is small (e.g., $\beta = 0$) or if β is large (e.g. if $\beta = (1-1/e)\text{OPT}$) then the guarantee becomes tight (e.g. $(1-1/e)\text{OPT}$). This is indeed expected from the greedy nature of the two phases of Algorithm 14. What is non-trivial about the result of Proposition 3 is that it provides a strong guarantee for any value of β, and not just cases that β is small or large. Similarly, we can provide near-optimality guarantees for Algorithm 15.

Proposition 4. *Let $S_{\text{tr}}^{(2)}, \{S_i^{(2)}\}_{i=1}^m$ be the output of Algorithm 15, and define γ as $\gamma := \frac{1}{m}\sum_{i=1}^m f_i(S_i^{(2)})$.*

If the set functions f_i are monotone and submodular, then

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)}) \geq \max\Big\{\gamma\,,\,(1-1/e)(\text{OPT}-2\gamma)+\gamma\Big\}.$$

Consequently, the solution obtained by Algorithm 15 is at least $1/2$*-optimal for any value of* γ*.*

Proof. Check Appendix 8.8. □

Similarly, we can show that $\gamma = \text{OPT}/2$ leads to (the worst) guarantee 1/2-OPT, while for large and small values of γ the bound in Proposition 4 approaches the optimal approximation $(1-1/e)\text{OPT}$.

We note that the values β in Proposition 3 (Algorithm 14) and γ in Proposition 4 (Algorithm 15) represent two different extremes. The value β represents the significance of the role of S_{tr} in solving Problem (8.7), and γ represents how significant the role of the sets $\{S_i\}_{i=1}^m$ can be. Even though the worst-case guarantees of Propositions 3 and 4 are obtained when $\beta, \gamma = \text{OPT}/2$, a coupled analysis of the algorithms show that in this case at least one of the algorithms should output a solution which is strictly better than 1/2-optimal. In other words, the outcomes of Algorithms 14 and 15 are dependent to one another, and the best performance is achieved when the maximum of the two is considered. This justifies why our main algorithm `Meta-Greedy` can perform strictly better than each of the Algorithms 14 and 15. Using a coupled analysis of the outcome of Algorithms 14 and 15, we can bound the performance of `Meta-Greedy` for different values of β and γ (see the proof of Theorem 30 in the appendix). In particular, we can show that the output of `Meta-Greedy` is at least 0.53-optimal. The proof of the following theorem carefully analyzes the interplay between the role of the inner and outer maximization problems in (8.7). We emphasize that the proof introduces new techniques applicable to other types of minimax submodular problems.

Theorem 30. *Consider the* `Meta-Greedy` *algorithm outlined in Algorithm 16. If the functions* f_i

are monotone and submodular, then we have

$$\max\Big\{\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)})\,,\ \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)})\Big\} \geq 0.53 \times \text{OPT}. \tag{8.8}$$

Proof. Check Appendix 8.9. □

Remark 7. *Note that for all the results in Propositions 3 and 4 as well as Theorem 30, for given output sets S_{tr} or $\{S_i\}_{i=1}^m$, the value of $\frac{1}{m}\sum_{i=1}^m f_i(S_{\text{tr}} \cup S_i)$ is a lower bound for the objective function value of Problem* (8.7) *evaluated at the output set S_{tr}. To be more precise, the accurate measure for evaluating the quality of the output set S_{tr} is $\frac{1}{m}\sum_{i=1}^m \big[\max_{S_i \in V, |S_i| \leq k-l} f_i(S_{\text{tr}} \cup S_i)\big]$ which is indeed larger than $\frac{1}{m}\sum_{i=1}^m f_i(S_{\text{tr}} \cup S_i)$. Hence, all the guarantees that have obtained in the statements above (as well as Theorem 31 below) would directly translate into the same guarantees when we evaluate the objective in* (8.7) *on the set S_{tr}.*

8.3.2. Randomized Algorithm

In this section, we consider greedy procedures in which the decision to alternate between the set S_{tr} (the outer maximization) and the sets $\{S_i\}_{i=1}^m$ (the inner maximization) is done based on a randomized scheme. The `Randomized meta-Greedy` procedure, outlined in Algorithm 17, provides a specific randomized order. In each round, with probability l/k we choose to perform a greedy update on S_{tr}, and with probability $1 - l/k$ we choose to perform a greedy update on *all* the S_i's, $i = 1, \cdots, m$. This procedure continues until either S_{tr} or $\{S_i\}_{i=1}^m$ hit their corresponding carnality constraint, in which case we continue to update the other set(s) greedily until they also become full.

The randomized update of Algorithm 17 is designed to optimally connect the expected increase the objective value at each round with the gap to OPT (as shown in the proof of Theorem 31). Hence, the `Randomized meta-Greedy` procedure is able to achieve in expectation a guarantee close to the tight value $(1 - 1/e)$OPT. However, due to the randomized nature of the algorithm, the sets S_{tr} or S_i might hit their carnality constraint earlier than expected. Analyzing the function value at this "stopping time" is another technical challenge that we resolve in the following theorem to obtain a guarantee that becomes slightly worse than $(1 - 1/e)$OPT depending on the values of $k - l$ and l.

Algorithm 17 `Randomized meta-Greedy`

Initialize the sets S_{tr} and $\{S_i\}_{i=1}^m$ to the empty set.
while $\mid S_i \mid < k-l$ and $\mid S_{\text{tr}} \mid < l$ **do**
  $e_i^* \leftarrow \arg\max_{e \in V} f_i(S_{\text{tr}} \cup S_i \cup \{e\}) - f_i(S_{\text{tr}} \cup S_i)$
  $e_{tr}^* \leftarrow \arg\max_{e \in V} \sum_{i=1}^m f_i(S_{\text{tr}} \cup S_i \cup \{e\}) - f_i(S_{\text{tr}} \cup S_i)$
  w.p. $\frac{l}{k}$**:** $S_{\text{tr}} = S_{\text{tr}} \cup \{e_{tr}^*\}$
  w.p. $\frac{k-l}{k}$**:** $S_i = S_i \cup \{e_i^*\}$, $\forall i = 1, \cdots, m$
end while
end
If S_{tr} or S_i's have not yet reached their cardinality limit then fill them greedily until their limit is reached
Return S_{tr} and S_i

Theorem 31. *Let the (random) sets S_{tr}, $\{S_i\}_{i=1}^m$ be the output of Algorithm 17. If the functions f_i are monotone and submodular, then*

$$\mathbb{E}\Big[\frac{1}{m}\sum_{i=1}^m f_i(S_{\text{tr}} \cup S_i)\Big] \geq \left(1 - \frac{1}{e} - b\right) \text{OPT},$$

where $b \to 0$ as $k-l$ and l grow. More precisely, letting $c = \max\{\frac{1}{k-l}, \frac{1}{l}\}$, we have $b = c + (\exp(3\sqrt{c \log 1/c}) - 1)/e = \mathcal{O}(\sqrt{c \log 1/c})$.

Proof. Check Appendix 8.10. □

Remark 8. *All presented algorithms are designed for the training phase and their output is the set S_{tr} with size l. The sets $\{S_i\}_{i=1}^m$ are only computed for algorithmic purposes. Given a new task at the test phase, the remaining $k-l$ task-specific elements will be added to S_{tr} using for instance greedy updates that require a total complexity of $O((k-l)n)$ in function evaluations. Also, the training complexity of the proposed algorithms is $O(kmn)$, however, certain phases can be implemented in parallel.*

8.4. Simulation Results

We provide two experimental setups to evaluate the performance of our proposed algorithms and compare with other baselines. Each setup involves a different set of tasks which are represented as submodular maximization problems subject to the k-cardinality constraint. In our experiments, we

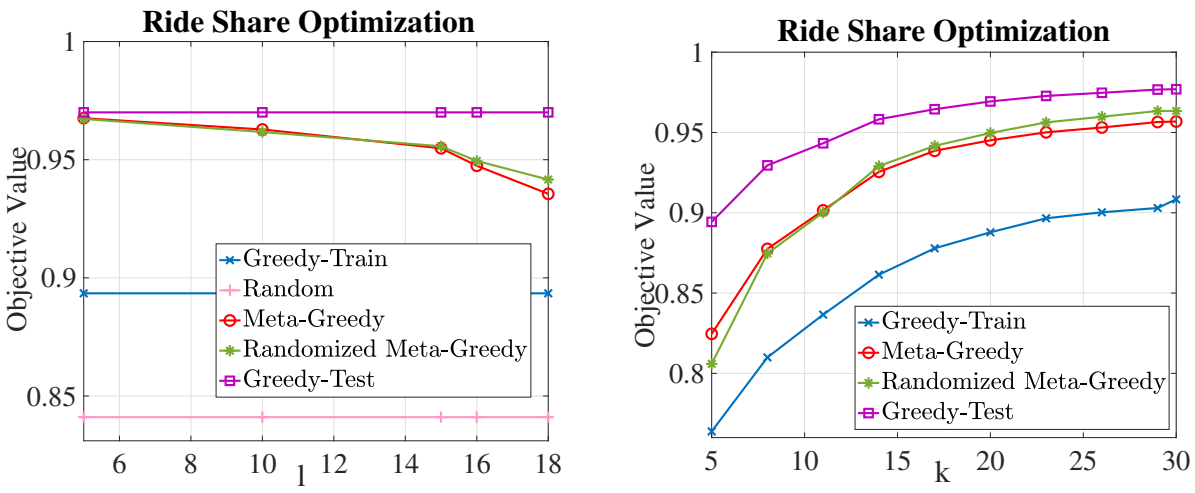

Figure 8.2: Performance for Ride Share Optimization.

consider the following algorithms: **Meta-Greedy** (Algorithm 16), **Randomized Meta-Greedy** (Algorithm 17), **Greedy-Train** (which chooses all the k elements during the training phase–see (8.4) and the discussion therein), **Greedy-Test** (which chooses all the k elements during the test phase–see (8.2) and the discussion therein), and **Random** (which chooses a random set of k elements). In the following, we briefly explain the data and tasks and refer the reader to the supplementary materials for more details.

Ride Share Optimization. We will formalize and solve a facility location problem on the Uber dataset (UberDataset). Our experiments were run on the portion of data corresponding to Uber pick-ups in Manhattan in the period of September 2014. This portion consists of $\sim 10^6$ data points each represented as a triplet $(latitude, longitude, DateTime)$. A customer and a driver are specified through their locations on the map. We use $u = (x_u, y_u)$ for a customer a and $r = (x_r, y_r)$ for a driver. We define the "convenience score" of a (customer, driver) pair as $c(u, r) = 2 - \frac{2}{1+e^{-200d(u,r)}}$, where $d(u, r)$ denotes the Manhattan distance (Mitrovic et al., 2018). Given a specific time a, we define a time slot T_a and picking inside the data set 10 points in half an hour prior to time a, and for each point we further pick 10 points in its 1 km neighborhood, which makes a total of 100 points (locations) on the map. A task $\mathcal{T}_i$ takes place at a corresponding time a_i, and by defining

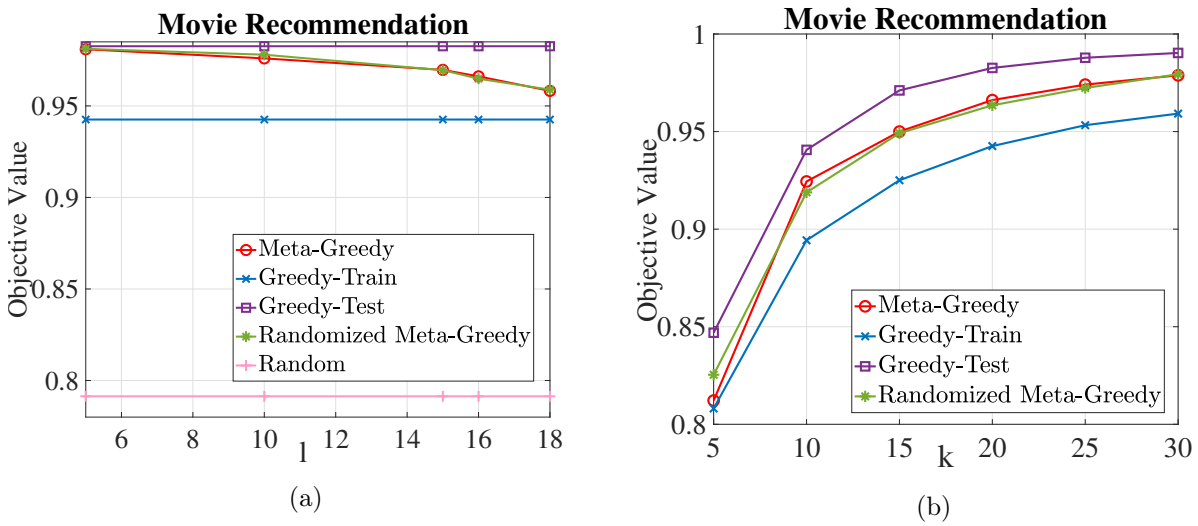

Figure 8.3: Performance for Movie Recommendation.

the set of locations T_{a_i} as above, we let f_i be a monotone submodular function defined over a set S of driver locations as $f_i(S) = \sum_{u \in T_{a_i}} \max_{r \in S} c(u, r)$. We pick 100,000 locations at random from the September 2014 Uber pick-up locations as a ground set. For training we form $m = 50$ tasks by picking for each task a random time in the *first* week of Sept. 2014. We test on $m = 50$ new tasks formed similarly from the *second* week of Sept. 2014 and report in the figures the average performance obtained at test tasks.

Figures 8.2a and 8.2b show the performance of our proposed algorithms against the baselines mentioned above. Figure (8.2a) shows the performance of all algorithms when we fix $k = 20$, and vary l from 5 to 18. Larger l means less computation at test time (since we need to further choose $k - l$ elements at test). However, we see that even for large values of l (e.g. $l = 16$), the performance of Meta-Greedy is still quite close to the ideal performance of Greedy-Test. Putting this together with the fact that the performance of Greedy-Train is not so good, we can conclude that adding a few personalized elements at test time significantly boosts performance to be even close to the ideal. In Figure (8.2b), we compare the performance of all the algorithms when k changes from 5 to 30, and l is 80% of k ($l = \lfloor 0.8k \rfloor$). As we can see, even when we just learn 20% of the set in test time, the performance of Meta-greedy is close to Test-Greedy. Also, when $k - l$ increases, Random-Meta-Greedy performs better than Meta-Greedy. This is in compliance with the results of

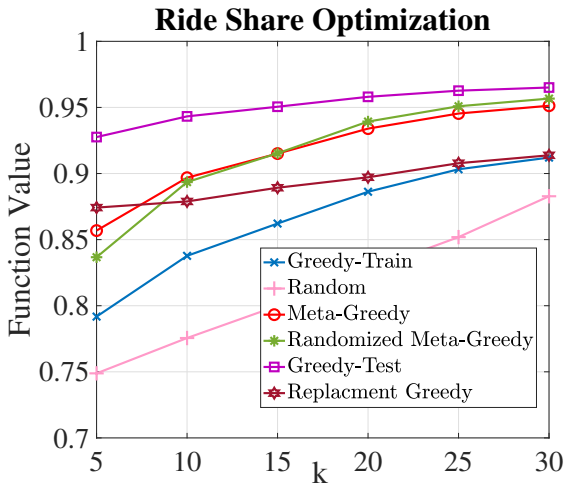

Figure 8.4: Comparison of two-stage framework and submodular meta-learning framework

Theorems 30, 31.

Movie Recommendation. In this application, we use the Movielens dataset (Harper and Konstan, 2015) which consists of 10^6 ratings (from 1 to 5) by 6041 users for 4000 movies. We pick the 2000 most rated movies, and 200 users who rated the highest number of movies (similar to (Stan et al., 2017b)). We partitioned the 200 users into 100 users for the training phase and 100 other users for the test phase. Each movie can belong to one of 18 genres. For each genre t we let G_t be the set of all movies with in genre t. For each user i, we let R_i be the set of all movie rated by the user, and for each movie $v \in R_i$ the corresponding rating is denoted by $r_i(v)$. Furthermore, for user i we define $f_i(S) = \sum_{t=1}^{18} w_{i,t}. \max_{v \in R_i \cap G_t \cap S} r_i(v)$ which is the weighted average over maximum rate that user i gives to movies from each genre and $w_{i,t}$ is proportion of movies in genre t which is rated by user i out of all the rating he provides. A task $\mathcal{T}_i$ involves 5 users $i_1, \cdots, i_5$ and the function assigned to the task is the average of $f_{i_1}, \cdots, f_{i_5}$. We formed $m = 50$ training tasks from the users in the training phase, and $m = 50$ test tasks from the users in the test phase. Figure (8.3a) (resp. 8.3b) has been obtained in a similar format as Figure 8.2a (resp. Figure 8.2b). We observe a very similar pattern as in the ride share experiments.

8.5. Comparison with Two-stage Submodular Optimization

Two-stage submodular optimization is another way to deal with limited computational power in test time. In this framework, at training time, a reduced ground set is learned which will be used as a ground set at test time. This procedure will reduce the computational time in test time. More formally, the two-stage submodular optimization framework aims to solve the following problem. Let $f_i : 2^{\mathcal{X}} \to \mathbb{R}_+$ for $i \in [m]$, be a monotone submodular function over ground set V. The goal is to find S with size at most q whose subests of size k maximize the sum of f_i for $i \in [m]$:

$$\max_{S \subseteq \mathcal{X}, |S| \leq q} \frac{1}{m} \sum_{i=1}^{m} \max_{S_i \subseteq S, |S_i| \leq k} f_i(S_i) \tag{8.9}$$

Once the set S is found, it will be used in the test phase (e.g., by running full greedy on S as the reduced ground set) to find k elements for a new task. This framework uses $\mathcal{O}(qk)$ function evaluations for each new test task; however, it poorly personalizes to a test task because the set S has been optimized only for the tasks at the training time. This intuition is indeed consistent with our experimental findings reported below. We further remark that the two-stage framework requires very high computational power in training. For example, the Replacement-Greedy algorithm (Stan et al., 2017b) requires computational complexity $\mathcal{O}(qkmn)$ (which is a factor q larger than the complexity of the algorithms in this paper). As a result of this issue, we were not able to run the state-of-the-art two-stage algorithms to solve (8.9) in the setting considered in our main simulation results (presented in Section 8.4). e.g., for ground set of size $n = 10^5$ our two-stage implementation would take a very long time.

We have considered the ride-sharing application discussed in Section 8.4 and let $n = 500$ (ground set size), $m = 50$ (number of tasks), and k changing from 5 to 30 (cardinality constraint) while $l = 80\%k$ (portion that will fill in the submodular meta-learning during training), and $q = 100$ (size of reduced ground set for two-stage framework). For solving the two-stage problem (8.9) we have used the Replacement-Greedy algorithm introduced in (Stan et al., 2017b). We choose these parameters based on the following two facts:

1. Because of the high computational cost of the Replacement Greedy algorithm in training for the ride-sharing application, we chose n to be 500.

2. We provide a fair comparison in terms of computational power at test time, which means both Meta-Greedy (our algorithm) and Replacement-Greedy have exactly *the same computational cost* at test time. Formally, $n(k-l) = qk$.

we report the result for the above setting in the Figure 8.4. A few comments are in order: (i) The two stage implementation reduces the ground set of size $n = 500$ to $q = 100$. When k is small, some of the popular elements found at training time would be good enough to warrant a good performance at test time. However, when k increases, the role of personalizing becomes more apparent. As we see, the performance of Replacement-Greedy does not improve much when we increase k and it is close to the performance of Greedy-Train (which chooses all the k elements during the training phase–see (8.4) and the discussion therein). However, since Meta-Greedy does (a small) task-specific optimization at test time, its performance becomes much better. We emphasize again that, in order to be fair, the comparison in Figure 8.4 has been obtained using *the same* computational power allowed at test time for both meta-learning and two-stage approaches.

8.6. Conclusion and Future Work

In this paper, we extended the notion of Model-Agnostic Meta-Learning (MAML) to discrete optimization and in particular to submodular maximization. We proposed a novel formulation in which we aim to find an initial solution set that can be quickly adapted to a new task at a relatively low computational cost. In our meta-learning framework, the process of selecting set items is done in two parts: In the first part, a set of items are selected offline according to prior experience and data. In the second part, which happens at test time, a set of elements that is personalized to the new revealed task is selected. For the proposed problem, we introduced a deterministic variant of the greedy algorithm which obtains a solution that is at least 0.53-optimal, when the tasks are monotone and submodular. We further presented a randomized algorithm that improves this result and obtains $(1 - 1/e - o(1))$-approximation in expectation. We also studied the performance of our proposed meta-learning framework and algorithms for two real-world applications: movie recommendation

and ride-sharing problems. Our numerical results indicate the advantage of our proposed scheme with respect to traditional learning procedures as well as methods based on two-stage submodular optimization.

There are numerous open directions that can be investigated along the lines of discrete meta-learning and user-specific adaptation for discrete problems (indeed, this work can be considered as a first step). Examples include extending the results to a more general setting when the tasks are (approximately) submodular but non-monotone, and considering the case that the tasks at training and test times are drawn according to two different probability distributions (possibly with bounded distance).

8.7. Proof of Proposition 3

Let S_{tr}, $\{S_i\}_{i=1}^m$ be the output of Algorithm 14 and S_{tr}^*, $\{S_i^*\}_{i=1}^m$ be the optimal solution for problem (8.7). We first show that the output of Algorithm 14 in phase 1 satisfies the following inequality:

$$\sum_{i=1}^{m} f_i(S_{\text{tr}}^* \cup S_i^*) - \sum_{i=1}^{m} f_i(S_{\text{tr}}) \leq \sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i^*) \tag{8.10}$$

To show (8.10) let $e^{(t)}$ be the t^{th} element of greedy procedure in phase 1, and $S_{\text{tr}}^{(t)}$ be the t^{th} set in this procedure, where $e^{(t)} = \arg\max_e \sum_{i=1}^{m} f_i(S_{\text{tr}}^{(t-1)} \cup e) - f_i(S_{\text{tr}}^{(t-1)})$. let $J^{(0)} = S_{\text{tr}}^*$ and define $J^{(t)}$ iteratively as follows. Let $D^{(t)} = J^{(t-1)} \setminus S_{\text{tr}}^{(t-1)}$ and define $o^{(t)}$ in the following way:

1. If $e^{(t)} \in D^{(t)}$, then let $o^{(t)} = e^{(t)}$.

2. Otherwise, if $e^{(t)} \notin D^{(t)}$, let $o^{(t)}$ be one of the elements of D^t chosen uniformly at random.

Define $J^{(t)} := J^{(t-1)} \cup e^{(t)} \setminus o^{(t)}$. We show this procedure in the following chain.

$$(S_{\text{tr}}^*, \{S_i^*\}_{i=1}^m) \xrightarrow[\{o_i^{(1)}\}]{\{e_i^{(1)}\}} (J^{(1)}, \{S_i^*\}_{i=1}^m) \dots \xrightarrow[\{o_i^{(l)}\}]{\{e_i^{(l)}\}} (J^{(l)}, \{S_i^*\}_{i=1}^m)$$

$$(S_{\text{tr}} = \emptyset, \{S_i^0\}_{i=1}^m = \emptyset) \xrightarrow{\{e_i^{(1)}\}} (S_{\text{tr}}^{(1)}, \{\emptyset\}_{i=1}^m) \dots \xrightarrow{\{e_i^{(l)}\}} (S_{\text{tr}}^{(l)}, \{\emptyset\}_{i=1}^m)$$

then we can write the following inequalities:

$$\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(t)}) - f_i(S_{\text{tr}}^{(t-1)}) = \sum_{i=1}^{m} f_i(S_{\text{tr}}^{(t-1)} \cup e^{(t)}) - f_i(S_{\text{tr}}^{(t-1)}) \tag{8.11}$$

$$\geq \sum_{i=1}^{m} f_i(S_{\text{tr}}^{(t-1)} \cup o_i^{(t)}) - f_i(S_{\text{tr}}^{(t-1)}) \tag{8.12}$$

$$\geq \sum_{i=1}^{m} f_i(S_i^* \cup J^{(t-1)}) - f_i(S_i^* \cup J^{(t-1)} \setminus o^{(t)}) \tag{8.13}$$

$$\geq \sum_{i=1}^{m} f_i(S_i^* \cup J^{(t-1)}) - f_i(S_i^* \cup J^{(t-1)} \setminus o_i^{(t)})$$

$$+ \sum_{i=1}^{m} -f_i(S_i^* \cup J^{(t)}) + f_i(S_i^* \cup J^{(t-1)} \setminus o_i^{(t)}) \tag{8.14}$$

$$= \sum_{i=1}^{m} f_i(S_i^* \cup J^{(t-1)}) - f_i(S_i^* \cup J^{(t)}) \tag{8.15}$$

where (8.12) follows from definition of $e^{(t)}$ and the greedy procedure and (8.13) follows from submodularity since in each step $S_{\text{tr}}^{(t-1)} \subseteq J^{(t-1)}$ and $o^{(t)} \notin S_{\text{tr}}^{(t-1)}$ and finally, equation (8.14) follows from the fact that $-f_i(J^{(t)} \cup S_i^*) + f_i(J^{(t-1)} \cup S_i^* \setminus o^{(t)}) \leq 0$. Then, by summing over t from 0 to l we get the following inequality:

$$\sum_{i=1}^{m} f_i(S_{\text{tr}}) = \sum_{i=1}^{m} f_i(S_{\text{tr}}^{(l)}) - f_i(S_{\text{tr}}^{(0)}) = \sum_{i=1}^{m} \sum_{t=0}^{l} f_i(S_{\text{tr}}^{(t)}) - f_i(S_{\text{tr}}^{(t-1)}) \tag{8.16}$$

$$\geq \sum_{i=1}^{m} \sum_{t=0}^{l} f_i(S_i^* \cup J^{(t-1)}) - f_i(S_i^* \cup J^{(t)}) \tag{8.17}$$

$$= \sum_{i=1}^{m} f_i(S_i^* \cup J^{(0)}) - f_i(S_i^* \cup J^{(l)}) \tag{8.18}$$

$$= \sum_{i=1}^{m} f_i(S_i^* \cup S_{\text{tr}}^*) - f_i(S_i^* \cup S_{\text{tr}}) \tag{8.19}$$

where the last equality comes from the process of defining J. Because, we only change one element by adding element found in greedy process and removing one element from the optimal set in each step and the size of $J^{(t)}$ is l in each step; therefore, after l step $J^{(l)} = S_{\text{tr}}$. By rearranging the terms and summing over i the claim in (8.10) follows.

Second, for the phase 2 of the algorithm 14 we can use the usual analysis of greedy(Krause and Golovin, 2014) for set S_i:

$$\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) - f_i(S_{\text{tr}}) \geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i^{opt}) - f_i(S_{\text{tr}})) \tag{8.20}$$

$$\geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i^{*}) - f_i(S_{\text{tr}})) \tag{8.21}$$

$$\geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S_{\text{tr}}^{*} \cup S_i^{*}) - 2f_i(S_{\text{tr}})) \tag{8.22}$$

where $S_i^{opt} = \underset{|S_i| \leq k-l}{\arg\max}\, f_i(S_{\text{tr}} \cup S_i)$ in the equation (8.20). Equation (8.20) follows from usual greedy analysis, equation (8.21) follows from definition of S_{tr}^{opt}, and equation (8.22) follows from equation (8.10).

Now divide both sides of (8.10) by $1/m$ and regroup the terms to obtain

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) \geq \left(1 - \frac{1}{e}\right)(\text{OPT} - 2\beta) + \beta, \tag{8.23}$$

where $\beta := \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}})$.

Finally, since $S_i \subseteq S_i \cup S_{\text{tr}}$ by monotonicity $f_i(S_i \cup S_{\text{tr}}) \geq f_i(S_{\text{tr}})$. Then, combing this observation with the result in (8.23) implies

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) \geq \max\Big\{\beta\,,\, (1-1/e)(\text{OPT} - 2\beta) + \beta\Big\}.$$

8.8. Proof of Proposition 4

Let S_{tr}, $\{S_i\}_{i=1}^m$ be the output of Algorithm 15 and S_{tr}^*, $\{S_i^*\}_{i=1}^m$ be the optimal solution for problem (8.7). We first show the following about the output of algorithm 15, phase 1.

$$\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S^*_i) - \sum_{i=1}^{m} f_i(S_i) \leq \sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S_i) \tag{8.24}$$

to show (8.24) consider the following:

let $e_i^{(t)} = \arg\max_e f_i(S_i^{(t-1)} \cup e) - f_i(S_i^{(t-1)})$. let $J_i^{(0)} = S_i^*$ and define $J_i^{(t)}$ iteratively as follows. Let $D_i^t = J_i^{(t-1)} \setminus S_i^{(t-1)}$ and define $o_i^{(t)}$ in the following way:

1. If $e_i^{(t)} \in D_i^t$, then $o^{(t)} = e_i^{(t)}$;

2. Otherwise, if $e_i^{(t)} \notin D_i^t$, let $o_i^{(t)}$ be one of the elements of D_i^t chosen uniformly at random;

Define $J_i^{(t)} := J_i^{(t-1)} \cup e_i^{(t)} \setminus o_i^{(t)}$.

$$(S^*_{\text{tr}}, \{S_i^*\}_{i=1}^m) \xrightarrow[\{o_i^{(1)}\}]{\{e_i^{(1)}\}} (S^*_{\text{tr}}, \{J_i^{(1)}\}_{i=1}^m) \dots \xrightarrow[\{o_i^{(k-l)}\}]{\{e_i^{(k-l)}\}} (S^*_{\text{tr}}, \{J_i^{(k-l)}\}_{i=1}^m)$$

$$(S_{\text{tr}} = \emptyset, \{S_i^0\}_{i=1}^m = \emptyset) \xrightarrow{\{e_i^{(1)}\}} (\emptyset, \{S_i^{(1)}\}_{i=1}^m) \dots \xrightarrow{\{e_i^{(k-l)}\}} (\emptyset, \{S_i^{(k-l)}\}_{i=1}^m)$$

then we can write the following inequalities:

$$\begin{aligned}
f_i(S_i^{(t)}) - f_i(S_i^{(t-1)}) &= f_i(S_i^{(t-1)} \cup e_i^{(t)}) - f_i(S_i^{(t-1)}) && (8.25)\\
&\geq f_i(S_i^{(t-1)} \cup o_i^{(t)}) - f_i(S_i^{(t-1)}) && (8.26)\\
&\geq f_i(S^*_{\text{tr}} \cup J_i^{(t-1)}) - f_i(S^*_{\text{tr}} \cup J_i^{(t-1)} \setminus o_i^{(t)}) && (8.27)\\
&\geq f_i(S^*_{\text{tr}} \cup J_i^{(t-1)}) - f_i(S^*_{\text{tr}} \cup J_i^{(t-1)} \setminus o_i^{(t)})\\
&\quad - f_i(S^*_{\text{tr}} \cup J_i^{(t)}) + f_i(S^*_{\text{tr}} \cup J_i^{(t-1)} \setminus o_i^{(t)}) && (8.28)\\
&= f_i(S^*_{\text{tr}} \cup J_i^{(t-1)}) - f_i(S^*_{\text{tr}} \cup J_i^{(t)}) && (8.29)
\end{aligned}$$

where (8.26) follows from definition of $e_i^{(t)}$ and the greedy procedure and (8.27) follows from the submodularity since in each step $S_i^{(t-1)} \subseteq J_i^{(t-1)}$ and $o_i^{(t)} \notin S_i^{(t-1)}$ and finally, equation (8.28) follows from the fact that $-f_i(S^*_{\text{tr}} \cup J_i^{(t)}) + f_i(S^*_{\text{tr}} \cup J_i^{(t-1)} \setminus o_i^{(t)}) \leq 0$ because of monotonicity. Then, by

summing over t from 0 to $k-l$ we get the following inequality:

$$f_i(S_i) = f_i(S_i^{(k-l)}) - f_i(S_i^{(0)}) = \sum_{t=0}^{k-l} f_i(S_i^{(t)}) - f_i(S_i^{(t-1)}) \tag{8.30}$$

$$\geq \sum_{t=0}^{k-l} f_i(S^*_{\text{tr}} \cup J_i^{(t-1)}) - f_i(S^*_{\text{tr}} \cup J_i^{(t)}) \tag{8.31}$$

$$= f_i(S^*_{\text{tr}} \cup J_i^{(0)}) - f_i(S^*_{\text{tr}} \cup J_i^{(k-l)}) \tag{8.32}$$

$$= f_i(S^*_{\text{tr}} \cup S_i^*) - f_i(S^*_{\text{tr}} \cup S_i) \tag{8.33}$$

where the last equality comes from the process of defining $J_i^{(k-l)}$; since, the size of $J_i^{(t)}$ is $k-l$ in each step and after $k-l$ step $J_i^{(k-l)} = S_i$. Then, by rearranging and summing over i we can obtain (8.24).

Second, for phase 2 of algorithm 15 we can use the usual analysis of greedy(Krause and Golovin, 2014) for set S_{tr} :

$$\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) - f_i(S_i) \geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S^{opt}_{\text{tr}} \cup S_i) - f_i(S_i)) \tag{8.34}$$

$$\geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S_i) - f_i(S_i)) \tag{8.35}$$

$$\geq (1 - \frac{1}{e})(\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S_i^*) - 2f_i(S_i)) \tag{8.36}$$

where $S^{opt}_{\text{tr}} = \underset{|S_{\text{tr}}| \leq l}{\arg\max} \sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i)$ in equation (8.34). Equation (8.34) follows from the usual greedy analysis, equation (8.35) follows from the definition of S^{opt}_{tr}, and equation (8.36) follows the from equation (8.24).

Now divide both sides of (8.36) by $1/m$ and regroup the terms to obtain

$$\frac{1}{m} \sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) \geq \left(1 - \frac{1}{e}\right)(\text{OPT} - 2\gamma) + \gamma, \tag{8.37}$$

where $\gamma := \frac{1}{m}\sum_{i=1}^{m} f_i(S_i)$.

Finally, since $S_i \subseteq S_i \cup S_{\text{tr}}$ by monotonicity $f_i(S_i \cup S_{\text{tr}}) \geq f_i(S_i)$. Then, by combing this result wit (8.37) we obtain

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}} \cup S_i) \geq \max\Big\{\gamma\,,\,(1-1/e)(\text{OPT}-2\gamma)+\gamma\Big\}.$$

The following shows the ratio of lower bound to optimum (a similar plot can be obtained for the lower bound of Proposition 3 when γ is replaced with β.). As we observe, in the worst case, the approximation factor is 0.5.

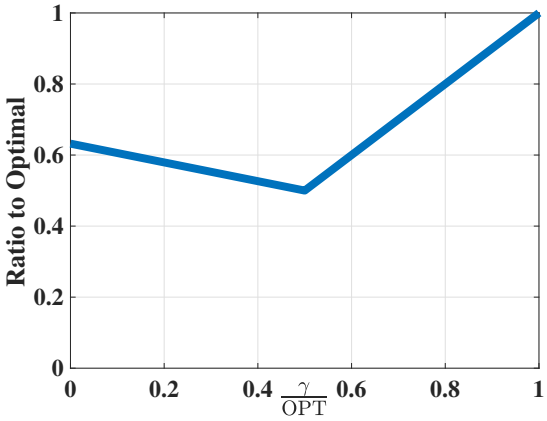

Figure 8.5: y-axis: The lower bound of Proposition 4 divided by OPT, x-axis: γ/OPT.

8.9. Proof of Theorem 30

Let $\theta_2 = \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)})$. Since $S_{\text{tr}}^{(2)}$ found greedily given $\{S_i\}_{i=1}^{m}$ we can write:

$$\theta_2 - \gamma \geq (\text{OPT}-\gamma)(1-\frac{1}{e}) \geq (\frac{1}{m}\sum_{i=1}^{m} f_i(S' \cup S_i^{(2)}) - \gamma)(1-\frac{1}{e}) \tag{8.38}$$

for every $\mid S^{'} \mid \leq l$. Also, we can write

$$\begin{aligned}
\mathrm{OPT} - \gamma &= \frac{1}{m}\sum_{i=1}^{m} f_i(S^*_{\mathrm{tr}} \cup S^*_i) - f_i(S^{(2)}_i) && (8.39)\\
&\leq \frac{1}{m}\sum_{i=1}^{m} f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i \cup S^*_i) - f_i(S^{(2)}_i) && (8.40)\\
&= \frac{1}{m}\sum_{i=1}^{m} f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i \cup S^*_i) + f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i) - f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i) - f_i(S^{(2)}_i) && (8.41)\\
&\leq \frac{1}{m}\sum_{i=1}^{m} f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i \cup S^*_i) - f_i(S^*_{\mathrm{tr}} \cup S^{(2)}_i) + \frac{\theta_2 - \gamma}{1 - 1/e} && (8.42)\\
&\leq \frac{1}{m}\sum_{i=1}^{m} f_i(S^{(2)}_i \cup S^*_i) - f_i(S^{(2)}_i) + \frac{\theta_2 - \gamma}{1 - 1/e} && (8.43)
\end{aligned}$$

where (8.42) comes from (8.38), and (8.43) comes from submodularity. We thus obtain

$$\mathrm{OPT} - \frac{\theta_2 - \gamma}{1 - 1/e} - \gamma \leq \frac{1}{m}\sum_{i=1}^{m} f_i(S^{(2)}_i \cup S^*_i) - f_i(S^{(2)}_i) \tag{8.44}$$

Also we can write for any set S' such that $\mid S^{'} \mid \leq l$:

$$\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m} f_i(S^{'} \cup S^*_i) - f_i(S^{'}) &\geq \frac{1}{m}\sum_{i=1}^{m} f_i(S^{'} \cup S^*_i \cup S_i) - f_i(S^{'} \cup S_i) && (8.45)\\
&\geq \frac{1}{m}\sum_{i=1}^{m} f_i(S^{'} \cup S^*_i \cup S_i) - f_i(S_i) + f_i(S_i) - f_i(S^{'} \cup S_i) && (8.46)\\
&\geq \frac{1}{m}\sum_{i=1}^{m} f_i(S^*_i \cup S_i) - f_i(S_i) + f_i(S_i) - f_i(S^{'} \cup S_i) && (8.47)\\
&\geq \mathrm{OPT} - \frac{\theta_2 - \gamma}{1 - 1/e} - \gamma + \frac{1}{m}\sum_{i=1}^{m} f_i(S_i) - f_i(S^{'} \cup S_i) && (8.48)\\
&\geq \mathrm{OPT} - 2\frac{\theta_2 - \gamma}{1 - 1/e} - \gamma && (8.49)
\end{aligned}$$

where (8.45) follows from submodularity, (8.47) follows from monotonicity, and (8.48) follows from (8.44), and (8.49) follows from (8.38). This results the following for any set S' such that $|S^{'}| \leq l$:

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S^{'} \cup S^*_i) - f_i(S^{'}) \geq \mathrm{OPT} - 2\frac{\theta_2 - \gamma}{1 - 1/e} - \gamma \tag{8.50}$$

Now, from (8.50) we can find a new bound for the performance of algorithm 16. From (8.50) we can write:

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^*) - f_i(S_{\text{tr}}^{(1)}) \geq \text{OPT} - 2\frac{\theta_2 - \gamma}{1 - 1/e} - \gamma \tag{8.51}$$

Also, since in Algorithm 1 the set $S_i^{(1)}$ is constructed greedily on the top of $S_{\text{tr}}^{(1)}$, we have:

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)}) - \beta \geq (\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^*) - \beta)(1 - \frac{1}{e}) \tag{8.52}$$

$$\geq (\text{OPT} - 2\frac{\theta_2 - \gamma}{1 - 1/e} - \gamma)(1 - \frac{1}{e}), \tag{8.53}$$

where (8.53) follows from (8.51). We thus obtain:

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)}) \geq (\text{OPT} - 2\frac{\theta_2 - \gamma}{1 - 1/e} - \gamma)(1 - \frac{1}{e}) + \beta \tag{8.54}$$

Using the same procedure as above, by defining $\theta_1 = \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)})$, we can prove:

$$\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)}) \geq (\text{OPT} - 2\frac{\theta_1 - \gamma}{1 - 1/e} - \gamma)(1 - \frac{1}{e}) + \beta \tag{8.55}$$

which results in the following lower bound:

$$\begin{aligned}
&\max\left\{\frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)} \cup S_i^{(1)}),\ \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(2)} \cup S_i^{(2)})\right\} \\
&\geq \max\Big\{\theta_1, \theta_2, (1-1/e)(\text{OPT}-\gamma) + \beta - 2(\theta_2 - \gamma), (1-1/e)(\text{OPT}-\beta) + \gamma - 2(\theta_1 - \beta)\Big\}. \quad (8.56)
\end{aligned}$$

Finally, given (8.54) and (8.56), the factor 0.53 is obtained as a result of the following procedure. Let β and γ given as $\beta := \frac{1}{m}\sum_{i=1}^{m} f_i(S_{\text{tr}}^{(1)})$ and $\gamma := \frac{1}{m}\sum_{i=1}^{m} f_i(S_i^{(2)})$. Then the left-hand-side term

in (8.8) is lower bounded by:

$$\min_{\theta_1,\theta_2} \max\Big\{\theta_1, \theta_2, (1-1/e)(\text{OPT}-\gamma)+\beta-2(\theta_2-\gamma), (1-1/e)(\text{OPT}-\beta)+\gamma-2(\theta_1-\beta)\Big\}$$
$$\text{subject to} \quad \theta_1 \geq \max\{\beta, (1-1/e)(\text{OPT}-2\beta)+\beta\}$$
$$\theta_2 \geq \max\{\gamma, (1-1/e)(\text{OPT}-2\gamma)+\gamma\}$$

Note that the constraints hold due to the results of Proposition 1 and 2. In particular, the above bound is always larger than $0.53 \times \text{OPT}$ for any value of β and γ.

8.10. Proof of Theorem 31

Consider round t in which $\mid S_{\text{tr}} \mid < l$ and $\mid S_i \mid < k-l$ the expected gain of the algorithm with probability $\frac{l}{k}$ is the maximum gain from adding an element $e^* = \arg\max_e \sum_{i=1}^{m} f_i(S^t_{\text{tr}} \cup e \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i)$ or with probability $\frac{k-l}{k}$ the gain is $\sum_{i=1}^{m} \max_{e_i} f_i(S^t_{\text{tr}} \cup e_i \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i)$ which can be written as follows.

$$\begin{aligned}
&\mathbb{E}[\sum_{i=1}^{m} f_i(S^{t+1}_{\text{tr}} \cup S^{t+1}_i) - f_i(S^t_{\text{tr}} \cup S^t_i) | S^t_{\text{tr}}, S^t_i] \\
&= \frac{l}{k} \max_e \sum_{i=1}^{m} f_i(S^t_{\text{tr}} \cup e \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i) + \frac{k-l}{k} \sum_{i=1}^{m} \max_{e_i} f_i(S^t_{\text{tr}} \cup e_i \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \quad (8.58)
\end{aligned}$$

assuming S^*_{tr}, S^*_i is optimal solution, we can also write:

$$\frac{1}{k}\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S^*_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \leq \frac{1}{k}\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S^*_i \cup S^t_{\text{tr}} \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \tag{8.59}$$

$$\leq \frac{1}{k}\sum_{e \in S^*_{\text{tr}} \setminus S^t_{\text{tr}}} \sum_{i=1}^{m} f_i(e \cup S^t_{\text{tr}} \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i)$$

$$+ \frac{1}{k}\sum_{i=1}^{m} \sum_{e \in S^*_i \setminus S^t_i} f_i(e \cup S^t_{\text{tr}} \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \tag{8.60}$$

$$\leq \frac{l}{k}\max_{e} \sum_{i=1}^{m} f_i(S^t_{\text{tr}} \cup e \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i)$$

$$+ \frac{k-l}{k}\sum_{i=1}^{m} \max_{e_i} f_i(S^t_{\text{tr}} \cup e_i \cup S^t_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \tag{8.61}$$

where (8.59) follows from monotonicity, and (8.60) follows from submodularity. Then, from (8.61) and (8.58) we conclude that:

$$\mathbb{E}[\sum_{i=1}^{m} f_i(S^{t+1}_{\text{tr}} \cup S^{t+1}_i) - f_i(S^t_{\text{tr}} \cup S^t_i) | S^t_{\text{tr}}, S^t_i] \leq \frac{1}{k}\sum_{i=1}^{m} f_i(S^*_{\text{tr}} \cup S^*_i) - f_i(S^t_{\text{tr}} \cup S^t_i) \tag{8.62}$$

In other words, the expected improvement in the objective (left-hand side of (8.62)) is at least $1/k$ times the gap of the current objective value to OPT (i.e. right-hand side of (8.62)). Note that (8.62) is only valid when $\mid S_{\text{tr}} \mid < l$ and $\mid S_i \mid < k - l$. Hence, by defining the stopping time τ as first time that either $\mid S_{\text{tr}} \mid = l$ or $\mid S_i \mid = k - l$, and a telescopic usages of the bounds in (8.62), we obtain the following bound:

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} f_i(S^\tau_{\text{tr}} \cup S^\tau_i)\right] \geq \text{OPT} \times \mathbb{E}\left[\left(1 - \left(1 - \frac{1}{k}\right)^\tau\right)\right]$$

The following theorem finds an upper bound on $\mathbb{E}[(1 - \frac{1}{k})^\tau]$ which finishes the proof.

Lemma 57. *If stopping time τ is first time that either $\mid S_{\text{tr}} \mid = l$ or $\mid S_i \mid = k - l$ then $\mathbb{E}[(1 - \frac{1}{k})^\tau] \leq c + \exp(-1 + \sqrt{3c.log(\frac{k}{c})})$ where $c = \frac{1}{\min\{l, k-l\}}$.*

Proof. let $u_1, u_2, \cdots$ be *i.i.d* random variables with distribution $u_i \sim \text{Bernoulli}(1 - l/k)$, i.e. $p(u_i =$

1) = $(k-l)/k$. The stopping time τ is the first time that $\sum_{i=1}^{\tau} u_i = k-l$ or $\tau - \sum_{i=1}^{\tau} u_i = l$. Let us define $X_r = \sum_{i=1}^{r} u_i$.

Furthermore, we define $\tau^{'} = r$ when r is the first time that $X_r = r-l$ and $\tau^{''} = r$ when r is the first time that $X_r = k-l$. Also, let $c = \frac{1}{\min\{l,k-l\}}$ as it was defined in the lemma. By this definition, $\tau = \min\{\tau^{''}, \tau^{'}\}$ and we can write the following about the probabilities of $\tau^{'}$ and $\tau^{''}$:

$$p(\tau^{'} = r) = \binom{r-1}{l-1}(\frac{k-l}{k})^{r-l}(\frac{l}{k})^{l}$$

$$p(\tau^{''} = r) = \binom{r-1}{k-l-1}(\frac{l}{k})^{r-k+l}(\frac{k-l}{k})^{k-l}$$

then, based on the definition of $\tau^{'}$ and $\tau^{''}$ we have the following properties for $\tau^{'}$ and $\tau^{''}$:

- if $r < k-l$ then $p(\tau^{''} = r) = 0$.
- if $r < l$ then $p(\tau^{'} = r) = 0$.
- if $r > k$ then $p(\tau^{'} \leq \tau^{''} | \tau^{'} = r) = 0$.
- if $r < k$ then $p(\tau^{'} \leq \tau^{''} | \tau^{'} = r) = 1$.
- if $r < k$ then $p(\tau^{'} \geq \tau^{''} | \tau^{''} = r) = 1$
- if $r > k$ then $p(\tau^{'} \geq \tau^{''} | \tau^{''} = r) = 0$.
- $p(\tau^{''} = r | \tau^{'} \geq \tau^{''}) = p(\tau = r | \tau^{'} \geq \tau^{''})$.
- $p(\tau^{'} = r | \tau^{'} \leq \tau^{''}) = p(\tau = r | \tau^{'} \leq \tau^{''})$.

Moreover using Bayes rule we can write:

- $$p(\tau^{'} = r | \tau^{'} \leq \tau^{''}) = \frac{p(\tau^{'} \leq \tau^{''} | \tau^{'} = r) p(\tau^{'} = r)}{p(\tau^{'} \leq \tau^{''})} = \frac{\mathbb{1}(r \leq k) p(\tau^{'} = r)}{p(\tau^{'} \leq \tau^{''})}.$$

•

$$p(\tau^{''} = r|\tau^{'} \geq \tau^{''}) = \frac{\mathbb{1}(r \leq k)p(\tau^{''} = r)}{p(\tau^{''} \leq \tau^{'})}.$$

Let $\bar{X}_r = r - X_r$ we can write $\bar{X}_r = \sum_{i=1}^{r} v_i$ where $v_1, v_2, v_3, \ldots$ are $i.i.d$ random variable with distribution $v_i \sim \text{Bernoulli}(l/k)$. Then, we can write the following using Chernoff bound:

$$\begin{aligned}
p(\tau^{'} = r) &\leq p(X_r = r - l) && (8.63)\\
&\leq p(\bar{X}_r \geq l) && (8.64)\\
&\leq p(\bar{X}_r \geq r(\frac{l}{k}) - (k - r)\frac{l}{k}) && (8.65)\\
&\leq \exp\left(-\frac{(k-r)^2(\frac{l}{k})^2}{3r(\frac{l}{k})}\right) && (8.66)\\
&= \exp\left(-\frac{(k-r)^2(l)}{3rk}\right) && (8.67)
\end{aligned}$$

Similarly:

$$\begin{aligned}
p(\tau^{''} = r) &\leq p(X_r = k - l) && (8.68)\\
&\leq p(X_r \geq k - l) && (8.69)\\
&\leq p(X_r \geq r(1 - \frac{l}{k}) - (k - r)(1 - \frac{l}{k})) && (8.70)\\
&\leq \exp\left(-\frac{(k-r)^2(1-\frac{l}{k})^2}{3r(1-\frac{l}{k})}\right) && (8.71)\\
&\leq \exp\left(-\frac{(k-r)^2(k-l)}{3rk}\right) && (8.72)\\
&\leq \exp\left(-\frac{(k-r)^2}{3rkc}\right) && (8.73)
\end{aligned}$$

then we can write the $\mathbb{E}[(1 - \frac{1}{k})^{\tau}]$ as follows:

$$\mathbb{E}[(1 - \frac{1}{k})^{\tau}] = \sum_{r=1}^{k}(1 - \frac{1}{k})^{r}p(\tau = r) \leq (1 - \frac{1}{k})^{k-\alpha\sqrt{c}} + \sum_{r=1}^{k-\alpha\sqrt{c}}(1 - \frac{1}{k})^{r}p(\tau = r) \quad (8.74)$$

Our goal is to find proper bound for (8.74). we focus on the second term in (8.75)-(8.81) and try to find proper bound for it.

$$\sum_{r=1}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r p(\tau=r) \tag{8.75}$$

$$= \sum_{r=1}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r (p(\tau^{'}=r|\tau^{'}<\tau^{''})p(\tau^{'}<\tau^{''}) + p(\tau^{''}=r|\tau^{'}\geq\tau^{''})p(\tau^{'}\geq\tau^{''})) \tag{8.76}$$

$$= \sum_{r=1}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r (p(\tau^{'}=r) + p(\tau^{''}=r)) \tag{8.77}$$

$$= \sum_{r=l}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r p(\tau^{'}=r) + \sum_{r=k-l}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r p(\tau^{''}=r) \tag{8.78}$$

$$\leq \sum_{r=l}^{k-\alpha\sqrt{c}} \exp\left(-\frac{(k-r)^2}{3rkc}\right) + \sum_{r=k-l}^{k-\alpha\sqrt{c}} \exp\left(-\frac{(k-r)^2}{3rkc}\right) \tag{8.79}$$

$$\leq (k-l)\exp\left(-\frac{(k-(k-\alpha\sqrt{c}))^2}{3k^2c}\right) + l\exp\left(-\frac{(k-(k-\alpha\sqrt{c}))^2 l}{3k^2}\right) \tag{8.80}$$

$$\leq (k-l)\exp\left(-\frac{(\alpha\sqrt{c})^2}{3k^2c}\right) + l\exp\left(-\frac{(\alpha\sqrt{c})^2 l}{3k^2}\right) \tag{8.81}$$

where (8.76) follows from law of total probability, (8.77) follows from bayes rule, (8.79) follows from Chernoff bound, (8.80) follows from the fact that $r<k$. Let $\alpha = 3\sqrt{\log(\frac{1}{c})}.k$. As result, we have:

$$\sum_{r=1}^{k-\alpha\sqrt{c}} (1-\frac{1}{k})^r p(\tau=r) \leq (k-l)c^3 + lc^{3cl} \tag{8.82}$$

Assume without loss of generality $k-l\leq l$ and $k-l\geq 2$. As a result, $c=\frac{1}{k-l}$. we want to show that $(k-l)c^3 + lc^{3cl} = c^2 + lc^{3cl} \leq c$. To show this, we show the following equivalent inequality :

$$l(k-l)^{-3cl} \leq c(1-c) = \frac{k-l-1}{(k-l)^2} \tag{8.83}$$

This holds since $k-l\geq 2$ we have $\frac{l}{(k-l)^3}(k-l)^{-3(cl-1)} \leq \frac{l}{(k-l)^3}2^{-3(\frac{l}{k-l}-1)} \leq \frac{l}{(k-l)^3\frac{l}{k-l}} = \frac{1}{(k-l)^2} \leq$

$\frac{k-l-1}{(k-l)^2}$. Moreover, we can bound the first term in (8.74) as follows:

$$(1-\frac{1}{k})^{k-\alpha\sqrt{c}} \leq \exp(-1+3\sqrt{c.\log(\frac{1}{c})}) \tag{8.84}$$

summing up we can find the following bound for $\mathbb{E}[(1-\frac{1}{k})^{\tau}]$ which finishes the proof.

$$\mathbb{E}[(1-\frac{1}{k})^{\tau}] \leq c+\exp(-1+3\sqrt{c.log(\frac{1}{c})}) \tag{8.85}$$

□

8.11. Counter-example for Submodularity of the Objective in (8.7)

In this section, we provide a counterexample for submodularity of the objective function in the equation (8.7). We consider a maximum coverage problem in which the function value is an area covered by a set of elements. We define the ground set $V = \{ABIJ, BCDI, ACDJ, IDEH$ $, HEFG, BCEH\}$ which has shown in Figure 8.6. Each element is a rectangle, and a function value of that element is an area covered by that element. We refer to each element (rectangle) by it's vertices.

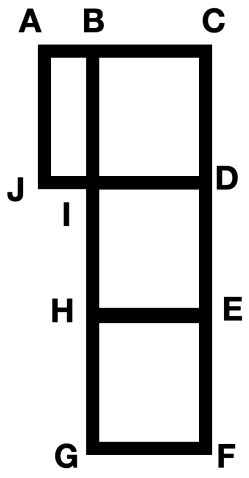

Figure 8.6: Counter Example of Submodularity

Let $AC = CD = DE = EF = 1$, and $BC = 0.75$. Also in (8.7) we let $m = 1$ and $k - l = 1$ which means that we are considering a single set function f defined as: $f(S) = \max_{e \in V} A(S \cup e)$, where $A(T)$ is a area of set T. Note that the area function A is monotone and submodular, however as we will show below, the function f is not submodular. To do so, we consider two sets $T_1 = \emptyset$ and $T_2 = \{ACDJ\}$ and add the element $IDEH$ to both sets and observe that f does not satisfy the diminishing returns property. Let us first compute the function value at T_1 and T_2 as follows:

$$f(T_1) = \max_{e \in V} A(e) = A(\{BCEH\}) = 1.5,$$

and

$$f(T_2) = \max_{e \in V} A(T_2 \cup e) = A(\{ACDJ, IDEH\}) = 1.75.$$

Similarly, we compute the function value at $T_1^{'} = T_1 \cup \{IDEH\}$, and $T_2^{'} = T_2 \cup \{IDEH\}$:

$$f(T_1^{'}) = \max_{e \in V} A(T_1^{'} \cup e) = A(\{IDEH, ACDJ\}) = 1.75,$$

and

$$f(T_2^{'}) = \max_{e \in V} A(T_2^{'} \cup e) = A(\{IDEH, ACDJ, EFGH\}) = 2.5.$$

We can now see that $T_1 \subseteq T_2$, but $f(T_2^{'}) - f(T_2) \not\leq f(T_1^{'}) - f(T_1)$. Therefore, f is not submodular.

Also let us make a remark about k-submodularity which studies functions of k subsets of the ground set that are disjoint sets. This class of functions is submodular in each orthant (Ohsaka and Yoshida, 2015). However, in the submodular meta-learning framework, sets can have overlap, and there is no restriction on the sets to be disjoint. Therefore, our framework is different from k-submodular maximization.

BIBLIOGRAPHY

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

Arman Adibi, Aryan Mokhtari, and Hamed Hassani. Submodular meta-learning. *arXiv preprint arXiv:2007.05852*, 2020.

Arman Adibi, Aritra Mitra, George J Pappas, and Hamed Hassani. Distributed statistical min-max learning in the presence of Byzantine agents. *arXiv preprint arXiv:2204.03187*, 2022a.

Arman Adibi, Aritra Mitra, George J Pappas, and Hamed Hassani. Distributed statistical min-max learning in the presence of byzantine agents. In *Proc. of the 61st IEEE Conference on Decision and Control*, pages 4179–4184, 2022b.

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. *Advances in neural information processing systems*, 24, 2011.

Mridul Agarwal, Vaneet Aggarwal, and Kamyar Azizzadenesheli. Multi-agent multi-armed bandits with limited communication. *arXiv preprint arXiv:2102.08462*, 2021.

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *arXiv preprint arXiv:1803.08917*, 2018.

Mohammad Alkousa, Darina Dvinskikh, Fedor Stonyakin, Alexander Gasnikov, and Dmitry Kovalev. Accelerated methods for composite non-bilinear saddle point problem. *arXiv preprint arXiv:1906.03620*, 2019.

Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. Structured robust submodular maximization: Offline and online algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3128–3137. PMLR, 2019.

Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. How to train your MAML. In *7th International Conference on Learning Representations, ICLR*, 2019.

Yossi Arjevani, Ohad Shamir, and Nathan Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pages 111–132. PMLR, 2020.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Mahmoud Assran, Arda Aytekin, Hamid Reza Feyzmahdavian, Mikael Johansson, and Michael G Rabbat. Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE*,

108(11):2013–2031, 2020.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1497–1514. SIAM, 2014.

Eric Balkanski, Baharan Mirzasoleiman, Andreas Krause, and Yaron Singer. Learning sparse combinatorial representations via two-stage submodular maximization. In *International Conference on Machine Learning*, pages 2207–2216, 2016.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 283–302. SIAM, 2019.

Rafael Barbosa, Alina Ene, Huy Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pages 1236–1244, 2015.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009a.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009b.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Citeseer.

Dimitri P Bertsekas and John N Tsitsiklis. Convergence rate and termination of asynchronous iterative algorithms. In *Proceedings of the 3rd International Conference on Supercomputing*, pages 461–470, 1989.

Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692. PMLR, 2018a.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692.

PMLR, 2018b.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher. A distributed algorithm for partitioned robust submodular maximization. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017a.

Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In *International Conference on Machine Learning*, pages 508–516. PMLR, 2017b.

Ilija Bogunovic, Junyao Zhao, and Volkan Cevher. Robust maximization of non-submodular objectives. In *International Conference on Artificial Intelligence and Statistics*, pages 890–899. PMLR, 2018.

Ilija Bogunovic, Andreas Krause, and Jonathan Scarlett. Corruption-tolerant gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1071–1081. PMLR, 2020.

Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Yann Bouteiller, Simon Ramstedt, Giovanni Beltrame, Christopher Pal, and Jonathan Binas. Reinforcement learning with random delays. In *International conference on learning representations*, 2020.

Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Bounded regret in stochastic multi-armed bandits. In *Conference on Learning Theory*, pages 122–134. PMLR, 2013.

Swapna Buccapatnam, Jian Tan, and Li Zhang. Information sharing in distributed stochastic bandits. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2605–2613. IEEE, 2015.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. SIAM, 2014.

Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5): 1384–1402, 2015.

Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3471–3481. PMLR, 2020a.

Ronshee Chawla, Abishek Sankararaman, and Sanjay Shakkottai. Multi-agent low-dimensional linear bandits. *arXiv preprint arXiv:2007.01442*, 2020b.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6): 1831–1879, 2014.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. *arXiv preprint arXiv:1802.08183*, 2018a.

Lin Chen, Mingrui Zhang, Hamed Hassani, and Amin Karbasi. Black box submodular maximization: Discrete and continuous settings. In *International Conference on Artificial Intelligence and Statistics*, pages 1058–1070. PMLR, 2020.

Lingjiao Chen, Zachary Charles, Dimitris Papailiopoulos, et al. Draco: Robust distributed training via redundant gradients. *arXiv preprint arXiv:1803.09877*, 2018b.

Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *ICML*, pages 903–912. PMLR, 2018c.

Mengjie Chen, Chao Gao, and Zhao Ren. Robust covariance matrix estimation via matrix depth. *arXiv preprint arXiv:1506.00691*, 2015.

Minshuo Chen, Yu Bai, H Vincent Poor, and Mengdi Wang. Efficient rl with impaired observability: Learning to act with delayed and missing state observations. *arXiv preprint arXiv:2306.01243*, 2023a.

Robert Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. *arXiv preprint arXiv:1707.01047*, 2017a.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017b.

Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. A lyapunov theory for finite-sample guarantees of asynchronous q-learning and td-learning variants. *arXiv preprint arXiv:2102.01567*, 2021.

Zaiwei Chen, Sheng Zhang, Thinh T Doan, John-Paul Clarke, and Siva Theja Maguluri. Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *Automatica*, 146:110623, 2022.

Zaiwei Chen, Siva Theja Maguluri, and Martin Zubeldia. Concentration of contractive stochastic approximation: Additive and multiplicative noise. *arXiv preprint arXiv:2303.15740*, 2023b.

Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proc. of the thirtieth annual ACM-SIAM symp. on discrete algorithms*, pages 2755–2771. SIAM, 2019.

Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran. Scalable and distributed submodular maximization with matroid constraints. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 435–442. IEEE, 2015.

Alon Cohen, Amit Daniely, Yoel Drori, Tomer Koren, and Mariano Schain. Asynchronous stochastic optimization robust to arbitrary delays. *Advances in Neural Information Processing Systems*, 34: 9024–9035, 2021.

Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Distribution-agnostic model-agnostic meta-learning. *arXiv preprint arXiv:2002.04766*, 2020.

Micah Corah and Nathan Michael. Efficient online multi-robot exploration via distributed sequential greedy assignment. In *Robotics: Science and Systems*, volume 13, 2017.

Micah Corah and Nathan Michael. Distributed submodular maximization on partition matroids for planning on large sensor networks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6792–6799. IEEE, 2018.

Micah Corah and Nathan Michael. Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice. *Autonomous Robots*, 43(2):485–501, 2019.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467. PMLR, 2017.

Arnak S. Dalalyan and Arshak Minasyan. All-in-one robust estimator of the gaussian mean. *The*

Annals of Statistics, 2022.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017a.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017b.

Jelena Diakonikolas, Constantinos Daskalakis, and Michael Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2746–2754. PMLR, 2021.

Josip Djolonga, Sebastian Tschiatschek, and Andreas Krause. Variational inference in mixed probabilistic submodular models. In *Advances in Neural Information Processing Systems*, pages 1759–1767, 2016.

Thinh T Doan. Finite-time analysis of markov gradient descent. *IEEE Transactions on Automatic Control*, 2022.

Thinh T Doan, Carolyn L Beck, and R Srikant. On the convergence rate of distributed gradient methods for finite-sum optimization under communication delays. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–27, 2017.

Bin Du, Kun Qian, Christian Claudel, and Dengfeng Sun. Jacobi-style iteration for distributed submodular maximization. *arXiv preprint arXiv:2010.14082*, 2020.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl$ˆ2$: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016.

Abhimanyu Dubey and Alex Pentland. Private and byzantine-proof cooperative decision-making. In *AAMAS*, pages 357–365, 2020a.

Abhimanyu Dubey and Alex Pentland. Differentially-private federated linear bandits. *arXiv preprint arXiv:2010.11425*, 2020b.

Abhimanyu Dubey et al. Kernel methods for cooperative multi-agent contextual bandits. In

International Conference on Machine Learning, pages 2740–2750. PMLR, 2020.

John C Duchi, Sorathan Chaturapruek, and Christopher Ré. Asynchronous stochastic convex optimization. *arXiv preprint arXiv:1508.00882*, 2015.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Sayna Ebrahimi, William Gan, Dian Chen, Giscard Biamby, Kamyar Salahi, Michael Laielli, Shizhan Zhu, and Trevor Darrell. Minimax active learning. *arXiv preprint arXiv:2012.10467*, 2020.

Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298, 2009.

Mathieu Even. Stochastic gradient descent under markovian sampling schemes. *arXiv preprint arXiv:2302.14428*, 2023.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092, 2020a.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Provably convergent policy gradient methods for model-agnostic meta-reinforcement learning. *arXiv preprint arXiv:2002.05135*, 2020b.

Alireza Fallah, Asuman Ozdaglar, and Sarath Pattathil. An optimal multistage stochastic gradient method for minimax problems. In *Proc. of the 59th IEEE Conference on Decision and Control*, pages 3573–3579, 2020c.

Farzan Farnia and David Tse. A minimax approach to supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4240–4248, 2016.

Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 570–579. IEEE, 2011.

Hamid Reza Feyzmahdavian, Arda Aytekin, and Mikael Johansson. A delayed proximal gradient method with linear convergence rate. In *2014 IEEE international workshop on machine learning for signal processing (MLSP)*, pages 1–6. IEEE, 2014.

Hamid Reza Feyzmahdavian, Arda Aytekin, and Mikael Johansson. An asynchronous mini-batch algorithm for regularized stochastic optimization. *IEEE Transactions on Automatic Control*, 61 (12):3740–3754, 2016.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, volume 23, pages 586–594, 2010.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and Shanmugavelayutham Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems*, pages 2697–2705, 2013.

Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Jean Tarbouriech, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits. *arXiv preprint arXiv:2002.03839*, 2020.

Bahman Gharesifard and Stephen L Smith. Distributed submodular maximization with limited information. *IEEE transactions on control of network systems*, 5(4):1635–1645, 2017.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramachandran. Communication efficient and Byzantine tolerant distributed learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2545–2550. IEEE, 2020a.

Avishek Ghosh, Raj Kumar Maity, and Arya Mazumdar. Distributed newton can communicate less and resist Byzantine workers. *arXiv preprint arXiv:2006.08737*, 2020b.

Avishek Ghosh, Abishek Sankararaman, and Kannan Ramchandran. Collaborative learning and personalization in multi-agent stochastic linear bandits. *arXiv preprint arXiv:2106.08902*, 2021.

Antonio Ginart, Melody Y Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. *arXiv preprint arXiv:1907.05012*, 2019.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Comm. of the ACM*, 63 (11):139–144, 2020.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014b.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas L. Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *6th International Conference on Learning Representations, ICLR*, 2018.

David Grimsman, Mohd Shabbir Ali, Joao P Hespanha, and Jason R Marden. The impact of information in greedy submodular maximization. *IEEE Transactions on Control of Network Systems*, 2018.

Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 1562–1578. PMLR, 2019.

Nirupam Gupta, Thinh T Doan, and Nitin H Vaidya. Byzantine fault-tolerance in decentralized optimization under 2f-redundancy. In *2021 American Control Conference (ACC)*, pages 3632–3637. IEEE, 2021.

Mert Gurbuzbalaban, Asuman Ozdaglar, and Pablo A Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.

Erfan Yazdandoost Hamedani and Necdet Serhat Aybat. A primal-dual algorithm for general convex-concave saddle point problems. *arXiv preprint arXiv:1803.01401*, 2, 2018.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *arXiv preprint arXiv:1708.03949*, 2017.

Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy++: When upper and lower bounds match. In *Advances in Neural Information Processing Systems*, pages 13087–13097, 2019.

Martin Hast, Karl Johan Åström, Bo Bernhardsson, and Stephen Boyd. Pid design by convex-concave optimization. In *2013 European Control Conference (ECC)*, pages 4460–4465. IEEE, 2013.

Jiafan He, Dongruo Zhou, Tong Zhang, and Quanquan Gu. Nearly optimal algorithms for linear contextual bandits with adversarial corruptions. *arXiv preprint arXiv:2205.06811*, 2022.

Yusuo Hu, Hua Chen, Jian-guang Lou, and Jiang Li. Distributed density estimation using non-parametric statistics. In *27th International Conference on Distributed Computing Systems (ICDCS'07)*, pages 28–28. IEEE, 2007.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

Adam Ibrahim, Waıss Azizian, Gauthier Gidel, and Ioannis Mitliagkas. Linear lower bounds and conditioning of differentiable games. In *International Conference on Machine Learning*, pages 4583–4593. PMLR, 2020.

Rishabh Iyer. A unified framework of constrained robust submodular optimization with applications, 2021.

Stefanie Jegelka and Jeff A Bilmes. Online submodular minimization for combinatorial structures. In *ICML*, pages 345–352. Citeseer, 2011.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, pages 4880–4889. PMLR, 2020.

Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu. Adversarial attacks on stochastic bandits. *arXiv preprint arXiv:1810.12188*, 2018.

Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

Sayash Kapoor, Kumar Kshitij Patel, and Purushottam Kar. Corruption-tolerant bandit learning. *Machine Learning*, 108(4):687–715, 2019.

Soummya Kar, H Vincent Poor, and Shuguang Cui. Bandit problems in networks: Asymptotically efficient distributed allocation rules. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1771–1778. IEEE, 2011.

Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. In *Advances in Neural Information Processing Systems*, pages 6853–6863, 2017.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319. PMLR, 2021.

Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International*

conference on machine learning, pages 2544–2553. PMLR, 2018.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

Katrin Kirchhoff and Jeff Bilmes. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, 2014.

Ravi Kumar Kolla, Krishna Jagannathan, and Aditya Gopalan. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795, 2018.

Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022a.

Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022b.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976a.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976b.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12), 2008a.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008b.

Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. *ACM Transactions on Parallel Computing (TOPC)*, 2(3):1–22, 2015.

Kananart Kuwaranancharoen, Lei Xin, and Shreyas Sundaram. Byzantine-resilient distributed

optimization of multi-dimensional functions. In *2020 American Control Conference (ACC)*, pages 4399–4404. IEEE, 2020.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Coordinated attacks against contextual bandits: Fundamental limits and defense mechanisms. *arXiv preprint arXiv:2201.12700*, 2022.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

Anusha Lalitha and Andrea Goldsmith. Bayesian algorithms for decentralized stochastic bandits. *arXiv preprint arXiv:2010.10569*, 2020.

Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

Gert RG Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3(Dec):555–582, 2002.

Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 167–172. IEEE, 2016.

Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. Distributed cooperative decision making in multi-agent multi-armed bandits. *Automatica*, 125:109445, 2021.

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

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *arXiv preprint arXiv:1812.00151*, 2018.

Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28:379–387, 2015.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

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

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, volume 33, pages 1544–1551, 2019a.

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4213–4220, 2019b.

Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *AISTATS*, pages 907–915. PMLR, 2019.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020a.

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020b.

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020c.

Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050. PMLR, 2019.

Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010.

Gabor Lugosi and Shahar Mendelson. Robust multivariate mean estimation: the optimality of trimmed mean. *The Annals of Statistics*, 49(1):393–410, 2021.

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017a.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017b.

David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic bandits. *arXiv preprint arXiv:1810.04468*, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

Stanislav Minsker. Uniform bounds for robust mean estimators. *arXiv preprint arXiv:1812.03523*, 2018.

Vahab Mirrokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 153–162, 2015.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *NIPS*, pages 2049–2057, 2013.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *The Journal of Machine Learning Research*, 17(1):8330–8373, 2016a.

Baharan Mirzasoleiman, Morteza Zadimoghaddam, and Amin Karbasi. Fast distributed submodular cover: Public-private data summarization. In *Advances in Neural Information Processing Systems*, pages 3594–3602, 2016b.

Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *International Conference on Machine Learning*, pages 2449–2458. PMLR, 2017.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of neural networks against noisy labels. *arXiv preprint arXiv:2011.07451*, 2020.

Aritra Mitra, Hamed Hassani, and George Pappas. Exploiting heterogeneity in robust federated best-arm identification. *arXiv preprint arXiv:2109.05700*, 2021a.

Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular+ concave. *Advances in Neural Information Processing Systems*, 34, 2021b.

Marko Mitrovic, Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Data summarization at scale: A two-stage submodular approach. *arXiv preprint arXiv:1806.02815*, 2018.

Slobodan Mitrović, Ilija Bogunovic, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. Streaming robust submodular maximization: A partitioned thresholding approach. *arXiv preprint arXiv:1711.02598*, 2017.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Decentralized submodular maximization: Bridging discrete and continuous settings. *arXiv preprint arXiv:1802.03825*, 2018.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of Machine Learning Research*, 21(105):1–49, 2020a.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020b.

Aryan Mokhtari, Asuman E Ozdaglar, and Sarath Pattathil. Convergence rate of o(1/k) for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020c.

Aryan Mokhtari, Asuman E Ozdaglar, and Sarath Pattathil. Convergence rate of o(1/k) for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020d.

Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. Technical report, 1985.

Dheeraj Nagaraj, Xian Wu, Guy Bresler, Prateek Jain, and Praneeth Netrapalli. Least squares regression with markovian data: Fundamental limits and algorithms. *Advances in neural information processing systems*, 33:16666–16676, 2020.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. *arXiv preprint arXiv:2007.02923*, 2020.

George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004a.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004b.

Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344, 2007.

John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 3581–3607. PMLR, 2022.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*, 2019.

Naoto Ohsaka and Yuichi Yoshida. Monotone k-submodular function maximization with size constraints. In *Advances in Neural Information Processing Systems*, pages 694–702, 2015.

James B Orlin, Andreas S Schulz, and Rajan Udwani. Robust monotone submodular function maximization. *Mathematical Programming*, 172(1):505–537, 2018.

Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 2022.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

Nikhil Ravi, Anna Scaglione, and Angelia Nedić. A case of distributed optimization in adversarial environment. In *ICASSP*, pages 5252–5256, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR*, 2017.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Proc. Systems*, 33: 21554–21565, 2020.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Omid Sadeghi and Maryam Fazel. Online continuous dr-submodular maximization with long-term budget constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 4410–4419. PMLR, 2020.

Shinsaku Sakaue. Differentiable greedy algorithm for monotone submodular maximization: Guarantees, gradient estimators, and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 28–36. PMLR, 2021.

Maziar Sanjabi, Meisam Razaviyayn, and Jason D Lee. Solving non-convex non-concave min-max games under polyak-lojasiewicz condition. *arXiv preprint arXiv:1812.02878*, 2018.

Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3 (3):1–35, 2019.

Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032, 2018.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2786–2790. IEEE, 2017.

Chengshuai Shi, Cong Shen, and Jing Yang. Federated multi-armed bandits with personalization. In *International Conference on Artificial Intelligence and Statistics*, pages 2917–2925. PMLR, 2021.

Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. ES-MAML: simple Hessian-free meta learning. In *8th International Conference on Learning Representations, ICLR*, 2020.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *arXiv preprint arXiv:2101.10382*, 2021.

Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.

Matthew Staib, Bryan Wilder, and Stefanie Jegelka. Distributionally robust submodular maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 506–516. PMLR, 2019.

Serban Stan, Morteza Zadimoghaddam, Andreas Krause, and Amin Karbasi. Probabilistic submodular maximization in sub-linear time. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3241–3250. PMLR, 06–11 Aug 2017a. URL http://proceedings.mlr.press/v70/stan17a.html.

Serban Stan, Morteza Zadimoghaddam, Andreas Krause, and Amin Karbasi. Probabilistic submodular maximization in sub-linear time. In *International Conference on Machine Learning*, pages 3241–3250. PMLR, 2017b.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates. *The Journal of Machine Learning Research*, 21(1):9613–9648, 2020.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2009.

Lili Su and Nitin H Vaidya. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In *Proc. of the 2016 ACM symposium on principles of distributed computing*, pages 425–434, 2016.

Lili Su and Nitin H Vaidya. Byzantine-resilient multiagent optimization. *IEEE Transactions on Automatic Control*, 66(5):2227–2233, 2020.

Shreyas Sundaram and Bahman Gharesifard. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3):1063–1076, 2018.

Kiran Koshy Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Efficient algorithms for smooth minimax optimization. *arXiv preprint arXiv:1907.01543*, 2019.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael J Todd. *Minimum-volume ellipsoids: Theory and algorithms*. SIAM, 2016.

Sebastian Tschiatschek, Aytunc Sahin, and Andreas Krause. Differentiable submodular maximization. *arXiv preprint arXiv:1803.01785*, 2018.

Paul Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995.

JN Tsitsiklis and B Vanroy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

UberDataset. Uber pickups in new york city. https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, pages 9712–9721. PMLR, 2020.

Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent multi-armed bandits. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 161–170, 2021.

Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent bandits over undirected graphs. *arXiv preprint arXiv:2203.00076*, 2022.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 2007.

Jan Vondrák. Submodularity in combinatorial optimization. 2007.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309*, 2019.

Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Using document summarization techniques for speech data subset selection. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 721–726, 2013.

Bryan Wilder. Equilibrium computation and robust optimization in zero sum games with submodular structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1658–1665, 2019.

Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

Yinjun Wu, Edgar Dobriban, and Susan Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pages 10355–10366. PMLR, 2020.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

Jiahao Xie, Chao Zhang, Zebang Shen, Chao Mi, and Hui Qian. Decentralized gradient tracking for continuous dr-submodular maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2897–2906, 2019.

Junchi Yang, Negar Kiyavash, and Niao He. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621*, 2020.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn.

Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2011.

Sihan Zeng, Thinh T Doan, and Justin Romberg. Finite-time convergence rates of decentralized stochastic approximation with applications in multi-agent and multi-task learning. *IEEE Transactions on Automatic Control*, 2022.

Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems*, pages 9210–9221, 2019.

Renbo Zhao. Optimal stochastic algorithms for convex-concave saddle-point problems. *arXiv preprint arXiv:1903.01687*, 2019.

Minyi Zhong and Christos G Cassandras. Distributed coverage control and data collection with mobile sensor networks. *IEEE Transactions on Automatic Control*, 56(10):2445–2455, 2011.

Zixin Zhong, Wang Chi Cheung, and Vincent Tan. Probabilistic sequential shrinking: A best arm identification algorithm for stochastic bandits with corruptions. In *International Conference on Machine Learning*, pages 12772–12781. PMLR, 2021.

Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

Lifeng Zhou, Vasileios Tzoumas, George J Pappas, and Pratap Tokekar. Distributed attack-robust submodular maximization for multi-robot planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2479–2485. IEEE, 2020a.

Tianyi Zhou and Jeff A Bilmes. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *ICLR (Poster)*, 2018.

Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33, 2020b.

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Curriculum learning by optimizing learning dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 433–441. PMLR, 2021.

Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Peter Glynn, Yinyu Ye, Li-Jia Li, and Li Fei-Fei. Distributed asynchronous optimization with unbounded delays: How slow can you go? In *International Conference on Machine Learning*, pages 5970–5979. PMLR, 2018.

Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. In

Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, pages 3–4, 2021.