Decision Making in Multi-agent Systems: from Cooperation to Competition

A

Dissertation

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Chuanhao Li

August 2023

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Chuanhao Li

This Dissertation has been read and approved by the examing committee:

Advisor: Hongning Wang

Advisor:

Committee Member: Aidong Zhang

Committee Member: Cong Shen

Committee Member: Haifeng Xu

Committee Member: Shangtong Zhang

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

August 2023

Abstract

In today's rapidly evolving technological landscape, multi-agent systems have become a fundamental paradigm for modeling complex interactions and decision making in various domains. The study of decision making within these systems has gained significant attention across fields like artificial intelligence, robotics, economics, and social sciences. This dissertation explores the intricacies of decision making in multi-agent systems, focusing on the cooperative, non-cooperative and competitive interactions among agents, and addresses the challenge of designing and analyzing multi-agent decision-making algorithms, with the aim to understand how individual agents with diverse capabilities, knowledge, and objectives can collectively achieve desirable outcomes.

For cooperative decision making that involves agents collaborating to achieve common objectives, effective coordination and information sharing are essential. This research investigates decision-making algorithms that facilitate collaboration and improve overall system performance under various challenging scenarios, such as heterogeneity, non-stationarity, and decentralized communication. In domains with limited resources, conflicting objectives, or strategic interactions, non-cooperative and strictly competitive agent behaviors become prevalent. In such settings, agents prioritize self-interest and individual objectives over collaborative efforts. This research analyzes and develops effective decision-making techniques on the system side to account for non-cooperative behaviors and guide agents towards desirable decision-making outcomes.

By understanding the dynamics of decision making in both cooperative and non-cooperative settings, this dissertation aims to enhance the overall performance and efficiency of multi-agent systems across a wide range of applications, and contribute to the advancement of decision-making algorithms, enabling better cooperation and addressing the challenges posed by non-cooperative behaviors in multi-agent systems.

To my grandparents, Guangsheng Li and Wanying Xu,
who sent me off on this remarkable journey, but couldn't witness my triumphant return.

Acknowledgement

I would like to extend my deepest gratitude to my PhD advisor, Professor Hongning Wang. His exceptional guidance, unwavering support, and invaluable mentorship have been instrumental in shaping my research and academic growth. I am truly grateful for his expertise, patience, and dedication, which have been pivotal in my success.

I am also indebted to my esteemed committee members, Professor Aidong Zhang, Professor Cong Shen, Professor Haifeng Xu, and Professor Shangtong Zhang, for their valuable insights, constructive feedback, and contributions to the refinement of this dissertation. Their expertise and commitment to academic excellence have significantly enriched my research and shaped the quality of this work.

I am immensely grateful to my collaborators both within and outside of UVA, including Professor Denis Nekipelov, Professor Mengdi Wang, Professor Huazheng Wang, Professor Qingyun Wu, and Fan Yao. Their collaboration, intellectual exchange, and shared expertise have enriched my research, expanded its scope, and led to significant advancements. I am fortunate to have had the opportunity to work alongside such talented individuals. I would also like to acknowledge the collective support and inspiration provided by my colleagues and friends. Their intellectual discussions, encouragement, and friendship have enhanced my academic experience and made it truly meaningful.

Last but most importantly, my heartfelt appreciation goes to my parents, Shiqiang Li and Qing Zhang, as well as my girlfriend, Huining Li, for their understanding, support, and love. Their belief in me and their sacrifices have been the foundation of my journey. I am grateful for the countless ways in which they have inspired and motivated me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

In today's rapidly evolving technological landscape, multi-agent systems have emerged as a fundamental paradigm for modeling complex interactions and decision making in various domains. These systems consist of multiple autonomous agents that interact with each other, aiming to achieve individual and collective goals. The study of decision making within such multi-agent systems is a crucial research area that has gained significant attention across diverse fields, including artificial intelligence, robotics, economics, and social sciences. This dissertation delves into the intricacies of decision making in multi-agent systems, exploring the spectrum of cooperative and competitive interactions that agents engage in. *By understanding the dynamics of decision making in both cooperative and non-cooperative settings, I aim to enhance the overall performance and efficiency of multi-agent systems across a wide range of applications.*

The fundamental challenge in designing and analyzing multi-agent decision-making systems lies in understanding how individual agents, each with its own capabilities, knowledge, and objectives, can collectively achieve desirable outcomes. Cooperative decision making lies at the heart of many real-world scenarios, where agents must collaborate to achieve common objectives. Examples can be found in various domains such as disaster response [1], autonomous vehicles [2], and multi-robot systems [3]. The ability to effectively coordinate actions and share information among agents is critical for successful cooperation. Consequently, this research investigates decision-making algorithms that facilitate collaboration among agents under different challenging scenarios, such as heterogeneity, non-stationarity, and decentralized communication, with the purpose of improving overall system performance. In addition to cooperation, non-cooperative and strictly competitive behaviors are another prevalent aspect of multi-agent systems, particularly in domains where limited resources, conflicting objectives, or strategic interactions exist. In contrast to agents in cooperative setting, now the agents operate in environments where self-interest and individual objectives take precedence over collaborative efforts. These agents seek to maximize their own utility or achieve personal goals without actively considering the well-being or interests of others. For instance, in economic settings like auction systems or marketplaces [4], the agents may engage in strategic bidding or pricing strategies to gain an advantage over their competitors. This research seeks to analyze and develop effective decision-making techniques on the system side to account for such behaviors and guide the agents towards desirable decision-making outcomes.

1.1 Challenges

In order to let the multi-agent system interact, learn, and adapt in complex environments, one fundamental challenge faced by these intelligent agents is the well-known *exploration-exploitation dilemma* [5], such that agents seeking to make optimal decisions need to determine: whether to invest resources and time into exploring new options that may yield greater long-term benefits or to exploit known strategies that have shown success in the past. Striking the right balance between exploration and exploitation becomes crucial to achieving efficient decision-making in various domains, ranging from recommender systems [6], mobile health [7], environment monitoring [8], automatic machine learning [9], cyber-physical systems [10, 11], etc. As illustrated in Figure 1.1, in addition to obtaining a good model estimation on dataset collected so far, the agents also need to efficiently explore the action space to acquire new data points from the environment. Otherwise, the learned model generalizes poorly on previously unseen data inputs, and thus leads to sub-optimal decisions.

In the context of multi-agent systems, the exploration-exploitation dilemma takes on a new dimension. Agents within such systems are not isolated decision-makers but instead interact with each other, forming complex networks of

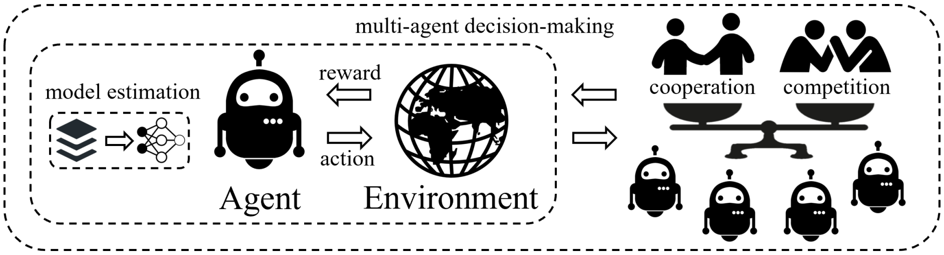


Figure 1.1: To make optimal decisions, agents need to trade-off between exploration and exploitation when interacting with the environment. Under multi-agent systems that are studied in this dissertation, challenges on new dimensions are introduced, as agents within such systems are not isolated decision-makers but instead interact with each other, forming complex networks of interactions and dependencies.

interactions and dependencies. These interactions introduce additional challenges that can affect the decision-making dynamics of both cooperative and non-cooperative agents, which will be discussed in more details in the following paragraphs.

Decision making with cooperative agents In cooperative multi-agent systems, where agents work together to achieve common goals, the exploration-exploitation dilemma is influenced by factors such as information sharing and coordination. Cooperative agents need to strike a balance between exploring new possibilities that may improve the collective performance and exploiting established strategies to maintain stability and avoid unnecessary risks. Achieving effective cooperation requires methods that enable agents to align their exploration and exploitation strategies. However, several practical challenges may impede such cooperation in multi-agent systems.

- First, for many applications, decision making systems face a non-stationary environment, i.e., the reward function of each agent changes over time, which induces changes in the task similarities between agents. For example, users of a recommender system may change their preferences dramatically over time due to various internal or external factors [12], and thus users sharing similar opinions now may develop diverse opinions in the future. Therefore, directly pooling their data together to learn a single model has negative impacts on the performance. Instead, the decision making system needs to carefully decide *when and with whom the agents should collaborate*, in order to enjoy improved performance compared with learning a separate model on each agent.
- Second, with the increasing number of decentralized applications, where data storage and computation are distributed to each agent, cooperative decision making under decentralized environment is becoming increasingly important. In this case, communication bandwidth becomes the main bottleneck, e.g., communication in a network of mobile devices can be slower than local computation by several orders of magnitude [13]. This gives rise to the conflict between the need of timely data/model aggregation for *cooperative decision making* and the need of *communication efficiency*, and thus a well-designed communication strategy becomes vital to strike the balance.

Decision making with non-cooperative agents Conversely, in non-cooperative multi-agent systems, where agents pursue self-interest, instead of a common goal in an altruistic manner, the decision making problem takes on a more competitive nature. For example, the agents may want to withhold some information due to concerns about cost, privacy and security, even if their interests align with that of the others, and thus, collectively they make sub-optimal decisions. In this case, it is essential to factor such self-interested behaviors into decision making, i.e., by carefully modeling their various non-cooperative or even competitive behaviors, and then designing strategies tailored to each situation. In this dissertation, the following non-cooperative behaviors are considered.

- First, the agents may withhold the actual feedback, and only reveals the sequence of chosen actions to the system, which gives rise to an intriguing challenge of learning agents's utility parameters from only the coarse and implicit feedback of "revealed preference" [14]. For example, it is widely observed that very few recommender system users would bother to provide detailed feedback (even not numerical ratings). This observation is also supported by the 90-9-1 Rule for online community engagement, and the "Lazy User Theory" [15] in the HCI community, which states that a user will most often choose the solution that will fulfill her information needs with the least cost/effort.

- Second, as self-interested entities, agents may refuse to participate in cooperation, unless the benefits of cooperation outweighs the potential risks, e.g., of privacy leakage. This is motivated by the practical observation that many clients in a decentralized/federated system are inherently self-interested and thus reluctant to share data without receiving explicit benefits from the server [16]. For instance, consider a recommendation platform (server) that wants its mobile app users (clients) to opt in its new recommendation service, which switches previous on-device local prediction model to a globally trained model. Although the new service is expected to improve the overall recommendation quality for all clients, particular clients may not be willing to participate, as the expected gain for them might not compensate their locally increased cost (e.g., communication bandwidth, added computation, lost control of their data, etc). In this case, additional actions have to be taken by the server to encourage participation, as it has no power to force clients.

- Third, in some situations, we may face strictly competitive agents, i.e., the actions of one agent negatively affect the utilities of the others. Take recommender systems of short videos or live streaming as an example. To maximize utility, the content creators may actively adjust their contents to attract more users. Therefore, how the system allocates the user traffic to the creators affects their content generation, and thus further affects social welfare, i.e., overall user satisfaction [4]. In this case, it is essential for the system to mediate creators' behaviors with properly designed allocation rule and steer the equilibrium outcome to optimize important societal objectives, such as social welfare for users.

1.2 Dissertation Overview

To explore the intricacies of different types of agent behaviors, and provide insights into effective strategies for agents to efficiently explore the environment and achieve optimal outcomes, this dissertation instantiates these challenges under scenarios where the environment repeatedly provides the system with a set of candidate actions to choose from, and possibly some side information (aka., context) [17, 18, 19]; and the system, whose goal is to maximize cumulative reward over time, can only observe the reward corresponding to the chosen action. This is often modeled as a *(contextual) bandit problem* [20, 21], which is a sub-class of episodic Markov decision processes (MDPs) [22] and exemplifies the well-known exploitation-exploration dilemma [5]. There is no explicit state transition. Each decision or action taken is considered in isolation, without any influence from previous actions or states. This means that the outcome of each action is independent and does not affect the future outcomes. However, we should note that solutions devised for bandit problems typically applies to more general episodic MDPs [22, 23, 24].

Bandit problems have gained popularity in a wide range of applications, such as recommender systems [17], display advertisement [18], clinical trials [19], mobile health [7], environment monitoring [8], automatic machine learning [9], cyber-physical systems [10, 11], etc. Despite the fact that many of these application scenarios involve multiple agents, most existing works formulate the problem under centralized/single-agent settings [5, 20, 25, 26]. In this dissertation, we aim to address the aforementioned challenges by developing novel decision making algorithms for both cooperative and non-cooperative multi-agent systems. Through a combination of theoretical analysis and empirical evaluations, we aim to contribute to the advancement of decision making algorithms that enable agents to navigate complex scenarios, foster cooperation, and steer equilibrium outcomes in competitive settings.

The rest of this dissertation is structured as follows. In Chapter 2, we investigate decision making with cooperative agents, with focus on two main aspects: cooperation in heterogeneous and non-stationary environments, and cooperation in decentralized environments. In Chapter 3, we investigate decision making with non-cooperative agents, which encompasses three scenarios: the system can only observe revealed preference feedback from another learning agent; the agents require incentives to participate in federated optimization; and the agents engage in competitive behavior under the context of content creation in recommender systems. These scenarios posed unique challenges that require us to explore novel decision-making algorithms. In Chapter 4, we summarize this dissertation and discuss future research directions.

Chapter 2

Decision Making with Cooperative Agents

In recent years, the field of multi-agent systems has witnessed significant advancements in the development of decision-making algorithms for cooperative agents [27, 28]. Cooperative decision making involves a group of agents working together to achieve a common goal by making coordinated choices. However, this collaborative setting introduces several challenges that must be addressed to ensure effective decision making. This chapter aims to explore two key challenges faced by cooperative agents: the presence of a heterogeneous and non-stationary environment, and the need for efficient communication in decentralized systems.

2.1 Cooperation in heterogeneous and non-stationary environments

Most existing contextual bandit algorithms impose strong assumptions on the mapping between context and reward [20, 29, 17]: typically it is assumed that the expected reward associated with a particular action is determined by a *time-invariant function* of the context vector. This overly simplified assumption restricts the application of contextual bandits in many important real-world scenarios, where a learner has to serve a population of users with possible mutual dependence and changing interest. This directly motivates recent efforts that postulate more general reward assumptions [30, 25, 31, 32]; among them, *non-stationary bandits* [12, 33, 34, 35, 36, 37] and *clustered bandits* [27, 38, 39, 40] received much attention.

In non-stationary bandits, the reward mapping function becomes time-variant. A typical non-stationary setting is the abruptly changing environment, a.k.a, a piecewise stationary environment, in which the environment undergoes abrupt changes at unknown time points but remains stationary between two consecutive change points [41, 42]. A working solution needs to either properly discount historical observations [43, 42, 36] or detect the change points and reset the model estimation accordingly [41, 34, 12]. In clustered bandits, grouping structures of bandit models are assumed, e.g., users in a group share the same bandit model. But instead of assuming an explicit dependency structure, e.g., leveraging existing social network among users [44, 30], clustered bandit algorithms aim to simultaneously cluster and estimate the bandit models during the sequential interactions with users [27, 38, 39, 40]. Its essence is thus to measure the relatedness between different bandit models. Typically, confidence bound of model parameter estimation [27] or reward estimation [39] is used for this purpose.

So far these two problems have been studied in parallel; but the key principles to solve them overlap considerably. On the one hand, mainstream solutions for piecewise stationary bandits detect change points in the underlying reward distribution by comparing the observed rewards [34] or the quality of estimated rewards [41, 12] in a window of consecutive observations. If change happens in the window, the designed statistics of interest would exceed a threshold with a high probability. This is essentially sequential hypothesis testing of a model's fitness [45]. On the other hand, existing solutions for clustered bandits evaluate if two bandit models share the same set of parameters [27, 38] or the same reward estimation on a particular arm [39]. This can also be understood as a goodness-of-fit test between models.

In this work, we take the first step to unify these two parallel strands of bandit research under the notion of *test of homogeneity*, and study non-stationarity in linear bandit with time-varying arm set, which distinguishes us from most existing work. We address both problems by testing whether the collection of observations in a bandit model follows the same distribution as that of new observations (i.e., change detection in non-stationary bandit algorithms) or of those in another bandit model (i.e., cluster identification in clustered bandit algorithms). Built upon our solution framework, bandit models can operate on individual users with much stronger flexibility, so that new bandit learning problems can be created and addressed. This enables us to study a new and challenging bandit problem in a *clustered non-stationary* environment, where the learner has to reset individual models when a change of reward distribution is detected, and merge them when they are determined as identical. This task of doing change detection while clustering is novel and important by itself [46], and has never been considered in bandit problem where the observations are non-IID in nature. Since our solution automatically detects changes and clustering structure, it has a much weaker assumption about the environment (e.g., it can be clustered, or non-stationary, or both). Furthermore, our solution enables data sharing across both users and time, when such structure exists in the environment, thus greatly reducing sample complexity in learning bandit models. Our rigorous regret analysis and extensive empirical evaluations demonstrate the value of this unified solution, especially its advantages in handling various environment assumptions.

2.1.1 Related works

Our work is closely related to the studies in non-stationary bandits and clustered bandits. In this section, we discuss the most representative solutions in each direction and highlight their connections.

Non-stationary bandits Instead of assuming a time-invariant environment, the reward mapping is allowed to change over time in this problem setting. Commonly imposed assumptions include slowly-varying environment [47, 48] and abruptly-changing environment [49, 12, 50]. We focus on the latter setting, which is also known as a piecewise stationary environment in literature [41, 42]. In a non-stationary setting, the main focus is to eliminate the distortion from out-dated observations, which follow a different reward distribution than that of the current environment. Popular solutions for the piecewise stationary environment actively detect change points and reset bandit models accordingly [41, 34, 35, 12, 51, 50, 37]. It should be noted that this dissertation studies non-stationarity in linear bandit with time-varying arm set [12, 48, 36, 52], which is different from the solutions for non-stationary MAB problem [41, 34, 35, 51, 50] or the non-stationary contextual MAB [53, 54, 37]. Therefore, their results do not apply to the setting considered in this dissertation. The closest work to our setting is [12], which maintains a pool of base linear bandit models and adaptively adds or selects from them via a change detector, which monitors how well each base bandit model predicts the new observations. This in essence boils down to a likelihood-ratio test for change in the bandit parameter. To the best of our knowledge, all the other studies for non-stationary linear bandit assume a slowly-varying environment and adopts strategies like sliding window [48], decaying weight [36] or periodical restart [52] to eliminate the distortion from out-dated observations.

Clustered bandits When serving a population of users, the vanilla linear bandit usually models the preference of each individual user in isolation, neglecting the correlation between users. In order to improve sample efficiency, such user correlation can be utilized to enable collaboration among each individual bandit models [55, 27, 38, 39, 44, 30]. Besides leveraging explicit structure among users, such as social networks [56, 44, 30, 57], recent efforts focus on online clustering of bandits via the interactions with users [27, 38, 39, 40]. For example, [27] assumed that observations from different users in the same cluster share the same underlying bandit parameter. Thus, they estimate the clustering structure among users based on the difference between their estimated bandit parameters. [38] used a similar idea to cluster items (arms) as well. [39] further studied arm-dependent clustering of users, by the projected difference between models on each arm. [40] proposed a phase-based algorithm to relax the uniform user frequency assumption in the analysis of [27]. Essentially, these algorithms measure the relatedness between users by evaluating the homogeneity of observations associated with individual models, though they have used various measures for this purpose.

In this section, we first formulate the problem setup. Then we describe two key components pertaining to non-stationary bandits and clustered bandits, and pinpoint the essential equivalence between them under the notion of homogeneity test, which becomes the cornerstone of our unified solution. Based on our construction of homogeneity test, we explain the proposed solution, followed by our theoretical analysis of the resulting upper regret bound of the proposed solution.

2.1.2 Clustered non-stationary bandit problem

To offer a unified approach that addresses the two target problems, we formulate a general bandit learning setting that encompasses both non-stationarity in individual models and existence of clustering structure.

Consider a learner that interacts with a set of n users, $\mathcal{U} = \{1, ..., n\}$. At each time $t = 1, 2, ..., T$, the learner receives an arbitrary user indexed by $i_t \in \mathcal{U}$, together with a set of available arms $C_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,K}\}$ to choose from, where $\mathbf{x}_{t,j} \in \mathbb{R}^d$ denotes the context vector associated with the arm indexed by j at time t (assume $\|\mathbf{x}_{t,j}\| \leq 1$ without loss of generality), and K denotes the size of arm pool C_t. After the learner chooses an arm $\mathbf{x}_t$, its reward $y_t \in \mathbb{R}$ is fed back from the user i_t. We follow the linear reward assumption [20, 29, 17] and use $\theta_{i_t,t}$ to denote the parameter of the reward mapping function in user i_t at time t (assume $\|\theta_{i_t,t}\| \leq 1$). Under this assumption, the reward at time t is $y_t = \mathbf{x}_t^\top \theta_{i_t,t} + \eta_t$, where η_t is Gaussian noise drawn from $N(0, \sigma^2)$. Interaction between the learner and users repeats, and the learner's goal is to maximize the accumulated reward it receives from all users in $\mathcal{U}$ up to time T.

Denote the set of time steps when user $i \in \mathcal{U}$ is served up to time T as $\mathcal{N}_i(T) = \{1 \leq t \leq T : i_t = i\}$. Among time steps $t \in \mathcal{N}_i(T)$, user i's parameter $\theta_{i,t}$ changes abruptly at arbitrary time steps $\{c_{i,1}, ..., c_{i,\Gamma_i(T)-1}\}$, but remain constant between any two consecutive change points. $\Gamma_i(T)$ denotes the total number of stationary periods in $\mathcal{N}_i(T)$. The set of unique parameters that $\theta_{i,t}$ takes for any user at any time is denoted as $\{\phi_k\}_{k=1}^m$ and their frequency of occurrences in T is $\{p_k\}_{k=1}^m$. Note that we do not impose any assumption on the distribution over the user, nor on the distribution over the unique bandit parameter appearing in each round. Also note that the ground-truth linear parameters, the set of change points, the number and frequencies of unique parameters are unknown to the learner. Moreover, the number of users, i.e., n, and the number of unique bandit parameters across users, i.e., m, are finite but arbitrary.

Our problem setting defined above is general. The non-stationary and clustering structure of an environment can be specified by different associations between $\{\theta_{i,t}\}_{i=1}^n$ and $\{\phi_k\}_{k=1}^m$ across users over time $t = 1, 2, ..., T$. For instance, by setting $n > m$ and $\Gamma_i(T) = 1, \forall i \in \mathcal{U}$, the problem reduces to the clustered bandits problem, which assumes sharing of bandit models among users with stationary reward distributions. By setting $n = 1$, $m > 1$ and $\Gamma_i(T) > 1, \forall i \in \mathcal{U}$, it reduces to the piecewise stationary bandits problem, which only concerns users with non-stationary reward distributions in isolation. To make our solution compatible with existing work in non-stationary bandits and clustered bandits, we also follow the three commonly made assumptions about the environment.

Assumption 1 (Change detectability)**.** *For any user $i \in \mathcal{U}$ and any change point c in user i, there exists $\Delta > 0$ such that at least ρ portion of arms satisfy: $|\mathbf{x}^\top \theta_{i,c-1} - \mathbf{x}^\top \theta_{i,c}| > \Delta$ [12].*

Assumption 2 (Separateness among $\{\phi_k\}_{k=1}^m$)**.** *For any two different unique parameters $\phi_i \neq \phi_j$, we have $\|\phi_i - \phi_j\| \geq \gamma > 0$ [27, 39, 40].*

Assumption 3 (Context regularity)**.** *At each time t, arm set C_t is generated i.i.d. from a sub-Gaussian random vector $X \in \mathbb{R}^d$, such that $\mathbb{E}[XX^\top]$ is full-rank with minimum eigenvalue $\lambda' > 0$; and the variance ς^2 of the random vector satisfies $\varsigma^2 \leq \frac{\lambda'^2}{8 \ln 4K}$ [27, 39, 40].*

The first assumption establishes the detectability of change points in each individual bandit models over time. The second assumption ensures separation within the global unique parameter set shared by all users, and the third assumption specifies the property of context vectors. Based on these assumptions, we establish the problem setup in this work and illustrate it on the left side of Figure 2.1.

2.1.3 Test statistic for homogeneity

As discussed in Section 2.1.1, the key problem in non-stationary bandits is to detect changes in the underlying reward distribution, and the key problem in clustered bandits is to measure the relatedness between different models. We view both problems as testing homogeneity between two sets of observations to unify these two seemingly distinct problems. For change detection, we test homogeneity between recent and past observations to evaluate whether there has been a change in the underlying bandit parameters for these two consecutive sets of observations. For cluster identification, we test homogeneity between observations of two different users to verify whether they share the same bandit parameter. On top of the test results, we operate the bandit models accordingly for model selection, model aggregation, arm selection, and model update.

We use $\mathcal{H}_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t_1}$ and $\mathcal{H}_2 = \{(\mathbf{x}_j, y_j)\}_{j=1}^{t_2}$ to denote two sets of observations, where $t_1, t_2 \geq 1$ are their cardinalities. $(\mathbf{X}_1, \mathbf{y}_1)$ and $(\mathbf{X}_2, \mathbf{y}_2)$ denote design matrices and feedback vectors of $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively, where each row of $\mathbf{X}$ is the context vector of a selected arm and the corresponding element in $\mathbf{y}$ is the observed reward for this arm. Under linear reward assumption, $\forall (\mathbf{x}_i, y_i) \in \mathcal{H}_1$, $y_i \sim N(\mathbf{x}_i^\top \theta_1, \sigma^2)$, and $\forall (\mathbf{x}_j, y_j) \in \mathcal{H}_2$, $y_j \sim N(\mathbf{x}_j^\top \theta_2, \sigma^2)$. The test of homogeneity between $\mathcal{H}_1$ and $\mathcal{H}_2$ can thus be formally defined as testing whether $\theta_1 = \theta_2$.

Because θ_1 and θ_2 are not observable, the test has to be performed on their estimates, for which maximum likelihood estimator (MLE) is a typical choice. Denote MLE for θ on a dataset $\mathcal{H}$ as $\vartheta = (\mathbf{X}^\top \mathbf{X})^- \mathbf{X}^\top \mathbf{y}$, where $(\cdot)^-$ stands

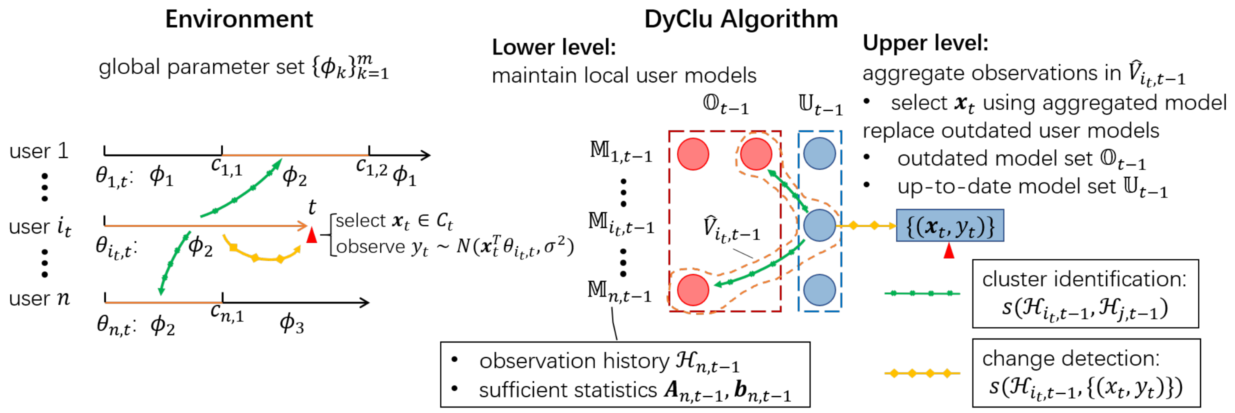

Environment
global parameter set $\{\phi_k\}_{k=1}^m$
user 1
user i_t
user n
$\theta_{1,t}$: ϕ_1
$c_{1,1}$
ϕ_2
$c_{1,2}$ ϕ_1
t
$\theta_{i_t,t}$:
ϕ_2
select $x_t \in C_t$
observe $y_t \sim N(x_t^T\theta_{i_t,t}, \sigma^2)$
$\theta_{n,t}$: ϕ_2
$c_{n,1}$
ϕ_3
DyClu Algorithm
Lower level:
maintain local user models
$\mathbb{O}_{t-1}$
$\mathbb{U}_{t-1}$
$\mathbb{M}_{1,t-1}$
$\mathbb{M}_{i_t,t-1}$
$\mathbb{M}_{n,t-1}$
$\hat{V}_{i_t,t-1}$
observation history $\mathcal{H}_{n,t-1}$
sufficient statistics $A_{n,t-1}, b_{n,t-1}$
Upper level:
aggregate observations in $\hat{V}_{i_t,t-1}$
select x_t using aggregated model
replace outdated user models
outdated model set $\mathbb{O}_{t-1}$
up-to-date model set $\mathbb{U}_{t-1}$
$\{(x_t, y_t)\}$
cluster identification: $s(\mathcal{H}_{i_t,t-1}, \mathcal{H}_{j,t-1})$
change detection: $s(\mathcal{H}_{i_t,t-1}, \{(x_t, y_t)\})$

Figure 2.1: Online bandit learning in a non-stationary and clustered environment. The environment setting is shown on the left side of the figure, where each user's reward mapping function undergoes a piecewise stationary process; and the reward mapping functions are globally shared across users. The proposed DyClu algorithm is illustrated on the right side of the figure. The model has a two-level hierarchy: at the lower level, individual users' bandit models are dynamically maintained; and at the upper level, a unified test of homogeneity is performed for the purpose of change detection and cluster identification among the lower-level user models.

for generalized matrix inverse. A straightforward approach to test homogeneity between $\mathcal{H}_1$ and $\mathcal{H}_2$ is to compare $\|\vartheta_1 - \vartheta_2\|$ against the estimation confidence on ϑ_1 and ϑ_2. The clustering methods by [27, 39] essentially followed this idea. However, theoretical guarantee on the false negative probability of this method only exists when the minimum eigenvalues of $\mathbf{X}_1^\top\mathbf{X}_1$ and $\mathbf{X}_2^\top\mathbf{X}_2$ are lower bounded by a predefined threshold. In other words, when one does not have sufficient observations in either $\mathcal{H}_1$ or $\mathcal{H}_2$, this test will not be effective.

To address this limitation, we choose the test statistic that has been proved to be *uniformly most powerful* for this type of problems [58, 59, 60]:

$$s(\mathcal{H}_1, \mathcal{H}_2) = \frac{||\mathbf{X}_1(\vartheta_1 - \vartheta_{1,2})||^2 + ||\mathbf{X}_2(\vartheta_2 - \vartheta_{1,2})||^2}{\sigma^2} \tag{2.1}$$

where $\vartheta_{1,2}$ denotes the estimator using data from both $\mathcal{H}_1$ and $\mathcal{H}_2$. The knowledge about σ^2 can be relaxed by replacing it with empirical estimate, which leads to Chow test that has an F-distribution [58].

When $s(\mathcal{H}_1, \mathcal{H}_2)$ is above a threshold υ, it suggests the pooled estimator deviates considerably from the individual estimators on two datasets. Thus, we conclude $\theta_1 \neq \theta_2$; otherwise, we conclude $\mathcal{H}_1$ and $\mathcal{H}_2$ are homogeneous. The choice of υ is critical, as it determines the type-I and type-II error probabilities of the test. Upper bounds of these two error probabilities are given below and their proofs are deferred to Section 2.1.7.

Theorem 2.1.1. *The test statistic $s(\mathcal{H}_1, \mathcal{H}_2)$ follows a non-central χ^2 distribution $s(\mathcal{H}_1, \mathcal{H}_2) \sim \chi^2(df, \psi)$, where the degree of freedom $df = rank(\mathbf{X}_1) + rank(\mathbf{X}_2) - rank(\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix})$, and the non-centrality parameter*

$$\psi = \frac{\begin{bmatrix}\mathbf{X}_1\theta_1\\ \mathbf{X}_2\theta_2\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix}(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)^- \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right]\begin{bmatrix}\mathbf{X}_1\theta_1\\ \mathbf{X}_2\theta_2\end{bmatrix}}{\sigma^2}$$

.

Lemma 2.1.2. *When $\theta_1 = \theta_2$, $\psi = 0$; the type-I error probability can be upper bounded by:*

$$P\big(s(\mathcal{H}_1, \mathcal{H}_2) > \upsilon | \theta_1 = \theta_2\big) \leq 1 - F(\upsilon; df, 0),$$

where $F(\upsilon; df, 0)$ denotes the cumulative density function of distribution $\chi^2(df, 0)$ evaluated at υ.

This lemma states that given two datasets $\mathcal{H}_1$ and $\mathcal{H}_2$ (hence the degree-of-freedom df is determined), the type-I error probability of this test only depends on the specified threshold υ.

Lemma 2.1.3. *When $\theta_1 \neq \theta_2$, $\psi \geq 0$; the type-II error probability can be upper bounded by,*

$$P\big(s(\mathcal{H}_1, \mathcal{H}_2) \leq \upsilon | \theta_1 \neq \theta_2\big) \leq \begin{cases} F\Big(\upsilon; d, \psi\Big), & \text{if } \mathbf{X}_1 \text{ and } \mathbf{X}_2 \text{ are full-rank.} \\ F(\upsilon; df, 0), & \text{otherwise.} \end{cases}$$

where $\psi = \frac{||\theta_1 - \theta_2||^2/\sigma^2}{1/\lambda_{min}(\mathbf{X}_1^\top \mathbf{X}_1) + 1/\lambda_{min}(\mathbf{X}_2^\top \mathbf{X}_2)}$.

Compared with the type-I error probability, this lemma shows that the type-II error probability also depends on the ground-truth parameters (θ_1, θ_2) and the variance of noise σ^2.

These error probabilities are the key concerns in our problem: in change detection, they correspond to the early and late detection of change points [12]; and in cluster identification, they correspond to missing a user model in the neighborhood and placing a wrong user model in the neighborhood [27]. Given it is impossible to completely eliminate these two types of errors in a non-deterministic algorithm, the uniformly most powerful property of the test defined in Eq (2.1) guarantees its sensitivity is optimal at any level of specificity.

2.1.4 DyClu algorithm

In the environment specified in Section 2.1.2, the user's reward mapping function is piecewise stationary (e.g., the line segments on each user's interaction trace in Figure 2.1), which calls for the learner to actively detect changes and re-initialize the estimator to avoid distortion from outdated observations [41, 34, 35, 12]. A limitation of these methods is that they do not attempt to reuse outdated observations because they implicitly assume each stationary period has an unique parameter. Our setting relaxes this by allowing existence of identical reward mappings across users and time (e.g., the orange line segments in Figure 2.1), which urges the learner to take advantage of this situation by identifying and aggregating observations with the same parameter to obtain a more accurate reward estimation.

Since neither the change points nor the grouping structure is known, in order to reuse past observations while avoiding distortion, the learner needs to accurately detect change points, stores observations in the interval between two consecutive detections together, and then correctly identify intervals with the same parameter as the current one. In this work, we propose to unify these two operations using the test in Section 2.1.3, which leads to our algorithm Dynamic Clustering of Bandits, or DyClu in short. DyClu forms a two-level hierarchy as shown in Figure 2.1: at the lower level, it stores observations in each interval and their sufficient statistics in a user model; at the upper level, it detects change in user's reward function to decide when to create new user models and clusters individual user models for arm selection. Detailed steps of DyClu are explained in Algorithm 1.

The lower level of DyClu manages observations associated with each user $i \in \mathcal{U}$ in user models, denoted by $\mathbb{M}_{i,t}$. Each user model $\mathbb{M}_{i,t} = (\mathbf{A}_{i,t}, \mathbf{b}_{i,t}, \mathcal{H}_{i,t})$ stores:

1. $\mathcal{H}_{i,t}$: a set of observations associated with user i since the initialization of $\mathbb{M}_{i,t}$ up to time t, where each element is a context vector and reward pair $(\mathbf{x}_k, y_k)$.
2. Sufficient statistics: $\mathbf{A}_{i,t} = \sum_{(\mathbf{x}_k, \cdot) \in \mathcal{H}_{i,t}} \mathbf{x}_k \mathbf{x}_k^\top$ and $\mathbf{b}_{i,t} = \sum_{(\mathbf{x}_k, y_k) \in \mathcal{H}_{i,t}} \mathbf{x}_k y_k$.

Every time DyClu detects change in a user's reward mapping function, a new user model is created to replace the previous one (line 15 in Algorithm 1). We refer to the replaced user models as outdated models and the others up-to-date ones. We denote the set of all outdated user models at time t as $\mathbb{O}_t$ and the up-to-date ones as $\mathbf{U}_t$. In Figure 2.1, the row of circles next to $\mathbb{M}_{1,t-1}$ represents all the user models for user 1, red ones denote outdated models and the blue one denotes up-to-date model.

The upper level of DyClu is responsible for managing the user models via change detection and model clustering. It replaces outdated models in each user and aggregates models across users and time for arm selection.

Change detection A one-sample homogeneity test is used to construct a test variable $e_{i_t,t} = \mathbf{1}\{s(\mathcal{H}_{i_t,t-1}, \{(\mathbf{x}_t, y_t)\}) > \upsilon^e\}$ to measure whether the user model $\mathbb{M}_{i_t,t-1}$ is 'admissible' to the new observation $(\mathbf{x}_t, y_t)$. υ^e is a chosen threshold for change detection. To make more reliable change detection, we use the empirical mean of $e_{i_t,t}$ in a sliding window of size $\min(|\mathcal{H}_{i_t,t-1}|, \tau)$ as the test statistic, denoted as $\hat{e}_{i_t,t} = \frac{1}{\min(|\mathcal{H}_{i_t,t-1}|,\tau)} \sum_k e_{i_t,k}$. Lemma 2.1.4 specifies the upper bound of early detection probability using $\hat{e}_{i,t}$, which is used for selecting threshold for it.

Algorithm 1 Dynamic Clustering of Bandits (DyClu)

```
1:  Input: sliding window size $\tau$, $\delta, \delta_e \in (0,1)$, threshold for change detection and neighbor identification $\upsilon^e$ and $\upsilon^c$, and regularization parameter $\lambda$
2:  Initialization: for each user model $\mathbb{M}_{i,0}, \forall i \in \mathcal{U}$: $\mathbf{A}_{i,0} = \mathbf{0} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_{i,0} = \mathbf{0} \in \mathbb{R}^d$, $\mathcal{H}_{i,0} = \emptyset$, $\hat{e}_{i,0} = 0$; the set of outdated user models $\mathbb{O}_0 = \emptyset$, and up-to-date user models $\mathbf{U}_0 = \{\mathbb{M}_{i,0}\}_{i \in \mathcal{U}}$
3:  for $t = 1, 2, ..., T$ do
4:      Observe user $i_t \in \mathcal{U}$, and set of available arms $C_t = \{x_{t,1}, ..., x_{t,K}\}$
5:      Choose arm $\mathbf{x}_t \in C_t$ by Eq 2.2:
            $$\arg\max_{\mathbf{x} \in C_t} \mathbf{x}^\top \hat{\theta}_{\hat{V}_{i_t,t-1}} + CB_{\hat{V}_{i_t,t-1}}(x)$$
6:      Observe reward $y_t$ from user $i_t$
7:      Compute $e_{i_t,t} = \mathbf{1}\left\{S(\mathcal{H}_{i_t,t-1}, (\mathbf{x}_t^\top, y_t)) > \upsilon^e\right\}$
8:      Update $\hat{e}_{i_t,t} = \sum_{\tilde{t}_{i_t}(\tau) < j \leq t : i_j = i_t} e_{i_t,j}$
9:      if $\hat{e}_{i_t,t} \leq 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}$ then
10:         if $e_{i_t,t} = 0$ then
11:             $\mathbb{M}_{i_t,t}$: $\mathcal{H}_{i_t,t} = \mathcal{H}_{i_t,t-1} \cup (\mathbf{x}_t, y_t)$, $\mathbf{A}_{i_t,t} = \mathbf{A}_{i_t,t-1} + \mathbf{x}_t\mathbf{x}_t^\top$, $\mathbf{b}_{i_t,t} = \mathbf{b}_{i_t,t-1} + \mathbf{x}_t y_t$
12:     else
13:         $\mathbb{O}_t = \mathbb{O}_{t-1} \cup \mathbb{M}_{i_t,t-1}$, $\hat{e}_{i_t,t} = 0$
14:         Replace $\mathbb{M}_{i_t,t-1}$ with $\mathbb{M}_{i_t,t} = (A_{i_t,t} = \mathbf{0}, b_{i_t,t} = \mathbf{0}, \mathcal{H}_{i,t} = \emptyset)$ in $\mathbf{U}_t$
15:     Compute $\hat{V}_{i_t,t} = \{\mathbb{M} \in \mathbf{U}_t \cup \mathbb{O}_t : S(\mathcal{H}_{i_t,t}, \mathcal{H}) \leq \upsilon^c\}$ and update $\hat{V}_{i,t}$ for $i \neq i_t$ accordingly.
```

Lemma 2.1.4. *From Lemma 2.1.2, type-1 error probability $P(e_{i,t} = 1) = 1 - F(\upsilon^e; 1, 0)$, and thus $\mathbb{E}[e_{i,t}] = 1 - F(\upsilon^e; 1, 0)$. Applying Hoeffding inequality gives,*

$$P\Big(\hat{e}_{i,t} > 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}\Big) \leq \delta_e$$

At any time step t, DyClu only updates $\mathbb{M}_{i_t,t-1}$ when $e_{i_t,t} = 0$ (line 10-12 in Algorithm 1). This guarantees that if the underlying reward distribution has changed, with a high probability we have $e_{i_t,t} = 1$, and thus the user model $\mathbb{M}_{i_t,t-1}$ will not be updated. This prevents any distortion in $\mathcal{H}_{i_t,t}$ by observations from different reward distributions.

When $\hat{e}_{i_t,t}$ exceeds the threshold specified by Lemma 2.1.4, DyClu will inform the lower level to move $\mathbb{M}_{i_t,t-1}$ to the outdated model set $\mathbb{O}_t = \mathbb{O}_{t-1} \cup \{\mathbb{M}_{i_t,t-1}\}$; and then create a new model $\mathbb{M}_{i_t,t} = (A_{i_t,t} = \mathbf{0}, b_{i_t,t} = \mathbf{0}, \mathcal{H}_{i_t,t} = \emptyset)$ for user i_t as shown in line 13-16 in Algorithm 1.

• **Clustering of user models.** In this step, DyClu finds the set of "neighborhood" user models $\hat{V}_{i_t,t}$ of current user model $\mathbb{M}_{i_t}, t$, where $\hat{V}_{i_t,t-1} = \{\mathbb{M} = (\mathbf{A}, \mathbf{b}, \mathcal{H}) \in \mathbf{U}_t \cup \mathbb{O}_t : s(\mathcal{H}_{i_t,t}, \mathcal{H}) \leq \upsilon^c\}$. Basically, DyClu executes homogeneity test between $\mathbb{M}_{i_t,t}$ and all other user models $\mathbb{M} \in \mathbf{U}_t \cup \mathbb{O}_t$ (both outdated and up-to-date) with a given threshold υ^c (line 17 in Algorithm 1). Lemma 2.1.2 and 2.1.3 again specify error probabilities of each decision.

When selecting an arm for user i_t at time t, DyClu aggregates the sufficient statistics of user models in neighborhood $\hat{V}_{i_t,t-1}$. Then it adopts the popular UCB strategy by [5, 17] to balance exploitation and exploration. Specifically, DyClu selects arm $\mathbf{x}_t$ that maximizes the UCB score computed by aggregated sufficient statistics as follows (line 5 in Algorithm 1),

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in C_t} \mathbf{x}^\top \hat{\theta}_{\hat{V}_{i_t,t-1}} + CB_{\hat{V}_{i_t,t-1}}(\mathbf{x}) \tag{2.2}$$

In Eq (2.2), $\hat{\theta}_{\hat{V}_{i_t,t-1}} = \mathbf{A}^{-1}_{\hat{V}_{i_t,t-1}} \mathbf{b}_{\hat{V}_{i_t,t-1}}$ is the ridge regression estimator using aggregated statistics $\mathbf{A}_{\hat{V}_{i_t,t-1}} = \lambda \mathbf{I}_d + \sum_{(\mathbf{A}_j, \mathbf{b}_j, \mathcal{H}_j) \in \hat{V}_{i_t,t-1}} \mathbf{A}_j$ and $\mathbf{b}_{\hat{V}_{i_t,t-1}} = \sum_{(\mathbf{A}_j, \mathbf{b}_j, \mathcal{H}_j) \in \hat{V}_{i_t,t-1}} \mathbf{b}_j$; the confidence bound of reward estimation for arm $\mathbf{x}$ is $CB_{\hat{V}_{i_t,t-1}}(\mathbf{x}) = \alpha_{\hat{V}_{i_t,t-1}} \sqrt{\mathbf{x}^\top \mathbf{A}^{-1}_{\hat{V}_{i_t,t-1}} \mathbf{x}}$, where $\alpha_{\hat{V}_{i_t,t-1}} = \sigma \sqrt{d \log{(1 + \frac{\sum_{(A_j, b_j, \mathcal{H}_j) \in \hat{V}_{i_t,t-1}} |\mathcal{H}_j|}{d\lambda})} + 2 \log \frac{1}{\delta}} + \sqrt{\lambda}$.

2.1.5 Regret analysis

Denote $R_T = \sum_{t=1}^{T} \theta_{i_t}^{\top}\mathbf{x}_t^* - \theta_{i_t}^{\top}\mathbf{x}_t$ as the accumulative regret, where $\mathbf{x}_t^* = \arg\max_{x_{t,j}\in C_t} \theta_{i_t}^{\top}\mathbf{x}_{t,j}$ is the optimal arm at time t. Our regret analysis relies on the high probability results by [20] and decomposition of "good" and "bad" events according to change detection and clustering results. The full proof, along with ancillary results and discussions, are deferred to the end of Section 2.1.

Theorem 2.1.5. *Under Assumptions 1, 2 and 3, the regret of DyClu is upper bounded by:*

$$R_T = O\Big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k}) + \sum_{i\in\mathcal{U}}\Gamma_i(T)\cdot C\Big)$$

where $C = \frac{1}{1-\delta^e} + \frac{\sigma^2}{\gamma^2\lambda'^2}\log\frac{d}{\delta'}$, *with a probability at least* $(1-\delta)(1-\frac{\delta_e}{1-\delta_e})(1-\delta')$.

Note that the first term matches the regret of the ideal case that the learner knows the exact change points and clustering structure of each user and time step, while the second term corresponds to the additional regret due to the interplay between errors in change detection and clustering, which is unique to our problem. To better understand this result, we discuss in the following paragraph how it compares with state-of-the-art bandit solutions in settings like non-stationary environment only or clustered environment only.

Case 1: Setting $m = 1$, $n = 1$ and $\Gamma_1(T) = 1$ reduces the problem to the basic linear bandit setting, because the environment consists of only one user with a stationary reward distribution for the entire time of interaction. With only one user who has a stationary reward distribution, we have $\sum_{k=1}^{1}\sqrt{p_k} = 1$ where p_k is frequency of occurrences of ϕ_k in T as defined in Section 2.1.2. In addition, since there is only one stationary period, the added regret caused by late detection does not exist; and the added regret due to the failure in clustering can be bounded by a constant, which only depends on environment variables. The upper regret bound of DyClu then becomes $O\big(\sigma d\sqrt{T\log^2 T}\big)$, which achieves the same order of regret as that in LinUCB [20]. **Case 2**: Setting $\Gamma_i(T) = 1, \forall i \in \mathcal{U}$ reduces the problem to the clustered bandit setting [27], because all users in the environment have a stationary reward distribution of their own. Similar to Case 1, the added regret caused by late detection becomes zero and the added regret due to the failure in clustering is bounded by a constant, which leads to the upper regret bound of $O\big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k}\big)$. DyClu achieves the same order of regret as that in CLUB [27]. **Case 3**: Setting $n = 1$ reduces the problem to a piecewise stationary bandit setting, because the environment consists of only one user with piecewise stationary reward distributions. For the convenience of comparison, we can rewrite the upper regret bound of DyClu in the form of $O\big(\sum_{k\in[m]} R_{Lin}(|N_k^{\phi}(T)|) + \Gamma_1(T)\big)$, where $R_{Lin}(t) = O\Big(d\sqrt{t\log^2 t}\Big)$ [20] and $N_k^{\phi}(T) = \Big\{1 \le t' \le T : \theta_{i_{t'},t'} = \phi_k\Big\}$ is the set of time steps up to time T when the user being served has the bandit parameter equal to ϕ_k. Detailed derivation of this is deferred to Section 2.1.7. Note that the upper regret bound of dLinUCB [12] for this setting is $O\big(\Gamma_1(T)R_{Lin}(S_{max}) + \Gamma_1(T)\big)$, where S_{max} denotes the maximum length of stationary periods. The regret of DyClu depends on the number of unique bandit parameters in the environment, instead of the number of stationary periods as in dLinUCB, because DyClu can reuse observations from previous stationary periods. This suggests DyClu has a tighter regret bound if different stationary periods share the same unique bandit parameters; for example, in situations where a future reward mapping function switches back to a previous one.

2.1.6 Experiment setup & results

We investigate the empirical performance of DyClu by comparing with a list of state-of-the-art baselines for both non-stationary bandits and clustered bandits on synthetic and real-world recommendation datasets.

Synthetic dataset We create a set of unique bandit parameters $\{\phi_k\}_{k=1}^{m}$ and arm pool $\{\mathbf{x}_j\}_{j=1}^{K}$ ($K = 1000$), where ϕ_k and $\mathbf{x}_j$ are first sampled from $N(\mathbf{0}_d, \mathbf{I}_d)$ with $d = 25$ and then normalized so that $\forall k, j$, $\|\phi_k\| = 1$ and $\|\mathbf{x}_j\| = 1$. When sampling $\{\phi_k\}_{k=1}^{m}$, the separation margin γ is set to 0.9 and enforced via rejection sampling. n users are simulated. In each user, we sample a series of time intervals from (S_{min}, S_{max}) uniformly; and for each time interval, we sample a unique parameter from $\{\phi_k\}_{k=1}^{m}$ as the ground-truth bandit parameter for this period. This creates asynchronous changes and clustering structure in users' reward functions. The users are served in a round-robin fashion. At time step $t = 1, 2, \ldots, T$, a subset of arms are randomly chosen and disclosed to the learner. Reward of the selected arm is

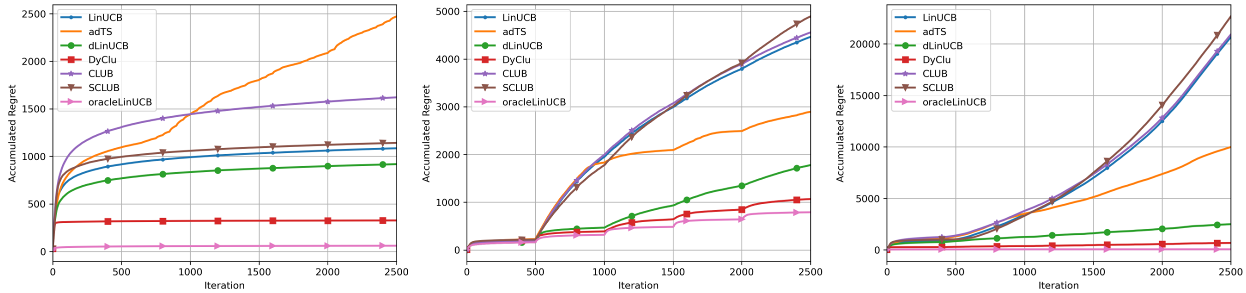


Figure 2.2: Accumulated regret on synthetic datasets with three different environment settings. Environment 1: $n = 100$ users share a global set of $m = 5$ unique bandit parameters, and each user remains stationary all the time. Environment 2: $n = 20$ user with fixed stationary period length 500; each period sample a unique bandit parameter. Environment 3: $n = 100$ users share a global set of $m = 5$ unique bandit parameters, and each user changes in a asynchronous manner.

generated by the linear function governed by the corresponding bandit parameter and context vector, with additional Gaussian noise sampled from $N(0, \sigma^2)$.

LastFM dataset The LastFM dataset is extracted from the music streaming service Last.fm [44], which contains 1892 users and 17632 items (artists). "Listened artists" of each user are treated as positive feedback. We followed [12] to preprocess the dataset and simulate a clustered non-stationary environment by creating 20 "hybrid users". We first discard users with less than 800 observations and then use PCA to reduce the dimension of TF-IDF feature vector to $d = 25$. We create hybrid users by sampling three real users uniformly and then concatenating their associated data points together. Hence, data points of the same real user would appear in different hybrid users, which is analogous to stationary periods that share the same unique bandit parameters across different users and time.

Baselines & hyper-parameters We compare DyClu with a set of state-of-the-art bandit algorithms: linear bandit LinUCB by [20], non-stationary bandit dLinUCB by [12] and adTS by [33], as well as clustered bandit CLUB by [27] and SCLUB by [40]. For experiments on synthetic dataset, we also include oracle-LinUCB for comparison, which runs an instance of LinUCB for each unique bandit parameter. Comparing with it helps us understand the added regret due to errors in change detection and clustering. We set the same regularization parameter $\lambda = 0.1$ for all the algorithms, and set the same sliding window size $\tau = 20$ for both dLinUCB and DyClu on synthetic dataset and $\tau = 50$ on LastFM dataset. The thresholds υ^e and υ^c for DyClu are essentially the upper-tail critical values of chi-square distributions $\chi^2(1)$ and $\chi^2(d)$, which directly control the type-I error probability for change detection and clustering, i.e. $1 - F(\upsilon^e; 1)$ and $1 - F(\upsilon^c; d)$ respectively. Their values affect the second term in the regret upper bound given in Theorem 3.5 (see Lemma 2.1.6 and Lemma 2.1.7 for details). In all our experiments, υ^e and υ^c are selected such that the corresponding significance level equals to 0.01, e.g., to make $F(\upsilon^c; 25) = 0.01$, we set $\upsilon^c = 44.314$.

Empirical comparisons on synthetic dataset We compare accumulated regret of all bandit algorithms under three environment settings, and the results are reported in Figure 2.2. Environment 1 simulates the clustered bandit setting in [27], where *no* change in the reward function is introduced. DyClu outperformed other baselines, including CLUB and SCLUB, demonstrating the quality of its identified clustering structure. Specifically, compared with adTS that incurs high regret as a result of too many false detections, the change detection in DyClu has much less false positives, as there is no change in each user's reward distribution. Environment 2 simulates the piecewise stationary bandit setting in [12]. Algorithms designed for stationary environment, e.g., CLUB, SCLUB, and LinUCB suffer from a linear regret after the first change point. DyClu achieved the best performance, with a wide margin from the second best, dLinUCB, which is designed for this environment. It shows the power of our change detection method against dLinUCB's. Environment 3 combines previous two settings with both non-stationarity and clustering structure. DyClu again outperformed others. It is worth noting that regret of all algorithms increased compared with Environment 1 due to the nonstationarity, but the increase in DyClu is the smallest. And in all settings, DyClu's performance is closest to the oracle-LinUCB's, which shows that DyClu can correctly cluster and aggregate observations from the dynamically changing users.

Table 2.1: Comparison of accumulated regret under different environment settings.

	n	m	S_{min}	S_{max}	T	σ	oracle.	LinU.	adTS	dLinU.	CLUB	SCLUB	DyClu
1	100	10	400	2500	2500	0.09	115	19954	9872	2432	20274	19989	853
2	100	50	400	2500	2500	0.09	489	20952	9563	2420	21205	21573	1363
3	100	100	400	2500	2500	0.09	873	21950	10961	2549	22280	22262	1958
4	100	10	200	400	2500	0.09	112	39249	36301	10831	39436	43836	3025
5	100	10	800	1000	2500	0.09	113	34385	13788	3265	34441	33514	1139
6	100	10	1200	1400	2500	0.09	112	24769	8124	2144	24980	23437	778
7	100	10	400	2500	2500	0.12	166	22453	10567	3301	22756	22683	1140
8	100	10	400	2500	2500	0.15	232	19082	10000	5872	19427	20664	1487
9	100	10	400	2500	2500	0.18	307	23918	11255	9848	24050	23677	1956

Sensitivity to environment settings According to our regret analysis, the performance of DyClu depends on environment parameters like the number of unique bandit parameters m, the number of stationary periods $\Gamma_i(T)$ for $i \in \mathcal{U}$, and variance of Gaussian noise σ^2. We investigate their influence on DyClu and baselines, by varying these parameters while keeping the others fixed. The accumulated regret under different settings are reported in Table 2.1. DyClu outperformed other baselines in all 9 different settings, and the changes of its regret align with our theoretical analysis. A larger number of unique parameters m leads to higher regret of DyClu as shown in setting 1, 2 and 3, since observations are split into more clusters with smaller size each. In addition, larger number of stationary periods incurs more errors in change detection, leading to an increased regret. This is confirmed by results in setting 4, 5 and 6. Lastly, as shown in setting 7, 8 and 9, larger Gaussian noise leads to higher regret, as it slows down convergence of reward estimation and change detection.

Empirical comparisons on LastFM We report normalized accumulative reward (ratio between baselines and uniformly random arm selection strategy [12]) on LastFM in Figure 2.3. In this environment, realizing both non-stationarity and clustering structure is important for an online learning algorithm to perform well. DyClu's improvement over other baselines confirms its quality in partitioning and aggregating relevant data points across users. The advantage of DyClu is more apparent at the early stage of learning, where each local user model has not collected sufficient amount of observations for individualized reward estimation; and thus change detection and clustering are more difficult there.

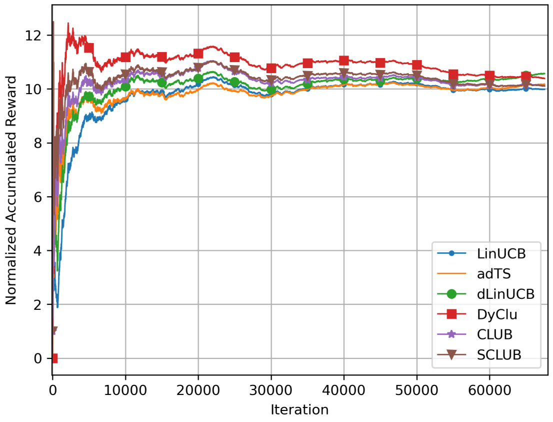


Figure 2.3: Comparison of accumulated reward normalized by a random policy on LastFM dataset.

2.1.7 Full proof of DyClu algorithm

Omitted proof in Section 2.1.3

The statistical test introduced in Section 2.1.3 falls under the category of χ^2 test of homogeneity. Specifically, it is used to test whether the parameters of linear regression models associated with two datasets are the same, assuming equal variance. The test statistic follows the noncentral χ^2-distribution $s(\mathcal{H}_1, \mathcal{H}_2) \sim \chi^2(df, \psi)$, where $df = rank(\mathbf{X}_1) + rank(\mathbf{X}_2) - rank(\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix})$ denotes the degree of freedom, and non-centrality parameter $\psi = \frac{1}{\sigma^2}\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix}^\top \begin{bmatrix}\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2\right)^- \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\end{bmatrix} \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix}$. Its proof is beyond the scope of this work. We refer the interested readers to statistics or econometrics literature like [58, 59].

Proof of Lemma 2.1.2. When datasets $\mathcal{H}_1$ and $\mathcal{H}_2$ are homogeneous, which means $\theta_1 = \theta_2$, the non-centrality parameter becomes:

$$\begin{aligned}
\psi &= \frac{1}{\sigma^2}\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix}^\top \begin{bmatrix}\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2\right)^- \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\end{bmatrix} \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix} \\
&= \frac{1}{\sigma^2}\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix}^\top \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix} - \frac{1}{\sigma^2}\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix}^\top \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2\right)^- \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix} \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix} \\
&= \frac{1}{\sigma^2}\left[\theta_1^\top(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)\theta_1 - \theta_1^\top(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)^-(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)\theta_1\right] \\
&= \frac{1}{\sigma^2}\left[\theta_1^\top(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)\theta_1 - \theta_1^\top(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2)\theta_1\right] = 0
\end{aligned}$$

Therefore, when $\theta_1 = \theta_2$, the test statistic $s(\mathcal{H}_1, \mathcal{H}_2) \sim \chi^2(df, 0)$. The type-I error probability can be upper bounded by $P(s(\mathcal{H}_1, \mathcal{H}_2) > \upsilon | \theta_1 = \theta_2) \leq 1 - F(\upsilon; df, 0)$, which concludes the proof of Lemma 2.1.2. □

Proof of Lemma 2.1.3. Similarly, using the cumulative density function of $\chi^2(df, \psi)$, we can show that the type-II error probability $P\big(s(\mathcal{H}_1, \mathcal{H}_2)\big) \leq \upsilon | \theta_1 \neq \theta_2) \leq F(\upsilon; df, \psi)$. As mentioned in Section 2.1.3, the value of ψ depends on the unknown parameters θ_1 and θ_2. From the definition of ψ, we know that $\theta_1 = \theta_2$ is only a sufficient condition for $\psi = 0$. The necessary and sufficient condition for $\psi = 0$ is that $\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix}$ is in the column space of $\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}$, e.g., there exists θ such that $\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix} = \begin{bmatrix}\mathbf{X}_1\theta \\ \mathbf{X}_2\theta\end{bmatrix}$. Only when both $\mathbf{X}_1$ and $\mathbf{X}_2$ have a full column rank, $\theta_1 = \theta_2$ becomes the necessary and sufficient condition for $\psi = 0$. This means when either $\mathbf{X}_1$ or $\mathbf{X}_2$ is rank-deficient, there always exists θ_1 and θ_2, and $\theta_1 \neq \theta_2$, that make $\psi = 0$. For example, assuming $\mathbf{X}_1$ is rank-sufficient and $\mathbf{X}_2$ is rank-deficient, then $\psi = 0$ as long as $\theta_1 - \theta_2$ is in the null space of $\mathbf{X}_2$.

To obtain a non-trivial upper bound of the type-II error probability, or equivalently a non-zero lower bound of the non-centrality parameter ψ, both $\mathbf{X}_1$ and $\mathbf{X}_2$ need to be rank-sufficient. Under this assumption, we can rewrite ψ in the following way to derive its lower bound.

Denote $\epsilon = \theta_2 - \theta_1$. Then $\theta_2 = \theta_1 + \epsilon$. We can decompose $\sigma^2\psi$ as:

$$\begin{aligned}
\sigma^2\psi &= \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2(\theta_1 + \epsilon)\end{bmatrix}^\top \begin{bmatrix}\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\mathbf{X}_1^\top\mathbf{X}_1 + \mathbf{X}_2^\top\mathbf{X}_2\right)^{-1} \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\end{bmatrix} \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2(\theta_1 + \epsilon)\end{bmatrix} \\
&= \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}\right)^{-1} \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right] \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix} \\
&\quad + \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}\right)^{-} \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right] \begin{bmatrix}0 \\ \mathbf{X}_2\epsilon\end{bmatrix} \\
&\quad + \begin{bmatrix}0 \\ \mathbf{X}_2\epsilon\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}\right)^{-1} \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right] \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_1\end{bmatrix} \\
&\quad + \begin{bmatrix}0 \\ \mathbf{X}_2\epsilon\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix} \left(\begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}\right)^{-1} \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right] \begin{bmatrix}0 \\ \mathbf{X}_2\epsilon\end{bmatrix}
\end{aligned}$$

Since $\begin{bmatrix}\mathbf{X}_1\theta_1\\ \mathbf{X}_2\theta_1\end{bmatrix}$ is in the column space of $\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix}$, the first term in the above result is zero. The second and third terms can be shown equal to zero as well using the property that matrix product is distributive w.r.t. matrix addition, which leaves us only the last term. Therefore, by substituting $\epsilon = \theta_2 - \theta_1$ back, we obtain:

$$\begin{aligned}\psi &= \frac{1}{\sigma^2}(\theta_1-\theta_2)^\top \mathbf{X}_2^\top \mathbf{X}_2(\mathbf{X}_1^\top \mathbf{X}_1 + \mathbf{X}_2^\top \mathbf{X}_2)^{-1}\mathbf{X}_1^\top \mathbf{X}_1(\theta_1-\theta_2)\\ &\geq \frac{1}{\sigma^2}\|\theta_1-\theta_2\|^2 \lambda_{\min}\Big(\mathbf{X}_2^\top \mathbf{X}_2(\mathbf{X}_1^\top \mathbf{X}_1 + \mathbf{X}_2^\top \mathbf{X}_2)^{-1}\mathbf{X}_1^\top \mathbf{X}_1\Big)\\ &\geq \frac{\|\theta_1-\theta_2\|^2/\sigma^2}{\frac{1}{\lambda_{\min}(\mathbf{X}_1^\top \mathbf{X}_1)}+\frac{1}{\lambda_{\min}(\mathbf{X}_2^\top \mathbf{X}_2)}}\end{aligned}$$

The first inequality uses the Rayleigh-Ritz theorem, and the second inequality uses the relation $\mathbf{Y}(\mathbf{X}+\mathbf{Y})^{-1}\mathbf{X} = (\mathbf{X}^{-1}+\mathbf{Y}^{-1})^{-1}$, where $\mathbf{X}$ and $\mathbf{Y}$ are both invertible matrices. This relation can be derived by taking inverse on both sides of the equation $\mathbf{X}^{-1}(\mathbf{X}+\mathbf{Y})\mathbf{Y}^{-1} = \mathbf{X}^{-1}\mathbf{X}\mathbf{Y}^{-1} + \mathbf{X}^{-1}\mathbf{Y}\mathbf{Y}^{-1} = \mathbf{Y}^{-1}+\mathbf{X}^{-1}$. □

Discussions The results above show that given two datasets $\mathcal{H}_1$ and $\mathcal{H}_2$, the type-I error probability of the homogeneity test only depends on the selection of threshold υ, while the type-II error probability also depends on the ground-truth parameters (θ_1, θ_2) and the variance of noise σ^2. If either $\mathbf{X}_1$ or $\mathbf{X}_2$ is rank-deficient, the type-II error probability will be trivially upper bounded by $F(\upsilon; df, 0)$, which means for a smaller upper bound of type-I error probability (i.e., $1 - F(\upsilon; df, 0)$), the upper bound of type-II error probability (i.e., $F(\upsilon; df, 0)$) will be large. Intuitively, for a certain level of type-I error, to ensure a smaller type-II error probability in the worst case, we at least need both $\mathbf{X}_1$ and $\mathbf{X}_2$ to be rank-sufficient and the value of $\frac{\|\theta_1-\theta_2\|^2/\sigma^2}{\frac{1}{\lambda_{\min}(\mathbf{X}_1^\top \mathbf{X}_1)}+\frac{1}{\lambda_{\min}(\mathbf{X}_2^\top \mathbf{X}_2)}}$ to be large. Similar idea is also found in [12, 27, 39], where they require the confidence bounds of the estimators (which is essentially equivalent to the condition on minimum eigenvalue $\lambda_{\min}(\mathbf{X}_1^\top \mathbf{X}_1)$ and $\lambda_{\min}(\mathbf{X}_2^\top \mathbf{X}_2)$ in our analysis) to be small enough, w.r.t. $\|\theta_1-\theta_2\|^2$, to ensure their change detection or cluster identification is accurate. Here we unify the analysis of these two tasks with this homogeneity test.

Omitted proof in Section 2.1.4

Proof of Lemma 2.1.4. Note that early detection corresponds to type-I error of the homogeneity test in Lemma 2.1.2, e.g., when change has not happened (thus $\mathcal{H}_{i_t,t-1}$ and $(\mathbf{x}_t, y_t)$ are homogeneous), but the test statistic exceeds the threshold υ^e: $e_{i_t,t} = \mathbf{1}\{s(\mathcal{H}_{i_t,t-1}, \{(\mathbf{x}_t, y_t)\}) > \upsilon^e\} = 1$. Therefore, we have $\mathbb{E}[e_{i_t,t}] \leq 1 - F(\upsilon^e; 1, 0)$. Then we can use Hoeffding inequality given in Lemma A.12 to upper bound the early detection probability using $\hat{e}_{i_t,t}$.

As the test variable $e_{i_t,t} \in \{0, 1\}$, it is $\frac{1}{2}$-sub-Gaussian. By applying Hoeffding inequality, we have:

$$P\big(\tau\hat{e}_{i_t,t} - \tau\mathbb{E}[e_{i_t,t}] \geq h\big) \leq \exp\Big(-\frac{2h^2}{\tau}\Big)$$

Then setting $\delta_e = \exp(-\frac{2h^2}{\tau})$ gives $h = \sqrt{\frac{\tau \log 1/\delta_e}{2}}$. Substituting this back and re-arrange the inequality gives us:

$$P\Big(\hat{e}_{i_t,t} < \mathbb{E}[e_{i_t,t}] + \sqrt{\frac{\log 1/\delta_e}{2\tau}}\Big) > 1 - \delta_e$$

Since $\mathbb{E}[e_{i_t,t}] \leq 1 - F(\upsilon^e; 1, 0)$, we have:

$$P\Big(\hat{e}_{i_t,t} < 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}\Big) > 1 - \delta_e$$

when change has not happened. □

Omitted proof in Section 2.1.5

In this section, we give the full proof of the upper regret bound in Theorem 2.1.5. We first define some additional notations necessary for the analysis and arrange the proof into three parts: 1) proof of Eq (2.3); 2) proof of Lemma 2.1.6; and 3) proof of Lemma 2.1.7. Specifically, Eq (2.3) provides an intermediate upper regret bound with three terms, and Lemma 2.1.6 and Lemma 2.1.7 further bound the second and third terms to obtain the final result in Theorem 2.1.5.

Consider a learner that has access to the ground-truth change points and clustering structure, or equivalently, the learner knows the index of the unique bandit parameter each observation is associated with (but it does not know the value of the parameter). For example, when serving user i_t at some time step t, the index of user i_t's unique bandit parameter for the moment is $k_{i_t,t}$, such that $\theta_{i_t,t} = \phi_{k_{i_t,t}}$. Then since this learner knows $k_{i_{t'},t'}$ for $t' \in [t]$, it can precisely group the observations associated with each unique bandit parameter ϕ_k for $k \in [m]$. Recall that we denote $N_k^\phi(t) = \left\{1 \leq t' \leq t : \theta_{i_{t'},t'} = \phi_k\right\}$ as the set of time steps up to time t when the user being served has the bandit parameter equal to ϕ_k, e.g., all the observations obtained at time steps in $N_k^\phi(t)$ have the same unique bandit parameter ϕ_k. Then an ideal reference algorithm would be the one that aggregates these observations to compute UCB score for each unique bandit parameter, and then select arm using the UCB score associated with the true bandit parameter in each time step. The regret of this ideal reference algorithm can be upper bounded by $\sum_{k=1}^m R_{Lin}(|N_k^\phi(T)|)$ where $R_{Lin}(|N_k^\phi(T)|) = O\Big(d\sqrt{|N_k^\phi(T)| \log^2 |N_k^\phi(T)|}\Big)$ [20].

However in our learning environment, such knowledge is not available to the learner; as a result, the learner does not know $N^\phi_{k_{i_t,t}}(t-1)$ when interacting with user i_t at time t; instead, it uses observations in the estimated neighborhood $\hat{V}_{i_t,t-1}$ as shown in Algorithm 1 (line 17). Denote the set of time steps associated with observations in $\hat{V}_{i_t,t-1}$ as $\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1)$, where $\tilde{k}_{i_t,t-1}$ is the index of the unique parameter associated with observations in $\mathcal{H}_{i_t,t-1}$. $\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1)$ is essentially an estimate of $N^\phi_{k_{i_t,t}}(t-1)$, obtained by running cluster identification w.r.t. $\mathcal{H}_{i_t,t-1}$ (whose true unique bandit parameter index is denoted by $\tilde{k}_{i_t,t-1}$). We define a 'good event' as $\Big\{\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1) = N^\phi_{k_{i_t,t}}(t-1)\Big\}$, which matches with the reference algorithm, and since there is a non-zero probability of errors in both change detection and cluster identification, we also have 'bad event' $\Big\{\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^\phi_{k_{i_t,t}}(t-1)\Big\}$, which incurs extra regret.

Recall the estimated neighborhood $\hat{V}_{i_t,t-1} = \{\mathbb{M} \in \mathbf{U}_{t-1} \cup \mathbb{O}_{t-1} : S(\mathcal{H}_{i_t,t-1}, \mathcal{H}) \leq \upsilon^c\}$. If $k_{i_t,t} \neq \tilde{k}_{i_t,t-1}$, it means there is a mismatch between user model $\mathbb{M}_{i_t,t-1}$ and the current ground-truth user parameter $\theta_{i_t,t}$, but the change detection module has failed to detect this. Then the obtained neighborhood is incorrect even if the cluster identification model made no mistake. Therefore, the bad event can be further decomposed into

$$\Big(\Big\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\Big\} \cap \Big\{\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^\phi_{k_{i_t,t}}(t-1)\Big\}\Big) \cup \Big(\Big\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\Big\} \cap \Big\{\hat{N}^\phi_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^\phi_{k_{i_t,t}}(t-1)\Big\}\Big).$$

The first part is a subset of event $\Big\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\Big\}$, which suggests late detection happens at time t. The second part indicates incorrectly estimated cluster for user i_t at time t.

Discussions Before moving on, we would like to provide some explanations about the use of $\Big\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\Big\}$ to denote the event that the user's underlying bandit parameter has changed, but the learner failed to detect it, i.e., late detection. Recall that $\tilde{k}_{i_t,t-1}$ is the index of the unique bandit parameter associated with observations in $\mathbb{M}_{i_t,t-1}$, i.e. $\mathcal{H}_{i_t,t-1}$, while $k_{i_t,t}$ is the index of the unique parameter that governs observation $(\mathbf{x}_t, y_t)$ from user i_t at time t. Our change detection mechanism in Algorithm 1 (line 9) is expected to replace model $\mathbb{M}_{i_t,t-1}$ if change has happened at time t, thus ensuring $\Big\{\tilde{k}_{i_t,t} = k_{i_t,t}\Big\}$ again. However, when it fails to detect the change, it will cause $\Big\{\tilde{k}_{i_t,t} \neq k_{i_t,t}\Big\}$, which means DyClu has failed to update the user model $\mathbb{M}_{i_t,t}$ to reflect the new behavior or preference that user i_t has switched to at time t.

With detailed derivation deferred to the end of this section, following the decomposition discussed above, we can obtain:

$$R_T \leq O\Big(\sigma d \sum_{k\in[m]} \sqrt{|N_k^{\phi}(T)| \log^2(|N_k^{\phi}(T)|)}\Big) + 2\sum_{i\in\mathcal{U}} \sum_{t\in\mathcal{N}_i(T)} \mathbf{1}\left\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\right\} \tag{2.3}$$
$$+ 2\sum_{t=1}^{T} \mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}$$

with a probability at least $1-\delta$.

In this upper regret bound, the first term matches the regret of the reference algorithm that has access to the exact change points and clustering structure of each user and time step. We can rewrite it using the frequency of unique model parameter ϕ_k as $O\big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k})\big)$ similar to Section A.4 in [27]. The second term is the added regret caused by the late detection of change points; and the third term is the added regret caused by the incorrect cluster identification for arm selection. The latter two terms can be further bounded by the following lemmas.

Lemma 2.1.6. *Under Assumption 1 and 3, by setting the sliding window size* $\tau \geq \frac{2\log 1/\delta_e}{\{[1-F(\upsilon^e;1,\psi^e)]\rho(1-\delta')-1+F(\upsilon^e;1,0)\}^2}$, *where* $\psi^e = \frac{\Delta^2/\sigma^2}{1+1/\left[\frac{\lambda'}{4}S_{min}-8\left(\log\frac{dS_{min}}{\delta'}+\sqrt{S_{min}\log\frac{dS_{min}}{\delta'}}\right)\right]}$, *the second term in Eq (2.3) can be upper bounded by:*

$$2\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)} \mathbf{1}\left\{\tilde{k}_{i_t,t-1} \neq k_{i_t}\right\} \leq 2\sum_{i\in\mathcal{U}}\Big(\Gamma_i(T)-1\Big)(\tau+\frac{2}{1-\delta_e})$$

with a probability at least $1-\frac{\delta_e}{1-\delta_e}$.

Lemma 2.1.7. *Define function* $g(\psi;d,\upsilon) = F(\upsilon;\psi,d)$*, and* $g^{-1}(\cdot|d,\upsilon)$ *as its inverse function. Under Assumption 2 and 3, the third term in Eq (2.3) can be upper bounded by:*

$$2\sum_{t=1}^{T}\mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\} \leq 2\sum_{i\in\mathcal{U}}\Gamma_i(T)O\Big(\frac{2\psi^c\sigma^2}{\gamma^2{\lambda'}^2}\log\frac{d}{\delta'}\Big)$$

with a probability at least $1-\delta'$*, where* $\psi^c = g^{-1}\big(\frac{p(1-F(\upsilon^c;d,0))}{1-p};d,\upsilon^c\big)$ *is a constant.*

Combining results in Eq (2.3), Lemma 2.1.6 and Lemma 2.1.7, we obtain the upper regret bound:

$$R_T \leq O\Big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k})\Big) + 2\sum_{i\in\mathcal{U}}\Big(\Gamma_i(T)-1\Big)(\tau+\frac{2}{1-\delta_e}) + 2\sum_{i\in\mathcal{U}}\Gamma_i(T)O\Big(\frac{2\psi^c\sigma^2}{\gamma^2{\lambda'}^2}\log\frac{d}{\delta'}\Big)$$
$$= O\Big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k}) + \sum_{i\in\mathcal{U}}\Gamma_i(T)\cdot C\Big)$$

where $C = \frac{1}{1-\delta^e} + \frac{\sigma^2}{\gamma^2{\lambda'}^2}\log\frac{d}{\delta'}$, with a probability at least $(1-\delta)(1-\frac{\delta_e}{1-\delta_e})(1-\delta')$. This finishes the proof of Theorem 2.1.5.

Derivation of Eq (2.3)

Recall that we define a 'good' event as $\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) = N^{\phi}_{k_{i_t,t}}(t-1)\right\}$, which means at time t, DyClu selects an arm using the UCB score computed with observations associated with $\phi_{k_{i_t,t}}$. And the 'bad' event is defined as its complement: $\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}$, which can be decomposed and then contained as shown below:

$$
\begin{aligned}
&\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\} \\
&= \left(\left\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\right) \cup \left(\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\right) \\
&\subseteq \left\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\right\} \cup \left(\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\right)
\end{aligned}
$$

where the event $\left\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\right\}$ means at time step t there is a late detection, and the event $\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}$ means there is no late detection, but the cluster identification fails to correctly cluster user models associated with $\phi_{k_{i_t,t}}$ together (for example, there might be models not belonging to this cluster, or models failed to be put into this cluster).

Under the 'good' event, arm $\mathbf{x}_t$ is selected using the UCB strategy by aggregating all existing observations associated with $\phi_{k_{i_t,t}}$, which is the unique bandit parameter for user i_t at time t. To simplify the notations, borrowing the notation used in Eq (2.8), we denote $\hat{\theta}_{N^{\phi}_{k_{i_t,t}}(t-1)} = \mathbf{A}^{-1}_{N^{\phi}_{k_{i_t,t}}(t-1)} \mathbf{b}_{N^{\phi}_{k_{i_t,t}}(t-1)}$, where $\mathbf{A}_{N^{\phi}_{k_{i_t,t}}(t-1)} = \lambda\mathbf{I} + \sum_{j \in N^{\phi}_{k_{i_t,t}}(t-1)} \mathbf{x}_j \mathbf{x}_j^\top$ and $\mathbf{b}_{N^{\phi}_{k_{i_t,t}}(t-1)} = \sum_{j \in N^{\phi}_{k_{i_t,t}}(t-1)} \mathbf{x}_j y_j$, as the ridge regression estimator constructed using these observations, and $CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}) = \alpha_{N^{\phi}_{k_{i_t,t}}(t-1)} \sqrt{\mathbf{x}^\top \mathbf{A}^{-1}_{N^{\phi}_{k_{i_t,t}}(t-1)} \mathbf{x}}$, where $\alpha_{N^{\phi}_{k_{i_t,t}}(t-1)} = \sigma\sqrt{d\log\left(1 + \frac{|N^{\phi}_{k_{i_t,t}}(t-1)|}{d\lambda}\right) + 2\log\frac{1}{\delta}} + \sqrt{\lambda}$ is the corresponding reward estimation confidence bound on $\mathbf{x}$.

Then we can upper bound the instantaneous regret r_t as follows,

$$
\begin{aligned}
r_t &= \langle \theta_{i_t,t}, \mathbf{x}_t^* \rangle - \langle \theta_{i_t,t}, \mathbf{x}_t \rangle \leq \langle \tilde{\theta}_{i_t,t}, \mathbf{x}_t \rangle - \langle \theta_{i_t,t}, \mathbf{x}_t \rangle \\
&= \langle \tilde{\theta}_{i_t,t} - \hat{\theta}_{\hat{V}_{i,t-1}}, \mathbf{x}_t \rangle + \langle \hat{\theta}_{\hat{V}_{i,t-1}} - \theta_{i_t,t}, \mathbf{x}_t \rangle \\
&\leq \begin{cases} 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t), & \text{if } \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) = N^{\phi}_{k_{i_t,t}}(t-1)\right\}. \\ 2, & \text{otherwise.} \end{cases}
\end{aligned}
$$

The first inequality is because $\langle \tilde{\theta}_{i_t,t}, \mathbf{x}_t \rangle$ is optimistic, where $\mathbf{x}_t \in C_t$ and $\tilde{\theta}_{i_t,t} \in \left\{\theta \in \mathbb{R}^d : \|\hat{\theta}_{\hat{V}_{i,t-1}} - \theta\|_{\mathbf{A}^{-1}_{\hat{V}_{i,t-1}}} \leq \alpha_{N^{\phi}_{k_{i_t,t}}(t-1)}\right\}$. For the second inequality, we split it into two cases according to the occurrence of the 'good' or 'bad' events. Recall that $\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1)$ denotes the set of time steps associated with observations in $\hat{V}_{i_t,t-1}$. Then under the 'good' event $\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) = N^{\phi}_{k_{i_t,t}}(t-1)\right\}$, with probability at least $1-\delta$, we have $\langle \tilde{\theta}_{i_t,t} - \hat{\theta}_{\hat{V}_{i,t-1}}, \mathbf{x}_t \rangle \leq CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t)$ and $\langle \hat{\theta}_{\hat{V}_{i,t-1}} - \theta_{i_t,t}, \mathbf{x}_t \rangle \leq CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t)$, so that $r_t \leq 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t)$. Under the 'bad' event when wrong cluster is used for arm selection, we simply bound r_t by 2 because $\|\theta_{i_t,t}\| \leq 1$ and $\|\mathbf{x}_t\| \leq 1$.

Then the accumulated regret R_T can be upper bounded by:

$$
\begin{aligned}
R_T &= \sum_{t=1}^{T} r_t \leq 2\sum_{t=1}^{T} \mathbf{1}\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) = N^{\phi}_{k_{i_t,t}}(t-1)\right\} CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t) + 2\sum_{t=1}^{T} \mathbf{1}\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\} \\
&\leq \sum_{t=1}^{T} 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t) + 2\sum_{t=1}^{T} \mathbf{1}\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\} \\
&\leq \sum_{t=1}^{T} 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t) + 2\sum_{t=1}^{T} \mathbf{1}\left\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\right\} + 2\sum_{t=1}^{T} \left(\mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\right) \\
&\leq \sum_{t=1}^{T} 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t) + 2\sum_{i\in\mathcal{U}} \sum_{t\in\mathcal{N}_i(T)} \mathbf{1}\left\{\tilde{k}_{i,t-1} \neq k_{i,t}\right\} \\
&\quad + 2\sum_{t=1}^{T} \left(\mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\right)
\end{aligned}
$$

The first term is essentially the upper regret bound of the reference algorithm mentioned earlier in this section, which can be further upper bounded with probability at least $1-\delta$ by:

$$
\sum_{t=1}^{T} 2CB_{N^{\phi}_{k_{i_t,t}}(t-1)}(\mathbf{x}_t) = \sum_{k\in[m]} \sum_{t\in N^{\phi}_k(T)} 2CB_{N^{\phi}_k(t-1)}(x_t) \leq \sum_{k\in[m]} R_{Lin}(|N^{\phi}_k(T)|)
$$

where $R_{Lin}(|N^{\phi}_k(T)|)$ is the high probability upper regret bound in [20] (Theorem 3), which is defined as:

$$
\begin{aligned}
R_{Lin}(|N^{\phi}_k(T)|) &= 4\sqrt{d|N^{\phi}_k(T)| \log{(\lambda + \frac{|N^{\phi}_k(T)|}{d})}} \left(\sigma\sqrt{2\log\frac{1}{\delta} + d\log{(1+\frac{|N^{\phi}_k(T)|}{d\lambda})}} + \lambda^{1/2}\right) \\
&= O\left(\sigma d\sqrt{|N^{\phi}_k(T)| \log^2 |N^{\phi}_k(T)|} + \sigma\sqrt{d|N^{\phi}_k(T)| \log\frac{|N^{\phi}_k(T)|}{\delta}}\right)
\end{aligned}
$$

Proof of Lemma 2.1.6

Now we have proved the intermediate regret upper bound in Eq (2.3). In this section, we continue to upper bound its second term $2\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)} \mathbf{1}\left\{\tilde{k}_{i,t-1} \neq k_{i,t}\right\}$, which essentially counts the total number of late detections in each user, e.g., there is a mismatch between $\mathbb{M}_{i_t,t-1}$ and the current ground-truth bandit parameter $\theta_{i_t,t}$, but the learner fails to detect this. To prove this lemma, we need the following lemmas that upper bound the probability of late detections.

As opposed to early detection in Lemma 2.1.4, late detection corresponds to type-II error of homogeneity test in Lemma 2.1.3. Therefore we have the following lemma.

Lemma 2.1.8. *When change has happened ($\tilde{k}_{i_t,t-1} \neq k_{i_t,t}$), we have*

$$
P\big(e_{i_t,t} = 1\big) \geq \rho(1-\delta^{'})\big[1 - F(\upsilon^e; 1, \psi^e)\big]
$$

where $\psi^e = \frac{\Delta^2/\sigma^2}{1+1/\left(\frac{\lambda^{'}}{4}S_{min}-8\left(\log\frac{dS_{min}}{\delta^{'}}+\sqrt{S_{min}\log\frac{dS_{min}}{\delta^{'}}}\right)\right)}$.

Proof of Lemma 2.1.8. Combining Lemma 2.1.3, Assumption 1 and 3, we can lower bound the probability that $e_{i_t,t} = 1$ when change has happened as:

$$\begin{aligned} P\big(e_{i_t,t} = 1\big) &= P\big(s(\mathcal{H}_{i_t,t-1}, \{\mathbf{x}_t, y_t\}) > \upsilon^e\big) \\ &\geq 1 - F\left(\upsilon^e; 1, \frac{[\mathbf{x}_t^\top(\theta_{i_t,c} - \theta_{i_t,c-1})]^2/\sigma^2}{1 + \mathbf{x}_t^\top(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i_t,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)^{-1}\mathbf{x}_t}\right) \\ &\geq 1 - F\left(\upsilon^e; 1, \frac{[\mathbf{x}_t^\top(\theta_{i_t,c} - \theta_{i_t,c-1})]^2/\sigma^2}{1 + \frac{||\mathbf{x}_t||^2}{\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i_t,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)}}\right) \\ &\geq \rho\left[1 - F\left(\upsilon^e; 1, \frac{\Delta^2/\sigma^2}{1 + \frac{1}{\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i_t,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)}}\right)\right] \end{aligned}$$

Since the minimum length of stationary period is S_{min}, by applying Lemma A.15, we can obtain the following lower bound on minimum eigenvalue when change happens as:

$$\lambda_{\min}\left(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i_t,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\right) \geq \frac{\lambda^{'}}{4}S_{min} - 8\left(\log\frac{dS_{min}}{\delta'} + \sqrt{S_{min}\log\frac{dS_{min}}{\delta'}}\right)$$

with probability at least $1 - \delta^{'}$.

Denote $\psi^e = \frac{\Delta^2/\sigma^2}{1+1/\left(\frac{\lambda'}{4}S_{min}-8\left(\log\frac{dS_{min}}{\delta'}+\sqrt{S_{min}\log\frac{dS_{min}}{\delta'}}\right)\right)}$. We obtain the following lower bound on the probability of detection:

$$P\big(e_{i_t,t} = 1\big) \geq \rho(1 - \delta^{'})\big[1 - F(\upsilon^e; 1, \psi^e)\big]$$

when change has happened.

□

Lemma 2.1.9. *When change has happened ($\tilde{k}_{i_t,t-1} \neq k_{i_t,t}$),*

$$P\left(\hat{e}_{i_t,t} \geq 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \geq 1 - \delta_e$$

if the size of sliding window $\tau \geq \frac{2\log 1/\delta_e}{\left\{[1-F(\upsilon^e;1,\psi^e)]\rho(1-\delta')-1+F(\upsilon^e;1,0)\right\}^2}$.

Proof of Lemma 2.1.9. Similarly to the proof of Lemma 2.1.4, applying Hoeffding inequality given in Lemma A.12, we have:

$$P\left(\hat{e}_{i_t,t} \leq \mathbb{E}[e_{i,t}] - \sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \leq \delta_e$$

$$P\left(\hat{e}_{i_t,t} \geq \mathbb{E}[e_{i,t}] - \sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \geq 1 - \delta_e$$

From Lemma 2.1.8, when change has happened, $\mathbb{E}[e_{i_t,t}] \geq \rho(1 - \delta^{'})\,[1 - F(\upsilon^e; 1, \psi^e)]$, with ψ^e being a variable dependent on environment as defined in Lemma 2.1.8. By substituting this into the above inequality, we have:

$$P\left(\hat{e}_{i,t} \geq \rho(1 - \delta^{'})\,[1 - F(\upsilon^e; 1, \psi^e)] - \sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \geq 1 - \delta_e$$

Then by rearranging terms above, we can find that if the sliding window size τ is selected to satisfy:

$$\tau \geq \frac{2\log 1/\delta_e}{\Big\{[1-F(\upsilon^e;1,\psi^e)]\rho(1-\delta')-1+F(\upsilon^e;1,0)\Big\}^2}$$

we can obtain:

$$P\left(\hat{e}_{i_t,t} \geq 1-F(\upsilon^e;1,0)+\sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \geq 1-\delta_e$$
$$P\left(\hat{e}_{i_t,t} \leq 1-F(\upsilon^e;1,0)+\sqrt{\frac{\log 1/\delta_e}{2\tau}}\right) \leq \delta_e$$

when change has happened ($\tilde{k}_{i_t,t-1} \neq k_{i_t,t}$).

□

Proof of Lemma 2.1.6. With results from Lemma 2.1.9, our solution to further upper bound the number of late detections in each stationary period is similar to [12] (Theorem 3.2). We include the proof here for the sake of completeness.

Denote the probability of detection when change has happened as $p_d = P\Big(\hat{e}_{i_t,t} \geq 1-F(\upsilon^e;1,0)+\sqrt{\frac{\log 1/\delta_e}{2\tau}}\Big)$, and from Lemma 2.1.9, we have $p_d \geq 1-\delta_e$. The probability distribution over the number of late detections when change has happened follows a geometric distribution: $P(n_{\text{late}}=k)=(1-p_d)^{k-1}p_d$. Then by applying Chebyshev's inequality, we have $P\Big(n_{\text{late}} \leq \frac{2}{1-\delta_e}\Big) \geq 1-\frac{\delta_e}{1-\delta_e}$.

Now we can upper bound the number of late detections $\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\Big\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\Big\}$ in user i. In Assumption 1 we have assumed that the total number of change points of user i is $\Gamma_i(T)-1$. Therefore, $\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\big\{\tilde{k}_{i_t,t-1} \neq k_{i_t}\big\} \leq \big(\Gamma_i(T)-1\big)(\tau+\frac{2}{1-\delta_e})$ with probability at least $1-\frac{\delta_e}{1-\delta_e}$. Then we can upper bound the second term in Eq (2.3) by:

$$2\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\Big\{\tilde{k}_{i_t,t-1} \neq k_{i_t,t}\Big\} \leq 2\sum_{i\in\mathcal{U}}\big(\Gamma_i(T)-1\big)(\tau+\frac{2}{1-\delta_e})$$

□

Proof of Lemma 2.1.7

The third term $\sum_{t=1}^{T}\mathbf{1}\Big\{\tilde{k}_{i_t,t-1}=k_{i_t,t}\Big\} \cap \Big\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\Big\}$ counts the total number of times that there is no late detection, but cluster identification module fails to correctly cluster user models. We upper bound this using a similar idea as [39], but is based on the properties of homogeneity test. For the proof of Lemma 2.1.7, we need the following lemmas related to probability of errors of cluster identification.

Lemma 2.1.10. *When the underlying bandit parameters $\phi_{\tilde{k}_{i,t-1}}$ and $\phi_{\tilde{k}_{j,t-1}}$ of two observation history $\mathcal{H}_{i,t-1}$ and $\mathcal{H}_{j,t-1}$ are the same, the probability that cluster identification fails to cluster them together corresponds to the type-I error probability given in Lemma 2.1.2, and it can be upper bounded by:*

$$P\Big(S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) > \upsilon^c \mid \phi_{\tilde{k}_{i,t-1}}=\phi_{\tilde{k}_{j,t-1}}\Big) \leq 1-F(\upsilon^c;df,0)$$

where $df = rank(\mathbf{X}_1)+rank(X_2)-rank(\begin{bmatrix}X_1\\X_2\end{bmatrix})$.

Corollary 2.1.10.1 (Lower bound $P\big(N^{\phi}_{k_{i_t,t}}(t-1) \subseteq \hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1)\big)$)**.** *Since $N^{\phi}_{k_{i_t,t}}(t-1)$ denotes the set of time indices associated with all observations whose underlying bandit parameter is $\phi_{k_{i_t,t}}$, and $\hat{N}^{\phi}_{\tilde{k}_{i,t-1}}(t-1)$ denotes those* *in the estimated neighborhood $\hat{V}_{i_t,t-1}$, when there is no late detection, i.e., we have $\tilde{k}_{i_t,t-1} = k_{i_t,t}$. It naturally follows Lemma 2.1.10 that:*

$$P\big(N^{\phi}_{k_{i_t}}(t-1) \subseteq \hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1)\big) \geq F(\upsilon^c; df, 0)$$

Lemma 2.1.11. *When the underlying bandit parameters $\phi_{\tilde{k}_{i,t-1}}$ and $\phi_{\tilde{k}_{j,t-1}}$ of two observation sequence $\mathcal{H}_{i,t-1}$ and $\mathcal{H}_{j,t-1}$ are not the same, the probability that cluster identification module clusters them together corresponds to the type-II error probability given in Lemma 2.1.3, which can be upper bounded by:*

$$P\Big(S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c | \phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\Big) \leq F(\upsilon^c; d, \psi^c)$$

under the condition that both $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ and $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ are at least $\frac{2\psi^c\sigma^2}{\gamma^2}$.

Proof of Lemma 2.1.11. Recall that type-II error probability of the homogeneity test can be upper bounded by $P\big(S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c | \phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\big) \leq F(\upsilon^c; df, \psi)$ as discussed in Section 2.1.3. If either design matrix of the two datasets is rank-deficient, the noncentrality parameter ψ is lower bounded by 0 (lower bound achieved when the difference between two parameters lies in the null space of rank-deficient design matrix). Therefore, a nontrivial upper bound of type-II error probability only exists when the design matrices of both datasets are rank-sufficient. In this case, combining Lemma 2.1.3 and Assumption 2 gives:

$$\begin{aligned}
&P\big(S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1})\big) \\
\leq &F\left(\upsilon^c; d, \frac{||\phi_{\tilde{k}_{i,t-1}} - \phi_{\tilde{k}_{j,t-1}}||^2/\sigma^2}{1/\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top) + 1/\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)}\right) \\
\leq &F\left(\upsilon^c; d, \frac{\gamma^2/\sigma^2}{1/\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top) + 1/\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)}\right)
\end{aligned}$$

Define $\psi^c > 0$; then by rearranging terms we obtain the conditions that, when both:

$$\lambda_{\min}\left(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\right) \geq \frac{2\psi^c\sigma^2}{\gamma^2} \quad \text{and} \quad \lambda_{\min}\left(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\right) \geq \frac{2\psi^c\sigma^2}{\gamma^2}$$

we have

$$P\big(S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c | \phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\big) \leq F(\upsilon^c; d, \psi^c)$$

□

Lemma 2.1.12. *If the cluster identification module clusters observation history $\mathcal{H}_{i,t-1}$ and $\mathcal{H}_{j,t-1}$ together, the probability that they actually have the same underlying bandit parameters is denoted as $P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c\big)$.*

$$P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c\big) \geq F(\upsilon^c; df, 0)$$

under the condition that both $\lambda_{\min}\big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\big)$ and $\lambda_{\min}\big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\big)$ are at least $\frac{2\psi^c\sigma^2}{\gamma^2}$, where $\psi^c = g^{-1}\big(\frac{p(1-F(\upsilon^c;d,0))}{1-p}; d, \upsilon^c\big)$.

Proof of Lemma 2.1.12. Compared with the type-I and type-II error probabilities given in Lemma 2.1.10 and 2.1.11, the probability $P(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c)$ also depends on the population being tested on. Two extreme examples would be 1) testing on a population that all user models have the same bandit parameter, and 2) every user model has an unique bandit parameter. Then in the former case $P(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c) = 1$ and in the latter case $P(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c) = 0$.

Denote the events $\big\{\phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\big\} \cap \big\{S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) > \upsilon^c\big\}$ as True Positive (TP), $\big\{\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}}\big\} \cap \big\{S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) \leq \upsilon^c\big\}$ as True Negative (TN), $\big\{\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}}\big\} \cap \big\{S(\mathcal{H}_{i,t-1}, \mathcal{H}_{j,t-1}) > \upsilon^c\big\}$ as False Positive

(FP), and $\big\{\phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\big\} \cap \big\{S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c\big\}$ as False Negative (FN) of cluster identification, respectively. We can rewrite the probabilities in Lemma 2.1.10, 2.1.11 and 2.1.12 as:

$$\begin{aligned}
&P\big(S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) > \upsilon^c | \phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}}\big) = \frac{P(\text{FP})}{P(\text{TN}+\text{FP})} \leq 1 - F(\upsilon^c; \mathit{df}, 0)\\
&P\big(S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c | \phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}\big) = \frac{P(\text{FN})}{P(\text{FN}+\text{TP})} \leq F(\upsilon^c; \mathit{df}, \psi^c)\\
&P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c\big) = \frac{P(\text{TN})}{P(\text{TN}+\text{FN})} \geq \frac{P(\text{TN})}{P(\text{TN})+P(\text{FN})} = \frac{1}{1+\frac{P(\text{FN})}{P(\text{TN})}}
\end{aligned}$$

We can upper bound $\frac{P(\text{FN})}{P(\text{TN})}$ by:

$$\frac{P(\text{FN})}{P(\text{TN})} \leq \frac{P(\text{TP}+\text{FN})}{P(\text{TN}+\text{FP})} \cdot \frac{F(\upsilon^c; \mathit{df}, \psi^c)}{F(\upsilon^c; \mathit{df}, 0)}$$

where $\frac{P(\text{TP+FN})}{P(\text{TN+FP})}$ denotes the ratio between the number of positive instances ($\phi_{\tilde{k}_{i,t-1}} \neq \phi_{\tilde{k}_{j,t-1}}$) and negative instances ($\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}}$) in the population, which can be upper bounded by $\frac{1-p}{p}$ where p denotes the lower bound of the portion that each unique bandit parameter occurs in all stationary periods, i.e. $p = 1$ means the same unique bandit parameter occurs in all stationary periods.

It is worth noting that when either design matrix of $\mathcal{H}_{i,t-1}$ or $\mathcal{H}_{j,t-1}$ does not have full column rank, $P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c\big) \geq 1/\big(1 + \frac{1-p}{p} \cdot \frac{F(\upsilon^c;\mathit{df},0)}{F(\upsilon^c;\mathit{df},0)}\big) \geq p$, which is then trivially lower bounded by the percentage of negative instances in the population.

Under the conditions given in Lemma 2.1.11, we have:

$$P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c\big) \geq 1/\Big(1 + \frac{1-p}{p} \cdot \frac{F(\upsilon^c; d, \psi^c)}{F(\upsilon^c; d, 0)}\Big)$$

Then by setting $\psi^c = g^{-1}\big(\frac{p(1-F(\upsilon^c;d,0))}{1-p}; d, \upsilon^c\big)$, we have:

$$P\big(\phi_{\tilde{k}_{i,t-1}} = \phi_{\tilde{k}_{j,t-1}} | S(\mathcal{H}_{i,t-1},\mathcal{H}_{j,t-1}) \leq \upsilon^c\big) \geq 1/\Big(1 + \frac{1-p}{p} \cdot \frac{F(\upsilon^c; d, \psi^c)}{F(\upsilon^c; d, 0)}\Big) = F(\upsilon^c; \mathit{df}, 0)$$

□

Corollary 2.1.12.1 (Lower bound $P\big(\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \subseteq N^{\phi}_{k_{i_t,t}}(t-1)\big)$)**.** *It naturally follows Lemma 2.1.12 that:*

$$P\big(\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \subseteq N^{\phi}_{k_{i_t,t}}(t-1)\big) \geq F(\upsilon^c; \mathit{df}, 0)$$

under the condition that both $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ *and* $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ *are at least* $\frac{2\psi^c\sigma^2}{\gamma^2}$*, where* $\psi^c = g^{-1}\big(\frac{p(1-F(\upsilon^c;d,0))}{1-p}; d, \upsilon^c\big)$*.*

Proof of Lemma 2.1.7. From Corollary 2.1.10.1 and Corollary 2.1.12.1, when both $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ and $\lambda_{\min}(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{j,t-1}} \mathbf{x}_k\mathbf{x}_k^\top)$ are at least $\frac{2\psi^c\sigma^2}{\gamma^2}$, with probability at least $F(\upsilon^c; \mathit{df}, 0)$, we have event $\Big\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\Big\}\cap$

$\left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) = N^{\phi}_{k_{i_t,t}}(t-1)\right\}$. Therefore, the third term in Eq (2.3) is upper bounded by:

$$\begin{aligned}
&2\sum_{t=1}^{T}\mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\\
\leq &2\sum_{t=1}^{T}\mathbf{1}\left\{\exists \mathcal{H} \in \mathbf{U}_{t-1} \cup \mathbb{O}_{t-1} : \lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}\\
\leq &2\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\left\{\lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}
\end{aligned}$$

Essentially, it counts the number of time steps in total when minimum eigenvalue of a user model $\mathbb{M}$'s correlation matrix is smaller than $\frac{2\psi^c\sigma^2}{\gamma^2}$. We further decompose the summation by considering each stationary period of each user.

$$\begin{aligned}
&2\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\left\{\lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}\\
=&2\sum_{i\in\mathcal{U}}\sum_{s\in[0,c_{i,1},..,c_{i,\Gamma_i(T)-1}]}\sum_{t\in S_{i,s}}\mathbf{1}\left\{\lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}
\end{aligned}$$

where $S_{i,s}$ denotes the s'th stationary period of user i.

Borrowing the notation from [39], denote A_t as a correlation matrix constructed through a series of rank-one updates using context vectors from $\{C_t\}_{t\in S}$, where S denotes the set of time steps we performed model update. Note that the choice of which context vector to select from C_t for $t \in S$ can be arbitrary. Then we denote the maximum number of updates it takes until $\lambda_{\min}(A_t)$ is lower bounded by η as $\text{HD}(\{C_t\}_{t\in S}, \eta) = \max\{t \in S : \exists \mathbf{x}_1 \in C_1, ..., \mathbf{x}_t \in C_t : \lambda_{\min}(A_t) \leq \eta\}$, where $A_t = \sum_{u\in S: u\leq t}\mathbf{x}_u\mathbf{x}_u^\top$. Therefore, we obtain:

$$\begin{aligned}
&\sum_{i\in\mathcal{U}}\sum_{t\in\mathcal{N}_i(T)}\mathbf{1}\left\{\lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}\\
=&\sum_{i\in\mathcal{U}}\sum_{s\in[0,c_{i,1},..,c_{i,\Gamma_i(T)-1}]}\sum_{t\in S_{i,s}}\mathbf{1}\left\{\lambda_{\min}\Big(\sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t-1}} \mathbf{x}_k\mathbf{x}_k^\top\Big) < \frac{2\psi^c\sigma^2}{\gamma^2}\right\}\\
\leq&\sum_{i\in\mathcal{U}}\sum_{s\in[0,c_{i,1},..,c_{i,\Gamma_i(T)-1}]}\text{HD}\Big(\{C_t\}_{t\in S_{i,s}}, \frac{2\psi^c\sigma^2}{\gamma^2}\Big)
\end{aligned}$$

Then similar to [39] (Lemma 1), by applying Lemma A.15 we can upper bound the third term in Eq (2.3):

$$\begin{aligned}
&2\sum_{t=1}^{T}\mathbf{1}\left\{\tilde{k}_{i_t,t-1} = k_{i_t,t}\right\} \cap \left\{\hat{N}^{\phi}_{\tilde{k}_{i_t,t-1}}(t-1) \neq N^{\phi}_{k_{i_t,t}}(t-1)\right\}\\
\leq&2\sum_{i\in\mathcal{U}}\sum_{s\in[0,c_{i,1},..,c_{i,\Gamma_i(T)-1}]}\text{HD}\Big(\{C_t\}_{t\in S_{i,s}}, \frac{2\psi^c\sigma^2}{\gamma^2}\Big)\\
\leq&2\sum_{i\in\mathcal{U}}\Gamma_i(T)O\Big(\frac{2\psi^c\sigma^2}{\gamma^2{\lambda'}^2}\log\frac{d}{\delta'}\Big)
\end{aligned}$$

with probability at least $1-\delta'$. □

2.2 Cooperation in decentralized environments: non-linear models

Most existing bandit solutions are designed under a centralized setting (i.e., data is readily available at a central server), including the works in non-stationary bandits and clustered bandits introduced in Section 2.1. In response to the increasing application scale and public concerns of privacy, there is a growing demand to keep data decentralized and push the learning of bandit models to the client side. This has led to the increasing research effort on *federated bandit learning* lately [28, 61, 62, 63, 64], i.e., N decentralized clients cooperate with limited communication bandwidth to minimize the overall cumulative regret incurred over a finite time horizon T, while keeping each client's raw data local. Compared with standard federated learning [65, 66] that works with fixed datasets, federated bandit learning is characterized by its online interactions with the environment, which continuously provides new data samples to the clients over time. This brings in new challenges in addressing the conflict between the need of timely data/model aggregation for regret minimization and the need of communication efficiency with decentralized data. A carefully designed model update method and communication strategy become vital to strike this balance.

Existing federated bandit learning solutions only partially addressed this challenge by considering simple bandit models, like context-free bandit [62] and contextual linear bandit [28, 61, 64], where closed-form solution for both local and global model update exists. Therefore, efficient communication for global bandit model update is realized by directly aggregating local sufficient statistics, such that the only concern left is how to control the communication frequency over time horizon T. However, such a solution framework does not apply to the more complicated bandit models that are often preferred in practice. Therefore, in this section, we propose algorithms that enable collaborative exploration of more complex function classes, such as generalized linear bandit (GLB) and kernelized contextual bandit.

2.2.1 Related works

Generalized linear bandit GLB, as an important extension of linear bandit models, has demonstrated encouraging performance in modeling binary rewards (such as clicks) that are ubiquitous in real-world applications [67]. The study of GLB under a centralized setting dates back to Filippi et al. [25], who proposed a UCB-type algorithm that achieved $\tilde{O}(d\sqrt{T})$ regret. Li et al. [68] later proposed two improvements: a similar UCB-type algorithm that improves the result of [25] by a factor of $O(\log T)$, which has been popularly used in practice as it avoids the projection step needed in [25]; and another impractical algorithm that further improves the result by a factor of $O(\sqrt{d})$ assuming fixed number of arms. To improve the time and space complexity of the aforementioned GLB algorithms, followup works adopted online regression methods. In particular, motivated by the online-to-confidence-set conversion technique from [69], Jun et al. [70] proposed both UCB and Thompsan sampling algorithms with online Newton step, and Ding et al. [71] proposed a Thompson sampling algorithm with online gradient descent, which, however, requires an additional context regularity assumption to obtain a sub-linear regret.

Kernelized/Gaussian process bandit By using kernels and Gaussian processes, studies in [8, 26, 72] further extend UCB-type algorithms to non-parametric reward functions in RKHS, i.e., the feature map associated with each arm is possibly infinite. To improve computation efficiency of these algorithms, Nyström approximation method is adopted. Specifically, Calandriello et al. [73] proposed an algorithm named BKB, which uses Ridge Leverage Score sampling (RLS) to re-sample a new dictionary from the updated dataset after each interaction with the environment. A recent work by Zenati et al. [74] further improved the computation efficiency of BKB by adopting an online sampling method to update the dictionary. However, both of them updated the dictionary at each time step to ensure the dictionary remains representative w.r.t. the growing dataset. Calandriello et al. [75] further proposed a variant of BKB, named BBKB, for batched Gaussian process optimization. BBKB only needs to update the dictionary occasionally according to an adaptive schedule, and thus addresses the issue mentioned above.

Federated bandits Recent years have witnessed increasing research efforts in distributed/federated bandit learning, i.e., multiple agents collaborate in pure exploration [76, 77, 78], or regret minimization [62, 28, 64]. They mainly differ in the relations of learning problems solved by the agents (i.e., homogeneous vs., heterogeneous) and the type of communication network (i.e., peer-to-peer (P2P) vs., star-shaped). Most of these works assume linear reward functions, and the clients communicate by transferring the $O(d^2)$ sufficient statistics. Korda et al. [79] considered a peer-to-peer (P2P) communication network and assumed that the clients form clusters, i.e., each cluster is associated with a unique bandit problem. However, they only focused on reducing *per-round* communication, and hence the communication cost is still linear over time. Huang et al. [63] considered a star-shaped communication network as us, but their

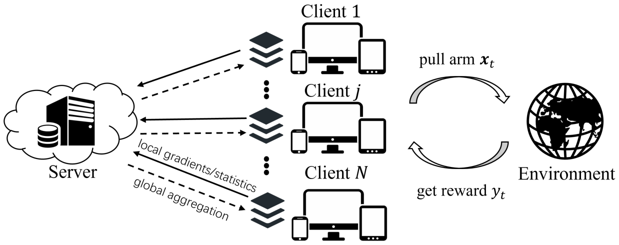


Figure 2.4: Illustration of federated bandit problem: a network of N clients sequentially taking actions and receiving feedback from the environment, and a server coordinating their communication.

proposed phase-based elimination algorithm only works in the fixed arm set setting. The closest works to ours are [28, 61, 64], which proposed event-triggered communication protocols to obtain sub-linear communication cost over time for distributed linear bandits with a time-varying arm set. In comparison, federated GLB and kernelized contextual bandits still remain under-explored.

Offline federated learning Another related line of research is the standard federated learning that considers offline supervised learning problems [66]. Since its debut in [65], FedAvg has become the most popularly used algorithm for offline federated learning. However, despite its popularity, several works [80, 81, 82] identified that FedAvg suffers from a *client-drift* problem when the clients' data are non-IID (which is an important signature of our case), i.e., local iterates in each client drift towards their local minimum. This leads to a sub-optimal convergence rate of FedAvg: for example, one has to suffer a sub-linear convergence rate for strongly convex and smooth losses, though a linear convergence rate is expected under a centralized setting. To alleviate this, Pathak and Wainwright [83] proposed an operator splitting procedure to guarantee linear convergence to a neighborhood of the global minimum. Later, Mitra et al. [82] introduced variance reduction techniques to guarantee exact linear convergence to the global minimum.

2.2.2 General problem formulation

Consider a learning system with 1) N clients responsible for taking actions and receiving corresponding reward feedback from the environment, e.g., each client being an edge device directly interacting with a user, and 2) a central server responsible for coordinating the communication between the clients for joint model estimation. This is illustrated in Figure 2.4.

At each time step $t = 1, 2, ..., T$, all N clients interact with the environment in a round-robin manner, i.e., each client $i \in [N]$ chooses an arm $\mathbf{x}_{t,i}$ from its time-varying candidate set $\mathcal{A}_{t,i} = \{\mathbf{x}_{t,i}^{(1)}, \mathbf{x}_{t,i}^{(2)}, \dots, \mathbf{x}_{t,i}^{(K)}\}$, where $\mathbf{x}_{t,i}^{(a)} \in \mathbb{R}^d$ denotes the context vector associated with the a-th arm for client i at time t. Without loss of generality, we assume $||\mathbf{x}_{t,i}^{(a)}||_2 \leq 1, \forall i, a, t$. Then client i receives the corresponding reward $y_{t,i} \in \mathbb{R}$ from the environment, which is drawn from the reward distribution governed by an unknown parameter $\theta_\star \in \mathbb{R}^d$ (assume $\|\theta_\star\| \leq S$), i.e., $y_{t,i} \sim p_{\theta_\star}(y|\mathbf{x}_{t,i}^{(a)})$. The interaction between the learning system and the environment repeats itself, and the goal of the learning system is to minimize the cumulative (pseudo) regret over all N clients in the finite time horizon T, i.e., $R_T = \sum_{t=1}^{T}\sum_{i=1}^{N} r_{t,i}$, where $r_{t,i} = \max_{\mathbf{x}\in\mathcal{A}_{t,i}} \mathbf{E}[y|\mathbf{x}] - \mathbf{E}[y_{t,i}|\mathbf{x}_{t,i}]$.

In a federated learning setting, the clients cannot directly communicate with each other, but through the central server, i.e., a star-shaped communication network. Raw data collected by each client $i \in [N]$, i.e., $\{(\mathbf{x}_{s,i}, y_{s,i})\}_{s\in[T]}$, is stored locally and cannot be shared with anyone else. Instead, the clients can only communicate the parameters of the learning algorithm, e.g., models, gradients, or sufficient statistics; and the communication cost is measured by the total number of times data being transferred across the system up to time T, which is denoted as C_T.

Federated Linear Bandit Prior works have studied communication-efficient federated linear bandit [28, 61], i.e., the reward function is a linear model $y_{t,i} = \mathbf{x}_{t,i}^\top\theta_\star + \eta_{t,i}$, where $\eta_{t,i}$ denotes zero-mean sub-Gaussian noise. Consider an imaginary centralized agent that has direct access to the data of all clients, so that it can compute the global sufficient statistics $A_t = \sum_{i\in[N]}\sum_{s\in[t]} \mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top, b_t = \sum_{i\in[N]}\sum_{s\in[t]} \mathbf{x}_{s,i}y_{s,i}$. Then the cumulative regret incurred by this distributed learning system can match that under a centralized setting, if all N clients select arms based on the global sufficient statistics $\{A_t, b_t\}$. However, it requires N^2T communication cost for the immediate sharing of each client's update to the sufficient statistics with all other clients, which is expensive for most applications.

To ensure communication efficiency, prior works like DisLinUCB [28] let each client i maintain a local copy $\{A_{t-1,i}, b_{t-1,i}\}$ for arm selection, which receives immediate local update using each newly collected data sample, i.e., $A_{t,i} = A_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top, b_{t,i} = b_{t-1,i} + \mathbf{x}_{t,i}y_{t,i}$. Then client i checks whether the event $(t - t_{\text{last}}) \log(\frac{\det A_{t,i}}{\det A_{t_{\text{last}}}}) > D$ is true, where t_{last} denotes the time step of last global update. If true, a new global update is triggered, such that the server will collect all clients' local update since t_{last}, aggregate them to compute $\{A_t, b_t\}$, and then synchronize the local sufficient statistics of all clients, i.e., set $\{A_{t,i}, b_{t,i}\} = \{A_t, b_t\}, \forall i \in [N]$.

2.2.3 Federated generalized linear bandit problem

In this work, we study federated bandit learning with generalized linear models, i.e., the conditional distribution of reward y given context vector $\mathbf{x}$ is drawn from the exponential family [25, 68]:

$$p_{\theta_\star}(y|\mathbf{x}) = \exp\left(\frac{y\mathbf{x}^\top\theta_\star - m(\mathbf{x}^\top\theta_\star)}{g(\tau)} + h(y,\tau)\right) \tag{2.4}$$

where $\tau \in \mathbb{R}^+$ is a known scale parameter. Given a function $f : \mathbb{R} \to \mathbb{R}$, we denote its first and second derivatives by $\dot{f}$ and $\ddot{f}$, respectively. It is known that $\dot{m}(\mathbf{x}^\top\theta_\star) = \mathbb{E}[y|\mathbf{x}] := \mu(\mathbf{x}^\top\theta_\star)$, which is called the inverse link function, and $\ddot{m}(\mathbf{x}^\top\theta_\star) = \mathbf{V}(y|\mathbf{x}^\top\theta_\star)$. Based on Eq.(2.4), the reward $y_{t,i}$ observed by client i at time t can be equivalently represented as $y_{t,i} = \mu(\mathbf{x}_{t,i}^\top\theta_\star) + \eta_{t,i}$, where $\eta_{t,i}$ denotes the sub-Gaussian noise. Then we denote the negative log-likelihood of $y_{i,t}$ given $\mathbf{x}_{i,t}$ as $l(\mathbf{x}_{t,i}^\top\theta_\star, y_{t,i}) = -\log p_{\theta_\star}(y_{t,i}|\mathbf{x}_{t,i}) = -y_{t,i}\mathbf{x}_{t,i}^\top\theta_\star + m(\mathbf{x}_{t,i}^\top\theta_\star)$. In addition, we adopt the following two assumptions about the reward, which are standard for GLB [25].

Assumption 4. *The link function μ is continuously differentiable on $(-S,S)$, k_μ-Lipschitz on $[-S,S]$, and $\inf_{z\in[-S,S]} \dot{\mu}(z) = c_\mu > 0$.*

Assumption 5. *$\mathbb{E}[\eta_{t,i}|\mathcal{F}_{t,i}] = 0, \forall t, i$, where $\mathcal{F}_{t,i} = \sigma\{\mathbf{x}_{t,i}, [\mathbf{x}_{s,j}, y_{s,j}]_{(s,j):s<t\cap j=i}\}$ denotes the σ-algebra generated by client i's previously pulled arms and observed rewards, and $\max_{t,i} |\eta_{t,i}| \leq R_{\max}$ for some constant $R_{\max} > 0$.*

GLB covers a wider range of non-linear parametric models, including linear, Poisson, logistic regression, etc. However, to enable joint model estimation for GLB, the learning system needs to solve distributed convex optimization problems for multiple times to adapt to the new data collected from the environment over time, and each requires iterative gradient/model aggregation among the clients. This is much more expensive compared with linear models, and it naturally leads to the question: whether a communication efficient solution to this challenging problem is still possible? In [84], we answered this question affirmatively by proposing the first provably communication efficient algorithm for distributed GLB.

New Challenges Compared with federated linear bandit discussed in Section 2.2.2, new challenges arise in designing a communication-efficient algorithm for federated GLB due to the absence of a closed form solution:

- *Iterative communication for global update:* compared with the global update for federated linear bandit that only requires one round of communication to share the sufficient statistics, now it takes multiple iterations of gradient aggregation to obtain converged global optimization. Moreover, as the clients collect more data samples over time during bandit learning, the required number of iterations for convergence also increases.

- *Drifting issue with local update:* during local model update, iterative optimization using only local gradient can push the updated model away from the global model, i.e., forget the knowledge gained during previous communications [85].

2.2.4 FedGLB-UCB algorithm

To ensure communication-efficient model updates for federated GLB, we propose to use online regression for local update, i.e., update each client's local model only with its newly collected data samples, and use offline regression for global update, i.e., solicit all clients' local gradients for joint model estimation. Based on the resulting sequence of

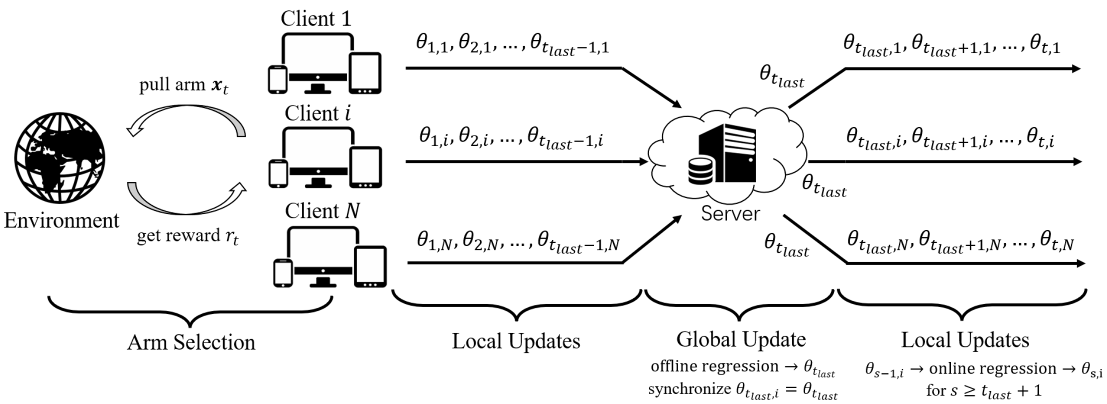

Client 1
pull arm $\boldsymbol{x}_t$
Client i
Environment
get reward r_t
Client N
$\theta_{1,1}, \theta_{2,1}, \dots, \theta_{t_{last}-1,1}$
$\theta_{1,i}, \theta_{2,i}, \dots, \theta_{t_{last}-1,i}$
$\theta_{1,N}, \theta_{2,N}, \dots, \theta_{t_{last}-1,N}$
$\theta_{t_{last}}$
$\theta_{t_{last}}$
$\theta_{t_{last}}$
Server
$\theta_{t_{last},1}, \theta_{t_{last}+1,1}, \dots, \theta_{t,1}$
$\theta_{t_{last},i}, \theta_{t_{last}+1,i}, \dots, \theta_{t,i}$
$\theta_{t_{last},N}, \theta_{t_{last}+1,N}, \dots, \theta_{t,N}$
Arm Selection
Local Updates
Global Update
offline regression → $\theta_{t_{last}}$
synchronize $\theta_{t_{last},i} = \theta_{t_{last}}$
Local Updates
$\theta_{s-1,i}$ → online regression → $\theta_{s,i}$
for $s \geq t_{last} + 1$

Figure 2.5: Illustration of FedGLB-UCB algorithm, which uses online regression for local update, i.e., immediately update each client's local model $\theta_{t,i}$ using its newly collected data sample, and uses offline regression for global update, i.e., synchronize all N clients to a globally updated model $\theta_{t_{\text{last}}}$ using all the data samples collected so far.

offline-and-online model updates, the confidence ellipsoid for $\theta_\star$ is constructed for each client to select arms using the OFUL principle. We name this algorithm Federated Generalized Linear Bandit with Upper Confidence Bound, or FedGLB-UCB for short. We illustrate its key components in Figure 2.5 and describe its procedures in Algorithm 2. In the following, we discuss about each component of FedGLB-UCB in details.

Algorithm 2 FedGLB-UCB

```
Input: threshold $D$, regularization parameter $\lambda > 0$, $\delta \in (0,1)$ and $c_\mu$.
Initialize $\forall i \in [N]$: $A_{0,i} = \frac{\lambda}{c_\mu}\mathbf{I} \in \mathbb{R}^{d\times d}, b_{0,i} = \mathbf{0} \in \mathbb{R}^d, \theta_{0,i} = \mathbf{0} \in \mathbb{R}^d, \Delta A_{0,i} = \mathbf{0} \in \mathbb{R}^{d\times d}$; $A_0 = \frac{\lambda}{c_\mu}\mathbf{I} \in \mathbb{R}^{d\times d}, b_0 = \mathbf{0} \in \mathbb{R}^d, \theta_0 = \mathbf{0} \in \mathbb{R}^d$, $t_{\text{last}} = 0$
for $t = 1, 2, \dots, T$ do
  for client $i = 1, 2, \dots, N$ do
    Observe arm set $\mathcal{A}_{t,i}$ for client $i$
    Select arm $\mathbf{x}_{t,i} \in \mathcal{A}_{t,i}$ by Eq.(2.8), and observe reward $y_{t,i}$
    Update client $i$: $A_{t,i} = A_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top$, $\Delta A_{t,i} = \Delta A_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top$
    if $(t - t_{\text{last}}) \log \frac{\det(A_{t,i})}{\det(A_{t,i} - \Delta A_{t,i})} < D$ then
      Client $i$: perform local update $\theta_{t,i} = \text{ONS-Update}(\theta_{t-1,i}, A_{t,i}, \nabla l(\mathbf{x}_{t,i}^\top \theta_{t-1,i}, y_{t,i}))$, $b_{t,i} = b_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top \theta_{t-1,i}$
    else
      Clients $\forall i \in [N]$: send $\Delta A_{t,i}$ to server, and reset $\Delta A_{t,i} = \mathbf{0}$
      Server: compute $A_t = A_{t_{\text{last}}} + \sum_{i=1}^N \Delta A_{t,i}$
      Server: perform global update $\theta_t = \text{AGD-Update}(\theta_{t_{\text{last}}}, J_t)$ (see Eq.(2.6) for the choice of $J_t$), $b_t = b_{t_{\text{last}}} + \sum_{i=1}^N \Delta A_{t,i}\theta_t$, and set $t_{\text{last}} = t$
      Clients $\forall i \in [N]$: set $\theta_{t,i} = \theta_t, A_{t,i} = A_t, b_{t,i} = b_t$
```

Local update As mentioned earlier, iterative optimization over local dataset $\{(\mathbf{x}_{s,i}, y_{s,i})\}_{s\in[t]}$ leads to the drifting issue that pushes the updated model to the local optimum. Due to the small size of this local dataset, the confidence ellipsoid centered at the converged model has increased width, which leads to increased regret in bandit learning. However, as we will prove in Section 2.2.5, completely disabling local update and restricting all clients to use the previous globally updated model for arm selection is also a bad choice, because the learning system will then need more frequent global updates to adapt to the growing dataset.

To enable local update while alleviating the drifting issue, we adopt online regression in each client, such that the local model estimation $\theta_{t,i}$ is only updated for one step using the sample $(\mathbf{x}_{t,i}, y_{t,i})$ collected at time t. Prior works [69, 70] showed that UCB-type algorithms with online regression can attain comparable cumulative regret to the standard UCB-type algorithms [20, 68], as long as the selected online regression method guarantees logarithmic online regret. As the negative log-likelihood loss defined in Section 2.2.3 is exp-concave and online Newton step (ONS) is known to attain logarithmic online regret in this case [86, 70], ONS is chosen for the local update of FedGLB-UCB and its description is given in Algorithm 3. At time step t, after client i pulls an arm $\mathbf{x}_{t,i} \in \mathcal{A}_{t,i}$ and observes the reward $y_{t,i}$, its model $\theta_{t-1,i}$ is immediately updated by the ONS update rule (line 9 in Algorithm 2), where $\nabla l(\mathbf{x}_{t,i}^{\top}\theta_{t-1,i}, y_{t,i})$ denotes the gradient w.r.t. $\theta_{t-1,i}$, and $A_{t,i}$ denotes the covariance matrix for client i at time t.

Algorithm 3 ONS-Update

```
Input: $\theta_{t-1,i}, A_{t,i}, \nabla l(\mathbf{x}_{t,i}^{\top}\theta_{t-1,i}, y_{t,i})$
$\theta'_{t,i} = \theta_{t-1,i} - \frac{1}{c_\mu} A_{t,i}^{-1} \nabla l(\mathbf{x}_{t,i}^{\top}\theta_{t-1,i}, y_{t,i})$
$\theta_{t,i} = \arg\min_{\theta \in \mathcal{B}_d(S)} ||\theta - \theta'_{t,i}||_{A_{t,i}}^2$
Output: $\theta_{t,i}$
```

Global update The global update of FedGLB-UCB requires communication among the N clients, which imposes communication cost in two aspects: 1) each global update for federated GLB requires multiple rounds of communication among N clients, i.e., iterative aggregation of local gradients; and 2) global update needs to be performed for multiple times over time horizon T, in order to adapt to the growing dataset collected by each client during bandit learning. Consider a particular time step $t \in [T]$ when global update happens, the distributed optimization objective is:

$$\min_{\theta \in \Theta} F_t(\theta) := \frac{1}{N} \sum_{i=1}^{N} F_{t,i}(\theta) \tag{2.5}$$

where $F_{t,i}(\theta) = \frac{1}{t}\sum_{s=1}^{t} l(\mathbf{x}_{s,i}^{\top}\theta, y_{s,i}) + \frac{\lambda}{2t}||\theta||_2^2$ denotes the *average* regularized negative log-likelihood loss for client $i \in [N]$, and $\lambda > 0$ denotes the regularization parameter. Based on Assumption 4, $\{F_{t,i}(\theta)\}_{i \in [N]}$ are $\frac{\lambda}{Nt}$-strongly-convex and $(k_\mu + \frac{\lambda}{Nt})$-smooth in θ (proof in Section 2.2.7), and we denote the unique minimizer of Eq.(2.5) as $\hat{\theta}_t^{\text{MLE}}$. In this case, it is known that the number of communication rounds J_t required to attain a specified sub-optimality ϵ_t, such that $F_t(\theta) - \min_{\theta \in \Theta} F_t(\theta) \leq \epsilon_t$, has a lower bound $J_t = \Omega\left(\sqrt{(k_\mu N t)/\lambda + 1} \log \frac{1}{\epsilon_t}\right)$ [87], which means J_t increases at least at the rate of $\sqrt{Nt}$. This lower bound is matched by the distributed version of accelerated gradient descent (AGD) [88]:

$$J_t \leq 1 + \sqrt{(k_\mu N t)/\lambda + 1} \log \frac{(k_\mu + \frac{2\lambda}{Nt})||\theta_t^{(1)} - \hat{\theta}_t^{\text{MLE}}||_2^2}{2\epsilon_t} \tag{2.6}$$

where the superscript (i) denotes the i-th iteration of AGD.

In order to minimize the number of communication rounds in one global update, AGD is chosen as the offline regression method for FedGLB-UCB, and its description is given in Algorithm 4 (subscript t is omitted for simplicity). However, other federated/distributed optimization methods can be readily used in place of AGD, as our analysis only requires the convergence result of the adopted method. We should note that ϵ_t is essential to the regret-communication trade-off during the global update at time t: a larger ϵ_t leads to a wider confidence ellipsoid, which increases regret, while a smaller ϵ_t requires more communication rounds J_t, which increases communication cost. In Section 2.2.5, we will discuss the proper choice of ϵ_t to attain desired trade-off between the two conflicting objectives.

To reduce the total number of global updates over time horizon T, we adopt the event-triggered communication from [28], such that global update is triggered if the following event is true for any client $i \in [N]$ (line 8):

$$(t - t_{\text{last}}) \log \frac{\det(A_{t,i})}{\det(A_{t,i} - \Delta A_{t,i})} > D \tag{2.7}$$

Algorithm 4 AGD-Update

```
**Input** : initial $\theta$, number of inner iterations $J$
**Initialization**: set $\theta^{(1)} = \vartheta^{(1)} = \theta$, and define the sequences $\{\upsilon_j := \frac{1+\sqrt{1+4\upsilon_{j-1}^2}}{2}\}_{j\in[J]}$ (with $\upsilon_0 = 0$), and $\{\gamma_j = \frac{1-\upsilon_j}{\upsilon_{j+1}}\}_{j\in[J]}$
**for** $j = 1, 2, \dots, J$ **do**
    **Clients** compute and send local gradient $\{\nabla F_i(\theta^{(j)})\}_{i\in[N]}$ to the server
    **Server** aggregates local gradients $\nabla F(\theta^{(j)}) = \frac{1}{N}\sum_{i=1}^{N} \nabla F_i(\theta^{(j)})$, and execute the following update rule to get $\theta^{(j+1)}$:
        • $\vartheta^{(j+1)} = \theta^{(j)} - \frac{1}{k_\mu + \frac{\lambda}{Nt}} \nabla F(\theta^{(j)})$
        • $\theta^{(j+1)} = (1-\gamma_j)\vartheta^{(j+1)} + \gamma_j \vartheta^{(j)}$
**Output:** $\arg\min_{\theta\in\mathcal{B}_d(S)} \|g_t(\theta^{(J+1)}) - g_t(\theta)\|_{A_t^{-1}}$
```

where $\Delta A_{t,i}$ denotes client i's local update to its covariance matrix since last global update at t_{last}, and $D > 0$ is the chosen threshold for the event-trigger. During the global update, the model estimation $\theta_{t,i}$, covariance matrix $A_{t,i}$ and vector $b_{t,i}$ for all clients $i \in [N]$ will be updated (line 11-14). We should note that the LHS of Eq.(2.7) is essentially an upper bound of the cumulative regret that client i's locally updated model has incurred since t_{last}. Therefore, this event-trigger guarantees that a global update only happens when effective regret reduction is possible.

Arm selection To balance exploration and exploitation during bandit learning, FedGLB-UCB uses the OFUL principle for arm selection [20], which requires the construction of a confidence ellipsoid for each client i. We propose a novel construction of the confidence ellipsoid based on the sequence of model updates that each client i has received up to time t: basically, there are 1) one global update at t_{last}, i.e., the joint offline regression across all clients' accumulated data till t_{last}: $\{(\mathbf{x}_{s,i}, y_{s,i})\}_{s\in[t_{\text{last}}], i\in[N]}$, which resets all clients' local models to $\theta_{t_{\text{last}}}$; and 2) multiple local updates from $t_{\text{last}} + 1$ to t, i.e., the online regression on client i's own data sequence $\{(\mathbf{x}_{s,i}, y_{s,i})\}_{s\in[t_{\text{last}}+1,t]}$ to get $\{\theta_{s,i}\}_{s\in[t_{\text{last}}+1,t]}$ step by step. This can be more easily understood by the illustration in Figure 2.5. The resulting confidence ellipsoid is centered at the ridge regression estimator $\hat{\theta}_{t,i} = A_{t,i}^{-1} b_{t,i}$ [69, 70], which is computed using the predicted rewards given by the past sequence of model updates $\{\theta_{t_{\text{last}}}\} \cup \{\theta_{s,i}\}_{s\in[t_{\text{last}}+1,t]}$ (see the update of $b_{t,i}$ in line 9 and 13 of Algorithm 2). Then at time step t, client i selects the arm that maximizes the UCB score:

$$\mathbf{x}_{t,i} = \arg\max_{\mathbf{x}\in\mathcal{A}_{t,i}} \mathbf{x}^\top \hat{\theta}_{t-1,i} + \alpha_{t-1,i} ||\mathbf{x}||_{A_{t-1,i}^{-1}} \tag{2.8}$$

where $\alpha_{t-1,i}$ is the parameter of the confidence ellipsoid given in Lemma 2.2.2. Note that compared with standard federated/distributed learning where clients only need to communicate gradients for joint model estimation, in our problem, due to the time-varying arm set, it is also necessary to communicate the confidence ellipsoid among clients, i.e., $A_t \in \mathbb{R}^{d\times d}$ and $b_t \in \mathbb{R}^d$ (line 14 in Algorithm 2), as the clients need to be prepared for all possible arms $\mathbf{x} \in \mathbb{R}^d$ that may appear in future for the sake of regret minimization.

2.2.5 Regret and communication cost analysis

In this section, we construct the confidence ellipsoid based on the *offline-and-online* estimators described in Section 2.2.4. Then we analyze the cumulative regret and communication cost of FedGLB-UCB, followed by theoretical comparisons with its different variants.

Construction of confidence ellipsoid Compared with prior works that convert a sequence of online regression estimators to confidence ellipsoid [69, 70], our confidence ellipsoid is built on the combination of an offline regression estimator $\theta_{t_{\text{last}}}$ for global update, and the subsequent online regression estimators $\{\theta_{s-1,i}\}_{s\in[t_{\text{last}}+1,t]}$ for local updates on each client i. This construction is new and requires proof techniques unique to our proposed solution. In the following, we highlight the key steps, and defer the details to Section 2.2.7.

To simplify the use of notations, we assume without loss of generality that the global update at t_{last} is triggered by the N-th client, such that no more new data will be collected at t_{last}, i.e., the first data sample obtained after the global update has index $t_{\text{last}}+1$. We start our construction by considering the following loss difference introduced by the global and local model updates: $\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[l(\mathbf{x}_{s,i}^\top\theta_{t_{\text{last}}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\big] + \sum_{s=t_{\text{last}}+1}^{t}\big[l(\mathbf{x}_{s,i}^\top\theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\big]$, where the first term is the loss difference between the globally updated model $\theta_{t_{\text{last}}}$ and $\theta_\star$, and the second term is between the sequence of locally updated models $\{\theta_{s-1,i}\}_{s\in[t_{\text{last}}+1,t]}$ and $\theta_\star$. This extends the definition of online regret used in the construction in [69, 70]; and due to the existence of offline regression, the obtained upper bounds in Lemma 2.2.1 are unique to our solution.

Lemma 2.2.1 (Upper Bound of Loss Difference)**.** *Denote the sub-optimality of the global model update procedure at time step t_{last} as $\epsilon_{t_{last}}$, such that $F_{t_{last}}(\theta) - \min_{\theta\in\mathcal{B}_d(S)} F_{t_{last}}(\theta) \leq \epsilon_{t_{last}}$. Then under Assumption 4 and 5, we have*

$$\sum_{s=1}^{t_{last}}\sum_{i=1}^{N}\big[l(\mathbf{x}_{s,i}^\top\theta_{t_{last}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\big] \leq B_1 \tag{2.9}$$

where $B_1 = Nt_{last}\epsilon_{t_{last}} + \frac{\lambda}{2}S^2$, and with probability at least $1-\delta$,

$$\sum_{s=t_{last}+1}^{t}\big[l(\mathbf{x}_{s,i}^\top\theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\big] \leq B_2 \tag{2.10}$$

where

$$B_2 = \frac{1}{2c_\mu}\sum_{s=t_{last}+1}^{t}\|\nabla l(\mathbf{x}_{s,i}^\top\theta_{s-1,i}, y_{s,i})\|^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2}\Big[\frac{1}{c_\mu}R_{\max}\sqrt{d\log\left(1+Nt_{last}c_\mu/d\lambda\right)+2\log\left(1/\delta\right)} + 2Nt_{last}\sqrt{\frac{2k_\mu}{\lambda c_\mu}+\frac{2}{Nt_{last}c_\mu}}\sqrt{\epsilon_{t_{last}}} + \sqrt{\frac{\lambda}{c_\mu}}S\Big]^2$$

respectively.

Specifically, B_1 corresponds to the convergence of the offline (distributed) optimization in previous global update; B_2 is essentially the online regret upper bound of ONS, with the major difference that it is initialized using the globally updated model $\theta_{t_{\text{last}}}$, instead of an arbitrary model as in standard ONS. Then due to the c_μ-strongly-convexity of $l(z,y)$ w.r.t. z, i.e., $l(\mathbf{x}_s^\top\theta, y_s) - l(\mathbf{x}_s^\top\theta_\star, y_s) \geq \big[\mu(\mathbf{x}_s^\top\theta_\star) - y_s\big]\mathbf{x}_s^\top(\theta-\theta_\star) + \frac{c_\mu}{2}\big[\mathbf{x}_s^\top(\theta-\theta_\star)\big]^2$, and by rearranging terms in Eq.(2.9) and Eq.(2.10), we have: $\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2 \leq \frac{2}{c_\mu}B_1 + \frac{2}{c_\mu}\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)$, and $\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2 \leq \frac{2}{c_\mu}B_2 + \frac{2}{c_\mu}\sum_{s=t_{\text{last}}+1}^{t}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)$, whose LHS is quadratic in $\theta_\star$. To further upper bound the RHS, we should note that the term $\frac{2}{c_\mu}\sum_{s=t_{\text{last}}+1}^{t}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)$ is standard in [69, 70] as $\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)$ is $\mathcal{F}_{s,i}$-measurable for online estimator $\theta_{s-1,i}$. However, this is not true for the term $\frac{2}{c_\mu}\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)$ as the offline regression estimator $\theta_{t_{\text{last}}}$ depends on all data samples collected till t_{last}; and thus we have to develop a different approach to bound it. This leads to Lemma 2.2.2 below, which provides the confidence ellipsoid for $\theta_\star$.

Lemma 2.2.2 (Confidence Ellipsoid of FedGLB-UCB)**.** *With probability at least $1-2\delta$, for all $t\in[T], i\in[N]$,*

$$\|\hat{\theta}_{t,i}-\theta_\star\|^2_{A_{t,i}} \leq \beta_{t,i} + \frac{\lambda}{c_\mu}S^2 - \|\mathbf{z}_{t,i}\|_2^2 + \hat{\theta}_{t,i}^\top b_{t,i} := \alpha_{t,i}^2$$

where $\mathbf{z}_{t,i} = [\mathbf{x}_{1,1}^\top\theta_{t_{last}}, \mathbf{x}_{1,2}^\top\theta_{t_{last}}, \dots, \mathbf{x}_{t_{last},N-1}^\top\theta_{t_{last}}, \mathbf{x}_{t_{last},N}^\top\theta_{t_{last}},\ \mathbf{x}_{t_{last}+1,i}^\top\theta_{t_{last},i}, \mathbf{x}_{t_{last}+2,i}^\top\theta_{t_{last}+1,i}, \dots, \mathbf{x}_{t,i}^\top\theta_{t-1,i}]^\top$, and $\beta_{t,i} = \frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{last}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big) + B_1 + \frac{4R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{last}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}\Big(\|\theta_{t_{last}}\|_2 + \|\theta_\star\|_2 + \sqrt{B_1}\Big) + \frac{4B_2}{c_\mu} + \frac{8R_{\max}^2}{c_\mu^2}\log\Big(\frac{N}{\delta}\sqrt{4+\frac{8}{c_\mu}B_2+\frac{64R_{\max}^4}{c_\mu^4\cdot 4\delta^2}}\Big) + 1.$

Table 2.2: Comparison between FedGLB-UCB and its variants with different design choices.

Global Upd.	Local Upd.	Setting	R_T	C_T
AGD	ONS	$D = \frac{T}{Nd\log(NT)}$	$\frac{k_\mu(k_\mu+R_{\max})}{c_\mu} d\sqrt{NT}\log(NT)$	$dN^2\sqrt{T}\log^2(NT)$
AGD	no update	$B = \sqrt{NT}$	$\frac{k_\mu R_{\max}}{c_\mu} d\sqrt{NT}\log(NT)$	$N^2T\log(NT)$
AGD	ONS	$B = d^2N\log(NT)$	$\frac{k_\mu(k_\mu+R_{\max})}{c_\mu} d\sqrt{NT}\log(NT)\log(T)$	$d^2N^{2.5}\sqrt{T}\log^2(NT)$
ONS	ONS	$B = \sqrt{NT}$	$\frac{k_\mu(k_\mu+R_{max})}{c_\mu} d(NT)^{3/4}\log(NT)$	$N^{1.5}\sqrt{T}$

Regret and communication cost From Lemma 2.2.2, we can see that $\alpha_{t,i}$ grows at a rate of $Nt_{\text{last}}\sqrt{\epsilon_{t_{\text{last}}}}$ through its dependence on the B_2 term. To make sure the growth rate of $\alpha_{t,i}$ matches that in standard GLB algorithms [68, 70], we set $\epsilon_{t_{\text{last}}} = \frac{1}{N^2 t_{\text{last}}^2}$, which leads to the following corollary.

Corollary 2.2.2.1 (Order of $\beta_{t,i}$)**.** *With* $\epsilon_{t_{last}} = \frac{1}{N^2 t_{last}^2}$*,* $\beta_{t,i} = O\big(\frac{d\log NT}{c_\mu^2}[k_\mu^2 + R_{\max}^2]\big)$*.*

Then using a similar argument as the proof for Theorem 4 of [28], we obtain the following upper bounds on R_T and C_T for FedGLB-UCB (proof in Section 2.2.7).

Theorem 2.2.3 (Regret and Communication Cost Upper Bound of FedGLB-UCB)**.** *Under Assumption 4, 5, and by setting* $\epsilon_t = \frac{1}{N^2t^2}, \forall t$ *and* $D = \frac{T}{Nd\log(NT)}$*, the cumulative regret* R_T *has upper bound*

$$R_T = O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT}\log(NT/\delta)\right),$$

with probability at least $1-2\delta$*. The corresponding communication cost* C_T[1] *has upper bound*

$$C_T = O\Big(dN^2\sqrt{T}\log^2(NT)\Big).$$

Theorem 2.2.3 shows that FedGLB-UCB recovers the standard $O\big(d\sqrt{NT}\log(NT)\big)$ rate in regret as in the centralized setting, while only incurring a communication cost that is sub-linear in T. Note that, to obtain $O\big(d\sqrt{NT}\log(NT)\big)$ regret for federated linear bandit, the DisLinUCB algorithm incurs a communication cost of $O(dN^{1.5}\log(NT))$ [28], which is smaller than that of FedGLB-UCB by a factor of $\sqrt{NT}\log(NT)$. As the frequency of global updates is the same for both algorithms (due to their use of the same event-trigger), this additional communication cost is caused by the iterative optimization procedure for the global update, which is required for GLB model estimation. Moreover, as we mentioned in Section 2.2.4, there is not much room for improvement here as the use of AGD already matches the lower bound up to a logarithmic factor.

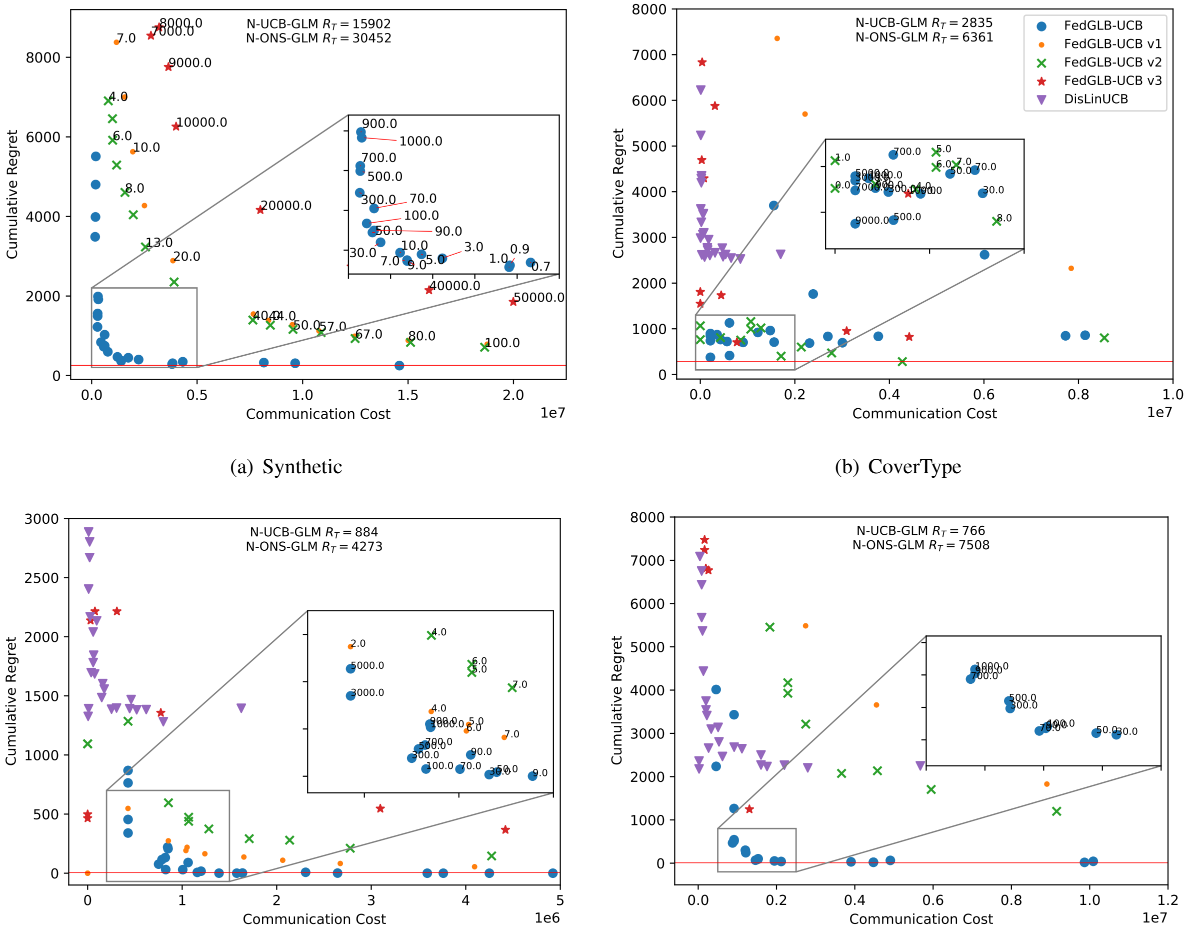
To facilitate the understanding of our algorithm design and investigate the impact of different components of FedGLB-UCB on its regret and communication efficiency trade-off, we propose and analyze three variants, which are also of independent interest, and report the results in Table 2.2. Detailed descriptions, as well as proof for these results can be found in Section 2.2.8. Note that all three variants perform global update according to a fixed schedule $\mathcal{S} = \{t_1 := \lfloor\frac{T}{B}\rfloor, t_2 := 2\lfloor\frac{T}{B}\rfloor, \dots, t_B := B\lfloor\frac{T}{B}\rfloor\}$, where B denotes the total number of global updates specified in advance to trade-off between R_T and C_T, and these variants differ in their global and local update strategies. This comparison demonstrates that our solution is proven to achieve a better regret-communication trade-off against these reasonable alternatives. For example, when using standard federated learning methods (which assume fixed dataset) for streaming data in real-world applications, it is a common practice to set some fixed schedule to periodically retrain the global model to fit the new dataset, and FedGLB-UCB$_1$ implements such behaviors. The design of FedGLB-UCB$_3$ is motivated by distributed online convex optimization that also deals with streaming data in a distributed setting.

2.2.6 Experiment setup & results

We performed extensive empirical evaluations of FedGLB-UCB on both synthetic and real-world datasets, and the results (averaged over 10 runs) are reported in Figure 2.6. We included the three variants of FedGLB-UCB (listed in

[1] This is measured by the *total number of times* data is transferred. Some works [28] measure C_T by the *total number of scalars* transferred, in which case, we have $C_T = O\Big(d^3N^{1.5}\log(NT) + d^2N^2T^{0.5}\log^2(NT)\Big)$.

Table 2.2), One-UCB-GLM, N-UCB-GLM [68] and N-ONS-GLM [70] as baselines, where One-UCB-GLM learns a shared bandit model across all clients, and N-UCB-GLM and N-ONS-GLM learn a separated bandit model for each client with no communication.



(a) Synthetic

(b) CoverType

(c) MagicTelescope

(d) Mushroom

Figure 2.6: Experiment results on synthetic and real world datasets.

Synthetic dataset We simulated the federated GLB setting defined in Section 2.2.3, with $T = 2000, N = 200, d = 10, S = 1$, $\mathcal{A}_t$ ($K = 25$) uniformly sampled from a ℓ_2 unit sphere, and reward $y_{t,,i} \sim \text{Bernoulli}(\mu(\mathbf{x}_{t,,i}^\top \theta_\star))$, with $\mu(z) = (1 + \exp(-z))^{-1}$. To compare the algorithms' R_T and C_T under different trade-off settings, we run FedGLB-UCB with different threshold value D (logarithmically spaced between 10^{-1} and 10^3) and its variants with different number of global updates B. Note that each dot in the result figure illustrates the C_T (x-axis) and R_T (y-axis) that a particular instance of FedGLB-UCB or its variants obtained by time T, and the corresponding value for D or B is labeled next to the dot. R_T of One-UCB-GLM is illustrated as the red horizontal line, and R_T of N-UCB-GLM and N-ONS-GLM are labeled on the top of the figure. We can observe that for FedGLB-UCB and its variants, R_T decreases as C_T increases, interpolating between the two extreme cases: independently learned bandit models by N-UCB-GLM, N-ONS-GLM; and the jointly learned bandit model by One-UCB-GLM. FedGLB-UCB significantly reduces C_T, while attaining low R_T, i.e., its regret is even comparable with One-UCB-GLM that requires at least $C_T = N^2 T$ (8×10^7 in this simulation) for gradient aggregation at each time step.

Real-world dataset The results above demonstrate the effectiveness of FedGLB-UCB when data is generated by a well-specified generalized linear model. To evaluate its performance in a more challenging and practical scenario, we performed experiments using real-world datasets: CoverType, MagicTelescope and Mushroom from the UCI Machine Learning Repository [89]. To convert them to contextual bandit problems, we pre-processed these datasets following the steps in prior works [25], with $T = 2000$ and $N = 20$. Moreover, to demonstrate the advantage of GLB over linear model, we included DisLinUCB [28] as an additional baseline. Since the parameters being communicated in DisLinUCB and FedGLB-UCB are different, to ensure a fair comparison of C_T in this experiment, we measure communication cost (x-axis) by the number of integers or real numbers transferred across the learning system (instead of the frequency of communications). Note that DisLinUCB has no $C_T \geq 3 \times 10^6$in Figure 2.6 because its global update is already happening in every round and cannot be increased further. As mentioned earlier, due to the difference in messages being sent, the communication in DisLinUCB's *per global update* is much smaller than that in FedGLB-UCB. However, because linear models failed to capture the complicated reward mappings in these three datasets, we can see that DisLinUCB is clearly outperformed by FedGLB-UCB and its variants. This shows that, by offering a larger variety of modeling choices, e.g., linear, Poisson, logistic regression, etc., FedGLB-UCB has more potential in dealing with the complicated data in real-world applications.

2.2.7 Full proof of FedGLB-UCB algorithm

Omitted proof in Section 2.2.5

We first need to establish the following lemma.

Lemma 2.2.4 (Confidence Ellipsoid Centered at Global Model)**.** *Consider time step $t \in [T]$ when a global update happens, such that the distributed optimization over N clients is performed to get the globally updated model θ_t. Denote the sub-optimality of the final iteration as ϵ_t, such that $F_t(\theta_t) - F_t(\hat{\theta}_t^{MLE}) \leq \epsilon_t$; then with probability at least $1-\delta$, for all $t \in [T]$,*

$$||\theta_t - \theta_\star||_{A_t} \leq \alpha_t$$

where $\alpha_t = Nt\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Ntc_\mu}}\sqrt{\epsilon_t} + \frac{R_{max}}{c_\mu}\sqrt{d\log\left(1 + Ntc_\mu/(d\lambda)\right) + 2\log\left(1/\delta\right)} + \sqrt{\frac{\lambda}{c_\mu}}S$*, and* $A_t = \frac{\lambda}{c_\mu}I + \sum_{i=1}^N \sum_{s=1}^t \mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top$.

Proof. Recall that the unique minimizer of Eq.(2.5) is denoted as $\hat{\theta}_t^{\text{MLE}}$, so by taking gradient w.r.t. θ we have, $g_t(\hat{\theta}_t^{\text{MLE}}) = \sum_{i=1}^N \sum_{s=1}^t \mathbf{x}_{s,i} y_{s,i}$, where we define $g_t(\theta) = \lambda\theta + \sum_{i=1}^N \sum_{s=1}^t \mu(\mathbf{x}_{s,i}^\top \theta)\mathbf{x}_{s,i}$. First, we start with standard arguments [25, 68] to show that $\|\theta_t - \theta_\star\|_{A_t} \leq \frac{1}{c_\mu}\|g_t(\theta_t) - g_t(\theta_\star)\|_{A_t^{-1}}$. Specifically, by Assumption 4 and the Fundamental Theorem of Calculus, we have

$$g_t(\theta_t) - g_t(\theta_\star) = G_t(\theta_t - \theta_\star)$$

where $G_t = \int_0^1 \nabla g_t(a\theta_t + (1-a)\theta_\star)da$. Again by Assumption 4, $\nabla g_t(\theta) = \lambda I + \sum_{s=1}^t \sum_{i=1}^N \mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top \dot{\mu}(\mathbf{x}_{s,i}^\top \theta)$ is continuous, and $\nabla g_t(\theta) \succcurlyeq \lambda I + c_\mu \sum_{s=1}^t \sum_{i=1}^N \mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top \succ 0$ for $\theta \in \mathcal{B}_d(S)$, so $G_t \succ 0$, i.e., G_t is invertible. Therefore, we have

$$\theta_t - \theta_\star = G_t^{-1}[g_t(\theta_t) - g_t(\theta_\star)]$$

Note that $G_t \succcurlyeq \lambda I + c_\mu \sum_{s=1}^t \sum_{i=1}^N \mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top = c_\mu A_t$, so $G_t^{-1} \preccurlyeq \frac{1}{c_\mu}A_t^{-1}$. Hence,

$$\begin{aligned}\|\theta_t - \theta_\star\|_{A_t} &= \|G_t^{-1}[g_t(\theta_t) - g_t(\theta_\star)]\|_{A_t} \leq \|\frac{1}{c_\mu}A_t^{-1}[g_t(\theta_t) - g_t(\theta_\star)]\|_{A_t} = \frac{1}{c_\mu}\|g_t(\theta_t) - g_t(\theta_\star)\|_{A_t^{-1}} \\ &\leq \frac{1}{c_\mu}\|g_t(\theta_t) - g_t(\hat{\theta}_t^{\text{MLE}})\|_{A_t^{-1}} + \frac{1}{c_\mu}\|g_t(\hat{\theta}_t^{\text{MLE}}) - g_t(\theta_\star)\|_{A_t^{-1}} \qquad (2.11)\end{aligned}$$

where the first term depends on the sub-optimality of the offline regression estimator θ_t to the unique minimizer $\hat{\theta}_t^{(\text{MLE})}$, and the second term is standard for GLB [68].

Recall from Algorithm 4 that $\theta_t = \arg\min_{\theta \in \mathcal{B}_d(S)} \|g_t(\tilde{\theta}_t) - g_t(\theta)\|_{A_t^{-1}}$, where $\tilde{\theta}_t$ denotes the AGD estimator before projection. Therefore, for the first term, using triangle inequality and the definition of $g_t(\cdot)$, we have

$$
\begin{aligned}
&||g_t(\theta_t) - g_t(\hat{\theta}_t^{(\text{MLE})})||_{A_t^{-1}} \leq ||g_t(\theta_t) - g_t(\tilde{\theta}_t)||_{A_t^{-1}} + ||g_t(\tilde{\theta}_t) - g_t(\hat{\theta}_t^{(\text{MLE})})||_{A_t^{-1}} \\
&\leq 2||g_t(\tilde{\theta}_t) - g_t(\hat{\theta}_t^{(\text{MLE})})||_{A_t^{-1}} = 2||\lambda\theta_t + \sum_{s=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{s,i}\mu(\mathbf{x}_{s,i}^\top \theta_t) - \sum_{s=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{s,i} y_{s,i}||_{A_t^{-1}} \\
&= 2||\sum_{s=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{s,i}[-y_{i,s} + \mu(\mathbf{x}_{s,i}^\top \theta_t)] + \lambda\theta_t||_{A_t^{-1}} = 2||Nt\nabla F_t(\theta_t)||_{A_t^{-1}}
\end{aligned}
$$

where the last equality is due to the definition of $F_t(\theta)$ in Eq.(2.5). We can further bound it using the property of Rayleigh quotient and the fact that $A_t \succcurlyeq \frac{\lambda}{c_\mu} I$, which gives us

$$
||g_t(\theta_t) - g_t(\hat{\theta}_t^{(\text{MLE})})||_{A_t^{-1}} \leq \frac{2Nt||\nabla F_t(\theta_t)||_2}{\sqrt{\lambda_{\text{min}}(A_t)}} \leq \frac{2Nt||\nabla F_t(\theta_t)||_2}{\sqrt{\lambda/c_\mu}}
$$

Based on Lemma A.19, $F_t(\theta)$ is $(k_\mu + \frac{\lambda}{Nt})$-smooth, which means

$$
\frac{1}{2k_\mu + 2\lambda/(Nt)} \|\nabla F_t(\theta_t)\|^2 \leq F_t(\theta_t) - F_t(\hat{\theta}_t^{\text{MLE}}) \leq \epsilon_t
$$

where the second inequality is by definition of ϵ_t. Putting everything together, we have the following bound for the first term

$$
\frac{1}{c_\mu} \|g_t(\theta_t) - g_t(\hat{\theta}_t^{\text{MLE}})\|_{A_t^{-1}} \leq 2Nt\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Ntc_\mu}}\sqrt{\epsilon_t}
$$

For the second term, similarly, based on the definition of $g_t(\cdot)$, we have

$$
\begin{aligned}
&\frac{1}{c_\mu}||g_t(\hat{\theta}_t^{\text{MLE}}) - g_t(\theta_\star)||_{A_t^{-1}} \\
&= \frac{1}{c_\mu}||\sum_{s=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{s,i} y_{s,i} - \sum_{s=1}^{t}\sum_{i=1}^{N} \mu(\mathbf{x}_{s,i}^\top \theta_\star)\mathbf{x}_{s,i} - \lambda\theta_\star||_{A_t^{-1}} \\
&\leq \frac{1}{c_\mu}||\sum_{s=1}^{t}\sum_{i=1}^{N} \mathbf{x}_{s,i}\eta_{s,i}||_{A_t^{-1}} + \sqrt{\frac{\lambda}{c_\mu}}S
\end{aligned}
$$

Then based on the self-normalized bound in Lemma A.17 (Theorem 1 of [20]), we have $|| \sum_{s=1}^{t} \sum_{i=1}^{N} \mathbf{x}_{s,i}\eta_{s,i}||_{A_t^{-1}} \leq R_{max}\sqrt{d \log{(1 + Ntc_\mu/d\lambda)} + 2\log{(1/\delta)}}, \forall t$, with probability at least $1-\delta$.

Substituting the upper bounds for these two terms back into Eq.(2.11), we have, with probability at least $1-\delta$,

$$
\begin{aligned}
\|\theta_t - \theta_\star\|_{A_t} &\leq \frac{1}{c_\mu}\|g_t(\theta_t) - g_t(\hat{\theta}_t^{(\text{MLE})})\|_{A_t^{-1}} + \frac{1}{c_\mu}\|g_t(\hat{\theta}_t^{(\text{MLE})}) - g_t(\theta_\star)\|_{A_t^{-1}} \\
&\leq 2Nt\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Ntc_\mu}}\sqrt{\epsilon_t} + \frac{R_{max}}{c_\mu}\sqrt{d\log{(1 + Ntc_\mu/(d\lambda))} + 2\log{(1/\delta)}} + \sqrt{\frac{\lambda}{c_\mu}}S
\end{aligned}
$$

which finishes the proof. □

Proof of Lemma 2.2.1. Denote the two terms for loss difference as $A_1 = \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} \big[l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i})\big]$, and $A_2 = \sum_{s=t_{\text{last}}+1}^{t} \big[l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i})\big]$. We can upper bound the term A_1 by

$$
\begin{aligned}
A_1 &= \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} \big[l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i})\big] - \frac{\lambda}{2}\|\theta_\star\|_2^2 + \frac{\lambda}{2}\|\theta_\star\|_2^2 \\
&\leq \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i}) - \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \hat{\theta}_{t_{\text{last}}}^{\text{MLE}}, y_{s,i}) - \frac{\lambda}{2}\|\hat{\theta}_{t_{\text{last}}}^{\text{MLE}}\|_2^2 + \frac{\lambda}{2}\|\theta_\star\|_2^2 \\
&\leq \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i}) + \frac{\lambda}{2}\|\theta_{t_{\text{last}}}\|_2^2 - \sum_{s=1}^{t_{\text{last}}} \sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \hat{\theta}_{t_{\text{last}}}^{\text{MLE}}, y_{s,i}) - \frac{\lambda}{2}\|\hat{\theta}_{t_{\text{last}}}^{\text{MLE}}\|_2^2 + \frac{\lambda}{2}S^2 \\
&\leq N t_{\text{last}} \epsilon_{t_{\text{last}}} + \frac{\lambda}{2}S^2 := B_1
\end{aligned}
$$

where the first inequality is because $\hat{\theta}_{t_{\text{last}}}^{\text{MLE}}$ minimizes Eq.(2.5), such that $\sum_{s=1}^{t_{\text{last}}} l(\mathbf{x}_{s,i}^\top \hat{\theta}_{t_{\text{last}}}^{\text{MLE}}, y_{s,i}) + \frac{\lambda}{2}\|\hat{\theta}_{t_{\text{last}}}^{\text{MLE}}\|_2^2 \leq \sum_{s=1}^{t_{\text{last}}} l(\mathbf{x}_{s,i}^\top \theta, y_{s,i}) + \frac{\lambda}{2}\|\theta\|_2^2$ for any $\theta \in \mathcal{B}_d(S)$, and the last inequality is because $F_{t_{\text{last}}}(\theta_{t_{\text{last}}}) - F_{t_{\text{last}}}(\hat{\theta}_{t_{\text{last}}}^{\text{MLE}}) \leq \epsilon_{t_{\text{last}}}$ by definition.

Now we start with standard arguments [70, 90] in order to bound the term A_2, which is essentially the online regret of ONS, except that its initial model is the globally updated model $\theta_{t_{\text{last}}}$. First, since $l(z, y)$ is c_μ-strongly-convex w.r.t. z, we have

$$
l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i}) \leq [\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}]\mathbf{x}_{s,i}^\top(\theta_{s-1,i} - \theta_\star) - \frac{c_\mu}{2}||\theta_{s-1,i} - \theta_\star||^2_{\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top} \tag{2.12}
$$

To further bound the RHS of Eq.(2.12), recall from the ONS local update rule in Algorithm 3 that, for each client $i \in [N]$ at the end of each time step $s \in [t_{\text{last}} + 1, t]$,

$$
\begin{aligned}
\theta'_{s,i} &= \theta_{s-1,i} - \frac{1}{c_\mu} A_{s,i}^{-1} \nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) \\
\theta_{s,i} &= \arg\min_{\theta \in \mathcal{B}_d(S)} ||\theta'_{s,i} - \theta||^2_{A_{s,i}}
\end{aligned}
$$

Then due to the property of generalized projection (Lemma 8 of [86]), we have

$$
\begin{aligned}
&||\theta_{s,i} - \theta_\star||^2_{A_{s,i}} \\
&\leq ||\theta_{s-1,i} - \theta_\star - \frac{1}{c_\mu} A_{s,i}^{-1} \nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})||^2_{A_{s,i}} \\
&\leq ||\theta_{s-1,i} - \theta_\star||^2_{A_{s,i}} - \frac{2}{c_\mu} \nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})^\top(\theta_{s-1,i} - \theta_\star) + \frac{1}{c_\mu^2} ||\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})||^2_{A_{s,i}^{-1}}
\end{aligned}
$$

By rearranging terms, we have

$$
\begin{aligned}
&\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})^\top(\theta_{s-1,i} - \theta_\star) \\
&\leq \frac{1}{2c_\mu} ||\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})||^2_{A_{s,j}^{-1}} + \frac{c_\mu}{2}\big(||\theta_{s-1,i} - \theta_\star||^2_{A_{s,i}} - ||\theta_{s,i} - \theta_\star||^2_{A_{s,i}}\big) \\
&= \frac{1}{2c_\mu} ||\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})||^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2} ||\theta_{s-1,i} - \theta_\star||^2_{A_{s-1,i}} \\
&\quad + \frac{c_\mu}{2}\big(||\theta_{s-1,i} - \theta_\star||^2_{A_{s,i}} - ||\theta_{s-1,i} - \theta_\star||^2_{A_{s-1,i}}\big) - \frac{c_\mu}{2} ||\theta_{s,i} - \theta_\star||^2_{A_{s,i}} \\
&= \frac{1}{2c_\mu} ||\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})||^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2} ||\theta_{s-1,i} - \theta_\star||^2_{A_{s-1,i}} + \frac{c_\mu}{2} ||\theta_{s-1,i} - \theta_\star||^2_{\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top} - \frac{c_\mu}{2} ||\theta_{s,i} - \theta_\star||^2_{A_{s,i}}
\end{aligned}
$$

Note that $\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) = \mathbf{x}_{s,i}[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}]$, so with the inequality above, we can further bound the RHS of Eq.(2.12):

$$\begin{aligned}
&l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i}) \leq [\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}]\mathbf{x}_{s,i}^\top(\theta_{s-1,i} - \theta_\star) - \frac{c_\mu}{2}||\theta_{s-1,i} - \theta_\star||^2_{\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top} \\
&\leq \frac{1}{2c_\mu}\|\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})\|^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2}\|\theta_{s-1,i} - \theta_\star\|^2_{A_{s-1,i}} - \frac{c_\mu}{2}\|\theta_{s,i} - \theta_\star\|^2_{A_{s,i}}
\end{aligned}$$

Then summing over $s \in [t_{\text{last}} + 1, t]$, we have

$$A_2 \leq \frac{1}{2c_\mu}\sum_{s=t_{\text{last}}+1}^{t} \|\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})\|^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2}\|\theta_{t_{\text{last}},i} - \theta_\star\|^2_{A_{t_{\text{last}},i}} - \frac{c_\mu}{2}\|\theta_{t,i} - \theta_\star\|^2_{A_{t,i}}$$

where $A_{t_{\text{last}},i} = A_{t_{\text{last}}}, \theta_{t_{\text{last}},i} = \theta_{t_{\text{last}}}, \forall i \in [N]$ due to the global update (line 15 in Algorithm 2).

We should note that the second term above itself essentially corresponds to a confidence ellipsoid centered at the globally updated model $\theta_{t_{\text{last}}}$, and its appearance in the upper bound for the loss difference (online regret) of local updates is because the local update is initialized by $\theta_{t_{\text{last}}}$. And based on Lemma 2.2.4, with probability at least $1 - \delta$,

$$\begin{aligned}
\|\theta_{t_{\text{last}},i} - \theta_\star\|_{A_{t_{\text{last}},i}} \leq 2Nt_{\text{last}}&\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Nt_{\text{last}}c_\mu}}\sqrt{\epsilon_{t_{\text{last}}}} \\
&+ \frac{1}{c_\mu}R_{\max}\sqrt{d\log\left(1 + Nt_{\text{last}}c_\mu/d\lambda\right) + 2\log\left(1/\delta\right)} + \sqrt{\frac{\lambda}{c_\mu}}S
\end{aligned}$$

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

$$\begin{aligned}
A_2 \leq \frac{1}{2c_\mu}&\sum_{s=t_{\text{last}}+1}^{t} \|\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})\|^2_{A_{s,i}^{-1}} + \frac{c_\mu}{2}\Big[2Nt_{\text{last}}\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Nt_{\text{last}}c_\mu}}\sqrt{\epsilon_{t_{\text{last}}}} \\
&+ \frac{1}{c_\mu}R_{\max}\sqrt{d\log\left(1 + Nt_{\text{last}}c_\mu/d\lambda\right) + 2\log\left(1/\delta\right)} + \sqrt{\frac{\lambda}{c_\mu}}S\Big]^2 := B_2
\end{aligned}$$

which finishes the proof for Lemma 2.2.1.

□

Proof of Lemma 2.2.2. Due to c_μ-strongly convexity of $l(z, y)$ w.r.t. z, we have $l(\mathbf{x}_{s,i}^\top \theta, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i}) \geq \big[\mu(\mathbf{x}_{s,i}^\top \theta_\star) - y_{s,i}\big]\mathbf{x}_{s,i}^\top(\theta - \theta_\star) + \frac{c_\mu}{2}\big[\mathbf{x}_{s,i}^\top(\theta - \theta_\star)\big]^2$. Substituting this to the LHS of Eq.(2.9) and Eq.(2.10), we have

$$\begin{aligned}
B_1 &\geq \sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i})\big] \\
&\geq \sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mu(\mathbf{x}_{s,i}^\top \theta_\star) - y_s\big]\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}} - \theta_\star) + \frac{c_\mu}{2}\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}} - \theta_\star)\big]^2 \\
B_2 &\geq \sum_{s=t_{\text{last}}+1}^{t}\big[l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top \theta_\star, y_{s,i})\big] \\
&\geq \sum_{s=t_{\text{last}}+1}^{t}\big[\mu(\mathbf{x}_{s,i}^\top \theta_\star) - y_s\big]\mathbf{x}_{s,i}^\top(\theta_{s-1,i} - \theta_\star) + \frac{c_\mu}{2}\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i} - \theta_\star)\big]^2
\end{aligned}$$

By rearranging the terms, we have

$$\begin{aligned}\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2 &\leq \frac{2}{c_\mu}B_1+\frac{2}{c_\mu}\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\\ \sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2 &\leq \frac{2}{c_\mu}B_2+\frac{2}{c_\mu}\sum_{s=t_{\text{last}}+1}^{t}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\end{aligned}$$

where the LHS is quadratic in $\theta_\star$. For the RHS, we will further upper bound the second term as shown below.

Upper Bound for $\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2$ Note that $\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)$ is $\mathcal{F}_{s,i}$-measurable, and $\eta_{s,i}$ is $\mathcal{F}_{s+1,i}$-measurable and conditionally R_{max}-sub-Gaussian. By applying Lemma A.18 (Corollary 8 of [69]) w.r.t. client i's filtration $\{\mathcal{F}_{s,i}\}_{s=t_{\text{last}}+1}^{\infty}$, where $\mathcal{F}_{s,i}=\sigma\big([\mathbf{x}_{k,j},\eta_{k,j}]_{k,j:k\leq t_{\text{last}}\cap j\leq N},[\mathbf{x}_{k,j},\eta_{k,j}]_{k,j:t_{\text{last}}+1\leq k\leq s-1\cap j=i},\mathbf{x}_{s,i}\big)$, and taking union bound over all $i\in[N]$, with probability at least $1-\delta$, for all $t\in[T],i\in[N]$,

$$\begin{aligned}&\sum_{s=t_{\text{last}}+1}^{t}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\leq\\ &\qquad R_{\max}\sqrt{2\big(1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2\big)\cdot\log\big(\frac{N}{\delta}\sqrt{1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2}\big)}\end{aligned}$$

Therefore,

$$\begin{aligned}1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2 &\leq 1+\frac{2}{c_\mu}B_2\\ &+\frac{2R_{\max}}{c_\mu}\sqrt{2\big(1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2\big)\cdot\log\big(\frac{N}{\delta}\sqrt{1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2}\big)}\end{aligned}\tag{2.13}$$

Then by applying Lemma 2 of [70], i.e., if $q^2\leq a+fq\sqrt{\log(\frac{q}{\delta/N})}$ then $q^2\leq 2a+f^2\log(\frac{\sqrt{4a+f^4/(4\delta^2)}}{\delta/N})$ (for $a,f\geq 0,q\geq 1$). And by setting $q=\sqrt{1+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2}$, $a=1+\frac{2}{c_\mu}B_2$, $f=\frac{2\sqrt{2}R_{\max}}{c_\mu}$, we have

$$\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2\leq 1+\frac{4B_2}{c_\mu}+\frac{8R_{\max}^2}{c_\mu^2}\log\left(\frac{N}{\delta}\sqrt{4+\frac{8}{c_\mu}B_2+\frac{64R_{\max}^4}{c_\mu^4\cdot 4\delta^2}}\right),\forall t,i\tag{2.14}$$

with probability at least $1-\delta$.

Upper Bound for $\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2$ Note that $\theta_{t_{\text{last}}}$ depends on all data samples in $\{(\mathbf{x}_{s,i},y_{s,i})\}_{s\in[t_{\text{last}}]}$ as a result of the offline regression method, and therefore $\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)$ is no longer $\mathcal{F}_{s,i}$-measurable for $s\in[1,t_{\text{last}})$.

Hence, we cannot use Lemma A.18 as before. Instead, we have

$$
\begin{aligned}
&\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star) = \Big(\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}\Big)^\top(\theta_{t_{\text{last}}}-\theta_\star)\\
&= \Big(\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}\Big)^\top\big(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top\big)^{-1}\big(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top\big)(\theta_{t_{\text{last}}}-\theta_\star)\\
&\leq \sqrt{\Big(\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}\Big)^\top\big(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top\big)^{-1}\Big(\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}\Big)\cdot(\theta_{t_{\text{last}}}-\theta_\star)^\top\big(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top\big)(\theta_{t_{\text{last}}}-\theta_\star)}\\
&= \sqrt{\|\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\eta_{s,i}\mathbf{x}_{s,i}\|^2_{(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)^{-1}}\cdot\|\theta_{t_{\text{last}}}-\theta_\star\|^2_{(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}}\\
&\leq R_{\max}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)\cdot\|\theta_{t_{\text{last}}}-\theta_\star\|^2_{(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}},
\end{aligned}
$$

with probability at least $1-\delta$, where the first inequality is due to the matrix-weighted Cauchy-Schwarz inequality in Lemma A.7, such that $x^\top A^{-1}Ay \leq \sqrt{x^\top A^{-1}x\cdot y^\top A^\top A^{-1}Ay} = \sqrt{x^\top A^{-1}x\cdot y^\top Ay}$ for symmetric PD matrix A, and the second inequality is obtained by applying the self-normalized bound in Lemma A.17 w.r.t. the filtration $\{\mathcal{F}_s\}_{s\in\{t_p\}_{p=1}^B}$, where $\mathcal{F}_s = \sigma\big([\mathbf{x}_{k,j},\eta_{k,j}]_{k,j:k\leq s-1\cap j\leq N}, [\mathbf{x}_{k,j},\eta_{k,j}]_{k,j:k=s\cap j\leq N-1}, \mathbf{x}_{s,N}\big)$ and $\{t_p\}_{p=1}^B$ denotes the sequence of time steps when global update happens, and B denotes the total number of global updates.

By substituting it back, we have

$$
\begin{aligned}
&\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2\\
&\leq \frac{2}{c_\mu}B_1 + \frac{2R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)\cdot\|\theta_{t_{\text{last}}}-\theta_\star\|^2_{I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top}}\\
&\leq \frac{2}{c_\mu}B_1 + \frac{2R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)\cdot\Big(\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2+\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2\Big)}
\end{aligned}
\tag{2.15}
$$

Then by setting $z=\sqrt{\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2+\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2}$, $a=\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2+\frac{2}{c_\mu}B_1$, as well as $b=\frac{2R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}$, and using Proposition 9 of [69], i.e. if $z^2\leq a+bz$ then $z\leq b+\sqrt{a}$ (for $a,b\geq 0$), we have

$$
\begin{aligned}
&\sqrt{\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2+\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2}\\
\leq&\frac{2R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}+\sqrt{\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2+B_1}
\end{aligned}
\tag{2.16}
$$

Taking square on both sides, and rearranging terms, we have

$$\begin{aligned}
&\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2\\
\leq&\frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)+B_1\\
&+\frac{4R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}\sqrt{\|\theta_{t_{\text{last}}}-\theta_\star\|_2^2+B_1}\\
\leq&\frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)+B_1\\
&+\frac{4R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}(\|\theta_{t_{\text{last}}}-\theta_\star\|_2+\sqrt{B_1})\\
\leq&\frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)+B_1\\
&+\frac{4R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}(\|\theta_{t_{\text{last}}}\|_2+\|\theta_\star\|_2+\sqrt{B_1})
\end{aligned}\tag{2.17}$$

Now putting everything together, we have the following confidence region for $\theta_\star$,

$$P\big(\forall t,i,\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\big[\mathbf{x}_{s,i}^\top(\theta_{t_{\text{last}}}-\theta_\star)\big]^2+\sum_{s=t_{\text{last}}+1}^{t}\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i}-\theta_\star)\big]^2\leq\beta_{t,i}\big)\geq 1-2\delta \tag{2.18}$$

where $\beta_{t,i}=\frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)+B_1+\frac{4R_{\max}}{c_\mu}\sqrt{2\log\big(\frac{1}{\delta}\sqrt{\det(I+\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\mathbf{x}_{s,i}\mathbf{x}_{s,i}^\top)}\big)}$ $(\|\theta_{t_{\text{last}}}\|_2+\|\theta_\star\|_2+\sqrt{B_1})+1+\frac{4B_2}{c_\mu}+\frac{8R_{\max}^2}{c_\mu^2}\log\big(\frac{N}{\delta}\sqrt{4+\frac{8}{c_\mu}B_2+\frac{64R_{\max}^4}{c_\mu^4\cdot 4\delta^2}}\big)$.

Denote $\mathbf{X}_{t,i}=\begin{bmatrix}\mathbf{x}_{1,1}^\top\\ \cdots\\ \mathbf{x}_{t_{\text{last}},N}^\top\\ \mathbf{x}_{i,t_{\text{last}}+1}^\top\\ \cdots\\ \mathbf{x}_{i,t}^\top\end{bmatrix}\in\mathbb{R}^{(Nt_{\text{last}}+t-t_{\text{last}})\times d}$, and $\mathbf{z}_{t,i}=\begin{bmatrix}\mathbf{x}_{1,1}^\top\theta_{t_{\text{last}}}\\ \cdots\\ \mathbf{x}_{t_{\text{last}},N}^\top\theta_{t_{\text{last}}}\\ \mathbf{x}_{i,t_{\text{last}}+1}^\top\theta_{t_{\text{last}},i}\\ \cdots\\ \mathbf{x}_{i,t}^\top\theta_{t-1,i}\end{bmatrix}\in\mathbb{R}^{Nt_{\text{last}}+t-t_{\text{last}}}$. We can rewrite the inequality above as

$$\begin{aligned}
&\|\mathbf{z}_{t,i}-\mathbf{X}_{t,i}\theta_\star\|_2^2+\frac{\lambda}{c_\mu}\|\theta_\star\|_2^2\leq\beta_{t,i}+\frac{\lambda}{c_\mu}\|\theta_\star\|_2^2\leq\beta_{t,i}+\frac{\lambda}{c_\mu}S^2\\
\Leftrightarrow&\|\mathbf{z}_{t,i}-\mathbf{X}_{t,i}\theta_\star\|_2^2+\frac{\lambda}{c_\mu}\|\theta_\star\|_2^2-\|\mathbf{z}_{t,i}-\mathbf{X}_{t,i}\hat{\theta}_{t,i}\|_2^2-\frac{\lambda}{c_\mu}\|\hat{\theta}_{t,i}\|_2^2+\|\mathbf{z}_{t,i}-\mathbf{X}_{t,i}\hat{\theta}_{t,i}\|_2^2+\frac{\lambda}{c_\mu}\|\hat{\theta}_{t,i}\|_2^2\\
&\leq\beta_{t,i}+\frac{\lambda}{c_\mu}S^2
\end{aligned}$$

where $\hat{\theta}_{t,i}=A_{t,i}^{-1}\mathbf{X}_{t,i}^\top\mathbf{z}_{t,i}$ denotes the Ridge regression estimator based on the predicted rewards given by the past sequence of model updates, and the regularization parameter is $\frac{\lambda}{c_\mu}$. Note that by expanding $\hat{\theta}_{t,i}$, we can show

$\hat{\theta}_{t,i}^\top A_{i,t}\hat{\theta}_{t,i} = \mathbf{z}_{i,t}^\top \mathbf{X}_{i,t}\hat{\theta}_{t,i}$, and $\hat{\theta}_{t,i}^\top A_{i,t}\theta_\star = \mathbf{z}_{i,t}^\top \mathbf{X}_{i,t}\theta_\star$. Therefore, we have

$$\|\hat{\theta}_{t,i} - \theta_\star\|_{A_{t,i}}^2 \leq \beta_{t,i} + \frac{\lambda}{c_\mu}S^2 - (\|\mathbf{z}_{t,i}\|_2^2 - \hat{\theta}_{t,i}^\top \mathbf{X}_{t,i}^\top \mathbf{z}_{t,i})$$

which finishes the proof of Lemma 2.2.2. □

Proof of Corollary 2.2.2.1. Under the condition that $\epsilon_{t_{\text{last}}} \leq \frac{1}{N^2 t_{\text{last}}^2}$,

$$\begin{aligned}
B_1 &\leq \frac{1}{N t_{\text{last}}} + \frac{\lambda}{2}S^2 = O(1) \\
B_2 &\leq \frac{1}{2c_\mu} \sum_{s=t_{\text{last}}+1}^{t} \|\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})\|_{A_{s,i}^{-1}}^2 \\
&\quad + \frac{c_\mu}{2}\Big[2\sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{N t_{\text{last}} c_\mu}} + \frac{1}{c_\mu} R_{\max}\sqrt{d \log\left(1 + N t_{\text{last}} c_\mu / d\lambda\right) + 2\log\left(1/\delta\right)} + \sqrt{\frac{\lambda}{c_\mu}}S\Big]^2
\end{aligned}$$

Note that $\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i}) = \mathbf{x}_{s,i}[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}]$. We can upper bound the squared prediction error by

$$\begin{aligned}
&\big[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}\big]^2 \\
&= \big[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - \mu(\mathbf{x}_{s,i}^\top \theta_\star) - \eta_{s,i}\big]^2 \\
&\leq 2\big[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - \mu(\mathbf{x}_{s,i}^\top \theta_\star)\big]^2 + 2\eta_{s,i}^2 \\
&\leq 2k_\mu^2\big[\mathbf{x}_{s,i}^\top(\theta_{s-1,i} - \theta_\star)\big]^2 + 2\eta_{s,i}^2 \\
&\leq 8k_\mu^2 S^2 + 2\eta_{s,i}^2
\end{aligned}$$

where the first inequality is due to AM-QM inequality, and the second inequality is due to the k_μ-Lipschitz continuity of $\mu(\cdot)$ according to Assumption 4. Since $|\eta_{s,i}| \leq R_{\max}$, $\big[\mu(\mathbf{x}_{s,i}^\top \theta_{s-1,i}) - y_{s,i}\big]^2 \leq k_\mu^2 S^2 + R_{\max}^2$. In addition, due to Lemma 11 of [20], i.e., $\sum_{s=t_{\text{last}}+1}^{t} \|\mathbf{x}_{s,i}\|_{A_{s,i}^{-1}}^2 \leq 2\log(\frac{\det(A_{t,i})}{\det(\lambda I)})$ Therefore,

$$\frac{1}{2c_\mu} \sum_{s=t_{\text{last}}+1}^{t} \|\nabla l(\mathbf{x}_{s,i}^\top \theta_{s-1,i}, y_{s,i})\|_{A_{s,i}^{-1}}^2 = O\big(\frac{d\log NT}{c_\mu}[k_\mu^2 S^2 + R_{\max}^2]\big)$$

so $B_2 = O\big(\frac{d\log NT}{c_\mu}[k_\mu^2 S^2 + R_{\max}^2]\big)$. Hence,

$$\beta_{t,i} = O(d\frac{R_{\max}^2}{c_\mu^2}\log NT + d\frac{k_\mu^2}{c_\mu^2}\log NT + d\frac{R_{\max}^2}{c_\mu^2}\log NT) = O(\frac{d\log NT}{c_\mu^2}[k_\mu^2 + R_{\max}^2])$$

which finishes the proof. □

Proof of Theorem 2.2.3. Since $\mu(\cdot)$ is k_μ-Lipschitz continuous, we have $\mu(\mathbf{x}_{t,\star}^\top \theta_\star) - \mu(\mathbf{x}_{t,i}^\top \theta_\star) \leq k_\mu(\mathbf{x}_{t,\star}^\top \theta_\star - \mathbf{x}_{t,i}^\top \theta_\star)$. Then we have the following upper bound on the instantaneous regret,

$$\begin{aligned}
\frac{r_{t,i}}{k_\mu} &\leq \mathbf{x}_{t,\star}^\top \theta_\star - \mathbf{x}_{t,i}^\top \theta_\star \leq \mathbf{x}_{t,i}^\top \tilde{\theta}_{t-1,i} - \mathbf{x}_{t,i}^\top \theta_\star \\
&= \mathbf{x}_{t,i}^\top(\tilde{\theta}_{t-1,i} - \hat{\theta}_{t-1,i}) + \mathbf{x}_{t,i}^\top(\hat{\theta}_{t-1,i} - \theta_\star) \\
&\leq \|\mathbf{x}_{t,i}\|_{A_{t-1,i}^{-1}} \|\tilde{\theta}_{t-1,i} - \hat{\theta}_{t-1,i}\|_{A_{t-1,i}} + \|\mathbf{x}_{t,i}\|_{A_{t-1,i}^{-1}} \|\hat{\theta}_{t-1,i} - \theta_\star\|_{A_{t-1,i}} \\
&\leq 2\alpha_{t-1,i} \cdot \|\mathbf{x}_{t,i}\|_{A_{t-1,i}^{-1}}
\end{aligned}$$

which holds for all $i \in [N], t \in [T]$, with probability at least $1-2\delta$. And $\tilde{\theta}_{t-1.i}$ denotes the optimistic estimate in the confidence ellipsoid that maximizes the UCB score when client i selects arm at time step t.

Now consider an imaginary centralized agent that has direct access to all clients' data, and we denote its covariance matrix as $\tilde{A}_{t,i} = \frac{\lambda}{c_\mu} I + \sum_{s=1}^{t-1}\sum_{j=1}^{N} \mathbf{x}_{s,j}\mathbf{x}_{s,j} + \sum_{j=1}^{i} \mathbf{x}_{t,j}\mathbf{x}_{t,j}^\top$, i.e., $\tilde{A}_{t,i}$ is immediately updated after any client obtains a new data sample from the environment. Then we can obtain the following upper bound for $r_{t,i}$, which is dependent on the determinant ratio between the covariance matrix of the imaginary centralized agent and that of client i, i.e., $\det(\tilde{A}_{t-1,i})/\det(A_{t-1,i})$.

$$r_{t,i} \leq 2k_\mu \alpha_{t-1,i}\sqrt{\mathbf{x}_{t,i}^\top A_{t-1,i}^{-1}\mathbf{x}_{t,i}} \leq 2k_\mu \alpha_{t-1,i}\sqrt{\mathbf{x}_{t,i}^\top \tilde{A}_{t-1,i}^{-1}\mathbf{x}_{t,i} \cdot \frac{\det(\tilde{A}_{t-1,i})}{\det(A_{t-1,i})}}$$

We refer to the time period in-between two consecutive global updates as an epoch, and denote the total number of epochs as $B \in \mathbb{R}$, i.e., the p-th epoch refers to the period from $t_{p-1}+1$ to t_p, for $p \in [B]$, where t_p denotes the time step when the p-th global update happens. Then the p-th epoch is called a 'good' epoch if the determinant ratio $\frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} \leq 2$, where A_{t_p} is the aggregated sufficient statistics computed at the p-th global update. Otherwise, it is called a 'bad' epoch. In the following, we bound the cumulative regret in 'good' and 'bad' epochs separately.

Suppose the p-th epoch is a good epoch, then for any client $i \in [N]$, and time step $t \in [t_{p-1}+1, t_p]$, we have $\frac{\det(\tilde{A}_{t-1,i})}{\det(A_{t-1,i})} \leq \frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} \leq 2$, because $A_{t-1,i} \succcurlyeq A_{t_{p-1}}$ and $\tilde{A}_{t-1,i} \preccurlyeq A_{t_p}$. Therefore, the instantaneous regret incurred by any client i at any time step t of a good epoch can be bounded by

$$r_{t,i} \leq 2\sqrt{2}k_\mu \alpha_{t-1,i}\sqrt{\mathbf{x}_{t,i}^\top \tilde{A}_{t-1,i}^{-1}\mathbf{x}_{t,i}}$$

with probability at least $1-2\delta$. Therefore, using standard arguments for UCB-type algorithms, e.g., Theorem 2 in [68], the cumulative regret for all the 'good epochs' is

$$\begin{aligned} REG_{good} &\leq 2\sqrt{2}k_\mu \alpha_{t-1,i}\sum_{t=1}^{T}\sum_{i=1}^{N} \|\mathbf{x}_{t,i}\|_{\tilde{A}_{t-1,i}^{-1}} \\ &= O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT}\log NT\right) \end{aligned}$$

which matches the regret upper bound of GLOC [70].

Now suppose the p-th epoch is bad. Then the cumulative regret incurred by all N clients during this 'bad epoch' can be upper bounded by:

$$\begin{aligned} &\sum_{t=t_{p-1}+1}^{t_p}\sum_{i=1}^{N} r_{t,i} \\ &\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}\sqrt{d\log(NT)})\sum_{t=t_{p-1}+1}^{t_p}\sum_{i=1}^{N}\min(1, ||\mathbf{x}_{t,i}||_{A_{t-1,i}^{-1}}) \\ &\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}\sqrt{d\log(NT)})\sum_{i=1}^{N}\sqrt{(t_p - t_{p-1})\log\frac{\det(A_{t_p-1,i})}{\det(A_{t_p-1,i} - \Delta A_{t_p-1,i})}} \\ &\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}N\sqrt{d\log(NT)D}) \end{aligned}$$

where the last inequality is due to the event-trigger design in Algorithm 2. Following the same argument as [28], there can be at most $R = \lceil d\log(1 + \frac{NTc_\mu}{\lambda d})\rceil = O\big(d\log(NT)\big)$ 'bad epochs', because $\det(A_{t_B}) \leq \det(\tilde{A}_{T,N}) \leq (\frac{\lambda}{c_\mu} + \frac{NT}{d})^d$.

Therefore, the cumulative regret for all the ‘bad epochs’ is

$$REG_{bad} = O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d^{1.5} \log^{1.5}(NT) N D^{0.5}\right)$$

Combining the regret upper bound for ‘good’ and ‘bad’ epochs, the cumulative regret

$$R_T = O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}(d\sqrt{NT}\log(NT) + d^{1.5}\log^{1.5}(NT)ND^{0.5})\right).$$

To obtain upper bound for the communication cost C_T, we first upper bound the total number of epochs B. Denote the length of an epoch, i.e., the number of time steps between two consecutive global updates, as $\alpha > 0$, so that there can be at most $\lceil \frac{T}{\alpha} \rceil$ epochs with length longer than α. For a particular epoch p with less than α time steps, we have $t_p - t_{p-1} < \alpha$. Moreover, due to the event-trigger design in Algorithm 2, we have $(t_p - t_{p-1}) \log \frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} > D$, which means $\log \frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} > \frac{D}{\alpha}$. Since $\sum_{p=1}^{B} \log \frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} \leq R$, the number of epochs with less than α time steps is at most $\lceil \frac{R\alpha}{D} \rceil$. Therefore, the total number of epochs.

$$B \leq \lceil \frac{T}{\alpha} \rceil + \lceil \frac{R\alpha}{D} \rceil$$

which is minimized it by choosing $\alpha = \sqrt{\frac{DT}{R}}$, so $B \leq \sqrt{\frac{TR}{D}} = O(d^{0.5}\log^{0.5}(NT)T^{0.5}D^{-0.5})$.

At the end of each epoch, FedGLB-UCB has a global update step that executes AGD among all N clients. As mentioned in Section 2.2.4, the number of iterations required by AGD has upper bound

$$J_t \leq 1 + \sqrt{\frac{k_\mu}{\lambda}Nt + 1} \log \frac{(k_\mu + \frac{2\lambda}{Nt})\|\theta_t^{(1)} - \hat{\theta}_t^{\text{MLE}}\|_2^2}{2\epsilon_t},$$

and under the condition that $\epsilon_t = \frac{1}{N^2 t^2}, \forall t \in [T]$, we have $J_t = O\big(\sqrt{NT}\log(NT)\big), \forall t \in [T]$. Moreover, each iteration of AGD involves communication with N clients, so the communication cost

$$C_T = O(d^{0.5}\log^{1.5}(NT)TN^{1.5}D^{-0.5})$$

In order to match the regret under centralized setting, we set the threshold $D = \frac{T}{Nd\log(NT)}$, which gives us $R_T = O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT}\log(NT))$, and $C_T = O(dN^2\sqrt{T}\log^2(NT))$. □

2.2.8 Full proof of variants of FedGLB-UCB algorithm

In this section, we describe and analyze the variants of FedGLB-UCB listed in Table 2.2. The first variant, FedGLB-UCB$_1$, completely disables local update, and we can see that it requires a linear communication cost in T to attain the $O(d\sqrt{NT}\log(NT))$ regret. As we mentioned in Section 2.2.4, this is because in the absence of local update, FedGLB-UCB$_1$ requires more frequent global updates, i.e., $\sqrt{NT}$ in total, to control the sub-optimality of the employed bandit model w.r.t the growing training set. The second variant, denoted as FedGLB-UCB$_2$, is exactly the same as FedGLB-UCB, except for its fixed communication schedule. This leads to additional $d\sqrt{N}$ global updates, as fixed update schedule cannot adapt to the actual quality of collected data. The third variant, denoted as FedGLB-UCB$_3$, uses ONS for both local and global update, such that only one round of gradient aggregation among N clients is performed for each global update, i.e., lazy ONS update over batched data. It incurs the least communication cost among all variants, but its regret grows at a rate of $(NT)^{3/4}$ due to the inferior quality of its lazy ONS update.

FedGLB-UCB$_1$: scheduled communication + no local update

Though many real-world applications are online problems in nature, i.e., the clients continuously collect new data samples from the users, standard federated/distributed learning methods do not provide a principled solution to adapt to the growing datasets. A common practice is to manually set a fixed global update schedule in advance, i.e., periodically update and deploy the model.

To demonstrate the advantage of FedGLB-UCB over this straightforward solution, we present and analyze the first variant FedGLB-UCB$_1$, which completely disables local update, and performs global update according to a fixed schedule $\mathcal{S} = \{t_1 := \lfloor \frac{T}{B} \rfloor, t_2 := 2\lfloor \frac{T}{B} \rfloor, \dots, t_B := B\lfloor \frac{T}{B} \rfloor\}$, where B is the total number of global updates up to time step T. The description of FedGLB-UCB$_1$ is presented in Algorithm 5.

Algorithm 5 FedGLB-UCB$_1$

```
Input: communication schedule $\mathcal{S}$, regularization parameter $\lambda > 0$, $\delta \in (0,1)$ and $c_\mu$.
Initialize $\forall i \in [N]$: $\theta_{0,i} = \mathbf{0} \in \mathbb{R}^d$, $A_{0,i} = \frac{\lambda}{c_\mu}\mathbf{I} \in \mathbb{R}^{d \times d}$, $\mathbf{X}_{0,i} = \mathbf{0} \in \mathbb{R}^{0 \times d}$, $\mathbf{y}_{0,i} = \mathbf{0} \in \mathbb{R}^0$, $t_{\text{last}} = 0$
for $t = 1, 2, ..., T$ do
  for client $i = 1, 2, ..., N$ do
    Observe arm set $\mathcal{A}_{t,i}$ for client $i$
    Select arm $x_{t,i} \in \mathcal{A}_{t,i}$ according to Eq. (2.19) and observe reward $y_{t,i}$
    Update client $i$: $\mathbf{X}_{t,i} = \begin{bmatrix} \mathbf{X}_{t-1,i} \\ \mathbf{x}_{t,i}^\top \end{bmatrix}$ ,$\mathbf{y}_{t,i} = \begin{bmatrix} \mathbf{y}_{t-1,i} \\ y_{t,i} \end{bmatrix}$
  if $t \notin \mathcal{S}$ then
    Clients: set $\theta_{t,i} = \theta_{t-1,i}, A_{t,i} = A_{t-1,i}, \forall i \in [N]$
  else
    Clients: send $\{\mathbf{X}_{t,i}^\top \mathbf{X}_{t,i}\}_{i \in [N]}$ to server
    Server compute $A_t = \frac{\lambda}{c_\mu}\mathbf{I} + \sum_{i=1}^N \mathbf{X}_{t,i}^\top \mathbf{X}_{t,i}$ and send $A_t$ to all clients.
    Clients: set $A_{t,i} = A_t$, for $i \in [N]$
    Server update global model $\theta_t = $ AGD-Update($\theta_{t_{\text{last}}}, J_t$), and set $t_{\text{last}} = t$
    Clients set local models $\theta_{t,i} = \theta_t, \forall i \in [N]$
```

In FedGLB-UCB$_1$, each client stores a local model $\theta_{t-1,i}$, and the corresponding covariance matrix $A_{t-1,i}$. Note that $\{\theta_{t-1,i}, A_{t-1,i}\}_{i \in [N]}$ are only updated at time steps $t \in \mathcal{S}$, and remain unchanged for $t \notin \mathcal{S}$. At time t, client i selects the arm that maximizes the following UCB score:

$$\mathbf{x}_{t,i} = \underset{\mathbf{x} \in \mathcal{A}_{t,i}}{\arg\max}\, \mathbf{x}^\top \theta_{t-1,i} + \alpha_{t-1,i} ||\mathbf{x}||_{A_{t-1,i}^{-1}} \tag{2.19}$$

where $\alpha_{t-1,i}$ is given in Lemma 2.2.4. The regret and communication cost of FedGLB-UCB$_1$ is given in the following theorem.

Theorem 2.2.5 (Regret and Communication Cost Upper Bound of FedGLB-UCB$_1$)**.** *Under the condition that $\epsilon_t = \frac{1}{N^2 t^2}$, and the total number of global synchronizations $B = \sqrt{NT}$, the cumulative regret R_T has upper bound*

$$R_T = O\left(\frac{k_\mu R_{\max} d}{c_\mu} \sqrt{NT} \log(NT/\delta)\right)$$

with probability at least $1 - \delta$. The cumulative communication cost has upper bound

$$C_T = O(N^2 T \log(NT))$$

Proof. First, based on Lemma 2.2.4 and under the condition that $\epsilon_t = \frac{1}{N^2 t^2}$, we have

$$\|\theta_t - \theta_\star\|_{A_t} \leq \alpha_t$$

holds $\forall t$, where $\alpha_t = \sqrt{\frac{2k_\mu}{\lambda c_\mu} + \frac{2}{Ntc_\mu}} + \frac{R_{max}}{c_\mu}\sqrt{d \log\left(1 + Ntc_\mu/(d\lambda)\right) + 2\log\left(1/\delta\right)} + \sqrt{\frac{\lambda}{c_\mu}}S = O(\frac{R_{max}}{c_\mu}\sqrt{d\log(Nt)})$, which matches the order in [68].

Similar to the proof of Theorem 2.2.3, we decompose all B epochs into 'good' and 'bad' epochs according to the log-determinant ratio: the p-th epoch, for $p \in [B]$, is a 'good' epoch if the determinant ratio $\frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} \leq 2$. Otherwise, it is a 'bad' epoch. In the following, we bound the cumulative regret in 'good' and 'bad' epochs separately.

Suppose epoch p is a good epoch, then for any client $i \in [N]$, and time step $t \in [t_{p-1}+1, t_p]$, we have $\frac{\det(\tilde{A}_{t-1,i})}{\det(A_{t-1,i})} \leq \frac{\det(A_{t_p})}{\det(A_{t_{p-1}})} \leq 2$, because $A_{t-1,i} = A_{t_{p-1}}$ and $\tilde{A}_{t-1,i} \preccurlyeq A_{t_p}$. Therefore, the instantaneous regret incurred by any client i at any time step t of a good epoch p can be bounded by

$$\begin{aligned} r_{t,i} &\leq 2k_\mu \alpha_{t_{p-1}} \sqrt{\mathbf{x}_{t,i}^\top A_{t-1,i} \mathbf{x}_{t,i}} \leq 2k_\mu \alpha_{t_{p-1}} \sqrt{\mathbf{x}_{t,i}^\top A_{t-1}^{-1} \mathbf{x}_{t,i} \cdot \frac{\det(\tilde{A}_{t-1,i})}{\det(A_{t-1,i})}} \\ &\leq 2\sqrt{2} k_\mu \alpha_T \sqrt{\mathbf{x}_{t,i}^\top A_{t-1}^{-1} \mathbf{x}_{t,i}} \end{aligned}$$

By standard arguments [20, 68], the cumulative regret in all good epochs is bounded by $O(\frac{k_\mu R_{\max}}{c_\mu} d\sqrt{NT} \log(NT/\delta))$ with probability at least $1-\delta$.

By Assumption 1, $\mu(\cdot)$ is Lipschitz continuous with constant k_μ, i.e., $|\mu(\mathbf{x}^\top \theta_1) - \mu(\mathbf{x}^\top \theta_2)| \leq k_\mu |\mathbf{x}^\top(\theta_1 - \theta_2)|$, so the instantaneous regret $r_{t,i}$ is uniformly bounded $\forall t \in [T], i \in [N]$ by $2k_\mu S$. Now suppose epoch p is bad, then we can upper bound the cumulative regret in this bad epoch by $2k_\mu S \frac{NT}{B}$, where $\frac{NT}{B}$ is the number of time steps in each epoch. Since there can be at most $O(d \log NT)$ bad epochs, the cumulative regret incurred in all bad epochs can be upper bounded by $O(\frac{NT}{B} k_\mu S d \log(NT))$. Combining both parts together, the cumulative regret upper bound is

$$R_T = O\left(\frac{NT}{B} k_\mu S d \log(NT) + \frac{k_\mu R_{\max} d}{c_\mu} \sqrt{NT} \log(NT)\right)$$

To recover the regret under centralized setting, we set $B = \sqrt{NT}$, so

$$R_T = O\left(\frac{k_\mu R_{max}}{c_\mu} d\sqrt{NT} \log(NT)\right)$$

Note that FedGLB-UCB$_1$ has $B = \sqrt{NT}$ global updates in total, and during each global update, there are J_t rounds of communications, for $t \in \mathcal{S}$. As mentioned earlier, for AGD to attain $\epsilon_t = \frac{1}{N^2 t^2}$ sub-optimality, the required number of inner iterations

$$J_t \leq 1 + \sqrt{\frac{k_\mu + \frac{\lambda}{Nt}}{\frac{\lambda}{Nt}}} \log \frac{(k_\mu + \frac{\lambda}{Nt} + \frac{\lambda}{Nt}) \|\theta_t^{(0)} - \hat{\theta}_t^{\text{MLE}}\|_2^2}{2\epsilon_t} = O\left(\sqrt{Nt} \log(Nt)\right)$$

Therefore, the communication cost over time horizon T is

$$\begin{aligned} C_T &= N \cdot \sum_{t \in \mathcal{S}} J_t \\ &= N \cdot \left[\sqrt{\sqrt{NT}} \log(\sqrt{NT}) + \sqrt{2\sqrt{NT}} \log(2\sqrt{NT}) + \cdots + \sqrt{\sqrt{NT} \cdot \sqrt{NT}} \log(\sqrt{NT} \cdot \sqrt{NT})\right] \\ &\leq N^{5/4} T^{1/4} \log(NT) \left[\sqrt{1} + \sqrt{2} + \cdots + \sqrt{\sqrt{NT}}\right] \\ &\leq N^{5/4} T^{1/4} \log(NT) \cdot \frac{3}{2} (\sqrt{NT} + \frac{1}{2})^{3/2} \\ &= O(N^2 T \log(NT)) \end{aligned}$$

which finishes the proof. □

FedGLB-UCB$_2$: scheduled communication

For the second variant FedGLB-UCB$_2$, we enabled local update on top of FedGLB-UCB$_1$. Therefore, compared with the original algorithm FedGLB-UCB, the only difference is that FedGLB-UCB$_2$ uses scheduled communication instead of event-triggered communication. Its description is given in Algorithm 6.

Algorithm 6 FedGLB-UCB$_2$

```
Input: communication schedule $\mathcal{S}$, regularization parameter $\lambda > 0$, $\delta \in (0,1)$ and $c_\mu$.
Initialize $\forall i \in [N]$: $A_{0,i} = \frac{\lambda}{c_\mu}\mathbf{I} \in \mathbb{R}^{d \times d}, b_{0,i} = \mathbf{0} \in \mathbb{R}^d, \theta_{0,i} = \mathbf{0} \in \mathbb{R}^d, \Delta A_{0,i} = \mathbf{0} \in \mathbb{R}^{d \times d}$; $A_0 = \frac{\lambda}{c_\mu}\mathbf{I} \in \mathbb{R}^{d \times d}, b_0 = \mathbf{0} \in \mathbb{R}^d, \theta_0 = \mathbf{0} \in \mathbb{R}^d$, $t_{\text{last}} = 0$
for $t = 1, 2, ..., T$ do
    for client $i = 1, 2, ..., N$ do
        Observe arm set $\mathcal{A}_{t,i}$ for client $i$
        Select arm $\mathbf{x}_{t,i} \in \mathcal{A}_{t,i}$ by Eq.(2.8), and observe reward $y_{t,i}$
        Update client $i$: $A_{t,i} = A_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top$, $\Delta A_{t,i} = \Delta A_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top$
    if $t \notin \mathcal{S}$ then
        Clients $\forall i \in [N]$: $\theta_{t,i} = \text{ONS-Update}(\theta_{t-1,i}, A_{t,i}, \nabla l(\mathbf{x}_{t,i}^\top \theta_{t-1,i}, y_{t,i}))$, $b_{t,i} = b_{t-1,i} + \mathbf{x}_{t,i}\mathbf{x}_{t,i}^\top \theta_{t-1,i}$
    else
        Clients $\forall i \in [N]$: send $\Delta A_{t,i}$ to server, and reset $\Delta A_{t,i} = \mathbf{0}$
        Server compute $A_t = A_{t_{\text{last}}} + \sum_{i=1}^N \Delta A_{t,i}$
        Server perform global model update $\theta_t = \text{AGD-Update}(\theta_{t_{\text{last}}}, J_t)$ (see Eq.(2.6) for choice of $J_t$), $b_t = b_{t_{\text{last}}} + \sum_{i=1}^N \Delta A_{t,i}\theta_t$, and set $t_{\text{last}} = t$
        Clients $\forall i \in [N]$: set $\theta_{t,i} = \theta_t, A_{t,i} = A_t, b_{t,i} = b_t$
```

The regret and communication cost of FedGLB-UCB$_2$ is given in the following theorem.

Theorem 2.2.6 (Regret and Communication Cost Upper Bound of FedGLB-UCB$_2$). *Under the condition that $\epsilon_t = \frac{1}{N^2 t^2}$, and the total number of global synchronizations $B = d^2 N \log(NT)$, the cumulative regret R_T has upper bound*

$$R_T = O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT}\log(NT/\delta)\sqrt{\log\frac{T}{d^2 N \log NT}}\right)$$

with probability at least $1-\delta$. The cumulative communication cost has upper bound

$$C_T = O(d^2 N^{2.5}\sqrt{T}\log^2(NT))$$

Proof. Compared with the analysis for FedGLB-UCB, the main difference in the analysis for FedGLB-UCB$_2$ is how we bound the regret incurred in 'bad epochs'. Using the same argument, the cumulative regret for the 'good epochs' is $REG_{good} = O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT}\log NT/\delta)$.

Now consider a particular bad epoch $p \in [B]$. Then the cumulative regret incurred by all N clients during this 'bad epoch' can be upper bounded by:

$$\begin{aligned}
&\sum_{t=t_{p-1}+1}^{t_p}\sum_{i=1}^{N} r_{t,i} \\
&\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}\sqrt{d\log(NT/\delta)})\sum_{t=t_{p-1}+1}^{t_p}\sum_{i=1}^{N}\min(1, ||\mathbf{x}_{t,i}||_{A_{t-1,i}^{-1}}) \\
&\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu}\sqrt{d\log(NT/\delta)})\sum_{i=1}^{N}\sqrt{(t_p - t_{p-1})\log\frac{\det(A_{t_p-1,i})}{\det(A_{t_p-1,i} - \Delta A_{t_p-1,i})}} \\
&\leq O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} dN\sqrt{\log(NT/\delta)}\sqrt{\frac{T}{B}\log(\frac{T}{B}}))
\end{aligned}$$

where the last inequality is because all epochs has length $\frac{T}{B}$ as defined by $\mathcal{S}$. Again, since there can be at most $O(d \log NT)$ 'bad epochs', the cumulative regret for the 'bad epochs' is upper bounded by

$$REG_{bad} = O(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d^2 \log^{1.5}(NT/\delta) N \sqrt{\frac{T}{B} \log(\frac{T}{B})}).$$

Combining the cumulative regret for both 'good' and 'bad' epochs, and setting $B = d^2 N \log(NT)$, we have

$$R_T = O\left(\frac{k_\mu(k_\mu + R_{\max})}{c_\mu} d\sqrt{NT} \log(NT/\delta) \sqrt{\log(\frac{T}{d^2 N \log NT})}\right)$$

Now that FedGLB-UCB$_2$ has $B = d^2 N \log(NT)$ global updates in total, and during each global update, there are $J_t = O(\sqrt{NT} \log(NT))$ rounds of communications, for $t \in \mathcal{S}$. Therefore, the communication cost over time horizon T is

$$\begin{aligned} C_T &= N \cdot \sum_{t \in \mathcal{S}} J_t = O(N \cdot d^2 N \log(NT) \cdot \sqrt{NT} \log(NT)) \\ &= O(d^2 N^{2.5} \sqrt{T} \log^2(NT)) \end{aligned}$$

which finishes the proof. □

FedGLB-UCB$_3$: scheduled communication + ONS for global update

The previous two variants both adopt iterative optimization method, i.e., AGD, for the global update, which introduces a $\sqrt{NT}\log(NT)$ factor in the communication cost. In this section, we try to avoid this by studying the third variant FedGLB-UCB$_3$ that adopts ONS for both local and global update, such that only one step of ONS is performed (based on all new data samples N clients collected in this epoch). It can be viewed as the ONS-GLM algorithm [70] with lazy batch update.

Algorithm 7 FedGLB-UCB$_3$

```
Input: communication schedule $\mathcal{S}$, regularization parameter $\lambda > 0$, $\delta \in (0,1)$ and $c_\mu$
Initialize $\forall i \in [N]$: $\theta_{0,i} = \mathbf{0} \in \mathbb{R}^d$, $A_{0,i} = \lambda \mathbf{I} \in \mathbb{R}^{d \times d}$, $V_{0,i} = \lambda \mathbf{I} \in \mathbb{R}^{d \times d}$, $b_{0,i} = \mathbf{0} \in \mathbb{R}^d$; $\theta_0 = \mathbf{0} \in \mathbb{R}^d$, $A_0 = \lambda \mathbf{I} \in \mathbb{R}^{d \times d}$, $V_0 = \lambda \mathbf{I} \in \mathbb{R}^{d \times d}$, $b_0 = \mathbf{0} \in \mathbb{R}^d$, $t_{\text{last}} = 0$
for $t = 1, 2, ..., T$ do
  for client $i = 1, 2, ..., N$ do
    Observe arm set $\mathcal{A}_{t,i}$ for client $i \in [N]$
    Select arm $\mathbf{x}_{t,i} = \arg\max_{\mathbf{x} \in \mathcal{A}_{t,i}} \mathbf{x}^\top \hat{\theta}_{t-1,i} + \alpha_{t-1,i} \|\mathbf{x}\|_{V_{t-1,i}^{-1}}$, where $\hat{\theta}_{t-1,i} = V_{t-1,i}^{-1} b_{t-1,i}$ and $\alpha_{t-1,i}$ is given in Lemma 2.2.8; and then observe reward $y_{t,i}$
    Compute loss $l(z_{t,i}, y_{t,i})$, where $z_{t,i} = \mathbf{x}_{t,i}^\top \theta_{t-1,i}$
    Update client $i$: $A_{t,i} = A_{t-1,i} + \nabla l(z_{t,i}, y_{t,i}) \nabla l(z_{t,i}, y_{t,i})^\top$, $V_{t,i} = V_{t-1,i} + \mathbf{x}_{t,i} \mathbf{x}_{t,i}^\top$
  if $t \notin \mathcal{S}$ then
    Clients $\forall i \in [N]$: $\theta_{t,i} = \text{ONS-Update}(\theta_{t-1,i}, A_{t,i}, \nabla l(z_{t,i}, y_{t,i}))$, $b_{t,i} = b_{t-1,i} + \mathbf{x}_{t,i} z_{t,i}$
  else
    Clients $\forall i \in [N]$: send gradient $\nabla F_{t,i}(\theta_{t_{\text{last}}}) = \sum_{s=t_{\text{last}}+1}^{t} \nabla l(\mathbf{x}_{s,i}^\top \theta_{t_{\text{last}}}, y_{s,i})$ and $\Delta V_{t,i} = V_{t,i} - V_{t_{\text{last}},i}$ to server
    Server $A_t = A_{t_{\text{last}}} + (\sum_{i=1}^N \nabla F_{t,i}(\theta_{t_{\text{last}}}))(\sum_{i=1}^N \nabla F_{t,i}(\theta_{t_{\text{last}}}))^\top$, $V_t = V_{t_{\text{last}}} + \sum_{i=1}^N \Delta V_{t,i}$, $b_t = b_{t_{\text{last}}} + \sum_{i=1}^N \Delta V_{t,i} \theta_{t_{\text{last}}}$, $\theta_t = \text{ONS-Update}(\theta_{t_{\text{last}}}, A_t, \sum_{i=1}^N \nabla F_{t,i}(\theta_{t_{\text{last}}}))$
    Clients $\forall i \in [N]$: $\theta_{t,i} = \theta_t$, $A_{t,i} = A_t$, $V_t = V_t$, $b_{t,i} = b_t$
    Set $t_{\text{last}} = t$
```

Recall that the update schedule is denoted as $\mathcal{S} = \{t_1 := \lfloor \frac{T}{B} \rfloor, t_2 := 2\lfloor \frac{T}{B} \rfloor, \dots, t_q := q\lfloor \frac{T}{B} \rfloor, \dots, t_B := B\lfloor \frac{T}{B} \rfloor\}$, where B denotes the total number of global updates up to T. Compared with [70], the main difference in our construction is that the loss function in the online regression problem may contain multiple data samples, i.e., for global update, or one single data sample, i.e., for local update. Then for a client $i \in [N]$ at time step t (suppose t is in the $(q+1)$-th epoch, so $t \in [t_q + 1, t_{q+1}]$), the sequence of loss functions observed by the online regression estimator till time t is:

$$\underbrace{\sum_{s=1}^{t_1}\sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \theta_0, y_{s,i}), \sum_{s=t_1+1}^{t_2}\sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \theta_{t_1}, y_{s,i}), \dots, \sum_{s=t_{q-1}+1}^{t_q}\sum_{i=1}^{N} l(\mathbf{x}_{s,i}^\top \theta_{t_{q-1}}, y_{s,i})}_{\text{global updates at } t_1, t_2, \dots, t_q}, \underbrace{l(\mathbf{x}_{t_q+1,i}^\top \theta_{t_q}, y_{t_q+1,i}), \dots, l(\mathbf{x}_{t,i}^\top \theta_{t-1,i}, y_{t,i})}_{\text{local updates at } t_q+1, \dots, t}$$

We can see that the first q terms correspond to the global ONS updates that are computed using the whole batch of data collected by N clients in each epoch, and the remaining $t - t_q$ terms are local ONS updates that are computed using each new data sample collected by client i in the $(q+1)$-th epoch.

To facilitate further analysis, we introduce a new set of indices for the data samples, so that we can unify the notations for the loss functions above. Imagine all the arm pulls are performed by an imaginary centralized agent, such that, in each time step $t \in [T]$, it pulls an arm for clients $1, 2, \dots, N$ one by one. Therefore, the sequence of data sample obtained by this imaginary agent can be denoted as $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_s, y_s), \dots, (\mathbf{x}_{NT}, y_{NT})$. Moreover, we denote n_p as the total number of data samples collected by all N clients till the p-th ONS update (including both global and local ONS update), and denote the updated model as θ_p, for $p \in [P]$. Note that P denotes the total number of updates up to time t (total number of terms in the sequence above), such that $P = q + t - t_q$. Then this sequence of loss functions can be rewritten as:

$$\underbrace{F_1(\theta_0), F_2(\theta_1), \dots, F_q(\theta_{q-1})}_{\text{global updates}}, \underbrace{F_{q+1}(\theta_q), \dots, F_P(\theta_{P-1})}_{\text{local updates}}$$

where $F_p(\theta_{p-1}) = \sum_{s=n_{p-1}+1}^{n_p} l(\mathbf{x}_s^\top \theta_{p-1}, y_s)$, for $p \in [P]$.

Online regret upper bound for lazily-updated ONS To construct the confidence ellipsoid based on this sequence of global and local ONS updates, we first need to upper bound the online regret that ONS incurs on this sequence of loss functions, which is given in Lemma 2.2.7.

Lemma 2.2.7 (Online regret upper bound). *Under the condition that the learning rate of ONS is set to* $\gamma = \frac{1}{2}\min(\frac{1}{4S\sqrt{k_\mu^2 S^2 + R_{\max}^2}}, \frac{c_\mu}{(k_\mu^2 S^2 + R_{\max}^2)\max_{p \in [P]}(n_p - n_{p-1})})$, *then the cumulative online regret over P steps*

$$\sum_{p=1}^{P} F_p(\theta_{p-1}) - F_p(\theta_\star) \leq B_P$$

where $B_P = \frac{1}{2\gamma}\sum_{p=1}^{P} ||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + 2\gamma\lambda S^2$.

Proof of Lemma 2.2.7. Recall from the proof of Corollary 2.2.2.1 that $|\mu(\mathbf{x}_s^\top \theta_{p-1}) - y_s| \leq \sqrt{k_\mu^2 S^2 + R_{\max}^2} := G, \forall s$. First, we need to show that $F_p(\theta_{p-1}) = \sum_{s=n_{p-1}+1}^{n_p} l(\mathbf{x}_s^\top \theta_{p-1}, y_s)$ is $\frac{c_\mu}{(n_p - n_{p-1})G^2}$-exp-concave, or equivalently, $\nabla^2 F_p(\theta_{p-1}) \succcurlyeq \frac{c_\mu}{(n_p - n_{p-1})G^2}\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top$ (Lemma 4.2 in [91]). Taking first and second order derivative of $F_p(\theta_{p-1})$ w.r.t. θ_{p-1}, we have

$$\nabla F_p(\theta_{p-1}) = \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s[-y_s + \mu(\mathbf{x}_s^\top \theta_{p-1})] = \mathbf{X}_p^\top[\mu(\mathbf{X}_p \theta_{p-1}) - \mathbf{y}_p],$$

$$\nabla^2 F_p(\theta_{p-1}) = \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s \mathbf{x}_s^\top \dot{\mu}(\mathbf{x}_s^\top \theta_{p-1})$$

where $\mathbf{X}_p = [\mathbf{x}_{n_{p-1}+1}, \mathbf{x}_{n_{p-1}+2}, \dots, \mathbf{x}_{n_p}]^\top \in \mathbb{R}^{(n_p - n_{p-1}) \times d}$, and $\mathbf{y}_p = [y_{n_{p-1}+1}, y_{n_{p-1}+2}, \dots, y_{n_p}]^\top \in \mathbb{R}^{n_p - n_{p-1}}$. Then due to Assumption 1, we have $\nabla^2 F_p(\theta_{p-1}) \succcurlyeq c_\mu \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s \mathbf{x}_s^\top = c_\mu \mathbf{X}_p^\top \mathbf{X}_p$. For any vector $u \in \mathbb{R}^d$, we can show that,

$$
\begin{aligned}
& u^\top \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top u \\
& = u^\top \mathbf{X}_p^\top [\mu(\mathbf{X}_p \theta_{p-1}) - \mathbf{y}_p][\mu(\mathbf{X}_p \theta_{p-1}) - \mathbf{y}_p]^\top \mathbf{X}_p u \\
& = \left[(\mathbf{X}_p u)^\top [\mu(\mathbf{X}_p \theta_{p-1}) - \mathbf{y}_p]\right]^2 \\
& \le \|\mathbf{X}_p u\|_2^2 \cdot \|\mu(\mathbf{X}_p \theta_{p-1}) - \mathbf{y}_p\|_2^2 \\
& \le u^\top \mathbf{X}_p^\top \mathbf{X}_p u \cdot (n_p - n_{p-1}) G^2
\end{aligned}
$$

where the first inequality is due to Cauchy-Schwarz inequality, and the second inequality is because $\|\mu(\mathbf{X}_p\theta_{p-1}) - \mathbf{y}_p\|_2^2 = \sum_{s=n_{p-1}+1}^{n_p}[-y_s+\mu(\mathbf{x}_s^\top\theta_{p-1})]^2 \le (n_p-n_{p-1})G^2$. Therefore, $\mathbf{X}_p^\top \mathbf{X}_p \succcurlyeq \frac{1}{(n_p-n_{p-1})G^2}\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top$, which gives us

$$
\nabla^2 F_p(\theta_{p-1}) \succcurlyeq \frac{c_\mu}{(n_p - n_{p-1})G^2} \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top
$$

Then due to Lemma 4.3 of [91], under the condition that $\gamma_p \le \frac{1}{2}\min(\frac{1}{4GS}, \frac{c_\mu}{(n_p-n_{p-1})G^2})$, we have

$$
\begin{aligned}
& F_p(\theta_{p-1}) - F_p(\theta_\star) \\
& \quad \le \nabla F_p(\theta_{p-1})^\top (\theta_{p-1} - \theta_\star) - \frac{\gamma_p}{2} (\theta_{p-1} - \theta_\star)^\top \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top (\theta_{p-1} - \theta_\star)
\end{aligned} \tag{2.20}
$$

Then we start to upper bound the RHS of the inequality above. Recall that the ONS update rule is:

$$
\begin{aligned}
\theta'_p &= \theta_{p-1} - \frac{1}{\gamma} A_p^{-1} \nabla F_p(\theta_{p-1}) \\
\theta_p &= \arg\min_{\theta \in \Theta} ||\theta'_p - \theta||_{A_p}^2
\end{aligned}
$$

where $A_p = \sum_{\rho=1}^{p} \nabla F_\rho(\theta_{\rho-1}) \nabla F_\rho(\theta_{\rho-1})^\top$, and γ is set to $\min_{p\in[P]} \gamma_p = \frac{1}{2}\min(\frac{1}{4GS}, \frac{c_\mu}{G^2 \max_{p\in[P]}(n_p - n_{p-1})})$. So we have

$$
\theta'_p - \theta_\star = \theta_{p-1} - \theta_\star - \frac{1}{\gamma} A_p^{-1} \nabla F_p(\theta_{p-1})
$$

Then due to the property of the generalized projection, and by substituting into the update rule, we have

$$
||\theta_p - \theta_\star||_{A_p}^2 \le ||\theta'_p - \theta_\star||_{A_p}^2 \le ||\theta_{p-1} - \theta_\star||_{A_p}^2 - \frac{2}{\gamma} (\theta_{p-1} - \theta_\star)^\top \nabla F_p(\theta_{p-1}) + \frac{1}{\gamma^2} ||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2
$$

By rearranging terms,

$$
\nabla F_p(\theta_{p-1})^\top (\theta_{p-1} - \theta_\star) \le \frac{1}{2\gamma} ||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + \frac{\gamma}{2} \big(||\theta_{p-1} - \theta_\star||_{A_p}^2 - ||\theta_p - \theta_\star||_{A_p}^2 \big)
$$

After summing over P steps, we have

$$
\sum_{p=1}^{P} \nabla F_p(\theta_{p-1})^\top (\theta_{p-1} - \theta_\star) \le \frac{1}{2\gamma} \sum_{p=1}^{P} ||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + \frac{\gamma}{2} \sum_{p=1}^{P} \big(||\theta_{p-1} - \theta_\star||_{A_p}^2 - ||\theta_p - \theta_\star||_{A_p}^2 \big)
$$

The second term can be simplified,

$$\begin{aligned}
&\sum_{p=1}^{P}\big(||\theta_{p-1}-\theta_\star||_{A_p}^2 - ||\theta_p-\theta_\star||_{A_p}^2\big)\\
&= ||\theta_0-\theta_\star||_{A_1}^2 + \sum_{p=2}^{P}\big(||\theta_{p-1}-\theta_\star||_{A_p}^2 - ||\theta_{p-1}-\theta_\star||_{A_{p-1}}^2\big) - ||\theta_P-\theta_\star||_{A_P}^2\\
&\leq ||\theta_0-\theta_\star||_{A_1}^2 + \sum_{p=2}^{P}\big(||\theta_{p-1}-\theta_\star||_{A_p}^2 - ||\theta_{p-1}-\theta_\star||_{A_{p-1}}^2\big)\\
&= ||\theta_0-\theta_\star||_{A_1}^2 + \sum_{p=2}^{P}||\theta_{p-1}-\theta_\star||_{\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top}^2\\
&= ||\theta_0-\theta_\star||_{A_1}^2 + \sum_{p=1}^{P}||\theta_{p-1}-\theta_\star||_{\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top}^2 - ||\theta_0-\theta_\star||_{\nabla F_1(\theta_0)\nabla F_1(\theta_0)^\top}^2\\
&= 4\lambda S^2 + \sum_{p=1}^{P}||\theta_{p-1}-\theta_\star||_{\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top}^2
\end{aligned}$$

which leads to

$$\begin{aligned}
\sum_{p=1}^{P}\nabla F_p(\theta_{p-1})^\top(\theta_{p-1}-\theta_\star) \leq \frac{1}{2\gamma}\sum_{p=1}^{P}||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + 2\gamma\lambda S^2\\
+\frac{\gamma}{2}\sum_{p=1}^{P}||\theta_{p-1}-\theta_\star||_{\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top}^2
\end{aligned}$$

By rearranging terms, we have

$$\begin{aligned}
&\sum_{p=1}^{P}\big[\nabla F_p(\theta_{p-1})^\top(\theta_{p-1}-\theta_\star) - \frac{\gamma}{2}||\theta_{p-1}-\theta_\star||_{\nabla F_p(\theta_{p-1})\nabla F_p(\theta_{p-1})^\top}^2\big]\\
&\leq \frac{1}{2\gamma}\sum_{p=1}^{P}||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + 2\gamma\lambda S^2
\end{aligned}$$

Combining with Eq.(2.20), we obtain the following upper bound for the P-step online regret

$$\sum_{p=1}^{P}F_p(\theta_{p-1}) - F_p(\theta_\star) \leq \frac{1}{2\gamma}\sum_{p=1}^{P}||\nabla F_p(\theta_{p-1})||_{A_p^{-1}}^2 + 2\gamma\lambda S^2$$

where $A_p = \sum_{\rho=1}^{p}\nabla F_\rho(\theta_{\rho-1})\nabla F_\rho(\theta_{\rho-1})^\top$. □

Corollary 2.2.7.1 (Order of B_P)**.** *Under the condition that* $\gamma = \frac{1}{2}\min(\frac{1}{4S\sqrt{k_\mu^2S^2+R_{\max}^2}}, \frac{c_\mu}{(k_\mu^2S^2+R_{\max}^2)\max_{p\in[P]}(n_p-n_{p-1})})$, *the online regret upper bound* $B_P = O(\frac{k_\mu^2+R_{max}^2}{c_\mu}d\log(n_P)\max_{p\in[P]}(n_p-n_{p-1}))$.

Proof of Corollary 2.2.7.1. Recall that $A_p = \sum_{\rho=1}^{p} \nabla F_\rho(\theta_{\rho-1}) \nabla F_\rho(\theta_{\rho-1})^\top$. Therefore, we have

$$\sum_{p=1}^{P} ||\nabla F_p(\theta_{p-1})||^2_{A_p^{-1}} \leq \log \frac{\det(A_P)}{\det(\lambda I)} = \log \frac{\det(\lambda I + \sum_{p=1}^{P} \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top)}{\det(\lambda I)}$$

$$\leq d \log \big(1 + \frac{1}{d\lambda} \sum_{p=1}^{P} \|\nabla F_p(\theta_{p-1})\|_2^2\big)$$

where the first inequality is due to Lemma 11 of [20], and the second due to the determinant-trace inequality (Lemma 10 of [20]), i.e., $\det(\lambda I + \sum_{p=1}^{P} \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top) \leq \big(\frac{tr(\lambda I + \sum_{p=1}^{P} \nabla F_p(\theta_{p-1}) \nabla F_p(\theta_{p-1})^\top)}{d}\big)^d = \big(\frac{d\lambda + \sum_{p=1}^{P} \|\nabla F_p(\theta_{p-1})\|_2^2}{d}\big)^d$. Also note that $\nabla F_p(\theta_{p-1}) = \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s \big[\mu(\mathbf{x}_s^\top \theta_{p-1}) - y_s\big]$, so we have

$$\sum_{p=1}^{P} ||\nabla F_p(\theta_{p-1})||_2^2 = \sum_{p=1}^{P} || \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s \big[\mu(\mathbf{x}_s^\top \theta_{p-1}) - y_s\big] ||_2^2$$

$$\leq G^2 \sum_{p=1}^{P} || \sum_{s=n_{p-1}+1}^{n_p} \mathbf{x}_s ||_2^2 \leq G^2 \sum_{p=1}^{P} (n_p - n_{p-1})^2 \leq G^2 n_P^2$$

where the second inequality is due to Jensen's inequality and the assumption that $\|\mathbf{x}_s\| \leq 1, \forall s$. Substituting this back gives us

$$\sum_{p=1}^{P} F_p(\theta_{p-1}) - F_p(\theta_\star) \leq \frac{1}{2\gamma} d \log \big(1 + \frac{1}{d\lambda} G^2 n_P^2\big) + 2\gamma\lambda S^2$$

$$= \frac{(k_\mu^2 S^2 + R_{\max}^2) \max_{p\in[P]}(n_p - n_{p-1})}{c_\mu} d \log \big(1 + \frac{1}{d\lambda}(k_\mu^2 S^2 + R_{\max}^2) n_P^2\big)$$

$$+ \frac{c_\mu}{(k_\mu^2 S^2 + R_{\max}^2) \max_{p\in[P]}(n_p - n_{p-1})} \lambda S^2$$

where the equality is because $\max_{p\in[P]}(n_p - n_{p-1})$ dominates $\gamma = \frac{1}{2} \min(\frac{1}{4GS}, \frac{c_\mu}{G^2 \max_{p\in[P]}(n_p - n_{p-1})})$. □

Construct confidence ellipsoid for FedGLB-UCB$_3$ With the online regret bound B_P in Lemma 2.2.7, the steps to construct the confidence ellipsoid largely follows that of Theorem 1 in [70], with the main difference in our batch update. We include the full proof here for the sake of completeness.

Lemma 2.2.8 (Confidence Ellipsoid for FedGLB-UCB$_3$). *Under the condition that the learning rate of ONS* $\gamma = \frac{1}{2} \min(\frac{1}{4S\sqrt{k_\mu^2 S^2 + R_{\max}^2}}, \frac{c_\mu}{(k_\mu^2 S^2 + R_{\max}^2) \max_{p\in[P]}(n_p - n_{p-1})})$, *we have* $\forall t \in [T], i \in [N]$

$$\|\theta_\star - \hat{\theta}_{t,i}\|_{V_{t,i}}^2 \leq \lambda S^2 + 1 + \frac{4}{c_\mu} B_P + \frac{8R_{max}^2}{c_\mu^2} \log (\frac{N}{\delta} \sqrt{4 + \frac{8}{c_\mu} B_P + \frac{64R_{max}^2}{c_\mu^4 \cdot 4\delta^2}})\}$$

$$- \hat{\theta}_{t,i}^\top b_{t,i} - \sum_{s=1}^{n_P} z_s^2 := \alpha_{t,i}^2$$

with probability at least $1 - \delta$.

Proof of Lemma 2.2.8. Due to c_μ-strongly convexity of $l(z, y)$ w.r.t. z, we have $l(\mathbf{x}_s^\top \theta_{p-1}, y_s) - l(\mathbf{x}_s^\top \theta_\star, y_s) \geq \big[\mu(\mathbf{x}_s^\top \theta_\star) - y_s\big]\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) + \frac{c_\mu}{2}\big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2$. Therefore,

$$\begin{aligned}
F_p(\theta_{p-1}) - F_p(\theta_\star) &= \sum_{s=n_{p-1}+1}^{n_p} l(\mathbf{x}_s^\top \theta_{p-1}, y_s) - l(\mathbf{x}_s^\top \theta_\star, y_s) \\
&\geq \sum_{s=n_{p-1}+1}^{n_p} \big[\mu(\mathbf{x}_s^\top \theta_\star) - y_s\big]\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) + \frac{c_\mu}{2} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2 \\
&= - \sum_{s=n_{p-1}+1}^{n_p} \eta_s \mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) + \frac{c_\mu}{2} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2
\end{aligned}$$

where η_s is the R-sub-Gaussian noise in the reward y_s. Summing over P steps we have

$$B_P \geq \sum_{p=1}^{P} F_p(\theta_{p-1}) - F_p(\theta_\star) \geq \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \eta_s \mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) + \frac{c_\mu}{2} \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2$$

By rearranging terms, we have

$$\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2 \leq \frac{2}{c_\mu} \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \eta_s \mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) + \frac{2}{c_\mu} B_P$$

Then as $\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)$ for $s \in [n_{p-1}+1, n_p]$ is $\mathcal{F}_s$-measurable for lazily updated online estimator θ_{p-1}, we can use Corollary 8 from [69], which leads to

$$\begin{aligned}
&\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \eta_s \mathbf{x}_s^\top(\theta_{p-1} - \theta_\star) \leq \\
&R_{max} \sqrt{\Big(2 + 2\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} (\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star))^2\Big) \cdot \log\Big(\frac{1}{\delta}\sqrt{1 + \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} (\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star))^2}\Big)}
\end{aligned}$$

Then we have

$$\begin{aligned}
&\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2 \leq \frac{2}{c_\mu} B_P \\
&+ \frac{2R_{max}}{c_\mu} \sqrt{\Big(2 + 2\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} (\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star))^2\Big) \cdot \log\Big(\frac{1}{\delta}\sqrt{1 + \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} (\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star))^2}\Big)}
\end{aligned}$$

Then by applying Lemma 2 from [70], we have

$$\sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2 \leq 1 + \frac{4}{c_\mu} B_P + \frac{8R_{max}^2}{c_\mu^2} \log\big(\frac{1}{\delta}\sqrt{4 + \frac{8}{c_\mu} B_P + \frac{64R_{max}^2}{c_\mu^4 \cdot 4\delta^2}}\big)$$

Therefore, we have the following confidence ellipsoid (regularized with parameter λ):

$$\{\theta : \sum_{p=1}^{P} \sum_{s=n_{p-1}+1}^{n_p} \big[\mathbf{x}_s^\top(\theta_{p-1} - \theta_\star)\big]^2 + \lambda\|\theta\|_2^2 \leq \lambda S^2 + 1 + \frac{4}{c_\mu} B_P + \frac{8R_{max}^2}{c_\mu^2} \log\big(\frac{1}{\delta}\sqrt{4 + \frac{8}{c_\mu} B_P + \frac{64R_{max}^2}{c_\mu^4 \cdot 4\delta^2}}\big)\}$$

And this can be rewritten as a ellipsoid centered at ridge regression estimator $\hat{\theta}_{t,i} = V_{t,i}^{-1} b_{t,i}$, where $V_{t,i} = \lambda I +$

$\sum_{p=1}^{P}\sum_{s=n_{p-1}+1}^{n_p}\mathbf{x}_s\mathbf{x}_s^\top$ and $b_{t,i} = \sum_{p=1}^{P}\sum_{s=n_{p-1}+1}^{n_p}\mathbf{x}_s z_s$ (recall that ONS's prediction at time s is denoted as $z_s = \mathbf{x}_s^\top\theta_{p-1}$), i.e., $\forall t \in [T]$

$$\|\theta_\star - \hat{\theta}_{t,i}\|_{V_{t,i}}^2 \leq \lambda S^2 + 1 + \frac{4}{c_\mu}B_P + \frac{8R_{max}^2}{c_\mu^2}\log\big(\frac{1}{\delta}\sqrt{4 + \frac{8}{c_\mu}B_P + \frac{64R_{max}^2}{c_\mu^4\cdot 4\delta^2}}\big)\} + \hat{\theta}_{t,i}^\top b_{t,i} - \sum_{s=1}^{n_P} z_s^2$$

with probability at least $1-\delta$. Then taking union bound over all N clients, we have, $\forall t \in [T], i \in [N]$

$$\|\theta_\star - \hat{\theta}_{t,i}\|_{V_{t,i}}^2 \leq \lambda S^2 + 1 + \frac{4}{c_\mu}B_P + \frac{8R_{max}^2}{c_\mu^2}\log\big(\frac{N}{\delta}\sqrt{4 + \frac{8}{c_\mu}B_P + \frac{64R_{max}^2}{c_\mu^4\cdot 4\delta^2}}\big)\} + \hat{\theta}_{t,i}^\top b_{t,i} - \sum_{s=1}^{n_P} z_s^2$$

with probability at least $1-\delta$. □

Regret and communication upper bounds for FedGLB-UCB$_3$ The regret and communication cost of FedGLB-UCB$_3$ is given in the following theorem.

Theorem 2.2.9 (Regret and Communication Cost Upper Bound of FedGLB-UCB$_3$)**.** *Under the condition that the learning rate of ONS* $\gamma = \frac{1}{2}\min(\frac{1}{4S\sqrt{k_\mu^2 S^2+R_{\max}^2}}, \frac{c_\mu}{(k_\mu^2 S^2+R_{\max}^2)\sqrt{NT}})$*, and the total number of global synchronizations* $B=\sqrt{NT}$*, the cumulative regret* R_T *has upper bound*

$$R_T = O\left(\frac{k_\mu(k_\mu + R_{max})}{c_\mu}dN^{3/4}T^{3/4}\log(NT/\delta)\right)$$

with probability at least $1-\delta$*. The cumulative communication cost has upper bound*

$$C_T = O(N^{1.5}\sqrt{T})$$

Proof. Similar to the proof for the previous two variants of FedGLB-UCB, we divide the epochs into 'good' and 'bad' ones according to the determinant ratio, and then bound their cumulative regret separately.

Recall that the instantaneous regret $r_{t,i}$ incurred by client $i \in [N]$ at time step $t \in [T]$ has upper bound

$$\begin{aligned}
\frac{r_{t,i}}{k_\mu} &\leq \mathbf{x}_{t,\star}^\top\theta_\star - \mathbf{x}_{t,i}^\top\theta_\star \leq \mathbf{x}_{t,i}^\top\tilde{\theta}_{i,t} - \mathbf{x}_{t,i}^\top\theta_\star \\
&= \mathbf{x}_{t,i}^\top(\tilde{\theta}_{i,t} - \hat{\theta}_{t,i}) + \mathbf{x}_{t,i}^\top(\hat{\theta}_{t,i} - \theta_\star) \\
&\leq \|\mathbf{x}_{t,i}\|_{V_{t,i}^{-1}}\|\tilde{\theta}_{i,t} - \hat{\theta}_{t,i}\|_{V_{t,i}} + \|\mathbf{x}_{t,i}\|_{V_{t,i}^{-1}}\|\hat{\theta}_{t,i} - \theta_\star\|_{V_{t,i}} \\
&\leq 2\alpha_{t,i}\|\mathbf{x}_{t,i}\|_{V_{t,i}^{-1}}
\end{aligned}$$

Note that due to the update schedule $\mathcal{S}$, we have $\max_{p\in[P]}(n_p - n_{p-1}) = \frac{NT}{B}$. Then based on Corollary 2.2.7.1, $\alpha_{t,i} = O(\frac{k_\mu + R_{max}}{c_\mu}\sqrt{d\log(NT)}\sqrt{\frac{NT}{B}})$, so we have, $\forall t \in [T], i \in [N]$,

$$r_{t,i} = O(\frac{k_\mu(k_\mu + R_{max})}{c_\mu}\sqrt{d\log(NT)}\sqrt{\frac{NT}{B}})\|\mathbf{x}_{t,i}\|_{A_{t,i}^{-1}}$$

with probability at least $1-\delta$.

Therefore, the cumulative regret for the 'good epochs' is $REG_{good} = O(\frac{k_\mu(k_\mu+R_{max})}{c_\mu}d\frac{NT}{\sqrt{B}}\log(NT))$.

Using the same argument as in the proof for FedGLB-UCB$_1$, the cumulative regret for each 'bad ' epoch is upper bounded by $2k_\mu S\frac{NT}{B}$. Since there can be at most $O(d\log NT)$ 'bad epochs', the cumulative regret for all the 'bad epochs' is upper bounded by

$$REG_{bad} = O(dNT\log(NT)\cdot\frac{k_\mu S}{B})$$

Combining the regret incurred in both 'good' and 'bad' epochs, we have

$$R_T = O\big(\frac{k_\mu(k_\mu + R_{max})}{c_\mu} d\frac{NT}{\sqrt{B}}\log(NT) + dNT\log(NT)\cdot\frac{k_\mu S}{B}\big)$$

To recover the regret in centralized setting, we can $B = NT$, which leads to $R_T = O(\frac{k_\mu(k_\mu+R_{max})}{c_\mu} d\sqrt{NT}\log(NT))$. However, this incurs communication cost $C_T = N^2T$. Alternatively, if we set $B = \sqrt{NT}$, we have $R_T = O(\frac{k_\mu(k_\mu+R_{max})}{c_\mu} dN^{3/4}T^{3/4}\log(NT))$, and $C_T = O(N^{1.5}\sqrt{T})$. □

2.2.9 Distributed kernelized contextual bandit problem

In this section, we further investigate the collaborative exploration of non-parametric functions lying in a reproducing kernel Hilbert space (RKHS) [26, 72], i.e., the expected reward is linear w.r.t. an action feature map of possibly infinite dimensions. The ability to learn non-parametric models has made kernelized bandit algorithms a powerful tool for optimizing black box functions based on noisy observations in various applications.

We consider the same star-shaped communication network as Section 2.2.2. To facilitate discussions in this section, we introduce slightly different notations for indexing the selected arms and observed rewards compared with the ones used in Section 2.2.3. Specifically, at each round $l \in [T]$, each client $i \in [N]$ chooses an arm $\mathbf{x}_t$ from a candidate set $\mathcal{A}_t$, and then receives the corresponding reward feedback $y_t = f(\mathbf{x}_t) + \eta_t \in \mathbb{R}$, where the subscript $t := N(l-1) + i$ indicates this is the t-th interaction between the learning system and the environment, and we refer to it as time step t [2]. Note that $\mathcal{A}_t$ is a time-varying subset of $\mathcal{A} \subseteq \mathbb{R}^d$ that is possibly infinite, f denotes the unknown reward function shared by all the clients, and η_t denotes the noise.

Denote the sequence of indices corresponding to the interactions between client i and the environment up to time t as $\mathcal{N}_t(i) = \{1 \leq s \leq t : i_s = i\}$ (if $s \bmod N = 0$, then $i_s = N$; otherwise $i_s = s \bmod N$) for $t = 1, 2, \dots, NT$. By definition, $|\mathcal{N}_{Nl}(i)| = l, \forall l \in [T]$, i.e., the clients have equal number of interactions at the end of each round l.

Kernelized Reward Function We consider an unknown reward function f that lies in a RKHS, denoted as $\mathcal{H}$, such that the reward can be equivalently written as

$$y_t = \theta_\star^\top \phi(\mathbf{x}_t) + \eta_t,$$

where $\theta_\star \in \mathcal{H}$ is an unknown parameter, and $\phi : \mathbb{R}^d \to \mathcal{H}$ is a known feature map associated with $\mathcal{H}$. We assume η_t is zero-mean R-sub-Gaussian conditioned on $\sigma\big((\mathbf{x}_s, \eta_s)_{s \in \mathcal{N}_{t-1}(i_t)}\big), \forall t$, which denotes the σ-algebra generated by client i_t's previously pulled arms and the corresponding noise. In addition, there exists a positive definite kernel $k(\cdot,\cdot)$ associated with $\mathcal{H}$, and we assume $\forall \mathbf{x} \in \mathcal{A}$ that, $\|\mathbf{x}\|_k \leq L$ and $\|f\|_k \leq S$ for some $L, S > 0$.

Regret and Communication Cost The goal of the learning system is to minimize the cumulative (pseudo) regret for all N clients, i.e., $R_{NT} = \sum_{t=1}^{NT} r_t$, where $r_t = \max_{\mathbf{x} \in \mathcal{A}_t} \phi(\mathbf{x})^\top\theta_\star - \phi(\mathbf{x}_t)^\top\theta_\star$. Meanwhile, the learning system also wants to keep the communication cost C_{NT} low, which is measured by the total number of scalars being transferred across the system up to time step NT.

2.2.10 DisKernelUCB algorithm

As a starting point to studying the communication efficient algorithm in Section 2.2.11 and demonstrate the challenges in designing a communication efficient distributed kernelized contextual bandit algorithm, here we first introduce and analyze a naive algorithm where the N clients collaborate on learning the exact parameters of kernel bandit, i.e., the mean and variance of estimated reward. We name this algorithm Distributed Kernel UCB, or DisKernelUCB for short, and its description is given in Algorithm 8.

[2] The meaning of index t is slightly different from prior works, e.g. DisLinUCB in [28], but this is only to simplify the use of notation and does not affect the theoretical results

Algorithm 8 Distributed Kernel UCB (DisKernelUCB)

```
1:  Input threshold $D > 0$
2:  Initialize $t_{\text{last}} = 0, \mathcal{D}_0(i) = \Delta\mathcal{D}_0(i) = \emptyset, \forall i \in [N]$
3:  for round $l = 1, 2, ..., T$ do
4:      for client $i = 1, 2, ..., N$ do
5:          Client $i$ chooses arm $\mathbf{x}_t \in \mathcal{A}_t$ according to Eq (2.21) and observes reward $y_t$, where $t = N(l-1) + i$
6:          Client $i$ updates $\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}, \mathbf{y}_{\mathcal{D}_t(i)}$, where $\mathcal{D}_t(i) = \mathcal{D}_{t-1}(i) \cup \{t\}$; and its upload buffer $\Delta\mathcal{D}_t(i) = \Delta\mathcal{D}_{t-1}(i) \cup \{t\}$
            // Global Synchronization
7:          if the event $\mathcal{U}_t(D)$ defined in Eq (2.22) is true then
8:              Clients $\forall j \in [N]$: send $\{(\mathbf{x}_s, y_s)\}_{s \in \Delta\mathcal{D}_t(j)}$ to server, and reset $\Delta\mathcal{D}_t(j) = \emptyset$
9:              Server: aggregates and sends back $\{(\mathbf{x}_s, y_s)\}_{s \in [t]}$; sets $t_{\text{last}} = t$
10:             Clients $\forall j \in [N]$: update $\mathbf{K}_{\mathcal{D}_t(j),\mathcal{D}_t(j)}, \mathbf{y}_{\mathcal{D}_t(i)}$, where $\mathcal{D}_t(j) = [t]$
```

Arm Selection For each round $l \in [T]$, when client $i \in [N]$ interacts with the environment, i.e., the t-th interaction between the learning system and the environment where $t = N(l-1) + i$, it chooses arm $\mathbf{x}_t \in \mathcal{A}_t$ based on the UCB of the mean estimator (line 5):

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{A}_t} \hat{\mu}_{t-1,i}(\mathbf{x}) + \alpha_{t-1,i}\hat{\sigma}_{t-1,i}(\mathbf{x}) \tag{2.21}$$

where $\hat{\mu}_{t,i}(\mathbf{x})$ and $\hat{\sigma}^2_{t,i}(\mathbf{x})$ denote client i's local estimated mean reward for arm $\mathbf{x} \in \mathcal{A}$ and its variance, and $\alpha_{t-1,i}$ is a carefully chosen scaling factor to balance exploration and exploitation (see Lemma 2.2.10 for proper choice).

To facilitate further discussion, for time step $t \in [NT]$, we denote the sequence of time indices for the data points that have been used to update client i's local estimate as $\mathcal{D}_t(i)$, which include both data points collected locally and those shared by the other clients. If the clients never communicate, $\mathcal{D}_t(i) = \mathcal{N}_t(i), \forall t, i$; otherwise, $\mathcal{N}_t(i) \subset \mathcal{D}_t(i) \subseteq [t]$, with $\mathcal{D}_t(i) = [t]$ recovering the centralized setting, i.e., each new data point collected from the environment immediately becomes available to all the clients in the learning system. The design matrix and reward vector for client i at time step t are denoted by $\mathbf{X}_{\mathcal{D}_t(i)} = [\mathbf{x}_s]^\top_{s \in \mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)| \times d}, \mathbf{y}_{t,i} = [y_s]^\top_{s \in \mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)|}$, respectively. By applying the feature map $\phi(\cdot)$ to each row of $\mathbf{X}_{\mathcal{D}_t(i)}$, we obtain $\mathbf{\Phi}_{\mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)| \times p}$, where p is the dimension of $\mathcal{H}$ and is possibly infinite. Since the reward function is linear in $\mathcal{H}$, client i can construct the Ridge regression estimator $\hat{\theta}_{t,i} = (\mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda I)^{-1}\mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\mathbf{y}_{t,i}$, where $\lambda > 0$ is the regularization coefficient. This gives us the estimated mean reward and variance in primal form for any arm $\mathbf{x} \in \mathcal{A}$, i.e., $\hat{\mu}_{t,i}(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{A}^{-1}_{t,i}\mathbf{b}_{t,i}$ and $\hat{\sigma}_{t,i}(\mathbf{x}) = \sqrt{\phi(\mathbf{x})^\top \mathbf{A}^{-1}_{t,i}\phi(\mathbf{x})}$, where $\mathbf{A}_{t,i} = \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda\mathbf{I}$ and $\mathbf{b}_{t,i} = \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\mathbf{y}_{t,i}$. Then using the kernel trick, we can obtain their equivalence in the dual form that only involves entries of the kernel matrix, and avoids directly working on $\mathcal{H}$ which is possibly infinite:

$$\hat{\mu}_{t,i}(\mathbf{x}) = \mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})^\top \big(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} + \lambda I\big)^{-1}\mathbf{y}_{\mathcal{D}_t(i)}$$
$$\hat{\sigma}_{t,i}(\mathbf{x}) = \lambda^{-1/2}\sqrt{k(\mathbf{x},\mathbf{x}) - \mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})^\top \big(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} + \lambda I\big)^{-1}\mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})}$$

where $\mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x}) = \mathbf{\Phi}_{\mathcal{D}_t(i)}\phi(\mathbf{x}) = [k(\mathbf{x}_s, \mathbf{x})]^\top_{s \in \mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)|}$, and $\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} = \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\mathbf{\Phi}_{\mathcal{D}_t(i)} = [k(\mathbf{x}_s, \mathbf{x}_{s'})]_{s,s' \in \mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)| \times |\mathcal{D}_t(i)|}$.

Communication Protocol To reduce the regret in future interactions with the environment, the N clients need to collaborate via communication, and a carefully designed communication protocol is essential in ensuring the communication efficiency. In prior works like DisLinUCB [28], after each round of interaction with the environment, client i checks whether the event $\{(|\mathcal{D}_t(i)| - |\mathcal{D}_{t_{\text{last}}}(i)|)\log(\frac{\det(\mathbf{A}_{t,i})}{\det(\mathbf{A}_{t_{\text{last}},i})}) > D\}$ is true, where t_{last} denotes the time step of last global synchronization. If true, a new global synchronization is triggered, such that the server will require all clients to upload their sufficient statistics since t_{last}, aggregate them to compute $\{\mathbf{A}_t, \mathbf{b}_t\}$, and then synchronize the aggregated sufficient statistics with all clients, i.e., set $\{\mathbf{A}_{t,i}, \mathbf{b}_{t,i}\} = \{\mathbf{A}_t, \mathbf{b}_t\}, \forall i \in [N]$.

Using kernel trick, we can obtain an equivalent event-trigger in terms of the kernel matrix,

$$\mathcal{U}_t(D) = \left\{ (|\mathcal{D}_t(i_t)| - |\mathcal{D}_{t_{\text{last}}}(i_t)|) \log \left(\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_t(i_t),\mathcal{D}_t(i_t)})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_t(i_t)\setminus\Delta\mathcal{D}_t(i_t),\mathcal{D}_t(i_t)\setminus\Delta\mathcal{D}_t(i_t)})} \right) > D \right\}. \tag{2.22}$$

where $D > 0$ denotes the predefined threshold value. If event $\mathcal{U}_t(D)$ is true (line 7), a global synchronization is triggered (line 7-10), where the local datasets of all N clients are synchronized to $\{(\mathbf{x}_s, y_s)\}_{s\in[t]}$. We should note that the transfer of raw data $(\mathbf{x}_s, y_s)$ is necessary for the update of the kernel matrix and reward vector in line 6 and line 10, which will be used for arm selection at line 5. This is an inherent disadvantage of kernelized estimation in distributed settings. Lemma 2.2.10 below shows that in order to obtain the optimal order of regret, DisKernelUCB incurs a communication cost linear in T (proof given in Section 2.2.14), which is expensive for an online learning problem.

Lemma 2.2.10 (Regret and Communication Cost of DisKernelUCB). *With threshold $D = \frac{T}{N\gamma_{NT}}$, $\alpha_{t,i} = \sqrt{\lambda}\|\theta_\star\| + R\sqrt{4\ln NT/\delta + 2\ln\det(\mathbf{I} + \mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}/\lambda)}$, we have*

$$R_{NT} = O\big(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\big),$$

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

$$C_{NT} = O(TN^2 d).$$

where $\gamma_{NT} := \max_{\mathcal{D}\subset\mathcal{A}:|\mathcal{D}|=NT} \frac{1}{2}\log\det(\mathbf{K}_{\mathcal{D},\mathcal{D}}/\lambda + \mathbf{I})$ is the maximum information gain after NT interactions [72]. It is problem-dependent and can be bounded for specific arm set $\mathcal{A}$ and kernel function $k(\cdot,\cdot)$. For example, $\gamma_{NT} = O(d\log(NT))$ for linear kernel and $\gamma_{NT} = O(\log(NT)^{d+1})$ for Gaussian kernel.

Remark 1. *In the distributed linear bandit problem, to attain $O(d\sqrt{NT}\ln(NT))$ regret, DisLinUCB [28] requires a total number of $O(N^{0.5}d\log(NT))$ synchronizations, and DisKernelUCB matches this result under linear kernel, as it requires $O(N^{0.5}\gamma_{NT})$ synchronizations. We should note that the communication cost for each synchronization in DisLinUCB is fixed, i.e., $O(Nd^2)$ to synchronize the sufficient statistics with all the clients, so in total $C_{NT} = O(N^{1.5}d^3\ln(NT))$. However, this is not the case for DisKernelUCB that needs to send raw data, because the communication cost for each synchronization in DisKernelUCB is not fixed, but depends on the number of unshared data points on each client. Even if the total number of synchronizations is small, DisKernelUCB could still incur $C_{NT} = O(TN^2d)$ in the worse case. Consider the extreme case where synchronization only happens once, but it happens near NT, then we still have $C_{NT} = O(TN^2d)$. The time when synchronization gets triggered depends on $\{\mathcal{A}_t\}_{t\in[NT]}$, which is out of the control of the algorithm. Therefore, in the following section, to improve the communication efficiency of DisKernelUCB, we propose to let each client communicate embedded statistics in some small subspace during each global synchronization.*

2.2.11 Approx-DisKernelUCB algorithm

In this section, we propose and analyze a new algorithm that improves the communication efficiency of DisKernelUCB using the Nyström approximation, such that the clients only communicate the embedded statistics during event-triggered synchronizations. We name this algorithm Approximated Distributed Kernel UCB, or Approx-DisKernelUCB for short. Its description is given in Algorithm 9.

Arm selection For each round $l \in [T]$, when client $i \in [N]$ interacts with the environment, i.e., the t-th interaction between the learning system and the environment where $t := N(l-1) + i$, instead of using the UCB for the exact estimator in Eq (2.21), client i chooses arm $\mathbf{x}_t \in \mathcal{A}_t$ that maximizes the UCB for the approximated estimator (line 5):

$$\mathbf{x}_t = \operatorname*{arg\,max}_{\mathbf{x}\in\mathcal{A}_{t,i}} \tilde{\mu}_{t-1,i}(\mathbf{x}) + \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}) \tag{2.23}$$

where $\tilde{\mu}_{t-1,i}(\mathbf{x})$ and $\tilde{\sigma}_{t-1,i}(\mathbf{x})$ are approximated using Nyeström method, and the statistics used to compute these approximations are much more efficient to communicate as they scale with the maximum information gain γ_{NT} instead of T.

Algorithm 9 Approximated Distributed Kernel UCB (Approx-DisKernelUCB)

```
1: Input: threshold $D > 0$, regularization parameter $\lambda > 0$, $\delta \in (0,1)$ and kernel function $k(\cdot,\cdot)$.
2: Initialize $\tilde{\mu}_{0,i}(\mathbf{x}) = 0$, $\tilde{\sigma}_{0,i}(\mathbf{x}) = \sqrt{k(\mathbf{x},\mathbf{x})}$, $\mathcal{N}_0(i) = \mathcal{D}_0(i) = \emptyset, \forall i \in [N]$; $\mathcal{S}_0 = \emptyset$, $t_{\text{last}} = 0$
3: for round $l = 1, 2, ..., T$ do
4:   for client $i = 1, 2, ..., N$ do
5:     [Client $i$] selects arm $\mathbf{x}_t \in \mathcal{A}_t$ according to Eq (2.23) and observes reward $y_t$, where $t := N(l-1) + i$
6:     [Client $i$] updates $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_{t_{\text{last}}}}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_{t_{\text{last}}}}$ and $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_{t_{\text{last}}}}^\top \mathbf{y}_{\mathcal{D}_t(i)}$ using $\big(\mathbf{z}(\mathbf{x}_t;\mathcal{S}_{t_{\text{last}}}), y_t\big)$; sets $\mathcal{N}_t(i) = \mathcal{N}_{t-1}(i) \cup \{t\}$, and $\mathcal{D}_t(i) = \mathcal{D}_{t-1}(i) \cup \{t\}$
       // Global Synchronization
7:     if the event $\mathcal{U}_t(D)$ defined in Eq (2.24) is true then
8:       [Clients $\forall i$] sample $\mathcal{S}_{t,i} = \text{RLS}(\mathcal{N}_t(i), \bar{q}, \tilde{\sigma}^2_{t_{\text{last}},i})$, and send $\{(\mathbf{x}_s, y_s)\}_{s \in \mathcal{S}_{t,i}}$ to server
9:       [Server] aggregates and sends $\{(\mathbf{x}_s, y_s)\}_{s \in \mathcal{S}_t}$ back to all clients, where $\mathcal{S}_t = \cup_{i \in [N]} \mathcal{S}_{t,i}$
10:      [Clients $\forall i$] compute and send $\{\mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}^\top \mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}, \mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}^\top \mathbf{y}_{\mathcal{N}_t(i)}\}$ to server
11:      [Server] aggregates $\sum_{i=1}^N \mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}^\top \mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}, \sum_{i=1}^N \mathbf{Z}_{\mathcal{N}_t(i);\mathcal{S}_t}^\top \mathbf{y}_{\mathcal{N}_t(i)}$ and sends it back
12:      [Clients $\forall i$] updates $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_t}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_t}$ and $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}_t}^\top \mathbf{y}_{\mathcal{D}_t(i)}$; sets $\mathcal{D}_t(i) = \cup_{i=1}^N \mathcal{N}_t(i) = [t]$ and $t_{\text{last}} = t$
```

Specifically, Nyström method works by projecting some original dataset $\mathcal{D}$ to the subspace defined by a small representative subset $\mathcal{S} \subseteq \mathcal{D}$, which is called the dictionary. The orthogonal projection matrix is defined as

$$\mathbf{P}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \left(\mathbf{\Phi}_{\mathcal{S}} \mathbf{\Phi}_{\mathcal{S}}^\top\right)^{-1} \mathbf{\Phi}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \mathbf{K}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Phi}_{\mathcal{S}} \in \mathbb{R}^{p \times p}$$

We then take eigen-decomposition of $\mathbf{K}_{\mathcal{S},\mathcal{S}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ to rewrite the orthogonal projection as $\mathbf{P}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top \mathbf{\Phi}_{\mathcal{S}}$, and define the Nyström embedding function

$$z(\mathbf{x};\mathcal{S}) = \mathbf{P}_{\mathcal{S}}^{1/2}\phi(\mathbf{x}) = \mathbf{\Lambda}^{-1/2}\mathbf{U}^\top \mathbf{\Phi}_{\mathcal{S}}\phi(\mathbf{x}) = \mathbf{K}_{\mathcal{S},\mathcal{S}}^{-1/2}\mathbf{K}_{\mathcal{S}}(\mathbf{x})$$

which maps the data point $\mathbf{x}$ from $\mathbb{R}^d$ to $\mathbb{R}^{|\mathcal{S}|}$.

Therefore, we can approximate the Ridge regression estimator in Section 2.2.10 as $\tilde{\theta}_{t,i} = \tilde{\mathbf{A}}_{t,i}^{-1}\tilde{\mathbf{b}}_{t,i}$, where $\tilde{\mathbf{A}}_{t,i} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I}$, and $\tilde{\mathbf{b}}_{t,i} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{y}_{\mathcal{D}_t(i)}$, and thus the approximated mean reward and variance in Eq (2.23) can be expressed as $\tilde{\mu}_{t,i}(\mathbf{x}) = \phi(\mathbf{x})^\top \tilde{\mathbf{A}}_{t,i}^{-1}\tilde{\mathbf{b}}_{t,i}$ and $\tilde{\sigma}_{t,i}(\mathbf{x}) = \sqrt{\phi(\mathbf{x})^\top \tilde{\mathbf{A}}_{t,i}^{-1}\phi(\mathbf{x})}$, and their kernelized representation are (see Section 2.2.15 for detailed derivation)

$$\begin{aligned}
\tilde{\mu}_{t,i}(\mathbf{x}) &= z(\mathbf{x};\mathcal{S})^\top \left(\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} + \lambda\mathbf{I}\right)^{-1} \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{y}_{\mathcal{D}_t(i)} \\
\tilde{\sigma}_{t,i}(\mathbf{x}) &= \lambda^{-1/2}\sqrt{k(\mathbf{x},\mathbf{x}) - z(\mathbf{x};\mathcal{S})^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}[\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} + \lambda\mathbf{I}]^{-1} z(\mathbf{x}|\mathcal{S})}
\end{aligned}$$

where $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} \in \mathbb{R}^{|\mathcal{D}_t(i)| \times |\mathcal{S}|}$ is obtained by applying $z(\cdot;\mathcal{S})$ to each row of $\mathbf{X}_{\mathcal{D}_t(i)}$, i.e., $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} = \mathbf{\Phi}_{\mathcal{D}_t(i)}\mathbf{P}_{\mathcal{S}}^{1/2}$. We can see that the computation of $\tilde{\mu}_{t,i}(\mathbf{x})$ and $\tilde{\sigma}_{t,i}(\mathbf{x})$ only requires the embedded statistics: matrix $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ and vector $\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top \mathbf{y}_{\mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{S}|}$, which, as we will show later, makes joint kernelized estimation among N clients much more efficient in communication.

After obtaining the new data point $(\mathbf{x}_t, y_t)$, client i immediately updates both $\tilde{\mu}_{t-1,i}(\mathbf{x})$ and $\tilde{\sigma}_{t-1,i}(\mathbf{x})$ using the newly collected data point $(\mathbf{x}_t, y_t)$, i.e., by projecting $\mathbf{x}_t$ to the finite dimensional RKHS spanned by $\mathbf{\Phi}_{\mathcal{S}_{t_{\text{last}}}}$ (line 6). Recall that, we use $\mathcal{N}_t(i)$ to denote the sequence of indices for data collected by client i, and denote by $\mathcal{D}_t(i)$ the sequence of indices for data that has been used to update client i's model estimation $\tilde{\mu}_{t,i}$. Therefore, both of them need to be updated to include time step t.

Communication Protocol With the approximated estimator, the size of message being communicated across the learning system is reduced. However, a carefully designed event-trigger is still required to minimize the total number of global synchronizations up to time NT. Since the clients can no longer evaluate the exact kernel matrices in Eq (2.22), we instead use the event-trigger in Eq (2.24), which can be computed using the approximated variance from last global synchronization as,

$$\mathcal{U}_t(D) = \left\{ \sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_{\text{last}}}(i)} \tilde{\sigma}^2_{t_{\text{last}},i}(\mathbf{x}_s) > D \right\} \tag{2.24}$$

Similar to Algorithm 8, if Eq (2.24) is true, global synchronization is triggered, where both the dictionary and the embedded statistics get updated. During synchronization, each client first samples a subset $\mathcal{S}_t(i)$ from $\mathcal{N}_t(i)$ (line 8) using Ridge Leverage Score sampling (RLS) [73, 75], which is given in Algorithm 10, and then sends $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_t(i)}$ to the server. The server aggregates the received local subsets to construct a new dictionary $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_t}$, where $\mathcal{S}_t = \cup_{i=1}^N \mathcal{S}_t(i)$, and then sends it back to all N clients (line 9). Finally, the N clients use this updated dictionary to re-compute the embedded statistics of their local data, and then synchronize it with all other clients via the server (line 10-12).

Algorithm 10 Ridge Leverage Score Sampling (RLS)

```
Input: dataset $\mathcal{D}$, scaling factor $\bar{q}$, (possibly delayed and approximated) variance function $\tilde{\sigma}^2(\cdot)$
Initialize new dictionary $\mathcal{S} = \emptyset$
for $s \in \mathcal{D}$ do
    Set $\tilde{p}_s = \bar{q}\tilde{\sigma}^2(\mathbf{x}_s)$
    Draw $q_s \sim$ Bernoulli($\tilde{p}_s$)
    If $q_s = 1$, add $s$ into $\mathcal{S}$
Output: $\mathcal{S}$
```

Intuitively, in Algorithm 9, the clients first agree upon a common dictionary $\mathcal{S}_t$ that serves as a good representation of the whole dataset at the current time t, and then project their local data to the subspace spanned by this dictionary before communication, in order to avoid directly sending the raw data as in Algorithm 8. Then using the event-trigger, each client monitors the amount of new knowledge it has gained through interactions with the environment from last synchronization. When there is a sufficient amount of new knowledge, it will inform all the other clients to perform a synchronization. As we will show in the following section, the size of $\mathcal{S}_t$ scales linearly w.r.t. the maximum information gain γ_{NT}, and therefore it improves both the local computation efficiency on each client, and the communication efficiency during the global synchronization.

2.2.12 Regret and communication cost analysis

Denote the sequence of time steps when global synchronization is performed, i.e., the event $\mathcal{U}_t(D)$ in Eq (2.24) is true, as $\{t_p\}_{p=1}^B$, where $B \in [NT]$ denotes the total number of global synchronizations. Note that in Algorithm 9, the dictionary is only updated during global synchronization, e.g., at time t_p, the dictionary $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_{t_p}}$ is sampled from the whole dataset $\{(\mathbf{x}_s, y_s)\}_{s\in[t_p]}$ in a distributed manner, and remains fixed for all the interactions happened at $t \in [t_p + 1, t_{p+1}]$. Moreover, at time t_p, all the clients synchronize their embedded statistics, so that $\mathcal{D}_{t_p}(i) = [t_p], \forall i \in [N]$.

Since Algorithm 9 enables local update on each client, for time step $t \in [t_p + 1, t_p]$, new data points are collected and added into $\mathcal{D}_t(i)$, such that $\mathcal{D}_t(i) \supseteq [t_p]$. This *decreases* the approximation accuracy of $\mathcal{S}_{t_p}$, as new data points may not be well approximated by $\mathcal{S}_{t_p}$. For example, in extreme cases, the new data could be orthogonal to the dictionary. To formally analyze the accuracy of the dictionary, we adopt the definition of ϵ-accuracy from [92]. Denote by $\bar{\mathbf{S}}_{t,i} \in \mathbb{R}^{|\mathcal{D}_t(i)|\times|\mathcal{D}_t(i)|}$ a diagonal matrix, with its s-th diagonal entry equal to $\frac{1}{\sqrt{\tilde{p}_s}}$ if $s \in \mathcal{S}_{t_p}$ and 0 otherwise. Then if

$$(1-\epsilon_{t,i})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda\mathbf{I}) \preceq \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \bar{\mathbf{S}}_{t,i}^\top \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda\mathbf{I} \preceq (1+\epsilon_{t,i})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda\mathbf{I}),$$

we say the dictionary $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_{t_p}}$ is $\epsilon_{t,i}$-accurate w.r.t. dataset $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{D}_t(i)}$.

As shown below, the accuracy of the dictionary for Nyström approximation is essential as it affects the width of the confidence ellipsoid, and thus affects the cumulative regret. Intuitively, in order to ensure its accuracy throughout the learning process, we need to 1) make sure the RLS procedure in line 8 of Algorithm 9 that happens at each global synchronization produces a representative set of data samples, and 2) monitor the extent to which the dictionary obtained in previous global synchronization has degraded over time, and when necessary, trigger a new global synchronization to update it. Compared with prior work that freezes the model in-between consecutive communications [75], the analysis of ϵ-accuracy for Approx-DisKernelUCB is unique to this work and the result is presented below.

Lemma 2.2.11. *With $\bar{q} = 6\frac{1+\epsilon}{1-\epsilon}\log(4NT/\delta)/\epsilon^2$, for some $\epsilon \in [0,1)$, and threshold $D > 0$, Algorithm 9 guarantees that the dictionary is accurate with constant $\epsilon_{t,i} := \big(\epsilon + 1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}\big)$, and its size $|\mathcal{S}_t| = O(\gamma_{NT})$ for all $t \in [NT]$.*

Based on Lemma 2.2.11, we can construct the following confidence ellipsoid for unknown parameter $\theta_\star$.

Lemma 2.2.12 (Confidence Ellipsoid of Approximated Estimator)**.** *Under the condition that $\bar{q} = 6\frac{1+\epsilon}{1-\epsilon}\log(4NT/\delta)/\epsilon^2$, for some $\epsilon \in [0,1)$, and threshold $D > 0$, with probability at least $1-\delta$, we have $\forall t, i$ that*

$$\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}} \leq \Big(\frac{1}{\sqrt{-\epsilon + 1/(\frac{1+\epsilon}{1-\epsilon}D)}} + 1\Big)\sqrt{\lambda}\|\theta_\star\| + 2R\sqrt{\ln NT/\delta + \gamma_{NT}} := \alpha_{t,i}.$$

Using Lemma 2.2.12, we obtain the regret and communication cost upper bound of Approx-DisKernelUCB, which is given in Theorem 2.2.13 below.

Theorem 2.2.13 (Regret and Communication Cost of Approx-DisKernelUCB)**.** *Under the same condition as Lemma 2.2.12, and by setting $D = \frac{1}{N}, \epsilon < \frac{1}{3}$, we have*

$$R_{NT} = O\big(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\big)$$

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

$$C_{NT} = O\big(N^2\gamma_{NT}^3\big)$$

Here we provide a proof sketch for Theorem 2.2.13, and the complete proof can be found in Section 2.2.15.

Proof Sketch. Similar to the analysis of DisKernelUCB in Section 2.2.10 and DisLinUCB from [28], the cumulative regret incurred by Approx-DisKernelUCB can be decomposed in terms of 'good' and 'bad' epochs, and bounded separately. Here an epoch refers to the time period in-between two consecutive global synchronizations, e.g., the p-th epoch refers to $[t_{p-1}+1, t_p]$. Now consider an imaginary centralized agent that has immediate access to each data point in the learning system, and denote by $A_t = \sum_{s=1}^t \phi_s\phi_s^\top$ for $t \in [NT]$ the matrix constructed by this centralized agent. We call the p-th epoch a good epoch if $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) \leq 1$, otherwise it is a bad epoch. Note that $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}{\det(\mathbf{I})}) + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_2],[t_2]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}) + \cdots + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[NT],[NT]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_B],[t_B]})}) = \ln(\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[NT],[NT]})) \leq 2\gamma_{NT}$, where the last equality is due to the matrix determinant lemma, and the last inequality is by the definition of the maximum information gain γ_{NT} in Lemma 2.2.10. Then based on the pigeonhole principle, there can be at most $2\gamma_{NT}$ bad epochs.

By combining Lemma 2.2.15 and Lemma 2.2.12, we can bound the cumulative regret incurred during all good epochs, i.e., $R_{good} = O(\sqrt{NT}\gamma_{NT})$, which matches the optimal regret attained by the KernelUCB algorithm in centralized setting. Our analysis deviates from that of DisKernelUCB in the bad epochs, because of the difference in the event-trigger. The event-trigger of DisKernelUCB bounds each client's regret in a bad epoch, i.e., $\sum_{t\in\mathcal{D}_{t_p}(i)\setminus\mathcal{D}_{t_{p-1}}(i)} \hat{\sigma}_{t-1,i}(\mathbf{x}_t) \leq \sqrt{(|\mathcal{D}_{t_p}(i)| - |\mathcal{D}_{t_{p-1}}(i)|)\log\big(\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_t(i_t),\mathcal{D}_t(i_t)})/\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_t(i_t)\setminus\Delta\mathcal{D}_t(i_t),\mathcal{D}_t(i_t)\setminus\Delta\mathcal{D}_t(i_t)})\big)} < \sqrt{D}$. However, the event trigger of Approx-DisKernelUCB algorithm only bounds part of it, i.e., $\sum_{t\in\mathcal{D}_{t_p}(i)\setminus\mathcal{D}_{t_{p-1}}(i)} \tilde{\sigma}_{t-1,i}(\mathbf{x}_t) \leq \sqrt{(|\mathcal{D}_{t_p}(i)| - |\mathcal{D}_{t_{p-1}}(i)|)D}$, which leads to $R_{bad} = O(\sqrt{T}\gamma_{NT}N\sqrt{D})$ that is slightly worse than that of DisKernelUCB, i.e., a $\sqrt{T}$ factor in place of the $\sqrt{\gamma_{NT}}$ factor. By setting $D = 1/N$, we have $R_{NT} = O(\sqrt{NT}\gamma_{NT})$. Note that, to make sure $\epsilon_{t,i} = \big(\epsilon + 1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}\frac{1}{N}}\big) \in [0,1)$ is still well-defined, we can set $\epsilon < 1/3$.

For communication cost analysis, we bound the total number of epochs B by upper bounding the total number of summations like $\sum_{s=t_{p-1}+1}^{t_p} \hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)$, over the time horizon NT. Using Lemma 2.2.15, our event-trigger in Eq (2.24) provides a lower bound $\sum_{s=t_{p-1}+1}^{t_p} \hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s) \geq \frac{1-\epsilon}{1+\epsilon}D$. Then in order to apply the pigeonhole principle, we continue to upper bound the summation over all epochs, $\sum_{p=1}^{B}\sum_{s=t_{p-1}+1}^{t_p}\hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)=\sum_{p=1}^{B}\sum_{s=t_{p-1}+1}^{t_p}\hat{\sigma}_{s-1}^2(\mathbf{x}_s)\frac{\hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)}{\hat{\sigma}_{s-1}^2(\mathbf{x}_s)}$ by deriving a uniform bound for the ratio $\frac{\hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)}{\hat{\sigma}_{s-1}^2(\mathbf{x}_s)}\leq\frac{\hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)}{\hat{\sigma}_{t_p}^2(\mathbf{x}_s)}\leq 1+\sum_{s=t_{p-1}+1}^{t_p}\hat{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)\leq 1+\frac{1+\epsilon}{1-\epsilon}\sum_{s=t_{p-1}+1}^{t_p}\tilde{\sigma}_{t_{p-1}}^2(\mathbf{x}_s)$ in terms of the communication threshold D on each client. This leads to the following upper bound about the total number of epochs $B\leq\frac{1+\epsilon}{1-\epsilon}[\frac{1}{D}+\frac{1+\epsilon}{1-\epsilon}(N+L^2/(\lambda D))]2\gamma_{NT}$, and with $D=1/N$, we have $C_{NT}\leq B\cdot N\gamma_{NT}^2=O(N^2\gamma_{NT}^3)$, which completes the proof. □

Remark 2. *Compared with DisKernelUCB's $O(TN^2d)$ communication cost, Approx-DisKernelUCB removes the linear dependence on T, but introduces an additional γ_{NT}^3 dependence due to the communication of the embedded statistics. In situations where $\gamma_{NT}\ll T^{1/3}d^{1/3}$, DisKernelUCB is preferable. As mentioned in Lemma 2.2.10, the value of γ_{NT}, which affects how much the data can be compressed, depends on the specific arm set of the problem and the kernel function of the choice. By Mercer's Theorem, one can represent the kernel using its eigenvalues, and γ_{NT} characterizes how fast its eigenvalues decay. Vakili et al. [93] showed that for kernels whose eigenvalues decay exponentially, i.e., $\lambda_m=O(\exp(-m^{\beta_e}))$, for some $\beta_e>0$, $\gamma_{NT}=O(\log^{1+\frac{1}{\beta_e}}(NT))$. In this case, Approx-DisKernelUCB is far more efficient than DisKernelUCB. This includes Gaussian kernel, which is widely used for GPs and SVMs. For kernels that have polynomially decaying eigenvalues, i.e., $\lambda_m=O(m^{-\beta_p})$, for some $\beta_p>1$, $\gamma_{NT}=O(T^{\frac{1}{\beta_p}}\log^{1-\frac{1}{\beta_p}}(NT))$. Then as long as $\beta_p>3$, Approx-DisKernelUCB still enjoys reduced communication cost.*

2.2.13 Experiment setup & results

In order to evaluate Approx-DisKernelUCB's effectiveness in reducing communication cost, we performed extensive empirical evaluations on both synthetic and real-world datasets, and the results (averaged over 3 runs) are reported in Figure 2.7, 2.8 and 2.9, respectively. We included DisKernelUCB, DisLinUCB [28], OneKernelUCB, and NKernelUCB [72] as baselines, where One-KernelUCB learns a shared bandit model across all clients' aggregated data where data aggregation happens immediately after each new data point becomes available, and N-KernelUCB learns a separated bandit model for each client with no communication. For all the kernelized algorithms, we used the Gaussian kernel $k(x,y)=\exp(-\gamma\|x-y\|^2)$. We did a grid search of $\gamma\in\{0.1,1,4\}$ for kernelized algorithms, and set $D=20$ for DisLinUCB and DisKernelUCB, $D=5$ for Approx-DisKernelUCB. For all algorithms, instead of using their theoretically derived exploration coefficient α, we followed the convention [17, 94] to use grid search for α in $\{0.1,1,4\}$.

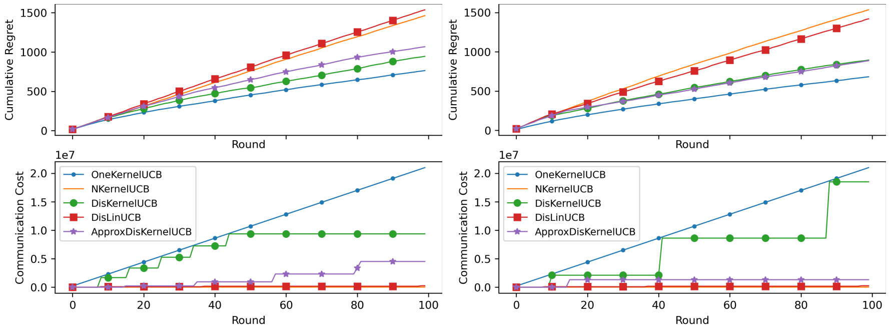
Synthetic dataset We simulated the distributed bandit setting defined in Section 2.2.9, with $d=20, T=100, N=100$ ($NT=10^4$ interactions in total). In each round $l\in[T]$, each client $i\in[N]$ (denote $t=N(l-1)+i$) selects an arm from candidate set $\mathcal{A}_t$, where $\mathcal{A}_t$ is uniformly sampled from a ℓ_2 unit ball, with $|\mathcal{A}_t|=20$. Then the corresponding reward is generated using one of the following reward functions:

$$f_1(\mathbf{x})=\cos(3\mathbf{x}^\top\theta_\star)$$
$$f_2(\mathbf{x})=(\mathbf{x}^\top\theta_\star)^3-3(\mathbf{x}^\top\theta_\star)^2-(\mathbf{x}^\top\theta_\star)+3$$

where the parameter $\theta_\star$ is uniformly sampled from a ℓ_2 unit ball.

UCI Datasets To evaluate Approx-DisKernelUCB's performance in a more challenging and practical scenario, we performed experiments using real-world datasets: MagicTelescope, Mushroom and Shuttle from the UCI Machine Learning Repository [89]. To convert them to contextual bandit problems, we pre-processed these datasets following the steps in [25]. In particular, we partitioned the dataset in to 20 clusters using k-means, and used the centroid of each cluster as the context vector for the arm and the averaged response variable as mean reward (the response variable is binarized by associating one class as 1, and all the others as 0). Then we simulated the distributed bandit learning problem in Section 2.2.9 with $|\mathcal{A}_t|=20$, $T=100$ and $N=100$ ($NT=10^4$ interactions in total).

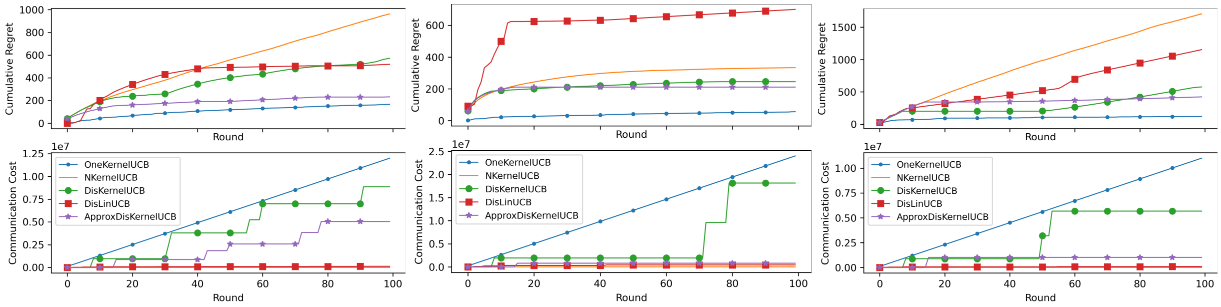
MovieLens and Yelp dataset Yelp dataset, which is released by the Yelp dataset challenge, consists of 4.7 million rating entries for 157 thousand restaurants by 1.18 million users. MovieLens is a dataset consisting of 25 million ratings between 160 thousand users and 60 thousand movies [95]. Following the pre-processing steps in [96], we built the rating matrix by choosing the top 2000 users and top 10000 restaurants/movies and used singular-value decomposition (SVD) to extract a 10-dimension feature vector for each user and restaurant/movie. We treated rating greater than 2 as positive. We simulated the distributed bandit learning problem in Section 2.2.9 with $T = 100$ and $N = 100$ ($NT = 10^4$ interactions in total). In each time step, the candidate set $\mathcal{A}_t$ (with $|\mathcal{A}_t| = 20$) is constructed by sampling an arm with reward 1 and nineteen arms with reward 0 from the arm pool, and the concatenation of user and restaurant/movie feature vector is used as the context vector for the arm (thus $d = 20$).



(a) $\cos(3\mathbf{x}^\top\theta_\star)$

(b) $(\mathbf{x}^\top\theta_\star)^3 - 3(\mathbf{x}^\top\theta_\star)^2 - (\mathbf{x}^\top\theta_\star) + 3$

Figure 2.7: Experiment results on synthetic datasets with different reward function $f(\mathbf{x})$.



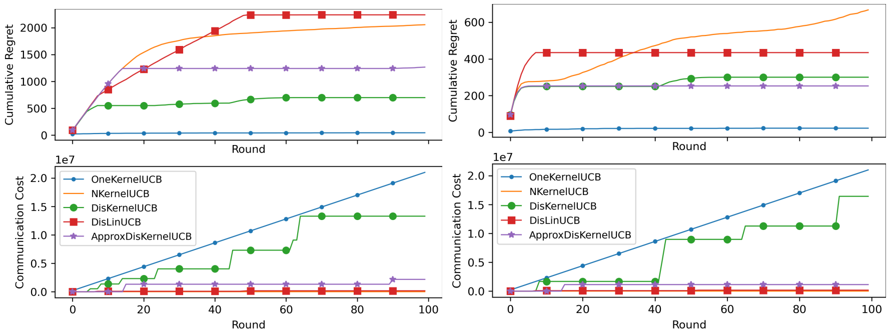
(a) MagicTelescope

(b) Mushroom

(c) Shuttle

Figure 2.8: Experiment results on UCI datasets.

Discussions When examining the experiment results presented in Figure 2.7, 2.8 and 2.9, we can first look at the cumulative regret and communication cost of OneKernelUCB and NKernelUCB, which correspond to the two extreme cases where the clients communicate in every time step to learn a shared model, and each client learns its own model independently with no communication, respectively. OneKernelUCB achieves the smallest cumulative regret in all experiments, while also incurring the highest communication cost, i.e., $O(TN^2d)$. This demonstrates the need of efficient data aggregation across clients for reducing regret. Second, we can observe that DisKernelUCB incurs the second highest communication cost in all experiments due to the transfer of raw data, as we have discussed in Remark 1, which makes it prohibitively expensive for distributed setting. On the other extreme, we can see that DisLinUCB incurs very small communication cost thanks to its closed-form solution, but fails to capture the complicated reward mappings in most of these datasets, e.g. in Figure 2.7(a), 2.8(b) and 2.9(a), it leads to even worse regret than NKernelUCB that learns a kernelized bandit model independently for each client. In comparison, the proposed Approx-DisKernelUCB algorithm enjoys the best of both worlds in most cases, i.e., it can take advantage of the superior modeling power of kernels to reduce regret, while only requiring a relatively low communication cost for clients to collaborate. On all the datasets, Approx-DisKernelUCB achieved comparable regret with DisKernelUCB that maintains exact kernelized estimators, and sometimes even getting very close to OneKernelUCB, e.g., in Figure 2.7(b) and 2.8(a), but its communication cost is only slightly higher than that of DisLinUCB.



(a) MovieLens

(b) Yelp

Figure 2.9: Experiment results on MovieLens & Yelp datasets.

2.2.14 Full proof of DisKernelUCB algorithm

Confidence Ellipsoid for DisKernelUCB

In this section, we construct the confidence ellipsoid for DisKernelUCB as shown in Lemma 2.2.14.

Lemma 2.2.14 (Confidence Ellipsoid for DisKernelUCB)**.** *Let $\delta \in (0,1)$. With probability at least $1-\delta$, for all $t \in [NT], i \in [N]$, we have*

$$\|\hat{\theta}_{t,i} - \theta_\star\|_{\mathbf{A}_{t,i}} \leq \sqrt{\lambda}\|\theta_\star\| + R\sqrt{2\ln(NT/\delta) + \ln(\det(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}/\lambda + \mathbf{I}))}.$$

Proof of Lemma 2.2.14. The analysis is rooted in [74] for kernelized contextual bandit, but with non-trivial extensions to address the dependencies due to the event-triggered distributed communication. This problem also exists in prior works of distributed linear bandit, but was not addressed rigorously (see Lemma H.1. of [28]). First, recall that the Ridge regression estimator

$$\begin{aligned}\hat{\theta}_{t,i} &= \mathbf{A}_{t,i}^{-1} \sum_{s \in \mathcal{D}_t(i)} \phi_s y_s = \mathbf{A}_{t,i}^{-1} \sum_{s \in \mathcal{D}_t(i)} \phi_s(\phi_s^\top \theta_\star + \eta_s) \\ &= \theta_\star - \lambda \mathbf{A}_{t,i}^{-1}\theta_\star + \mathbf{A}_{t,i}^{-1} \sum_{s \in \mathcal{D}_t(i)} \phi_s \eta_s,\end{aligned}$$

and thus, we have

$$\begin{aligned}\|\mathbf{A}_{t,i}^{1/2}(\hat{\theta}_{t,i} - \theta_\star)\| &= \|-\lambda \mathbf{A}_{t,i}^{-1/2}\theta_\star + \mathbf{A}_{t,i}^{-1/2} \sum_{s \in \mathcal{D}_t(i)} \phi_s \eta_s\| \\ &\leq \|\lambda \mathbf{A}_{t,i}^{-1/2}\theta_\star\| + \|\mathbf{A}_{t,i}^{-1/2} \sum_{s \in \mathcal{D}_t(i)} \phi_s \eta_s\| \\ &\leq \sqrt{\lambda}\|\theta_\star\| + \|\mathbf{A}_{t,i}^{-1/2} \sum_{s \in \mathcal{D}_t(i)} \phi_s \eta_s\|\end{aligned} \tag{2.25}$$

where the first inequality is due to the triangle inequality, and the second is due to the property of Rayleigh quotient, i.e., $\|\mathbf{A}_{t,i}^{-1/2}\theta_\star\| \leq \|\theta_\star\|\sqrt{\lambda_{max}(\mathbf{A}_{t,i}^{-1})} \leq \|\theta_\star\|\frac{1}{\sqrt{\lambda}}$.

Difference from standard argument Note that the second term may seem similar to the ones appear in the self-normalized bound in previous works of linear and kernelized bandits [20, 72, 74]. However, a main difference is that $\mathcal{D}_t(i)$, i.e., the sequence of indices for the data points used to update client i, is constructed using the event-trigger as defined in Eq (2.22) . The event-trigger is data-dependent, and thus it is a delayed and permuted version of the original sequence $[t]$. It is delayed in the sense that the length $|\mathcal{D}_t(i)| < t$ unless t is the global synchronization step. It is permuted in the sense that every client receives the data in a different order, i.e., before the synchronization, each client first updates using its local new data, and then receives data from other clients at the synchronization. This prevents us from directly applying Lemma 3.1 of [74], and requires a more careful treatment as shown in the following paragraph.

First, we should note that during the time steps of global synchronization, i.e., $t \in \{t_p\}_{p\in[B]}$, we have $\mathcal{D}_t(i) = [t], \forall i \in [N]$, which recovers the case under centralized setting, i.e., the centralized agent that has access to all data points in the learning system. Therefore, analogous to the proof of RARELY SWITCHING OFUL algorithm in Appendix D of [20], with the standard argument in [74], we have

$$\|\mathbf{A}_{t,i}^{-1/2} \sum_{s\in\mathcal{D}_t(i)} \phi_s \eta_s\| \leq R\sqrt{2\ln(1/\delta) + \ln(\det(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}/\lambda + \mathbf{I}))}$$

for all $t \in \{t_p\}_{p\in[B]}$ and $i \in [N]$, with probability at least $1-\delta$. If our proposed algorithm has no local update, or use the 'hallucinating update' as in [75], then this would suffice. However, the existence of local update requires us to obtain self-normalized bounds for the local models that have been updated using each client's newly collected data after the synchronization step, which leads to the issue mentioned in the previous paragraph. Therefore, we need to address this issue by a union bound over all possible time steps of global synchronization and all clients.

Specifically, consider some time step $t \notin \{t_p\}_{p\in[B]}$ and client i. We denote the time step of the most recent global synchronization to t as t_{last}, and define the filtration $\{\mathcal{F}_s\}_{s=0}^{t_{\text{last}}} \cup \{\mathcal{F}_{s,i}\}_{s=t_{last}+1}^{\infty}$, where the σ-algebra $\mathcal{F}_s = \sigma\big((\mathbf{x}_\tau, \eta_\tau)_{\tau\in[s]}\big)$ for $s \in [0, t_{last}]$, and $\mathcal{F}_{s,i} = \sigma\big((\mathbf{x}_\tau, \eta_\tau)_{\tau\in[t_{\text{last}}]}, (\mathbf{x}_\tau, \eta_\tau)_{\tau\in[t_{\text{last}},s], i_\tau=i}\big)$ for $s \geq t_{\text{last}} + 1$. By applying the standard argument for self-normalized bound using the filtration constructed above and then an union bound over N clients, we have

$$\|\mathbf{A}_{t,i}^{-1/2} \sum_{s\in\mathcal{D}_t(i)} \phi_s \eta_s\| \leq R\sqrt{2\ln(N/\delta) + \ln(\det(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}/\lambda + \mathbf{I}))}$$

for all $t > t_{\text{last}}$ and $i \in [N]$, with probability at least $1-\delta$. As the time step of global synchronization t_{last} is data-dependent, and thus can take any value in $[T]$, we apply another union bound, which finishes the proof.

□

Proof of regret upper bound in Lemma 2.2.10

Based on Lemma 2.2.14 and the arm selection rule in Eq (2.21), we have

$$\begin{aligned} f(\mathbf{x}_t^\star) &\leq \hat{\mu}_{t-1,i_t}(\mathbf{x}_t^\star) + \alpha_{t-1,i_t}\hat{\sigma}_{t-1,i_t}(\mathbf{x}_t^\star) \leq \hat{\mu}_{t-1,i_t}(\mathbf{x}_t) + \alpha_{t-1,i_t}\hat{\sigma}_{t-1,i_t}(\mathbf{x}_t), \\ f(\mathbf{x}_t) &\geq \hat{\mu}_{t-1,i_t}(\mathbf{x}_t) - \alpha_{t-1,i_t}\hat{\sigma}_{t-1,i_t}(\mathbf{x}_t), \end{aligned}$$

and thus $r_t = f(\mathbf{x}_t^\star) - f(\mathbf{x}_t) \leq 2\alpha_{t-1,i_t}\hat{\sigma}_{t-1,i_t}(\mathbf{x}_t)$, for all $t \in [NT]$, with probability at least $1-\delta$. Then following similar steps as DisLinUCB of [28], we can obtain the regret and communication cost upper bound of DisKernelUCB.

We call the time period in-between two consecutive global synchronizations as an epoch, i.e., the p-th epoch refers to $[t_{p-1}+1, t_p]$, where $p \in [B]$ and $0 \leq B \leq NT$ denotes the total number of global synchronizations. Now consider an imaginary centralized agent that has immediate access to each data point in the learning system, and denote by $A_t = \sum_{s=1}^{t} \phi_s \phi_s^\top$ and $\mathbf{K}_{[t],[t]}$ for $t \in [NT]$ the covariance matrix and kernel matrix constructed by this centralized agent. Then similar to [28], we call the p-th epoch a good epoch if

$$\ln\left(\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}\right) \leq 1,$$

otherwise it is a bad epoch. Note that $\ln(\det(I + \lambda^{-1}\mathbf{K}_{[NT],[NT]})) \leq 2\gamma_{NT}$ by definition of γ_{NT}, i.e., the maximum information gain. Since $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}{\det(\mathbf{I})}) + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_2],[t_2]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}) + \cdots + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[NT],[NT]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_B],[t_B]})}) = \ln(\det(I + \lambda^{-1}\mathbf{K}_{[NT],[NT]})) \leq 2\gamma_{NT}$, and due to the pigeonhole principle, there can be at most $2\gamma_{NT}$ bad epochs.

If the instantaneous regret r_t is incurred during a good epoch, we have

$$\begin{aligned} r_t &\leq 2\alpha_{t-1,i_t} \|\phi_t\|_{\mathbf{A}_{t-1,i_t}^{-1}} \leq 2\alpha_{t-1,i_t} \|\phi_t\|_{\mathbf{A}_{t-1}^{-1}} \sqrt{\|\phi_t\|^2_{\mathbf{A}_{t-1,i_t}^{-1}} / \|\phi_t\|^2_{\mathbf{A}_{t-1}^{-1}}} \\ &= 2\alpha_{t-1,i_t} \|\phi_t\|_{\mathbf{A}_{t-1}^{-1}} \sqrt{\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[t-1],[t-1]})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{t-1}(i_t),\mathcal{D}_{t-1}(i_t)})}} \\ &\leq 2\sqrt{e}\alpha_{t-1,i_t} \|\phi_t\|_{\mathbf{A}_{t-1}^{-1}} \end{aligned}$$

where the second inequality is due to Lemma A.8, and the last inequality is due to the definition of good epoch, i.e., $\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t-1],[t-1]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{t-1}(i_t),\mathcal{D}_{t-1}(i_t)})} \leq \frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})} \leq e$.

Define $\alpha_{NT} := \sqrt{\lambda}\|\theta_\star\| + \sqrt{2\ln(NT/\delta) + \ln(\det(\mathbf{K}_{[NT],[NT]}/\lambda + \mathbf{I}))}$. Then using standard arguments, the cumulative regret incurred in all good epochs can be bounded by,

$$\begin{aligned} R_{good} &= \sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) \leq 1\} \sum_{t=t_{p-1}}^{t_p} r_t \leq \sum_{t=1}^{NT} 2\sqrt{e}\alpha_{t-1,i_t} \|\phi_t\|_{\mathbf{A}_{t-1}^{-1}} \\ &\leq 2\sqrt{e}\alpha_{NT} \sum_{t=1}^{NT} \|\phi_t\|_{\mathbf{A}_{t-1}^{-1}} \leq 2\sqrt{e}\alpha_{NT} \sqrt{NT \cdot 2\ln(\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{[NT],[NT]}))} \\ &\leq 2\sqrt{e}\alpha_{NT}\sqrt{NT \cdot 4\gamma_{NT}} = O\Big(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\Big) \end{aligned}$$

where the third inequality is due to Cauchy-Schwartz and Lemma A.9, and the forth is due to the definition of maximum information gain γ_{NT}.

Then we look at the regret incurred during bad epochs. Consider some bad epoch p, and the cumulative regret incurred during this epoch can be bounded by

$$\begin{aligned} \sum_{t=t_{p-1}+1}^{t_p} r_t &= \sum_{i=1}^{N} \sum_{t \in \mathcal{D}_{t_p}(i) \setminus \mathcal{D}_{t_{p-1}}(i)} r_t \leq 2\alpha_{NT} \sum_{i=1}^{N} \sum_{t \in \mathcal{D}_{t_p}(i) \setminus \mathcal{D}_{t_{p-1}}(i)} \|\phi_t\|_{\mathbf{A}_{t-1,i}^{-1}} \\ &\leq 2\alpha_{NT} \sum_{i=1}^{N} \sqrt{(|\mathcal{D}_{t_p}(i)| - |\mathcal{D}_{t_{p-1}}(i)|) \sum_{t \in \mathcal{D}_{t_p}(i) \setminus \mathcal{D}_{t_{p-1}}(i)} \|\phi_t\|^2_{\mathbf{A}_{t-1,i}^{-1}}} \\ &\leq 2\alpha_{NT} \sum_{i=1}^{N} \sqrt{2(|\mathcal{D}_{t_p}(i)| - |\mathcal{D}_{t_{p-1}}(i)|) \ln(\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{t_p}(i),\mathcal{D}_{t_p}(i)})}{\det(\mathbf{I} + \lambda^{-1}K_{\mathcal{D}_{t_{p-1}}(i),\mathcal{D}_{t_{p-1}}(i)})})} \\ &\leq 2\sqrt{2}\alpha_{NT}N\sqrt{D} \end{aligned}$$

where the last inequality is due to our event-trigger in Eq (2.22). Since there can be at most $2\gamma_{NT}$ bad epochs, the cumulative regret incurred in all bad epochs

$$R_{bad} \leq 2\gamma_{NT} \cdot 2\sqrt{2}\alpha_{NT}N\sqrt{D} = O\Big(ND^{0.5}(\|\theta_\star\|\gamma_{NT} + \gamma_{NT}^{1.5})\Big)$$

Combining cumulative regret incurred during both good and bad epochs, we have

$$R_{NT} = R_{good} + R_{bad} = O\big((\sqrt{NT} + N\sqrt{D\gamma_{NT}})(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\big)$$

Proof of communication upper bound in Lemma 2.2.10

For some $\alpha > 0$, there can be at most $\lceil \frac{NT}{\alpha} \rceil$ epochs with length larger than α. With our event-trigger design, we have $(|\mathcal{D}_{t_p}(i_{t_p})| - |\mathcal{D}_{t_{p-1}}(i_{t_p})|) \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}K_{[t_{p-1}],[t_{p-1}]})}) \geq (|\mathcal{D}_{t_p}(i_{t_p})| - |\mathcal{D}_{t_{p-1}}(i_{t_p})|) \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{t_p}(i_{t_p}),\mathcal{D}_{t_p}(i_{t_p})})}{\det(\mathbf{I}+\lambda^{-1}K_{\mathcal{D}_{t_{p-1}}(i_{t_p}),\mathcal{D}_{t_{p-1}}(i_{t_p})})}) \geq D$ for any epoch $p \in [B]$, where i_{t_p} is the client who triggers the global synchronization at time step t_p. Then if the length of certain epoch p is smaller than α, i.e., $t_p - t_{p-1} \leq \alpha$, we have $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}K_{[t_{p-1}],[t_{p-1}]})}) \geq \frac{ND}{\alpha}$. Since $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}{\det(\mathbf{I})}) + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_2],[t_2]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_1],[t_1]})}) + \cdots + \ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_B],[t_B]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{B-1}],[t_{B-1}]})}) \leq \ln(\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[NT],[NT]})) \leq 2\gamma_{NT}$, the total number of such epochs is upper bounded by $\lceil \frac{2\gamma_{NT}\alpha}{ND} \rceil$. Combining the two terms, the total number of epochs can be bounded by,

$$B \leq \lceil \frac{NT}{\alpha} \rceil + \lceil \frac{2\gamma_{NT}\alpha}{ND} \rceil$$

where the LHS can be minimized using the AM-GM inequality, i.e., $B \leq \sqrt{\frac{NT}{\alpha}\frac{2\gamma_{NT}\alpha}{ND}} = \sqrt{\frac{2\gamma_{NT}T}{D}}$. To obtain the optimal order of regret, we set $D = O(\frac{T}{N\gamma_{NT}})$, so that $R_{NT} = O\big(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\big)$. And the total number of epochs $B = O(\sqrt{N}\gamma_{NT})$. However, we should note that as DisKernelUCB communicates all the unshared raw data at each global synchronization, the total communication cost mainly depends on when the last global synchronization happens. Since the sequence of candidate sets $\{\mathcal{A}_t\}_{t\in[NT]}$, which controls the growth of determinant, is an arbitrary subset of $\mathcal{A}$, the time of last global synchronization could happen at the last time step $t = NT$. Therefore, $C_T = O(N^2Td)$ in such a worst case.

2.2.15 Full proof of Approx-DisKernelUCB algorithm

Derivation of the Approximated Mean and Variance in Section 2.2.11

For simplicity, subscript t is omitted in this section. The approximated Ridge regression estimator for dataset $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{D}}$ is formulated as

$$\tilde{\theta} = \arg\min_{\theta\in\mathcal{H}} \sum_{s\in\mathcal{D}} \Big((\mathbf{P}_{\mathcal{S}}\phi_s)^\top\theta - y_s\Big)^2 + \lambda\|\theta\|_2^2$$

where $\mathcal{D}$ denotes the sequence of time indices for data in the original dataset, $\mathcal{S} \subseteq \mathcal{D}$ denotes the time indices for data in the dictionary, and $\mathbf{P}_{\mathcal{S}} \in \mathbb{R}^{p\times p}$ denotes the orthogonal projection matrix defined by $\mathcal{S}$. Then by taking derivative and setting it to zero, we have $(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})\tilde{\theta} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{y}_{\mathcal{D}}$, and thus $\tilde{\theta} = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}$, where $\tilde{\mathbf{A}} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I}$ and $\tilde{\mathbf{b}} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{y}_{\mathcal{D}}$.

Hence, the approximated mean reward and variance for some arm $\mathbf{x}$ are

$$\tilde{\mu}_{t,i}(\mathbf{x}) = \phi(\mathbf{x})^\top\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}$$
$$\tilde{\sigma}_{t,i}(\mathbf{x}) = \sqrt{\phi(\mathbf{x})^\top\tilde{\mathbf{A}}^{-1}\phi(\mathbf{x})}$$

To obtain their kernelized representation, we rewrite

$$(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})\tilde{\theta} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{y}_{\mathcal{D}}$$
$$\Leftrightarrow\ \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{y}_{\mathcal{D}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\tilde{\theta}) = \lambda\tilde{\theta}$$
$$\Leftrightarrow\ \tilde{\theta} = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\rho$$

where $\rho := \frac{1}{\lambda}(\mathbf{y}_{\mathcal{D}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\tilde{\theta}) = \frac{1}{\lambda}(\mathbf{y}_{\mathcal{D}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\rho)$. Solving this equation, we get $\rho = (\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{y}_{\mathcal{D}}$. Note that $\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}} = \mathbf{P}_{\mathcal{S}}$, since projection matrix $\mathbf{P}_{\mathcal{S}}$ is idempotent. Moreover, we have $(\mathbf{\Phi}^\top\mathbf{\Phi} + \lambda\mathbf{I})\mathbf{\Phi}^\top = \mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda\mathbf{I})$, and $(\mathbf{\Phi}^\top\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^\top = \mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda\mathbf{I})^{-1}$. Therefore, we can rewrite the approximated mean for some arm $\mathbf{x}$ as

$$\begin{aligned}\tilde{\mu}(\mathbf{x}) &= \phi(\mathbf{x})^\top \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{y}_{\mathcal{D}} \\ &= (\mathbf{P}_{\mathcal{S}}^{1/2}\phi(\mathbf{x}))^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}^{1/2})^\top[\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}^{1/2}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}^{1/2})^\top + \lambda\mathbf{I}]^{-1}\mathbf{y}_{\mathcal{D}} \\ &= (\mathbf{P}_{\mathcal{S}}^{1/2}\phi(\mathbf{x}))^\top(\mathbf{P}_{\mathcal{S}}^{1/2}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}^{1/2} + \lambda\mathbf{I})^{-1}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}^{1/2})^\top\mathbf{y}_{\mathcal{D}} \\ &= z(\mathbf{x};\mathcal{S})^\top\left(\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D};\mathcal{S}} + \lambda\mathbf{I}\right)^{-1}\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top\mathbf{y}_{\mathcal{D}}\end{aligned}$$

To derive the approximated variance, we start from the fact that $(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})\phi(\mathbf{x}) = \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda\phi(\mathbf{x})$, so

$$\begin{aligned}\phi(\mathbf{x}) &= (\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x}) \\ &= \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x})\end{aligned}$$

Then we have

$$\begin{aligned}&\phi(\mathbf{x})^\top\phi(\mathbf{x}) \\ =&\big\{\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x})\big\}^\top \\ &\quad\big\{\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x})\big\} \\ =&\phi(\mathbf{x})^\top\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) \\ &+ 2\lambda\phi(\mathbf{x})^\top\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x}) \\ &+ \lambda\phi(\mathbf{x})^\top(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\lambda\mathbf{I}(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x}) \\ =&\phi(\mathbf{x})^\top\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x}) + \lambda\phi(\mathbf{x})^\top(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x})\end{aligned}$$

By rearranging terms, we have

$$\begin{aligned}\tilde{\sigma}^2(\mathbf{x}) =&\phi(\mathbf{x})^\top(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\phi(\mathbf{x}) \\ =&\frac{1}{\lambda}\{\phi(\mathbf{x})^\top\phi(\mathbf{x}) - \phi(\mathbf{x})^\top\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top(\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}}\mathbf{P}_{\mathcal{S}}\phi(\mathbf{x})\} \\ =&\frac{1}{\lambda}\{k(\mathbf{x},\mathbf{x}) - z(\mathbf{x};\mathcal{S})^\top\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D};\mathcal{S}}[\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D};\mathcal{S}} + \lambda\mathbf{I}]^{-1}z(\mathbf{x}|\mathcal{S})\}\end{aligned}$$

Omitted proof for Lemma 2.2.11 and Lemma 2.2.12 in Section 2.2.12

Proof of Lemma 2.2.11. In the following, we analyze the $\epsilon_{t,i}$-accuracy of the dictionary for all t, i.

At the time steps when global synchronization happens, i.e., t_p for $p \in [B]$, $\mathcal{S}_{t_p}$ is sampled from $[t_p] = \mathcal{D}_{t_p}(i)$ using approximated variance $\tilde{\sigma}^2_{t_{p-1},i}$. In this case, the accuracy of the dictionary only depends on the RLS procedure, and Calandriello et al. [75] have already showed that the following guarantee on the accuracy and size of dictionary holds $\forall t \in \{t_p\}_{p\in[B]}$.

Lemma 2.2.15 (Lemma 2 of [75])**.** *Under the condition that $\bar{q} = 6\frac{1+\epsilon}{1-\epsilon}\log(4NT/\delta)/\epsilon^2$, for some $\epsilon \in [0,1)$, with probability at least $1-\delta$, we have $\forall t \in \{t_p\}_{p\in[B]}$ that the dictionary $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_t}$ is ϵ-accurate w.r.t. $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{D}_t(i)}$, and $\frac{1-\epsilon}{1+\epsilon}\sigma_t^2(\mathbf{x}) \leq \tilde{\sigma}_t^2(\mathbf{x}) \leq \frac{1+\epsilon}{1-\epsilon}\sigma_t^2(\mathbf{x}), \forall \mathbf{x} \in \mathcal{A}$. Moreover, the size of dictionary $|\mathcal{S}_t| \leq 3(1+L^2/\lambda)\frac{1+\epsilon}{1-\epsilon}\bar{q}\tilde{d}$, where $\tilde{d} := Tr(\mathbf{K}_{[NT],[NT]}(\mathbf{K}_{[NT],[NT]} + \lambda\mathbf{I})^{-1})$ denotes the effective dimension of the problem, and it is known that $\tilde{d} = O(\gamma_{NT})$ [72].*

Lemma 2.2.15 guarantees that for all $t \in \{t_p\}_{p\in[B]}$, the dictionary has a constant accuracy, i.e., $\epsilon_{t,i} = \epsilon, \forall i$. In addition, since the dictionary is fixed for $t \notin \{t_p\}_{p\in[B]}$, its size $\mathcal{S}_t = O(\gamma_{NT}), \forall t \in [NT]$.

Then for time steps $t \notin \{t_p\}_{p\in[B]}$, due to the local update, the accuracy of the dictionary will degrade. However, thanks to our event-trigger in Eq (2.24), the extent of such degradation can be controlled, i.e., a new dictionary update will be triggered before the previous dictionary becomes completely irrelevant. This is shown in Lemma 2.2.16 below.

Lemma 2.2.16. *Under the condition that* $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{S}_{t_p}}$ *is* ϵ*-accurate w.r.t.* $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{D}_{t_p}(i)}$, $\forall t \in [t_p{+}1, t_{p+1}], i \in [N]$, $\mathcal{S}_{t_p}$ *is* $\left(\epsilon + 1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}\right)$*-accurate w.r.t.* $\mathcal{D}_t(i)$.

Combining Lemma 2.2.15 and Lemma 2.2.16 finishes the proof of Lemma 2.2.11. □

Proof of Lemma 2.2.16. Similar to [73], we can rewrite the ϵ-accuracy condition of $\mathcal{S}_{t_p}$ w.r.t. $\mathcal{D}_t(i)$ for $t \in [t_p{+}1, t_{p+1}]$ as

$$\begin{aligned}
&(1-\epsilon_{t,i})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I}) \preceq \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \bar{\mathbf{S}}_{t,i}^\top \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I} \preceq (1+\epsilon_{t,i})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})\\
\Leftrightarrow & -\epsilon_{t,i}(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I}) \preceq \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \bar{\mathbf{S}}_{t,i}^\top \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} - \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} \preceq \epsilon_{t,i}(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})\\
\Leftrightarrow & -\epsilon_{t,i}\mathbf{I} \preceq (\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \bar{\mathbf{S}}_{t,i}^\top \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} - \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2} \preceq \epsilon_{t,i}\mathbf{I}\\
\Leftrightarrow & \|(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \bar{\mathbf{S}}_{t,i}^\top \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} - \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)})(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}\| \leq \epsilon_{t,i}\\
\Leftrightarrow & \|\sum_{s\in\mathcal{D}_{t_p}} (\frac{q_s}{\tilde{p}_s} - 1)\psi_s\psi_s^\top + \sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} (0-1)\psi_s\psi_s^\top\| \leq \epsilon_{t,i}
\end{aligned}$$

where $\psi_s = (\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}\phi_s$. Notice that the second term in the norm has weight -1 because the dictionary $\mathcal{S}_{t_p}$ is fixed after t_p. With triangle inequality, now it suffices to bound

$$\|\sum_{s\in\mathcal{D}_{t_p}} (\frac{q_s}{\tilde{p}_s} - 1)\psi_{s,j}\psi_s^\top + \sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} (0-1)\psi_s\psi_s^\top\| \leq \|\sum_{s\in\mathcal{D}_{t_p}} (\frac{q_s}{\tilde{p}_s} - 1)\psi_s\psi_s^\top\| + \|\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} \psi_s\psi_s^\top\|.$$

We should note that the first term corresponds to the approximation accuracy of $\mathcal{S}_{t_p}$ w.r.t. the dataset $\mathcal{D}_{t_p}$. And under the condition that it is ϵ-accurate w.r.t. $\mathcal{D}_{t_p}$, we have $\|\sum_{s\in\mathcal{D}_{t_p}} (\frac{q_s}{\tilde{p}_s} - 1)\psi_s\psi_s^\top\| \leq \epsilon$. The second term measures the difference between $\mathcal{D}_t(i)$ compared with $\mathcal{D}_{t_p}$, which is unique to our work. We can bound it as follows.

$$\begin{aligned}
&\|\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} \psi_s\psi_s^\top\|\\
=&\|(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}(\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} \phi_s\phi_s^\top)(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}\|\\
=&\max_{\phi\in\mathcal{H}} \frac{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}(\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} \phi_s\phi_s^\top)(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1/2}\phi}{\phi^\top\phi}\\
=&\max_{\phi\in\mathcal{H}} \frac{\phi^\top(\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_p}} \phi_s\phi_s^\top)\phi}{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})\phi}\\
=&1 - \min_{\phi\in\mathcal{H}} \frac{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_{t_p}}^\top \mathbf{\Phi}_{\mathcal{D}_{t_p}} + \lambda \mathbf{I})\phi}{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})\phi}\\
=&1 - \frac{1}{\max_{\phi\in\mathcal{H}} \frac{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_{t_p}}^\top \mathbf{\Phi}_{\mathcal{D}_{t_p}} + \lambda \mathbf{I})^{-1}\phi}{\phi^\top(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1}\phi}}\\
=&1 - \frac{1}{\max_{\mathbf{x}} \frac{\sigma_{t_p,i}^2(\mathbf{x})}{\sigma_{t,i}^2(\mathbf{x})}}
\end{aligned}$$

We can further bound the term $\frac{\sigma^2_{t_p,i}(\mathbf{x})}{\sigma^2_{t,i}(\mathbf{x})}$ using the threshold of the event-trigger in Eq (2.24). For any $\mathbf{x} \in \mathbb{R}^d$,

$$\frac{\sigma^2_{t_p,i}(\mathbf{x})}{\sigma^2_{t,i}(\mathbf{x})} \leq 1 + \sum_{s \in \mathcal{D}_t(i) \setminus \mathcal{D}_{t_p}} \hat{\sigma}^2_{t_p,i}(\mathbf{x}_s) \leq 1 + \frac{1+\epsilon}{1-\epsilon} \sum_{s \in \mathcal{D}_t(i) \setminus \mathcal{D}_{t_p}} \tilde{\sigma}^2_{t_p,i}(\mathbf{x}_s) \leq 1 + \frac{1+\epsilon}{1-\epsilon} D$$

where the first inequality is due to Lemma A.10, the second is due to Lemma 2.2.15, and the third is due to the event-trigger in Eq (2.24). Putting everything together, we have that if $\mathcal{S}_{t_p}$ is ϵ-accurate w.r.t. $\mathcal{D}_{t_p}$, then it is $\big(\epsilon + 1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}\big)$-accurate w.r.t. dataset $\mathcal{D}_t(i)$, which finishes the proof. □

Proof of Lemma 2.2.12. To prove Lemma 2.2.12, we need the following lemma.

Lemma 2.2.17. *We have* $\forall t, i$ *that*

$$\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}} \leq \Big(\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\| + \sqrt{\lambda}\Big)\|\theta_\star\| + R\sqrt{4 \ln N/\delta + 2 \ln \det((1+\lambda)\mathbf{I} + \mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)})}$$

with probability at least $1 - \delta$.

Now we are ready to prove Lemma 2.2.12 by further bounding the term $\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}})\|$. Recall that $\bar{\mathbf{S}}_{t,i} \in \mathbb{R}^{|\mathcal{D}_t(i)| \times |\mathcal{D}_t(i)|}$ denotes the diagonal matrix, whose s-th diagonal entry equals to $\frac{q_s}{\sqrt{\tilde{p}_s}}$, where $q_s = 1$ if $s \in \mathcal{S}_{t_p}$ and 0 otherwise (note that for $s \notin \mathcal{S}_{t_p}$, we set $\tilde{p}_s = 1$, so $q_s/\tilde{p}_s = 0$). Therefore, $\forall s \in \mathcal{D}_t(i) \setminus \mathcal{D}_{t_p}$, $q_s = 0$, as the dictionary is fixed after t_p. We can rewrite $\mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \bar{\mathbf{S}}^\top_{t,i} \bar{\mathbf{S}}_{t,i} \mathbf{\Phi}_{\mathcal{D}_t(i)} = \sum_{s \in \mathcal{D}_t(i)} \frac{q_s}{\tilde{p}_s} \phi_s \phi_s^\top$, where $\phi_s := \phi(\mathbf{x}_s)$. Then by definition of the spectral norm $\|\cdot\|$, and the properties of the projection matrix $\mathbf{P}_{\mathcal{S}_{t_p}}$, we have

$$\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}})\| = \sqrt{\lambda_{\max}\big(\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}})^2 \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\big)} = \sqrt{\lambda_{\max}\big(\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}}) \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\big)}. \tag{2.26}$$

Moreover, due to Lemma 2.2.16, we know $\mathcal{S}_{t_p}$ is $\epsilon_{t,i}$-accurate w.r.t. $\mathcal{D}_t(i)$ for $t \in [t_p + 1, t_{p+1}]$, where $\epsilon_{t,i} = \big(\epsilon + 1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}\big)$, so we have $\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}} \preceq \frac{\lambda}{1-\epsilon_{t,i}}(\mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1}$ by the property of ϵ-accuracy (Proposition 10 of [73]). Therefore, by substituting this back to Eq (2.26), we have

$$\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}_{t_p}})\| \leq \sqrt{\lambda_{\max}\big(\frac{\lambda}{1-\epsilon_{t,i}} \mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda \mathbf{I})^{-1} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)}\big)} \leq \sqrt{\frac{\lambda}{-\epsilon + \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}}}$$

which finishes the proof. □

Proof of Lemma 2.2.17. Recall that the approximated kernel Ridge regression estimator for $\theta_\star$ is defined as

$$\tilde{\theta}_{t,i} = \tilde{\mathbf{A}}^{-1}_{t,i} \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{y}_{\mathcal{D}_t(i)}$$

where $\mathbf{P}_{\mathcal{S}}$ is the orthogonal projection matrix for the Nyström approximation, and $\tilde{\mathbf{A}}_{t,i} = \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{\Phi}_{\mathcal{D}_t(i)} \mathbf{P}_{\mathcal{S}} + \lambda \mathbf{I}$. Then our goal is to bound

$$\begin{aligned}
&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i} (\tilde{\theta}_{t,i} - \theta_\star) \\
=&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i} (\tilde{\mathbf{A}}^{-1}_{t,i} \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{y}_{\mathcal{D}_t(i)} - \theta_\star) \\
=&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i} [\tilde{\mathbf{A}}^{-1}_{t,i} \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} (\mathbf{\Phi}_{\mathcal{D}_t(i)} \theta_\star + \eta_{\mathcal{D}_t(i)}) - \theta_\star] \\
=&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i} (\tilde{\mathbf{A}}^{-1}_{t,i} \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \mathbf{\Phi}_{\mathcal{D}_t(i)} \theta_\star - \theta_\star) + (\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i} \tilde{\mathbf{A}}^{-1}_{t,i} \mathbf{P}_{\mathcal{S}} \mathbf{\Phi}^\top_{\mathcal{D}_t(i)} \eta_{\mathcal{D}_t(i)}
\end{aligned}$$

Bounding the first term To bound the first term, we begin with rewriting

$$\begin{aligned}
&\tilde{\mathbf{A}}_{t,i}(\tilde{\mathbf{A}}_{t,i}^{-1}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}\theta_\star - \theta_\star) \\
=&\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}\theta_\star - \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}\mathbf{P}_{\mathcal{S}}\theta_\star - \lambda\theta_\star \\
=&\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\theta_\star - \lambda\theta_\star
\end{aligned}$$

and by substituting this into the first term, we have

$$\begin{aligned}
&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i}(\tilde{\mathbf{A}}_{t,i}^{-1}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}\theta_\star - \theta_\star) \\
=&(\tilde{\theta}_{t,i} - \theta_\star)^\top \mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\theta_\star - \lambda(\tilde{\theta}_{t,i} - \theta_\star)^\top\theta_\star \\
=&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i}^{1/2}\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\theta_\star - \lambda(\tilde{\theta}_{t,i} - \theta_\star)^\top\tilde{\mathbf{A}}_{t,i}^{1/2}\tilde{\mathbf{A}}_{t,i}^{-1/2}\theta_\star \\
\leq&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\big(\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\theta_\star\| + \lambda\|\theta_\star\|_{\tilde{\mathbf{A}}_{t,i}^{-1}}\big) \\
\leq&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\big(\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\|\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\|\|\theta_\star\| + \sqrt{\lambda}\|\theta_\star\|\big) \\
\leq&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\big(\|\mathbf{\Phi}_{\mathcal{D}_t(i)}(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\| + \sqrt{\lambda}\big)\|\theta_\star\|
\end{aligned}$$

where the first inequality is due to Cauchy Schwartz, and the last inequality is because $\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\| = \sqrt{\mathbf{\Phi}_{\mathcal{D}_t(i)}\mathbf{P}_{\mathcal{S}}(\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\mathbf{\Phi}_{\mathcal{D}_t(i)}\mathbf{P}_{\mathcal{S}} + \lambda\mathbf{I})^{-1}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top} \leq 1$.

Bounding the second term By applying Cauchy-Schwartz inequality to the second term, we have

$$\begin{aligned}
&(\tilde{\theta}_{t,i} - \theta_\star)^\top \tilde{\mathbf{A}}_{t,i}\tilde{\mathbf{A}}_{t,i}^{-1}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \eta_{\mathcal{D}_t(i)} \\
\leq&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \eta_{\mathcal{D}_t(i)}\| \\
=&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{A}_{t,i}^{1/2}\mathbf{A}_{t,i}^{-1/2}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \eta_{\mathcal{D}_t(i)}\| \\
\leq&\|\tilde{\theta}_{t,i} - \theta_\star\|_{\tilde{\mathbf{A}}_{t,i}}\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{A}_{t,i}^{1/2}\|\|\mathbf{A}_{t,i}^{-1/2}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top \eta_{\mathcal{D}_t(i)}\|
\end{aligned}$$

Note that $\mathbf{P}_{\mathcal{S}}\mathbf{A}_{t,i}\mathbf{P}_{\mathcal{S}} = \mathbf{P}_{\mathcal{S}}(\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\mathbf{\Phi}_{\mathcal{D}_t(i)} + \lambda\mathbf{I})\mathbf{P}_{\mathcal{S}} = \tilde{\mathbf{A}}_{t,i} + \lambda(\mathbf{P}_{\mathcal{S}} - \mathbf{I})$ and $\mathbf{P}_{\mathcal{S}} \preceq \mathbf{I}$, so we have

$$\begin{aligned}
\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{A}_{t,i}^{1/2}\| &= \sqrt{\|\tilde{\mathbf{A}}_{t,i}^{-1/2}\mathbf{P}_{\mathcal{S}}\mathbf{A}_{t,i}^{1/2}\mathbf{A}_{t,i}^{1/2}\mathbf{P}_{\mathcal{S}}\tilde{\mathbf{A}}_{t,i}^{-1/2}\|} \leq \sqrt{\|\tilde{\mathbf{A}}_{t,i}^{-1/2}(\tilde{\mathbf{A}}_{t,i} + \lambda(\mathbf{P}_{\mathcal{S}} - \mathbf{I}))\tilde{\mathbf{A}}_{t,i}^{-1/2}\|} \\
&= \sqrt{\|\mathbf{I} + \lambda\tilde{\mathbf{A}}_{t,i}^{-1/2}(\mathbf{P}_{\mathcal{S}} - \mathbf{I}))\tilde{\mathbf{A}}_{t,i}^{-1/2}\|} \leq \sqrt{1 + \lambda\|\tilde{\mathbf{A}}_{t,i}^{-1}\|\|\mathbf{P}_{\mathcal{S}} - \mathbf{I})\|} \\
&\leq \sqrt{1 + \lambda\cdot\lambda^{-1}\cdot 1} = \sqrt{2}
\end{aligned}$$

Then using the self-normalized bound derived for Lemma 2.2.14, the term $\|\mathbf{A}_{t,i}^{-1/2}\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\eta_{\mathcal{D}_t(i)}\| = \|\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\eta_{\mathcal{D}_t(i)}\|_{\mathbf{A}_{t,i}^{-1}}$ can be bounded by

$$\begin{aligned}
\|\mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\eta_{\mathcal{D}_t(i)}\|_{\mathbf{A}_{t,i}^{-1}} &\leq R\sqrt{2\ln(NT/\delta) + \ln(\det(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)}/\lambda + \mathbf{I}))} \\
&\leq R\sqrt{2\ln(NT/\delta) + 2\gamma_{NT}}
\end{aligned}$$

for $\forall t, i$, with probability at least $1 - \delta$. Combining everything finishes the proof. □

Omitted proof of Theorem 2.2.13 in Section 2.2.12

Regret analysis Consider some time step $t \in [t_{p-1}+1, t_p]$, where $p \in [B]$. Due to Lemma 2.2.12, i.e., the confidence ellipsoid for approximated estimator, and the fact that $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{A}_{t,i}} \tilde{\mu}_{t-1,i}(\mathbf{x}) + \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x})$, we have

$$\begin{aligned} f(\mathbf{x}_t^\star) &\leq \tilde{\mu}_{t-1,i}(\mathbf{x}_t^\star) + \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t^\star) \leq \tilde{\mu}_{t-1,i}(\mathbf{x}_t) + \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t), \\ f(\mathbf{x}_t) &\geq \tilde{\mu}_{t-1,i}(\mathbf{x}_t) - \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t), \end{aligned}$$

and thus $r_t = f(\mathbf{x}_t^\star) - f(\mathbf{x}_t) \leq 2\alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t)$, where

$$\alpha_{t-1,i} = \left(\frac{1}{\sqrt{-\epsilon + \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}}} + 1\right)\sqrt{\lambda}\|\theta_\star\| + R\sqrt{4\ln NT/\delta + 2\ln\det((1+\lambda)\mathbf{I} + \mathbf{K}_{\mathcal{D}_{t-1}(i),\mathcal{D}_{t-1}(i)})}.$$

Note that, different from Section 2.2.14 the $\alpha_{t-1,i}$ term now depends on the threshold D and accuracy constant ϵ, as a result of the approximation error. As we will see in the following paragraphs, their values need to be set properly in order to bound $\alpha_{t-1,i}$.

We begin the regret analysis of Approx-DisKernelUCB with the same decomposition of good and bad epochs as in Section 2.2.14, i.e., we call the p-th epoch a good epoch if $\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) \leq 1$, otherwise it is a bad epoch. Moreover, due to the pigeon-hold principle, there can be at most $2\gamma_{NT}$ bad epochs.

As we will show in the following paragraphs, using Lemma 2.2.15, we can obtain a similar bound for the cumulative regret in good epochs as that in Section 2.2.14, but with additional dependence on D and ϵ. The proof mainly differs in the bad epochs, where we need to use the event-trigger in Eq (2.24) to bound the cumulative regret in each bad epoch. Compared with Eq (2.22), Eq (2.24) does not contain the number of local updates on each client since last synchronization., and as mentioned in Section 2.3.10, this introduces a $\sqrt{T}$ factor in the regret bound for bad epochs in place of the $\sqrt{\gamma_{NT}}$ term in Section 2.2.14.

Cumulative Regret in Good Epochs Let's first consider some time step t in a good epoch p, i.e., $t \in [t_{p-1}+1, t_p]$, and we have the following bound on the instantaneous regret

$$\begin{aligned} r_t &\leq 2\alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t) \leq 2\alpha_{t-1,i}\tilde{\sigma}_{t_{p-1},i}(\mathbf{x}_t) \leq 2\alpha_{t-1,i}\frac{1+\epsilon}{1-\epsilon}\sigma_{t_{p-1},i}(\mathbf{x}_t) \\ &= 2\alpha_{t-1,i}\frac{1+\epsilon}{1-\epsilon}\sqrt{\phi_t^\top A_{t_{p-1}}^{-1}\phi_t} \leq 2\alpha_{t-1,i}\frac{1+\epsilon}{1-\epsilon}\sqrt{\phi_t^\top A_{t-1}^{-1}\phi_t}\sqrt{\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t-1],[t-1]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}} \\ &\leq 2\sqrt{e}\frac{1+\epsilon}{1-\epsilon}\alpha_{t-1,i}\sqrt{\phi_t^\top A_{t-1}^{-1}\phi_t} \end{aligned}$$

where the second inequality is because the (approximated) variance is non-decreasing, the third inequality is due to Lemma 2.2.15, the forth is due to Lemma A.8, and the last is because in a good epoch, we have $\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t-1],[t-1]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})} \leq \frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})} \leq e$ for $t \in [t_{p-1}+1, t_p]$.

Therefore, the cumulative regret incurred in all good epochs, denoted by R_{good}, is upper bounded by

$$\begin{aligned} R_{good} &\leq 2\sqrt{e}\frac{1+\epsilon}{1-\epsilon}\sum_{t=1}^{NT}\alpha_{t-1,i}\sqrt{\phi_t^\top A_{t-1}^{-1}\phi_t} \leq 2\sqrt{e}\frac{1+\epsilon}{1-\epsilon}\alpha_{NT}\sqrt{NT\cdot\sum_{t=1}^{NT}\phi_t^\top A_{t-1}^{-1}\phi_t} \\ &\leq 2\sqrt{e}\frac{1+\epsilon}{1-\epsilon}\alpha_{NT}\sqrt{NT\cdot 2\gamma_{NT}} \end{aligned}$$

where $\alpha_{NT} := \left(\frac{1}{\sqrt{-\epsilon+\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}D}}} + 1\right)\sqrt{\lambda}\|\theta_\star\| + R\sqrt{4\ln NT/\delta + 2\ln\det((1+\lambda)\mathbf{I} + \mathbf{K}_{[NT],[NT]})}$.

Cumulative Regret in Bad Epochs The cumulative regret incurred in this bad epoch is

$$\begin{aligned}
&\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{t=t_{p-1}+1}^{t_p} r_t \\
&\leq 2\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{t=t_{p-1}+1}^{t_p} \alpha_{t-1,i}\tilde{\sigma}_{t-1,i}(\mathbf{x}_t) \\
&\leq 2\alpha_{NT}\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{i=1}^{N} \sum_{t\in\mathcal{N}_{t_p}(i)\setminus\mathcal{N}_{t_{p-1}}(i)} \tilde{\sigma}_{t-1,i}(\mathbf{x}_t) \\
&\leq 2\alpha_{NT}\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{i=1}^{N} \sqrt{(|\mathcal{N}_{t_p}(i)| - |\mathcal{N}_{t_{p-1}}(i)|) \sum_{t\in\mathcal{N}_{t_p}(i)\setminus\mathcal{N}_{t_{p-1}}(i)} \tilde{\sigma}^2_{t-1,i}(\mathbf{x}_t)} \\
&\leq 2\alpha_{NT}\sqrt{D}\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{i=1}^{N} \sqrt{(|\mathcal{N}_{t_p}(i)| - |\mathcal{N}_{t_{p-1}}(i)|)} \\
&\leq 2\alpha_{NT}\sqrt{D}\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sum_{i=1}^{N} \sqrt{\frac{t_p - t_{p-1}}{N}} \\
&\leq 2\alpha_{NT}\sqrt{DN}\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\} \sqrt{t_p - t_{p-1}} \\
&\leq 2\alpha_{NT}\sqrt{DN}\sqrt{\sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\}(t_p - t_{p-1}) \cdot \sum_{p=1}^{B} \mathbb{1}\{\ln(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_p],[t_p]})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[t_{p-1}],[t_{p-1}]})}) > 1\}} \\
&\leq 2\alpha_{NT}\sqrt{DN}\sqrt{2NT\gamma_{NT}}
\end{aligned}$$

where the third inequality is due to the Cauchy-Schwartz inequality, the forth is due to our event-trigger in Eq (2.24), the fifth is due to our assumption that clients interact with the environment in a round-robin manner, the sixth is due to the Cauchy-Schwartz inequality again, and the last is due to the fact that there can be at most $2\gamma_{NT}$ bad epochs.

Combining cumulative regret incurred during both good and bad epochs, we have

$$R_{NT} \leq R_{good} + R_{bad} \leq 2\sqrt{e}\frac{1+\epsilon}{1-\epsilon}\alpha_{NT}\sqrt{NT \cdot 2\gamma_{NT}} + 2\alpha_{NT}\sqrt{DN}\sqrt{2NT\gamma_{NT}}$$

Communication cost analysis Consider some epoch p. We know that for the client i who triggers the global synchronization, we have

$$\frac{1+\epsilon}{1-\epsilon}\sum_{s=t_{p-1}+1}^{t_p} \sigma^2_{t_{p-1}}(\mathbf{x}_s) \geq \sum_{s=t_{p-1}+1}^{t_p} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) \geq \sum_{s\in\mathcal{D}_{t_p(i)}\setminus\mathcal{D}_{t_{p-1}(i)}} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) \geq D$$

Then by summing over B epochs, we have

$$BD < \frac{1+\epsilon}{1-\epsilon}\sum_{p=1}^{B}\sum_{s=t_{p-1}+1}^{t_p} \sigma^2_{t_{p-1}}(\mathbf{x}_s) \leq \frac{1+\epsilon}{1-\epsilon}\sum_{p=1}^{B}\sum_{s=t_{p-1}+1}^{t_p} \sigma^2_{s-1}(\mathbf{x}_s)\frac{\sigma^2_{t_{p-1}}(\mathbf{x}_s)}{\sigma^2_{s-1}(\mathbf{x}_s)}.$$

Now we need to bound the ratio $\frac{\sigma^2_{t_{p-1}}(\mathbf{x}_s)}{\sigma^2_{s-1}(\mathbf{x}_s)}$ for $s \in [t_{p-1}+1, t_p]$.

$$\frac{\sigma^2_{t_{p-1}}(\mathbf{x}_s)}{\sigma^2_{s-1}(\mathbf{x}_s)} \leq \Big[1 + \sum_{\tau=t_{p-1}+1}^{s} \sigma^2_{t_{p-1}}(\mathbf{x}_\tau)\Big] \leq \Big[1 + \frac{1+\epsilon}{1-\epsilon}\sum_{\tau=t_{p-1}+1}^{s} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_\tau)\Big]$$

Note that for the client who triggers the global synchronization, we have $\sum_{s\in\mathcal{D}_{t_p-1}(i)\setminus\mathcal{D}_{t_{p-1}}(i)} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) < D$, i.e., one time step before it triggers the synchronization at time t_p. Due to the fact that the (approximated) posterior variance cannot exceed L^2/λ, we have $\sum_{s\in\mathcal{D}_{t_p}(i)\setminus\mathcal{D}_{t_{p-1}}(i)} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) < D + L^2/\lambda$. For the other $N-1$ clients, we have

$\sum_{s\in\mathcal{D}_{t_p}(i)\setminus\mathcal{D}_{t_{p-1}}(i)} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) < D$. Summing them together, we have

$$\sum_{s=t_{p-1}+1}^{t_p} \tilde{\sigma}^2_{t_{p-1}}(\mathbf{x}_s) < (ND + L^2/\lambda)$$

for the p-th epoch. By substituting this back, we have

$$\frac{\sigma^2_{t_{p-1}}(\mathbf{x}_s)}{\sigma^2_{s-1}(\mathbf{x}_s)} \leq \Big[1 + \frac{1+\epsilon}{1-\epsilon}(ND + L^2/\lambda)\Big]$$

Therefore,

$$\begin{aligned} BD &< \frac{1+\epsilon}{1-\epsilon}\Big[1 + \frac{1+\epsilon}{1-\epsilon}(ND + L^2/\lambda)\Big] \sum_{p=1}^{B} \sum_{s=t_{p-1}+1}^{t_p} \sigma^2_{s-1}(\mathbf{x}_s) \\ &\leq \frac{1+\epsilon}{1-\epsilon}\Big[1 + \frac{1+\epsilon}{1-\epsilon}(ND + L^2/\lambda)\Big] 2\gamma_{NT} \end{aligned}$$

and thus the total number of epochs $B < \frac{1+\epsilon}{1-\epsilon}[\frac{1}{D} + \frac{1+\epsilon}{1-\epsilon}(N + L^2/(\lambda D))]2\gamma_{NT}$.

By setting $D = \frac{1}{N}$, we have

$$\begin{aligned} \alpha_{NT} &= \left(\frac{1}{\sqrt{-\epsilon + \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}\frac{1}{N}}}} + 1\right)\sqrt{\lambda}\|\theta_\star\| + R\sqrt{4\ln N/\delta + 2\ln\det((1+\lambda)\mathbf{I} + \mathbf{K}_{[NT],[NT]})} \\ &\leq \left(\frac{1}{\sqrt{-\epsilon + \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}}}} + 1\right)\sqrt{\lambda}\|\theta_\star\| + R\sqrt{4\ln N/\delta + 2\ln\det((1+\lambda)\mathbf{I} + \mathbf{K}_{[NT],[NT]})} \end{aligned}$$

because $N \geq 1$. Moreover, to ensure $-\epsilon + \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}} > 0$, we need to set the constant $\epsilon < 1/3$. Therefore,

$$R_{NT} = O\Big(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT})\Big)$$

and the total number of global synchronizations $B = O(N\gamma_{NT})$. Since for each global synchronization, the communication cost is $O(N\gamma^2_{NT})$, we have

$$C_{NT} = O\Big(N^2\gamma^3_{NT}\Big)$$

2.3 Cooperation in decentralized environments: asynchronous communication

Constraints from real-world applications should also be taken into consideration when designing the communication strategy. For example, the clients often have various response time and even occasional unavailability in reality, due to the differences in their computational and communication capacities. This hampers global synchronization employed in existing federated bandit solutions [28, 61], which requires the server to first send a synchronization signal to all clients, wait and collect their returned local updates, and finally send the aggregated update back to every client.

To address this challenge, we proposed the first asynchronous communication framework for federated bandit learning [64], and it has been further improved and extended to more general problem settings in some recent works [97, 98, 24]. We designed an event-triggered mechanism that offers a flexible way to balance between the regret-minimization and communication-efficiency dilemma. Communication with a client happens only when the last communicated update to the client becomes irrelevant to the latest one; and we prove only by then effective regret reduction can be expected in this client because of the communication. Under this asynchronous communication, each client sends local update to and receives aggregated update from the server independently from other clients, with no need for global synchronization. This improves our method's robustness against possible delays and temporary unavailability of clients.

2.3.1 General problem formulation

We consider the same star-shaped communication network as Section 2.2.2. At each time step $t \in [T]$, a client $i_t \in [N]$ becomes active (assume $P(i_t = i) > 0, \forall i \in [N]$) and chooses an arm $\mathbf{x}_t$ from a candidate set $\mathcal{A}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,K}\} \subseteq \mathbb{R}^d$, and then receives the corresponding reward feedback $y_t = f(\mathbf{x}_t) + \eta_t \in \mathbb{R}$. Note that $\mathcal{A}_t$ is time-varying and assumed to be chosen by an oblivious adversary, f denotes the unknown reward function shared by all clients, and η_t denotes zero-mean sub-Gaussian noise with known variance σ^2. at time step t.

The learning system's goal is to minimize the cumulative (pseudo) regret for all N clients, i.e., $R_T = \sum_{t=1}^T r_t$, where $r_t = \max_{\mathbf{x} \in \mathcal{A}_t} f(\mathbf{x}) - f(\mathbf{x}_t)$. Meanwhile, the system also needs to keep the communication cost C_T low, which is measured by the *total number of scalars* being transferred across the system up to time T.

2.3.2 Asynchronous communication for federated linear bandit

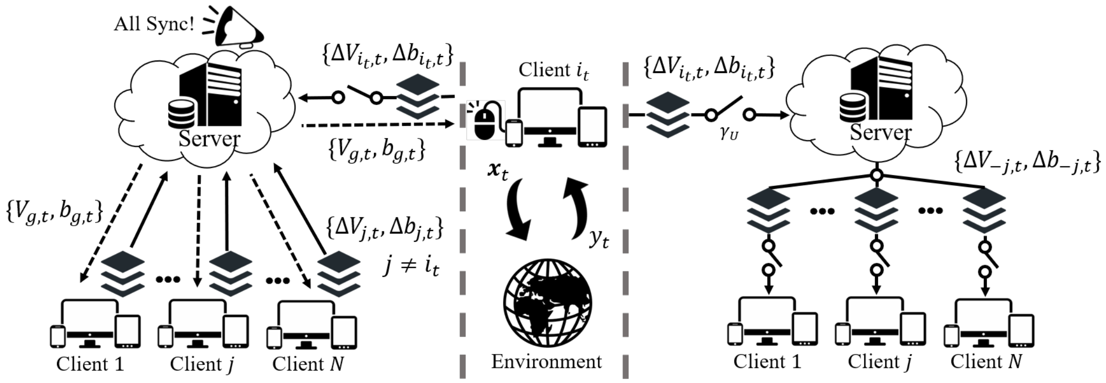
To balance the two conflicting objectives, i.e., cumulative regret R_T and communication cost C_T, as well as ensuring robustness against stragglers in the system, we introduce an asynchronous event-triggered communication framework as illustrated in Figure 2.10(b). In this section, we study this communication framework under federated linear bandit problem, which assumes the rewards received by all the clients are generated by the linear function:

$$f(\mathbf{x}) = \theta^\top \mathbf{x}, \quad \forall i \in [N] \tag{2.27}$$

where $\theta \in \mathbb{R}^d$ is the unknown parameter and we assume $\|\theta\| \leq 1$ and $\|\mathbf{x}\| \leq 1$. Despite its simplicity, this setting is commonly adopted in existing works for federated bandits [28, 61]. Later in this chapter, we will also introduce extension to more general function classes, such as generalized linear bandits and kernelized bandits. Moreover, we adopt the context regularity assumption from [27, 40, 99], which imposes a variance condition on the stochastic process generating $\mathbf{x}_{t,a}$ (for heterogeneous clients, it is imposed on global features $\mathbf{x}_{t,a}^{(g)}$). This suggests the informativeness of each observation in expectation.

Assumption 6 (Context regularity)**.** *At each time t, the context vector $\mathbf{x}_{t,a} \in \mathcal{A}_t$ for each arm $a \in [K]$ is independently generated from a random process, such that $\mathbb{E}_{t-1}[\mathbf{x}_{t,a}\mathbf{x}_{t,a}^\top] := \mathbb{E}[\mathbf{x}_{t,a}\mathbf{x}_{t,a}^\top | \{i_s, \mathcal{A}_s, \eta_s\}_{s\in[t-1]}] = \Sigma_c \succeq \lambda_c I, \forall t \in [T]$ where the constant $\lambda_c > 0$. Let also, for any fixed unit vector $z \in \mathbb{R}^d$, the random variable $(z^\top \mathbf{x}_{t,a})^2$ be conditionally sub-Gaussian with variance parameter $v^2 \leq \lambda_c^2/(8 \log 4K)$.*

We begin our discussion with an important observation about the instantaneous regret of linear bandit algorithms. Denote the sufficient statistics (for θ) collected from all clients by time t as $V_t = \sum_{\tau=1}^t \mathbf{x}_\tau \mathbf{x}_\tau^\top$ and $b_t = \sum_{\tau=1}^t \mathbf{x}_\tau y_\tau$. In a centralized setting, at each time step $t \in [T]$, $\{V_{t-1}, b_{t-1}\}$ are readily available to make an informed choice of arm $\mathbf{x}_t \in \mathcal{A}_t$. It is known that the instantaneous regret r_t incurred by the mentioned linear bandit algorithms is directly related to the width of the confidence ellipsoid in the direction of $\mathbf{x}_t$. Specifically, from Theorem 3 in [20], with probability at least $1 - \delta$, the instantaneous regret r_t incurred by LinUCB can be upper bounded by $r_t \leq 2\alpha_{t-1}\sqrt{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t}$, where $\alpha_{t-1} = O\left(\sqrt{d \log \frac{T}{\delta}}\right)$. However, as data is decentralized in our problem, $\{V_{t-1}, b_{t-1}\}$ are not readily available to client i_t. Instead, the client only has a delayed copy, denoted by $\{V_{i_t,t-1}, b_{i_t,t-1}\}$, which contains its own interactions with the environment on top of the last communication with the server. Therefore, now the instantaneous regret $r_t \leq 2\alpha_{i_t,t-1}\sqrt{\mathbf{x}_t^\top V_{i_t,t-1}^{-1}\mathbf{x}_t} = 2\alpha_{i_t,t-1}\sqrt{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t}\sqrt{\Gamma_{t-1}}$, where $\Gamma_{t-1} = \frac{\mathbf{x}_t^\top V_{i_t,t-1}^{-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t}$ measures how much wider the confidence ellipsoid at client i_t's estimation in the direction of $\mathbf{x}_t$ is, compared with that under a centralized setting. The value of Γ_{t-1} depends on how frequent local updates are aggregated and shared. Also note that $\Gamma_{t-1} \geq 1$, as $V_{t-1} \succeq V_{i_t,t-1}, \forall t$, which suggests the regret in the decentralized setting is at best the same as that in the centralized setting. Equality is attained when all the clients are synchronized in every time step.



(a) Synchronous Communication

(b)Asynchronous Communication

Figure 2.10: Comparison between the synchronous and asynchronous event-triggered communications for federated linear bandit. The former requires all clients to upload their latest data at once and then download the aggregated data, while latter performs both upload and download on a per-client basis.

Based on this observation, we can balance regret and communication cost by controlling the value of Γ_{t-1}. However, in the decentralized setting, neither the server nor the clients has direct access to $\{V_{t-1}, b_{t-1}\}$, and the closest thing one can get is the aggregated sufficient statistics managed by the server, which we denote as $\{V_{g,t-1}, b_{g,t-1}\}$. Hence, we take an indirect approach by first ensuring $\{V_{g,t-1}, b_{g,t-1}\}$ do not deviate too much from $\{V_{t-1}, b_{t-1}\}$, and then $\{V_{i,t-1}, b_{i,t-1}\}$ do not deviate too much from $\{V_{g,t-1}, b_{g,t-1}\}$ for each client $i \in [N]$. The former leads to the 'upload' event, i.e., each client decides whether to upload independently, and the latter leads to the 'download' event, i.e., the server decides whether to send its latest statistics to each client independently as well.

In the proposed communication framework shown in Figure 2.10(b), each client $i \in [N]$ stores a local copy of its sufficient statistics $\{V_{i,t-1}, b_{i,t-1}\}$, and also an 'upload' buffer $\{\Delta V_{i,t-1}, \Delta b_{i,t-1}\}$, i.e., local updates that have not been sent to the server. At each time step t, client $i_t \in [N]$ interacts with the environment, and updates $V_{i_t,t} = V_{i_t,t-1} + \mathbf{x}_t\mathbf{x}_t^\top, b_{i_t,t} = b_{i_t,t-1} + \mathbf{x}_t y_t$, $\Delta V_{i_t,t} = \Delta V_{i_t,t-1} + \mathbf{x}_t\mathbf{x}_t^\top, \Delta b_{i_t,t} = \Delta b_{i_t,t-1} + \mathbf{x}_t y_t$ with the new observation $(\mathbf{x}_t, y_t)$. Then it executes Algorithm 11, by first checking the following condition (line 2):

'Upload' event Client i_t sends $\{\Delta V_{i_t,t}, \Delta b_{i_t,t}\}$ to the server if event:

$$\mathcal{U}_t(\gamma_U) = \left\{ \frac{\det(V_{i_t,t})}{\det(V_{i_t,t} - \Delta V_{i_t,t})} > \gamma_U \right\} \tag{2.28}$$

happens, and then sets $\Delta V_{i,t} = \mathbf{0}, \Delta b_{i,t} = \mathbf{0}$. Otherwise, $\{\Delta V_{i,t}, \Delta b_{i,t}\}$ remain unchanged. The server stores the aggregated sufficient statistics $\{V_{g,t-1}, b_{g,t-1}\}$ over the local updates received from the clients, and also maintains 'download' buffers $\{\Delta V_{-j,t-1}, \Delta b_{-j,t-1}\}$ for each client $j \in [N]$, i.e., the aggregated updates that have not been sent to client j. Specifically, after the server receives $\{\Delta V_{i_t,t}, \Delta b_{i_t,t}\}$ via the 'upload' from client i_t, it updates $V_{g,t} = V_{g,t-1} + \Delta V_{i_t,t}, b_{g,t} = b_{g,t-1} + \Delta b_{i_t,t}$, and $\Delta V_{-j,t} = \Delta V_{-j,t-1} + \Delta V_{i_t,t}, \Delta b_{-j,t} = \Delta b_{-j,t-1} + \Delta b_{i_t,t}$ for all clients $j \neq i_t$. Then it checks the following condition for each client $j \neq i_t$ (line 7):

'Download' event The server sends $\{\Delta V_{j,t}, \Delta b_{j,t}\}$ to client j if event:

$$\mathcal{D}_{t,j}(\gamma_D) = \left\{ \frac{\det(V_{g.t})}{\det(V_{g,t} - \Delta V_{-j,t})} > \gamma_D \right\} \tag{2.29}$$

happens, and then sets $\Delta V_{-j,t} = \mathbf{0}, \Delta b_{-j,t} = \mathbf{0}$. Otherwise, $\{\Delta V_{-j,t}, \Delta b_{-j,t}\}$ remain unchanged. After client j receives $\{\Delta V_{-j,t}, \Delta b_{-j,t}\}$ via the 'download' communication, it updates $V_{j,t} = V_{j,t-1} + \Delta V_{-j,t}, b_{j,t} = b_{j,t-1} + \Delta b_{-j,t}$.

The following lemma specifies an upper bound of Γ_{t-1} by executing Algorithm 11, which depends on the thresholds $\{\gamma_U, \gamma_D\}$ and the number of clients N.

Algorithm 11 Asynchronous Communication Protocol

```
Input: thresholds $\gamma_U, \gamma_D \geq 1$
if Event $\mathcal{U}_t(\gamma_U)$ in Eq (2.28) happens then
    Upload $\Delta V_{i_t,t}, \Delta b_{i_t,t}$ (client $i_t \rightarrow$ server)
    Update server: $V_{g,t}$ += $\Delta V_{i_t,t}, b_{g,t}$ += $\Delta b_{i_t,t}$, $\Delta V_{-j,t}$ += $\Delta V_{i_t,t}, \Delta b_{-j,t}$ += $\Delta b_{i_t,t}, \forall j \neq i_t$
    Client $i_t$ sets $\Delta V_{i_t,t} = \mathbf{0}$, $\Delta b_{i_t,t} = \mathbf{0}$
    for $j = 1, \dots, N$ do
        if Event $\mathcal{D}_{t,j}(\gamma_D)$ in Eq (2.29) happens then
            Download $\Delta V_{-j,t}, \Delta b_{-j,t}$ (server $\rightarrow$ client $j$)
            Update client $j$: $V_{j,t}$ += $\Delta V_{-j,t}, b_{j,t}$ += $\Delta b_{-j,t}$
            Server sets $\Delta V_{-j,t} = \mathbf{0}, \Delta b_{-j,t} = \mathbf{0}$
```

Lemma 2.3.1. *Denote the total number of observations that have been used to update* $\{V_{i,t}, b_{i,t}\}$ *as* τ_i*. With Assumption 6, the 'upload' and 'download' events defined in Eq (2.28) and Eq (2.29), when* $\tau_{i_t} \geq \tau_{min} := \lceil \frac{64}{3\lambda_c} \log(\frac{2NTd}{\delta}) \rceil$*, with probability at least* $1 - \delta$*,* $\Gamma_{t-1} \leq \frac{8\gamma_D}{\lambda_c}[1 + (N-1)(\gamma_U - 1)], \forall t$*.*

Proof of Lemma 2.3.1 is given in Section 2.3.6. The main idea is to use $\det(V_{g,t-1})$ as an intermediate between $\det(V_{i_t,t-1})$ and $\det(V_{t-1})$, which are separately controlled by the 'download' and 'upload' events. When setting $\gamma_D = \gamma_U = 1, \Gamma_{t-1} = 1, \forall t \in [T]$, which means global synchronization happens at each time step, it recovers the regret incurred in the centralized setting.

Synchronous vs. asynchronous communication As shown in Figure 2.10(a), in the synchronous protocol (Appendix G in [28]), when a synchronization round is triggered by a client i_t, the server asks *all* the clients to upload their local updates (illustrated as solid lines), aggregates them, and then sends the aggregated update back (illustrated as dashed lines). This 'two-way' communication is vulnerable to delays or unavailability of clients, which are common in a distributed setting. In comparison, our asynchronous communication, as shown in Figure 2.10(b), is more robust because the server only concerns the clients whose 'download' condition has been met, which does not need other clients' acknowledgement. In addition, when the clients have distinct availability of new observations, which is usually the case for most applications, synchronizing all N clients leads to inefficient communication as some clients may have very few new observations since last synchronization. We will show later that this unfortunately leads to an increased rate in N in the upper bound of C_T, compared with our asynchronous communication.

2.3.3 Async-LinUCB algorithm

Based on the asynchronous event-triggered communication, we design the Asynchronous LinUCB Algorithm (Async-LinUCB) for homogeneous clients. Detailed steps are explained in Algorithm 12.

Arm selection To balance between exploration and exploitation during interactions with the environment, at each time step $t = 1, \dots, T$, client i_t selects an arm $\mathbf{x}_t \in \mathcal{A}_t$ using the the UCB strategy based on its local copy of sufficient statistics $\{V_{i_t,t-1}, b_{i_t,t-1}\}$. Specifically, client i_t pulls arm $\mathbf{x}_t$ that maximizes the following UCB score (line 8),

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{A}_t} \mathbf{x}^\top \hat{\theta}_{i_t,t-1}(\lambda) + \mathrm{CB}_{i_t,t-1}(\mathbf{x}) \tag{2.30}$$

where $\hat{\theta}_{i_t,t-1}(\lambda) = V_{i_t,t-1}(\lambda)^{-1} b_{i_t,t-1}$ is the ridge regression estimator with regularization parameter λ; $V_{i_t,t-1}(\lambda) = V_{i_t,t-1} + \lambda I$; and the confidence bound of reward estimation for arm $\mathbf{x}$ is $\mathrm{CB}_{i_t,t-1}(\mathbf{x}) = \alpha_{i_t,t-1} ||\mathbf{x}||_{V_{i_t,t-1}(\lambda)^{-1}}$, where $\alpha_{i_t,t-1} = \sigma \sqrt{\log \frac{\det V_{i_t,t-1}(\lambda)}{\det \lambda I} + 2 \log 1/\delta} + \sqrt{\lambda}$. After client i_t observes reward y_t and updates locally (line 9), it proceeds with the asynchronous event-triggered communication (line 10), and sends updates accordingly.

Algorithm 12 Async-LinUCB

```
**Input:** thresholds $\gamma_U, \gamma_D \geq 1$, $\sigma, \lambda > 0$, $\delta \in (0,1)$
Initialize server: $V_{g,0} = \mathbf{0}_{d,d}$, $b_{g,0} = \mathbf{0}_d$
**for** $t = 1, 2, ..., T$ **do**
    Observe arm set $\mathcal{A}_t$ for client $i_t \in [N]$
    **if** client $i_t$ is new **then**
        Initialize client $i_t$: $V_{i_t,t-1} = \mathbf{0}_{d,d}$, $b_{i_t,t-1} = \mathbf{0}_d$, $\Delta V_{i_t,t-1} = \mathbf{0}_{d,d}$, $\Delta b_{i_t,t-1} = \mathbf{0}_d$
        Initialize server's download buffer for client $i_t$: $\Delta V_{-i_t,t-1} = V_{g,t-1}$, $\Delta b_{-i_t,t-1} = b_{g,t-1}$
    Pull arm $\mathbf{x}_t \in \mathcal{A}_t$ by Eq (2.30) and observe $y_t$
    Update client $i_t$: $V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^T$, $b_{i_t,t} \mathrel{+}= \mathbf{x}_t y_t$, $\Delta V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^T$, $\Delta b_{i_t,t} + = \mathbf{x}_t y_t$
    Event-triggered Communications (Algorithm 11)
```

2.3.4 Regret and communication cost analysis

The upper bounds of cumulative regret R_T and communication cost C_T incurred by Async-LinUCB are given in Theorem 2.3.2 (complete proof is provided in Section 2.3.6).

Theorem 2.3.2 (Regret and Communication). *With Assumption 6, and the communication thresholds γ_U, γ_D, then the accumulative regret* [3]

$$R_T = \tilde{O}\left(d\sqrt{T}\log\frac{T}{\delta}\min(\sqrt{N}, \sqrt{\gamma_D[1 + (N-1)(\gamma_U - 1)]})\right)$$

with probability at least $1 - \delta$, and the communication cost

$$C_T = O\big(d^3 N \log T / \log \min(\gamma_U, \gamma_D)\big).$$

The thresholds γ_U, γ_D can be flexibly adjusted to trade-off between R_T and C_T, e.g., interpolate between the two extreme cases: clients never communicate ($R_T = O(N^{1/2} d\sqrt{T}\log T)$); and clients are synchronized in every time step ($R_T = O(d\sqrt{T}\log T)$). In practice, depending on whether the application at hand is performance-critical or communication-critical, one can first specify the scaling factor for the regret bound or the communication bound and solve for valid values of γ_U, γ_D. Details about threshold selection and the corresponding theoretical results are provided in Section 2.3.6. For simplicity, we fix $\gamma_U = \gamma_D = \gamma$ in the following discussions, but one can choose different values to have a finer control especially for applications where the cost of upload and download communication differs. Based on Theorem 2.3.2, to attain $R_T = \tilde{O}(N^{1/4} d\sqrt{T}\log T)$, Async-LinUCB needs $C_T = O(N^{3/2} d^3 \log T)$ (by setting $\gamma = \exp(N^{-\frac{1}{2}})$). To attain the same R_T, the corresponding C_T of Sync-LinUCB [4] is smaller than ours by a factor of $O(N^{1/4})$ only under uniform client distribution ($P(i_t = i) = \frac{1}{N}, \forall i, t$), while under non-uniform client distribution, which is almost always the case in practice, it is higher than ours by a factor of $O(N^{1/4})$. The description and theoretical analysis of Sync-LinUCB under uniform and non-uniform client distribution are given in Section 2.3.6.

2.3.5 Experiment setup & results

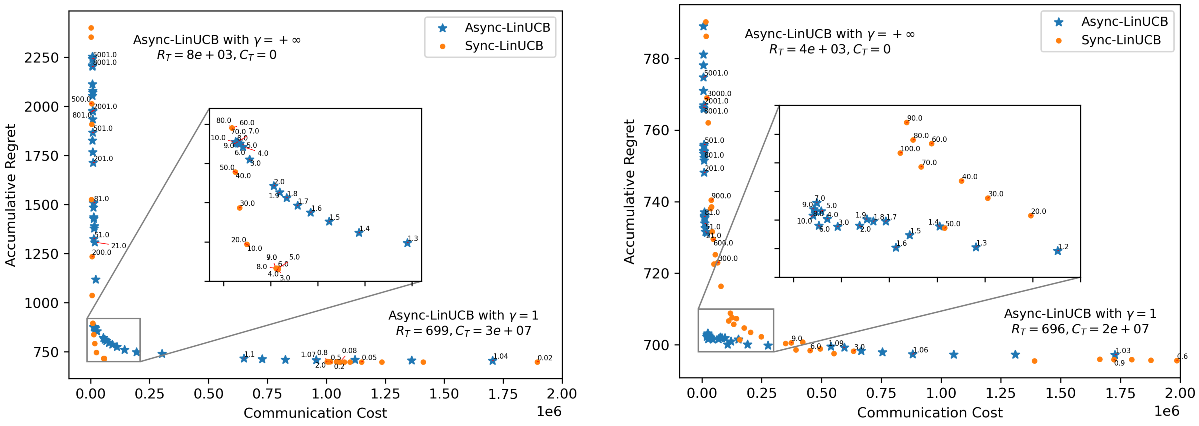
We performed extensive empirical evaluations of Async-LinUCB on synthetic datasets (we set $\gamma_U = \gamma_D = \gamma$ in all experiments), and included Sync-LinUCB [28] as baseline.

We first conducted simulation experiment in to validate our theoretical comparison between Async-LinUCB and Sync-LinUCB (see Section 2.3.4), i.e., how well the algorithms balance regret R_T and communication cost C_T under uniform and non-uniform client distributions.

Synthetic dataset We simulated the federated linear bandit problem setting in Section 2.3.2, with $T = 30000$, $N = 1000$, and $\mathcal{A}_t$ ($K = 25$) uniformly sampled from a ℓ_2 ball. To compare how the algorithms balance R_T and C_T under uniform ($P(i_t = i) = \frac{1}{N}, \forall i, t$) and non-uniform client distributions ($P(i_t)$ is an arbitrary point on probability simplex), we fixed $d = 25$, and ran Async-LinUCB and Sync-LinUCB with a large range of threshold values (logarithmically

[3] $\tilde{O}(\cdot)$ omits the logarithmic regret term incurred during the initial $\tau_{\min}$ time steps on each client

[4] Sync-LinUCB refers to DisLinUCB algorithm in Appendix G of [28] adapted to our setting.



(a) Homogeneous (uniform client distribution)

(b) Homogeneous (non-uniform client distribution)

Figure 2.11: Experiment results on synthetic dataset.

spaced between 10^{-2} and 10^3). Experiment results (averaged over 10 runs) on synthetic dataset are shown in Figure 2.11(a)-2.11(b). Note that in the scatter plots, each dot denotes the cumulative communication cost (x-axis) and regret (y-axis) that an algorithm (Async-LinUCB or Sync-LinUCB) with certain threshold value (labeled next to the dot) has obtained at iteration T.

Discussions From both Figure 2.11(a) and Figure 2.11(b), we can see that as the threshold value increases, C_T decreases and R_T increases, and that the use of event-triggered communication significantly reduces C_T while attaining low R_T, compared with synchronizing all the clients at each time step (Async-LinUCB with $\gamma = 1$). In Figure 2.11(a), Sync-LinUCB has lower C_T than Async-LinUCB under the same R_T, and in Figure 2.11(b), Async-LinUCB has lower C_T than Sync-LinUCB under the same R_T, which conform with our theoretical results that Sync-LinUCB has inefficient communication under non-uniform client distribution.

2.3.6 Full proof of Async-LinUCB algorithm

Proof of Lemma 2.3.1 in Section 2.3.2

To show that $\Gamma_{t-1} \leq \frac{8\gamma_D}{\lambda_c}[1 + (N-1)(\gamma_U - 1)]$, we first need the following lemma.

Lemma 2.3.3. *Denote the number of observations that have been used to update $\{V_{i,t}, b_{i,t}\}$ as τ_i, i.e., $V_{i,t} = \lambda I + \sum_{s=1}^{\tau_i} \mathbf{x}_s \mathbf{x}_s^\top$. Then under Assumption 6, with probability at least $1-\delta$, we have:*

$$\lambda_{\min}(V_{i,t}) \geq \lambda + \frac{\lambda_c \tau_i}{8}$$

$\forall \tau_i \in \{\tau_{min}, \tau_{min}+1, \dots, T\}, i \in [N]$, where $\tau_{min} = \lceil \frac{64}{3\lambda_c} \log(\frac{2NTd}{\delta}) \rceil$.

Proof of Lemma 2.3.3. This proof is based on standard matrix martingale arguments, and is included here for the sake of completeness.

Consider the random variable $(z^\top \mathbf{x}_{s,a})^2$, where $z \in \mathbb{R}^d$ is an arbitrary vector such that $\|z\|_2 \leq 1$ and $\mathbf{x}_{s,a} \in \mathcal{A}_s = \{\mathbf{x}_{s,1}, \mathbf{x}_{s,2}, \dots, \mathbf{x}_{s,K}\}$. Then by Assumption 6, $(z^\top \mathbf{x}_{s,a})^2$ is sub-Gaussian with variance parameter v^2. Now we follow the same argument as Claim 1 of [27] to derive a lower bound for $\lambda_{\min}(\Sigma_s)$. First we construct $Z_a = (z^\top \mathbf{x}_{s,a})^2 - \mathbb{E}_{s-1}[(z^\top \mathbf{x}_{s,a})^2]$, for $a \in [K]$. Due to (conditional) sub-Gaussianity, we have

$$P_{s-1}(Z_a < -h) \leq P_{s-1}(|Z_a| > h) \leq 2e^{-\frac{h^2}{2v^2}}$$

Then by union bound, and the fact that $\mathbb{E}_{s-1}[(z^\top \mathbf{x}_{s,a})^2] = z^\top \Sigma_c z \geq \lambda_c$, we have:

$$P_{s-1}\big(\min_{a\in[K]}(z^\top \mathbf{x}_{s,a}) \geq \lambda_c - h\big) \geq (1 - 2e^{-\frac{h^2}{2v^2}})^K$$

Therefore,

$$\mathbb{E}_{s-1}((z^\top \mathbf{x}_s)^2) \geq \mathbb{E}_{s-1}(\min_{a\in[K]}(z^\top \mathbf{x}_{s,a})^2) \geq (\lambda_c - h)(1 - 2e^{-\frac{h^2}{2v^2}})^K$$

Then by seting $h = \sqrt{2v^2 \log{(4K)}}$, we have $(1 - 2e^{-\frac{h^2}{2v^2}})^K = (1 - \frac{1}{2K})^K \geq \frac{1}{2}$ because $K \geq 1$, and $(\lambda_c - h) \geq \frac{\lambda_c}{2}$ because of the assumption on v^2. Now we have $z^\top \Sigma_s z = \mathbb{E}_{s-1}((z^\top x_s)^2) \geq \frac{1}{4}\lambda_c, \forall z$, so $\lambda_{\min}(\Sigma_s) \geq \frac{1}{4}\lambda_c$.

Then we are ready to lower bound $\lambda_{\min}(V_{i,t})$ as shown below. Specifically, consider the sequence $Y_{\tau_i} := \sum_{s=1}^{\tau_i}[\mathbf{x}_s \mathbf{x}_s^\top - \Sigma_s]$, for $\tau_i = 1, 2, \dots$. And $\{Y_{\tau_i}\}_{\tau_i=1,2,\dots}$ is a matrix martingale, because $\mathbb{E}[\|Y_{\tau_i}\|_{op}] < +\infty$ and $\mathbb{E}_{\tau_i-1}[Y_{\tau_i}] = \sum_{s=1}^{\tau_i-1}[\mathbf{x}_s \mathbf{x}_s - \Sigma_s] + \mathbb{E}_{\tau_i-1}[\mathbf{x}_{\tau_i}\mathbf{x}_{\tau_i}^\top - \Sigma_{\tau_i}] = Y_{\tau_i-1}$. Then with the Matrix Freedman inequality (Lemma A.16), we have

$$P(\|\sum_{s=1}^{\tau_i}(x_s x_s^\top - \Sigma_s)\|_{op} \geq u) \leq 2d\exp(\frac{-u^2/2}{w^2 + 2u/3}) \tag{2.31}$$

where $\|\cdot\|_{op}$ denotes the operator norm. This can be rewritten as $P(-\|\sum_{s=1}^{\tau_i}\Sigma_s - \sum_{s=1}^{\tau_i} x_s x_s^\top\|_{op} > -u) \geq 1 - 2d\exp(\frac{-u^2/2}{w^2+2u/3})$. Then, we have

$$\begin{aligned}
&1 - 2d\exp(\frac{-u^2/2}{w^2 + 2u/3}) \leq P(-\|\sum_{s=1}^{\tau_i}\Sigma_s - \sum_{s=1}^{\tau_i} x_s x_s^\top\|_{op} > -u) \leq P(-\lambda_{\min}(\sum_{s=1}^{\tau_i}\Sigma_s - \sum_{s=1}^{\tau_i} x_s x_s^\top) > -u) \\
&\leq P(-\lambda_{\min}(\sum_{s=1}^{\tau_i}\Sigma_s) + \lambda_{\min}(\sum_{s=1}^{\tau_i} x_s x_s^\top) > -u) \leq P(-\sum_{s=1}^{\tau_i}\lambda_{\min}(\Sigma_s) + \lambda_{\min}(\sum_{s=1}^{\tau_i} x_s x_s^\top) > -u) \\
&\leq P(\lambda_{\min}(\sum_{s=1}^{\tau_i} x_s x_s^\top) > \frac{\tau_i \lambda_c}{4} - u)
\end{aligned}$$

where the third and forth inequalities are due to Weyl's inequality, i.e., $\lambda_{\min}(A + B) \geq \lambda_{\min}(A) + \lambda_{\min}(B)$ for symmetric matrices A and B, and the fifth inequality is due to $\lambda_{\min}(\Sigma_s) \geq \frac{1}{4}\lambda_c$.

By setting $u = \frac{\lambda_c \tau_i}{8}$ and $w^2 = \frac{\tau_i}{12}$, we have $P(\lambda_{\min}(\sum_{s=1}^{\tau_i} x_s x_s^\top) > \frac{\lambda_c \tau_i}{8}) \geq 1 - 2d\exp(\frac{-\lambda_c \tau_i}{64/3})$. Then when $\tau_i \geq \frac{64}{3\lambda_c}\log(\frac{2Td}{\delta}) := \tau_{\min}$, we have $P(\lambda_{\min}(\sum_{s=1}^{\tau_i} x_s x_s^\top) > \frac{\lambda_c \tau_i}{8}) \geq 1 - \frac{\delta}{T}$. By taking a union bound over all $\tau_i \in \{\tau_{min}, \tau_{min}+1, \dots, T\}$, we have $P(\lambda_{\min}(V_{i,t}) > \lambda + \frac{\lambda_c \tau_i}{8}) \geq 1 - \delta$. Then we take union bound over all $i \in [N]$, which finishes the proof of Lemma 2.3.3. □

Proof of Lemma 2.3.1. Under Lemma A.1, we have

$$\Gamma_{t-1} = \frac{\mathbf{x}_t^\top V_{i_t,t-1}^{-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t} \leq \frac{\lambda_{max}(V_{i_t,t-1})}{\lambda_{min}(V_{i_t,t-1})}\frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t}$$

Then when $\tau_{i_t} \geq \tau_{min}$, with Lemma 2.3.3 and the fact that $\lambda_{max}(V_{i_t,t-1}) \leq \lambda + \tau_{i_t}$, we have

$$\Gamma_{t-1} \leq \frac{\lambda + \tau_{i_t}}{\lambda + \tau_{i_t}\lambda_c/8} \cdot \frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t} \leq \frac{8}{\lambda_c} \cdot \frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t}$$

where the second inequality is because, for bounded context vector ($\|\mathbf{x}_{t,a}\|_2 \leq 1$), $\lambda_c \leq \frac{1}{d} < 8$, so $\frac{\lambda_c}{8} < 1$. In this case, $r_t \leq 2\alpha_{i_t,t-1}\sqrt{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t}\sqrt{\frac{8}{\lambda_c}\frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t}}$. Note that when $\tau_{i_t} < \tau_{min}$, we can simply bound r_t by the constant $2LS$, and in total this added regret is $O(\frac{64N}{3\lambda_c}\log(\frac{2NTd}{\delta}))$, which is negligible compared with the $O(\sqrt{T})$ term in the upper bound of R_T.

Now we need to show that

$$\frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t} \leq \gamma_D[1 + (N-1)(\gamma_U - 1)]$$

In order to do this, we need the following two facts:

- $V_{i_t,t-1} - \Delta V_{i_t,t-1} = V_{g,t-1} - \Delta V_{-i_t,t-1}$, because they both equal to the copy of sufficient statistics in the most recent communication between the client i_t and the server.
- Due to Lemma A.3, and our design of the 'upload' and 'download' triggering events in Eq (2.28) and Eq (2.29), at the beginning of time $t \in [T]$, the inequalities

$$\sup_{\mathbf{x}} \frac{\mathbf{x}^\top(V_{j,t-1})\mathbf{x}}{\mathbf{x}^\top(V_{j,t-1} - \Delta V_{j,t-1})\mathbf{x}} \leq \frac{\det(V_{j,t-1})}{\det(V_{j,t-1} - \Delta V_{j,t-1})} \leq \gamma_U \tag{2.32}$$

  and

$$\sup_{\mathbf{x}} \frac{\mathbf{x}^\top(V_{g,t-1})\mathbf{x}}{\mathbf{x}^\top(V_{g,t-1} - \Delta V_{-j,t-1})\mathbf{x}} \leq \frac{\det(V_{g,t-1})}{\det(V_{g,t-1} - \Delta V_{-j,t-1})} \leq \gamma_D \tag{2.33}$$

  hold $\forall j \in [N], \forall t \in [T]$.

Then by decomposing $\frac{\mathbf{x}_t^\top V_{t-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{i_t,t-1}\mathbf{x}_t}$, we have:

$$\begin{aligned}
&\frac{\mathbf{x}_t^\top(V_{t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{i_t,t-1})\mathbf{x}_t} = \frac{\mathbf{x}_t^\top(V_{g,t-1} + \sum_{j=1}^N \Delta V_{j,t-1})\mathbf{x}_t}{x_t^\top(V_{i_t,t-1} - \Delta V_{i_t,t-1} + \Delta V_{i_t,t-1})\mathbf{x}_t} \\
&\leq \frac{\mathbf{x}_t^\top(V_{g,t-1} + \sum_{j\neq 1} \Delta V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{i_t,t-1} - \Delta V_{i_t,t-1})\mathbf{x}_t} = \frac{\mathbf{x}_t^\top(V_{g,t-1})\mathbf{x}_t + \sum_{j\neq 1}\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{g,t-1} - \Delta V_{-i_t,t-1})\mathbf{x}_t}
\end{aligned}$$

And the term $\sum_{j\neq 1}\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t$ can be further upper bounded by:

$$\begin{aligned}
&\sum_{j\neq 1}\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t = \mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t \cdot \sum_{j\neq i_t}\frac{\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t} \\
&\leq \mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t \cdot \sum_{j\neq i_t}\frac{\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{g,t-1} - \Delta V_{-j,t-1})\mathbf{x}_t} = \mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t \cdot \sum_{j\neq i_t}\frac{\mathbf{x}_t^\top(\Delta V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{j,t-1} - \Delta V_{j,t-1})\mathbf{x}_t} \\
&= \mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t \cdot \sum_{j\neq i_t}\Big[\frac{\mathbf{x}_t^\top(V_{j,t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{j,t-1} - \Delta V_{j,t-1})\mathbf{x}_t} - 1\Big] \leq \mathbf{x}_t^\top V_{g,t-1}\mathbf{x}_t \cdot (N-1)(\gamma_U - 1)
\end{aligned}$$

where the last inequality is due to Eq (2.32). Then by substituting this back, and using Eq (2.33), we have

$$\frac{\mathbf{x}_t^\top(V_{t-1})\mathbf{x}_t}{\mathbf{x}_t^\top(V_{i_t,t-1})\mathbf{x}_t} \leq \frac{\mathbf{x}_t^\top(V_{g,t-1})\mathbf{x}_t[1 + (N-1)(\gamma_U - 1)]}{\mathbf{x}_t^\top(V_{g,t-1} - \Delta V_{-i_t,t-1})\mathbf{x}_t} \leq \gamma_D[1 + (N-1)(\gamma_U - 1)]$$

which finishes the proof of Lemma 2.3.1. □

Discussion Compared with the synchronous method, our asynchronous method needs an additional context regularity assumption in the proof of Lemma 2.3.1. This is because we allow each client to decide on its own whether to upload, based on how much its local data, i.e., $V_{j,t-1}$, has deviated from its last communicated data with the server, i.e., $V_{j,t-1} - \Delta V_{j,t-1}$, and they are unaware of other clients' new data. More specifically, as we mentioned in Section 2.3.2, to guarantee each individual client's sufficient statistics $\{V_{i,t-1}, b_{i,t-1}\}$ do not deviate too much from $\{V_{t-1}, b_{t-1}\}$, we need to first ensue $\{V_{g,t-1}, b_{g,t-1}\}$ do not deviate too much from $\{V_{t-1}, b_{t-1}\}$ via asynchronous upload, and then $\{V_{i,t-1}, b_{i,t-1}\}$ do not deviate too much from $\{V_{g,t-1}, b_{g,t-1}\}$ via asynchronous download. To better understand this, we can look at the following upper bound of Γ_{t-1}:

$$
\begin{aligned}
\Gamma_{t-1} = \frac{\mathbf{x}_t^\top V_{i_t,t-1}^{-1}\mathbf{x}_t}{\mathbf{x}_t^\top V_{t-1}^{-1}\mathbf{x}_t} \leq \frac{\det(V_{i_t,t-1}^{-1})}{\det(V_{t-1}^{-1})} = \frac{\det(V_{t-1})}{\det(V_{i_t,t-1})} &= \frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{i_t,t-1} - \Delta V_{i_t,t-1} + \Delta V_{i_t,t-1})} \\
&= \frac{\det(V_{g,t-1} + \Delta V_{i_t,t-1})}{\det(V_{i_t,t-1} - \Delta V_{i_t,t-1} + \Delta V_{i_t,t-1})} \cdot \frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1} + \Delta V_{i_t,t-1})} \\
&\leq \frac{\det(V_{g,t-1})}{\det(V_{i_t,t-1} - \Delta V_{i_t,t-1})} \cdot \frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1})}
\end{aligned}
$$

where the first inequality is due to Lemma A.3, and the second inequality is due to Lemma A.4. Note that according to our 'download' event, the first term $\frac{\det(V_{g,t-1})}{\det(V_{i_t,t-1} - \Delta V_{i_t,t-1})} \leq \gamma_D$. The difficulty mainly lies in the second term $\frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1})}$, which essentially measures the difference in the volume of confidence ellipsoid between the data under the ideal centralized setting and the data actually available to the server. In the synchronous method, the ratio $\frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1})}$ is simply pushed to 1 at every global synchronization step, since N clients will simultaneously upload their local updates $\{\Delta V_{j,t-1}\}_{j \in [N]}$ to the server. However, in our case, this ratio is jointly controlled by the asynchronous uploads from each individual client who decides on its own whether to upload, based on the locally available data. Ideally, to make sure the upload is effective in terms of regret reduction, each client $j \in [N]$ should directly compute the value of $\frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1})}$ or its upper bound, and decide whether this ratio has grown too large, i.e., the server's data has become out-of-date, such that sending local updates to the server is necessary. Unfortunately, with data being decentralized, this information is unavailable to any client. Instead, each client j only knows the upper bound of $\frac{\det(V_{g,t-1} + \Delta V_{j,t-1})}{\det(V_{g,t-1})}$:

$$
\frac{\det(V_{g,t-1} + \Delta V_{j,t-1})}{\det(V_{g,t-1})} \leq \frac{\det(V_{g,t-1} - \Delta V_{-j,t-1} + \Delta V_{j,t-1})}{\det(V_{g,t-1} - \Delta V_{-j,t-1})} = \frac{\det(V_{j,t-1})}{\det(V_{j,t-1} - \Delta V_{j,t-1})} \leq \gamma_U.
$$

The information gap due to each client's unawareness about what other clients have in their upload buffers makes it difficult to obtain a non-trivial upper bound of $\frac{\det(V_{g,t-1} + \sum_{j=1}^{N} \Delta V_{j,t-1})}{\det(V_{g,t-1})}$. In the worst case scenario where new data of the clients are very different from each other, this leads to a trivial upper bound that is exponential in N, i.e., updating $V_{g,t-1}$ with the new data $\Delta V_{j,t-1}$ of each client $j \in [N]$ can scale up the determinant of $V_{g,t-1}$ by γ_U. Assumption 6 is a sufficient condition to circumvent this, but may not be a necessary condition. Finding a sufficient and necessary condition to relax Assumption 1 will be an important future direction of this work.

Proof of Theorem 2.3.2 in Section 2.3.4

Regret analysis Based on the discussion in Section 2.3.2 that the instantaneous regret r_t directly depends on Γ_{t-1}, we can upper bound the accumulative regret of Async-LinUCB by

$$
\begin{aligned}
R_T = \sum_{t=1}^{T} r_t &\leq \sum_{t=1}^{T} O\left(\sqrt{d \log \frac{T}{\delta}}\right) \sqrt{\mathbf{x}_t^\top V_{t-1}^{-1} \mathbf{x}_t} \sqrt{\Gamma_{t-1}} \\
&\leq O\left(\sqrt{d \log \frac{T}{\delta}}\right) \sqrt{\sum_{t=1}^{T} x^\top V_{t-1}^{-1} x} \sqrt{\sum_{t=1}^{T} \Gamma_{t-1}} \leq O\left(\sqrt{d \log \frac{T}{\delta}}\right) \sqrt{\log \frac{det(V_{T-1})}{det(\lambda I)}} \sqrt{\sum_{t=1}^{T} \Gamma_{t-1}}
\end{aligned}
$$

where the second inequality is by the Cauchy–Schwarz inequality, and the third is based on Lemma 11 in [20]. Then using the upper bound of Γ_{t-1} given in Lemma 2.3.1, the accumulative regret

$$\begin{aligned} R_T &= O\left(d\sqrt{T}\log(T/\delta)\min(\sqrt{N}, \sqrt{\gamma_D[1+(N-1)(\gamma_U-1)]}) + \frac{64N}{3\lambda_c}\log(\frac{2NTd}{\delta})\right) \\ &= \tilde{O}\left(d\sqrt{T}\log(T/\delta)\min(\sqrt{N}, \sqrt{\gamma_D[1+(N-1)(\gamma_U-1)]})\right), \end{aligned}$$

with probability at least $1-\delta$, where $\tilde{O}(\cdot)$ omits the logarithmic term.

Communication cost analysis As discussed in Section 2.3.2, clients collaborate by transferring updates of the sufficient statistics, i.e., $\{\Delta V \in \mathbb{R}^{d\times d}, \Delta b \in \mathbb{R}^d\}$. Therefore, each time of communication incurs a cost of (d^2+d), i.e., size of the statistics. To analyze C_T, we denote the sequence of time steps when either 'upload' or 'download' is triggered up to time T as $\{t_1, t_2, \dots, t_{C_{T,i}}\}$, where $C_{T,i}$ is the total number of communications between client i and the server. Then the corresponding sequence of local covariance matrices is $\{\lambda I, V_{i,t_1}, V_{i,t_2}, \dots, V_{i,t_{C_{T,i}}}\}$. We can decompose

$$\log\frac{\det V_{i,t_{C_{T,i}}}}{\det \lambda I} = \log\frac{\det V_{i,t_1}}{\det \lambda I} + \log\frac{\det V_{i,t_2}}{\det V_{i,t_1}} + \dots \log\frac{\det V_{i,t_{C_{T,i}}}}{\det V_{i,t_{C_{T,i}-1}}} \leq \log\frac{\det V_{T-1}}{\det \lambda I}$$

Since the matrices in the sequence trigger either Eq (2.28) or Eq (2.29), each term in this summation is lower bounded by $\log\min(\gamma_U, \gamma_D)$. When $\min(\gamma_U, \gamma_D) > 1$, by the pigeonhole principle, $C_{T,i} \leq \frac{\log\det(V_{T-1}) - d\log\lambda}{\log\min(\gamma_U,\gamma_D)}$; as a result, the communication cost for N clients is $C_T = (d^2+d)\sum_{i=1}^N C_{T,i} \leq Nd^2\frac{\log\det(V_{T-1})-d\log\lambda}{\log\min(\gamma_U,\gamma_D)}$.

Synchronous communication method

The synchronous method DisLinUCB (Appendix G in [28]) imposes a stronger assumption about the appearance of clients: i.e., they assume all N clients interact with the environment in a round-robin fashion (so $\mathcal{N}_i(T) = \frac{T}{N}$ [5]). For the sake of completeness, we present the formal description of this algorithm adapted to our problem setting in Algorithm 13 (which is referred to as Synchronous LinUCB algorithm, or Sync-LinUCB for short), and provide the corresponding theoretical analysis about its regret R_T and communication cost C_T under both uniform and non-uniform client distribution. In particular, in this setting we no longer assume uniform appearance of clients.

In our problem setting (Section 2.3.1), other than assuming each client has a nonzero probability to appear in each time step, we do not impose any further assumption on the clients' distribution or its frequency of interactions with the environment. This is more general than the setting considered in [28], since the clients now may have distinct availability of new observations. We will see below that this will cause additional communication cost for Sync-LinUCB, compared with the case where all the clients interact with the environment in a round-robin fashion, i.e., all N clients have equal number of observations. Intuitively, when one single client accounts for the majority of the interactions with the environment and always triggers the global synchronization, all the other $N-1$ clients are forced to upload their local data despite the fact that they have very few new observations since the last synchronization. This directly leads to a waste of communication. Below we give the analysis of R_T and C_T of sync-LinUCB considering both uniform and non-uniform client distribution.

Regret analysis Most part of the proof for Theorem 4 in [28] extends to the problem setting considered in this work (with slight modifications due to the difference in the meaning of T as mentioned in the footnote). Since now only one client interacts with the environment in each time step, the accumulative regret for the 'good epochs' is $REG_{good} = O(d\sqrt{T}\log(T))$. Denote the first time step of a certain 'bad epoch' as t_s and the last as t_e. The accumulative regret for this 'bad epoch' can be upper bounded by: $O(\sqrt{d\log T})\sum_{i=1}^N \sum_{\tau\in\mathcal{N}_i(t_e)\setminus\mathcal{N}_i(t_s)} \min(1, ||\mathbf{x}_\tau||_{V_{i,\tau-1}^{-1}}) \leq O(\sqrt{d\log T})\sum_{i=1}^N \sqrt{\Delta t_{i,t_e}\log\frac{\det(V_{i,t_e-1}+\lambda I)}{\det(V_{i,t_e-1}-\Delta V_{i,t_e-1}+\lambda I)}} \leq O(\sqrt{d\log T}N\sqrt{D})$. And using the same argument as in the original proof, there can be at most $R = O(d\log T)$ 'bad epochs', so that accumulative regret for the 'bad epochs'

[5] It is worth noting the difference in the meaning of T between our work and [28]. In our work, T is the total number of interactions for all N clients, while for [28], T is the total number of interactions for each client.

Algorithm 13 Synchronous LinUCB Algorithm

```
Input: threshold $D$, $\sigma, \lambda > 0$, $\delta \in (0,1)$
Initialize server: $V_{g,0} = \mathbf{0}_{d\times d} \in \mathbb{R}^{d\times d}$, $b_{g,0} = \mathbf{0}_d \in \mathbb{R}^d$
for $t = 1, 2, ..., T$ do
    Observe arm set $\mathcal{A}_t$ for client $i_t \in [N]$
    if client $i_t$ is new then
        Initialize client $i_t$: $V_{i_t,t-1} = \mathbf{0}_{d\times d}$, $b_{i_t,t-1} = \mathbf{0}_d$, $\Delta V_{i_t,t-1} = \mathbf{0}_{d\times d}$, $\Delta b_{i_t,t-1} = \mathbf{0}_d$, $\Delta t_{i_t,t-1} = 0$
    Select arm $\mathbf{x}_t \in \mathcal{A}_t$ by Eq (2.8) and observe reward $y_t$
    Update client $i_t$: $V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^T$, $b_{i_t,t} \mathrel{+}= \mathbf{x}_t y_t$, $\Delta V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^T$, $\Delta b_{i_t,t} + = \mathbf{x}_t y_t$, $\Delta t_{i_t,t} \mathrel{+}= 1$
    # Check whether global synchronization is triggered
    if $\Delta t_{i_t,t} \log \frac{\det(V_{i_t,t}+\lambda I)}{\det(V_{i_t,t}-\Delta V_{i_t,t}+\lambda I)} > D$ then
        for $i = 1, \dots, N$ do
            Upload $\Delta V_{i,t}, \Delta b_{i,t}$ ($i \to$ server)
            Client $i$ reset $\Delta V_{i,t} = \mathbf{0}$, $\Delta b_{i,t} = \mathbf{0}$, $\Delta t_{i,t} = 0$
            Update server: $V_{g,t} \mathrel{+}= \Delta V_{i,t}, b_{g,t} \mathrel{+}= \Delta b_{i,t}$
        for $i = 1, \dots, N$ do
            Download $V_{g,t}, b_{g,t}$ (server $\to i$)
            Update client $i$: $V_{i,t} = V_{g,t}, b_{j,t} = b_{g,t}$
```

is upper bounded by $REG_{bad} = O(d^{1.5}\log^{1.5}(T)N\sqrt{D})$. Therefore, with the threshold D, the accumulative regret is $R_T = O(d\sqrt{T}\log(T)) + O(d^{1.5}\log^{1.5}(T)N\sqrt{D})$.

For the analysis of communication cost C_T, we consider the settings of uniform and non-uniform client distributions separately in the following two paragraphs.

Communication cost of Sync-LinUCB under uniform client distribution Denote the length of an epoch as α, so that there can be at most $\lceil \frac{T}{\alpha} \rceil$ epochs with length longer than α. For an epoch with less than α time steps, similarly, we denote the first time step of this epoch as t_s and the last as t_e, i.e., $t_e - t_s < \alpha$. Then since the users appear in a uniform manner, the number of interactions for any user $i \in [N]$ satisfies $\Delta t_{i,t_e} < \frac{\alpha}{N}$. Therefore, $\log \frac{\det(V_{t_e})}{\det(V_{t_s})} > \frac{DN}{\alpha}$. Following the same argument as in the original proof, the number of epochs with less than α time steps is at most $\lceil \frac{R\alpha}{DN} \rceil$. Then $C_T = Nd^2 \cdot (\lceil \frac{T}{\alpha} \rceil + \lceil \frac{R\alpha}{DN} \rceil)$, because at the end of each epoch, the synchronization round incurs $2N$ communication cost. We minimize C_T by choosing $\alpha = \sqrt{\frac{DTN}{R}}$, so that $C_T = O(Nd^2 \cdot \sqrt{\frac{TR}{DN}})$. Note that this result is the same as [28] (we can see this by simply substituting T in our result with TN), because T in our work denotes the total number of iterations for all N clients.

Communication cost of Sync-LinUCB under non-uniform client distribution However, for most applications in reality, the client distribution can hardly be uniform, i.e., the clients have distinct availability of new observations. Then the global synchronization of Sync-LinUCB leads to a waste of communication in this more common situation. Specifically, when considering epochs with less than α time steps, the number of interactions for any client $i \in [N]$ can be equal to $t_e - t_s$ in the worst case, i.e., all the interactions with the environment in this epoch are done by this single client. In this case, $\Delta t_{i,t_e} < \alpha$, which is different from the case of uniform client distribution. Therefore, $\log \frac{\det(V_{t_e})}{\det(V_{t_s})} > \frac{D}{\alpha}$. The number of epochs with less than α time steps is at most $\lceil \frac{R\alpha}{D} \rceil$. Then $C_T = Nd^2 \cdot (\lceil \frac{T}{\alpha} \rceil + \lceil \frac{R\alpha}{D} \rceil)$. Similarly, we choose $\alpha = \sqrt{\frac{DT}{R}}$ to minimize C_T, so that $C_T = O(Nd^2 \cdot \sqrt{\frac{TR}{D}})$. We can see that this is larger than the communication cost under a uniform client distribution by a factor of $\sqrt{N}$.

Comparison between Async-LinUCB and Sync-LinUCB

In this section, we provide more details about the theoretical results of Async-LinUCB, and add the corresponding results of Sync-LinUCB for comparison (see Table 2.3). Depending on the application, the thresholds γ_U and γ_D of Async-LinUCB can be flexibly adjusted to get various trade-off between R_T and C_T. For all the discussions below, we constrain $\gamma_U = \gamma_D = \gamma$ for simplicity. However, when necessary, different values can be chosen for γ_U and γ_D for different clients. This gives our algorithm much more flexibility in practice, i.e., allows for a fine-grained control of every single edge in the communication network, compared with Sync-LinUCB. For example, for users who are less willing to participate in frequent uploads and downloads, a higher threshold can be chosen for their corresponding clients to reduce communication, and vice versa.

Table 2.3: Upper bounds for R_T and C_T under different thresholds.

Algorithm	Threshold	R_T	C_T (uniform)	C_T (non-uniform)
Async-LinUCB	$\gamma = 1$	$d\sqrt{T}\log T$	Nd^2T	Nd^2T
	$\gamma = \exp(N^{-1})$	$d\sqrt{T}\log T$	$N^2d^3\log T$	$N^2d^3\log T$
	$\gamma = \exp(N^{-\frac{1}{2}})$	$N^{\frac{1}{4}}d\sqrt{T}\log T$	$N^{\frac{3}{2}}d^3\log T$	$N^{\frac{3}{2}}d^3\log T$
	$\gamma = +\infty$	$N^{\frac{1}{2}}d\sqrt{T}\log T$	0	0
Sync-LinUCB	$D = T/(N^2 d\log T)$	$d\sqrt{T}\log T$	$N^{\frac{3}{2}}d^3\log T$	$N^2d^3\log T$
	$D = T/(N^{\frac{3}{2}} d\log T)$	$N^{\frac{1}{4}}d\sqrt{T}\log T$	$N^{\frac{5}{4}}d^3\log T$	$N^{\frac{7}{4}}d^3\log T$

When setting $\gamma = +\infty$, all communications in the learning system are blocked; and in this case, $C_T = 0$ and $R_T = \tilde{O}(N^{\frac{1}{2}}d\sqrt{T}\log T)$, which recovers the regret of running an instance of LinUCB for each client independently. When setting $\gamma = 1$, the upload and download events are always triggered, i.e., synchronize all N clients in each time step. And in this case $C_T = NTd^2$ and $R_T = \tilde{O}(d\sqrt{T}\log T)$, which recovers the regret in the centralized setting.

What we prefer is to strike a balance between these two extreme cases, i.e., reduce the communication cost without sacrificing too much on regret. Specifically, we should note that T is the dominating variable for almost all applications instead of N or d. Since even without communication $R_T = \tilde{O}(N^{\frac{1}{2}}d\sqrt{T}\log T)$ already matches the minimax lower bound $\Omega(d\sqrt{T})$ in T (up to a logarithmic factor) and d, we are mostly interested in the case where C_T's rate in T is improved from $O(T)$ to $O(\log T)$.

For example, we can set Async-LinUCB's upper bound of the communication cost $C_T \leq Nd^3\frac{\log T}{\log\gamma}$ to be $N^{\frac{3}{2}}d\log T$, and thus $\gamma = \exp(N^{-\frac{1}{2}})$. Then by substituting γ into the upper bound of R_T, we have

$$R_T = \tilde{O}\left(\sqrt{(N-1)\gamma^2 + (2-N)\gamma}d\sqrt{T}\log T\right) = \tilde{O}\left(\sqrt{(N-1)e^{2N^{-\frac{1}{2}}} + (2-N)e^{N^{-\frac{1}{2}}}}d\sqrt{T}\log T\right)$$

Since $\lim_{N\to\infty}\frac{\sqrt{(N-1)e^{2N^{-\frac{1}{2}}}+(2-N)e^{N^{-\frac{1}{2}}}}}{N^{\frac{1}{4}}} = 1$, we know $\sqrt{(N-1)e^{2N^{-\frac{1}{2}}} + (2-N)e^{N^{-\frac{1}{2}}}} = O(N^{\frac{1}{4}})$. Therefore, $R_T = \tilde{O}(N^{\frac{1}{4}}d\sqrt{T}\log T)$. And similarly, by setting $\gamma = \exp(N^{-1})$, Async-LinUCB has $C_T = N^2d^3\log T$ and $R_T = \tilde{O}(d\sqrt{T}\log T)$. For both choices of γ, at the cost of an increased rate in N, we have improved C_T's rate in the dominating variable T from $O(T)$ to $O(\log T)$.

For comparison, we choose the threshold D for Sync-LinUCB such that its upper bound of R_T matches that of Async-LinUCB; and we include the corresponding results in Table 2.3 as well. We can see that Async-LinUCB's upper bound of C_T is not influenced by whether the client distribution is uniform or not, while Sync-LinUCB is, as we have shown in Section 2.3.6. Specifically, under the same regret $R_T = O(N^{\frac{1}{4}}d\sqrt{T}\log T)$, in terms of C_T's rate in N, Sync-LinUCB is slightly better than Async-LinUCB (by a factor of $O(N^{\frac{1}{4}})$) under the ideal case of uniform client distribution, and slightly worse than Async-LinUCB (by a factor of $O(N^{\frac{1}{4}})$) under non-uniform client distribution.

2.3.7 Asynchronous communication for kernelized contextual bandit

In [100], we proposed the first algorithm for distributed kernel bandit that has sub-linear communication cost. We achieved this via a Nyström embedding function [101] shared among all the clients, such that the clients only need to transfer the embedded statistics for joint kernelized estimation. Nevertheless, the update of the Nyström embedding function, as well as the communication of the embedded statistics, relies on a synchronization round that requires participation of all the clients. To improve algorithm's robustness against stragglers (i.e., slower clients) in the system, we investigate the asynchronous communication for kernelized contextual bandit, such that the server can readily perform model update when communication from a client is received, with no need to wait for others.

The main bottleneck in addressing this limitation of [100] lies in computing Nyström approximation under *asynchronous communication*. Specifically, during synchronization step, Approx-DiskernelUCB algorithm in [100] first samples a small set of representative data points (i.e., the dictionary) from all clients, and then lets each client project their local data to the subspace spanned by this dictionary and share statistics about the projected data with others. However, new challenges arise in both algorithmic design and theoretical analysis when extending their solution to asynchronous communication, since a 'fresh' re-sample from the data of all clients is no longer possible, and each client has a different copy of the dictionary due to the asynchronous communication with the server, such that their local data will be projected to different subspaces, and thus causes difficulty in joint kernel estimation. In this work, we address these challenges and propose the first asynchronous algorithm for federated kernelized contextual bandits. Compared with prior works in federated bandits, our algorithm simultaneously enjoys the modeling capacity of non-parametric models and the improved robustness against delays and unavailability of clients, making it suitable for a wider range of applications.

Kernelized reward function Following [26, 100], we assume the unknown reward function f lies in the RKHS, denoted as $\mathcal{H}$, such that the reward can be equivalently written as $y_t = \theta_\star^\top \phi(\mathbf{x}_t) + \eta_t$, where $\theta_\star \in \mathcal{H}$ is an unknown parameter vector and $\phi : \mathbb{R}^d \to \mathcal{H}$ is a known feature map associated with $\mathcal{H}$. We assume that η_t is zero-mean R-sub-Gaussian conditioned on $\sigma\big((i_s, \mathbf{x}_s, \eta_s)_{s\in[t-1]}, i_t, \mathbf{x}_t\big)$, i.e., the σ-algebra generated by previous clients, their pulled arms, and the corresponding noises. In addition, there exists a positive definite kernel $k(\cdot,\cdot)$ associated with $\mathcal{H}$, and we assume $\forall \mathbf{x} \in \mathcal{A} := \cup_{t\in[T]}\mathcal{A}_t$ that, $\|\mathbf{x}\|_k \leq L$ and $\|f\|_k \leq S$ for some $L, S > 0$.

Throughout this section, we use $\mathcal{D} \subseteq [T]$ to denote a set of time steps and $|\mathcal{D}|$ as its size. The design matrix and reward vector constructed using data collected at these time steps, i.e., $\{\mathbf{x}_s, y_s\}_{s\in\mathcal{D}}$, are denoted as $\mathbf{X}_\mathcal{D} = [\mathbf{x}_s]_{s\in\mathcal{D}}^\top \in \mathbb{R}^{|\mathcal{D}|\times d}$ and $\mathbf{y}_\mathcal{D} = [y_s]_{s\in\mathcal{D}}^\top \in \mathbb{R}^{|\mathcal{D}|}$. Applying feature map $\phi(\cdot)$ to each row of $\mathbf{X}_\mathcal{D}$, we have $\mathbf{\Phi}_\mathcal{D} \in \mathbb{R}^{|\mathcal{D}|\times p}$, where p denotes the dimension of $\mathcal{H}$ and is possibly infinite.

Kernel Ridge regression Since the reward function f is linear in $\mathcal{H}$, one can construct the Ridge regression estimator for $\theta_\star$ as,

$$\hat{\theta} = (\mathbf{\Phi}_\mathcal{D}^\top \mathbf{\Phi}_\mathcal{D} + \lambda \mathbf{I})^{-1} \mathbf{\Phi}_\mathcal{D}^\top \mathbf{y}_\mathcal{D}$$

where $\lambda > 0$ is the regularization parameter. This gives us the following estimated mean reward and standard deviation in the primal form for any arm $\mathbf{x} \in \mathcal{A}$:

$$\begin{aligned}\hat{\mu}(\mathbf{x}) &= \phi(\mathbf{x})^\top \left(\mathbf{\Phi}_\mathcal{D}^\top \mathbf{\Phi}_\mathcal{D} + \lambda \mathbf{I}\right)^{-1} \left(\mathbf{\Phi}_\mathcal{D}^\top \mathbf{y}_\mathcal{D}\right) \\ \hat{\sigma}(\mathbf{x}) &= \sqrt{\phi(\mathbf{x})^\top \left(\mathbf{\Phi}_\mathcal{D}^\top \mathbf{\Phi}_\mathcal{D} + \lambda \mathbf{I}\right)^{-1} \phi(\mathbf{x})}.\end{aligned}$$

Note that directly working with the possibly infinite-dimension $\hat{\theta} \in \mathbb{R}^p$ is impractical. Instead, using the kernel trick [26, 100], we can obtain an equivalent dual form that only involves entries of the kernel matrix:

$$\begin{aligned}\hat{\mu}(\mathbf{x}) &= \mathbf{K}_\mathcal{D}(\mathbf{x})^\top \left(\mathbf{K}_{\mathcal{D},\mathcal{D}} + \lambda \mathbf{I}\right)^{-1} \mathbf{y}_\mathcal{D} \\ \hat{\sigma}(\mathbf{x}) &= \lambda^{-1/2} \sqrt{k(\mathbf{x},\mathbf{x}) - \mathbf{K}_\mathcal{D}(\mathbf{x})^\top \left(\mathbf{K}_{\mathcal{D},\mathcal{D}} + \lambda \mathbf{I}\right)^{-1} \mathbf{K}_\mathcal{D}(\mathbf{x})}\end{aligned} \tag{2.34}$$

where $\mathbf{K}_\mathcal{D}(\mathbf{x}) = \mathbf{\Phi}_\mathcal{D}\phi(\mathbf{x}) = [k(\mathbf{x}_s, \mathbf{x})]_{s\in\mathcal{D}}^\top \in \mathbb{R}^{|\mathcal{D}|}$ and $\mathbf{K}_{\mathcal{D},\mathcal{D}} = \mathbf{\Phi}_\mathcal{D}^\top \mathbf{\Phi}_\mathcal{D} = [k(\mathbf{x}_s, \mathbf{x}_{s'})]_{s,s'\in\mathcal{D}} \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{D}|}$.

Nyström approximation Though Eq (2.34) avoids directly working in $\mathcal{H}$, it requires computing the inverse of $\mathbf{K}_{\mathcal{D},\mathcal{D}}$, which is expensive in terms of both computation cost [73], i.e., $O(T^3)$ as $|\mathcal{D}| = O(T)$, and communication cost [100], i.e., $O(T)$ as $\{(\mathbf{x}_s, y_s)\}_{s\in\mathcal{D}}$ needs to be transferred across the clients. Therefore, Nyström method is used to approximate Eq (2.34), so clients can share embedded statistics, which improves communication efficiency.

As [75, 100], we project the original dataset $\mathcal{D}$[6] to the subspace defined by a small representative subset $\mathcal{S} \subseteq \mathcal{D}$, i.e., the dictionary, and the orthogonal projection matrix is defined as

$$\mathbf{P}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \left(\mathbf{\Phi}_{\mathcal{S}} \mathbf{\Phi}_{\mathcal{S}}^\top\right)^{-1} \mathbf{\Phi}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \mathbf{K}_{\mathcal{S},\mathcal{S}}^{-1} \mathbf{\Phi}_{\mathcal{S}} \in \mathbb{R}^{p \times p}.$$

Taking eigen-decomposition of $\mathbf{K}_{\mathcal{S},\mathcal{S}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, we can rewrite the orthogonal projection as $\mathbf{P}_{\mathcal{S}} = \mathbf{\Phi}_{\mathcal{S}}^\top \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top \mathbf{\Phi}_{\mathcal{S}}$, and define the Nyström embedding function as

$$z(\mathbf{x}; \mathcal{S}) = \mathbf{P}_{\mathcal{S}}^{1/2} \phi(\mathbf{x}) = \mathbf{\Lambda}^{-1/2}\mathbf{U}^\top \mathbf{\Phi}_{\mathcal{S}} \phi(\mathbf{x}) = \mathbf{K}_{\mathcal{S},\mathcal{S}}^{-1/2} \mathbf{K}_{\mathcal{S}}(\mathbf{x}),$$

which maps the data point $\mathbf{x}$ from $\mathbb{R}^d$ to $\mathbb{R}^{|\mathcal{S}|}$. Therefore, we can approximate the Ridge regression estimator on dataset $\mathcal{D}$ as $\tilde{\theta} = \left(\mathbf{P}_{\mathcal{S}} \mathbf{\Phi}_{\mathcal{D}}^\top \mathbf{\Phi}_{\mathcal{D}} \mathbf{P}_{\mathcal{S}} + \lambda \mathbf{I}\right)^{-1} \left(\mathbf{P}_{\mathcal{S}} \mathbf{\Phi}_{\mathcal{D}}^\top \mathbf{y}_{\mathcal{D}}\right)$, and Eq (2.34) as

$$\begin{aligned} \tilde{\mu}(\mathbf{x}) &= z(\mathbf{x}; \mathcal{S})^\top \left(\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}} + \lambda \mathbf{I}\right)^{-1} \mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{y}_{\mathcal{D}} \\ \tilde{\sigma}(\mathbf{x}) &= \lambda^{-1/2} \sqrt{k(\mathbf{x}, \mathbf{x}) - z(\mathbf{x}; \mathcal{S})^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}} [\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}} + \lambda \mathbf{I}]^{-1} z(\mathbf{x}|\mathcal{S})} \end{aligned} \tag{2.35}$$

where $\mathbf{Z}_{\mathcal{D};\mathcal{S}} \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{S}|}$ is obtained by applying $z(\cdot; \mathcal{S})$ to each row of $\mathbf{X}_{\mathcal{D}}$, i.e., $\mathbf{Z}_{\mathcal{D};\mathcal{S}} = \mathbf{\Phi}_{\mathcal{D}} \mathbf{P}_{\mathcal{S}}^{1/2}$. Note that the computation of $\tilde{\mu}(\mathbf{x})$ and $\tilde{\sigma}(\mathbf{x})$ only requires the embedded statistics, i.e., matrix $\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and vector $\mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{y}_{\mathcal{D}} \in \mathbb{R}^{|\mathcal{S}|}$, which makes joint kernelized estimation among N clients much more efficient in communication compared with Eq (2.34).

2.3.8 Async-KernelUCB algorithm

In this section, we propose and analyze the first asynchronous algorithm for distributed kernelized contextual bandit problem that addresses the aforementioned challenges, and name the resulting algorithm Async-KernelUCB, with its description given in Algorithm 14.

We denote the embedded statistics used in the computation of Eq (2.35) by $\tilde{\mathbf{A}}(\mathcal{D}; \mathcal{S}) := \mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{Z}_{\mathcal{D};\mathcal{S}}$ and $\tilde{\mathbf{b}}(\mathcal{D}; \mathcal{S}) := \mathbf{Z}_{\mathcal{D};\mathcal{S}}^\top \mathbf{y}_{\mathcal{D}}$, to explicitly emphasize they are computed by projecting the data points from dataset $\mathcal{D}$ to the subspace spanned by dictionary $\mathcal{S}$. We denote the sequence of time steps corresponding to the interactions between client i and the environment up to time t as $\mathcal{N}_t(i) = \{1 \leq s \leq t : i_s = i\}$ for $t \in [T]$. Throughout this section, we reserve k as the index for communication, and use $t_k \in [T]$ to denote the time step when the k-th communication happens. Moreover, as each client has a different copy of the embedding function and embedded statistics due to asynchronous communication, we use $\underline{k}(i)$ to denote the index of client i's latest communication with the server, up to the k-th one: if client i triggers the k-th communication, then $\underline{k}(i) = k$.

Arm selection At each round $t \in [T]$, client $i_t \in [N]$ selects arm $\mathbf{x}_t$ from the candidate set $\mathcal{A}_t$ by maximizing the following upper confidence bound (line 5)

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{A}_t} \tilde{\mu}_{\underline{k}(i_t)}(\mathbf{x}) + \alpha \tilde{\sigma}_{\underline{k}(i_t)}(\mathbf{x}) \tag{2.36}$$

where $\tilde{\mu}_{\underline{k}(i_t)}(\mathbf{x})$ and $\tilde{\sigma}_{\underline{k}(i_t)}(\mathbf{x})$ are approximated mean and standard deviation of arm $\mathbf{x}$'s reward, computed using statistics $\tilde{\mathbf{A}}(\mathcal{D}_{\underline{k}(i_t)}, \mathcal{S}_{\underline{k}(i_t)})$ and $\tilde{\mathbf{b}}(\mathcal{D}_{\underline{k}(i_t)}, \mathcal{S}_{\underline{k}(i_t)})$ that client i_t received from the server during the $\underline{k}(i_t)$-th communication. Proper choice of α is given in Lemma 2.3.7.

Event-triggered asynchronous communication After the interaction at time step t, $\tilde{\mu}_{\underline{k}(i_t)}(\cdot)$ and $\tilde{\sigma}_{\underline{k}(i_t)}(\cdot)$ of active client i_t will only be updated if the following event is true (line 7):

$$\sum_{s \in \mathcal{N}_t(i_t) \setminus \mathcal{N}_{t_{\underline{k}(i_t)}}(i_t)} \tilde{\sigma}^2_{\underline{k}(i_t)}(\mathbf{x}_s) > D, \tag{2.37}$$

[6] Throughout this work, we will often use the set of indices $\mathcal{D}$ (or $\mathcal{S}$) to refer to the actual dataset $\{\mathbf{x}_s, y_s\}_{s\in\mathcal{D}}$ (or dictionary $\{\mathbf{x}_s, y_s\}_{s\in\mathcal{S}}$) for simplicity.

Algorithm 14 Asynchronous KernelUCB (Async-KernelUCB)

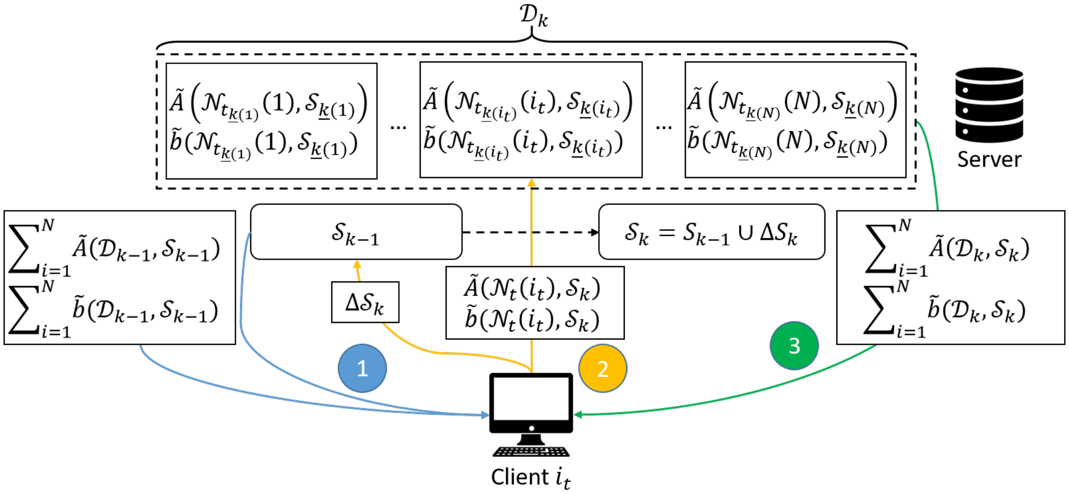
```
1: Input: $\alpha$, $\bar{q}$, communication threshold $D > 0$, regularization parameter $\lambda > 0$, $\delta \in (0, 1)$ and kernel function $k(\cdot, \cdot)$.
2: Initialize approximated mean and variance $\tilde{\mu}_0(\mathbf{x}) = 0$, $\tilde{\sigma}_0(\mathbf{x}) = \lambda^{-1/2}\sqrt{k(\mathbf{x}, \mathbf{x})}$, dataset $\mathcal{D}_0 = \emptyset$, dictionary $\mathcal{S}_0 = \emptyset$, index of communication $k = 0$, and $\mathcal{N}_0(i) = \emptyset$ for each client $i \in [N]$
3: for $t = 1, 2, ..., T$ do
4:     Client $i_t \in [N]$ becomes active, and observes arm set $\mathcal{A}_t$
5:     [Client $i_t$] Choose arm $\mathbf{x}_t \in \mathcal{A}_t$ according to Eq (2.36), and observe reward $y_t$
6:     // Set $\mathcal{N}_t(i_t) = \mathcal{N}_{t-1}(i_t) \cup \{t\}$, and $\mathcal{N}_t(i) = \mathcal{N}_{t-1}(i)$ for $i \neq i_t$
7:     if $\sum_{s \in \mathcal{N}_t(i_t) \setminus \mathcal{N}_{t_{\underline{k}(i_t)}}(i_t)} \tilde{\sigma}^2_{\underline{k}(i_t)}(\mathbf{x}_s) > D$ then
           // Denote $\Delta\mathcal{D}_k = \mathcal{N}_t(i_t) \setminus \mathcal{N}_{t_{\underline{k}(i_t)}}(i_t)$, and set $k = k + 1$
8:         [Server → Client $i_t$] Send $\{\mathbf{x}_s, y_s\}_{s \in \mathcal{S}_{k-1}}$, $\tilde{\mathbf{A}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1}), \tilde{\mathbf{b}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1})$ to client $i_t$
9:         [Client $i_t$] Select $\Delta\mathcal{S}_k \subseteq \Delta\mathcal{D}_k$ via RLS sampling with probability $\bar{q}\tilde{\sigma}^2_{k-1}(\cdot)$
           // Set $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \Delta\mathcal{S}_k$
10:        [Client $i_t$] Compute $\tilde{\mathbf{A}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$, $\tilde{\mathbf{b}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$
11:        [Client $i_t$ → Server] Send $\{\mathbf{x}_s, y_s\}_{s \in \Delta\mathcal{S}_k}$, $\tilde{\mathbf{A}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$ and $\tilde{\mathbf{b}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$ to server
           // Set $\mathcal{D}_k = \mathcal{D}_{k-1} \cup \Delta\mathcal{D}_k$
12:        [Server] Compute $\tilde{\mathbf{A}}(\mathcal{D}_k; \mathcal{S}_k), \tilde{\mathbf{b}}(\mathcal{D}_k; \mathcal{S}_k)$ according to Eq (2.38)
13:        [Server → Client $i_t$] Send $\tilde{\mathbf{A}}(\mathcal{D}_k; \mathcal{S}_k), \tilde{\mathbf{b}}(\mathcal{D}_k; \mathcal{S}_k)$ to client $i_t$
14:        [Client $i_t$] Update $\tilde{\mu}_k(\cdot)$ and $\tilde{\sigma}_k(\cdot)$ using $\tilde{\mathbf{A}}(\mathcal{D}_k; \mathcal{S}_k), \tilde{\mathbf{b}}(\mathcal{D}_k; \mathcal{S}_k)$ according to Eq (2.35)
```

where $D > 0$ denotes the communication threshold. This measures whether sufficient amount of new information has been collected by client i_t since its lastest (the $\underline{k}(i_t)$-th) communication with the server. If true, communication between client i_t and the server is triggered (line 8-14), where the update procedure described in the following paragraphs will be performed. And this procedure is also illustrated in Figure 2.12.

Dictionary and embedded statistics update During the k-th communication, the server first sends its latest dictionary $\mathcal{S}_{k-1}$, as well as its latest embedded statistics $\tilde{\mathbf{A}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1})$ and $\tilde{\mathbf{b}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1})$, to client i_t (line 8), which is illustrated as the blue lines in Figure 2.12. Then client i_t selects a subset $\Delta\mathcal{S}_k$ from the data it has collected since its lastest communication (line 9), i.e., $\Delta\mathcal{D}_k$, which will be used to incrementally update dictionary $\mathcal{S}_{k-1}$. This is done by sampling $q_{k,s} \sim \mathcal{B}(\tilde{p}_{k,s})$ for each data point with time index $s \in \Delta\mathcal{D}_k$, where $\tilde{p}_{k,s} := \bar{q}\tilde{\sigma}^2_{k-1}(\mathbf{x}_s)$. This can be considered as a variant of Ridge leverage score (RLS) sampling [75, 100]. It is worth noting that the only purpose of sending $\tilde{\mathbf{A}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1})$ and $\tilde{\mathbf{b}}(\mathcal{D}_{k-1}; \mathcal{S}_{k-1})$ is to enable RLS sampling with the latest $\tilde{\sigma}^2_{k-1}(\cdot)$. Otherwise, client i_t, whose lastest communication with the server can be long time ago, would include unnecessary data points into $\Delta\mathcal{S}_k$ due to its unawareness of server's current status. We will demonstrate in the proof of Lemma 2.3.6 that our design here is necessary to obtain a compact dictionary under asynchronous communication. With the dictionary updated, client i_t computes the embeddings of its new local data, i.e., $\tilde{\mathbf{A}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$ and $\tilde{\mathbf{b}}(\Delta\mathcal{D}_k; \mathcal{S}_k)$, and sends them, as well as $\Delta\mathcal{S}_k$, to the server (the yellow lines in Figure 2.12).

As shown in Figure 2.12, the server stores: 1) the last received embedded statistics from each client $i \in [N]$, i.e., $\tilde{\mathbf{A}}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_{\underline{k}(i)}) \in \mathbb{R}^{|\mathcal{S}_{\underline{k}(i)}| \times |\mathcal{S}_{\underline{k}(i)}|}$ and $\tilde{\mathbf{b}}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_{\underline{k}(i)}) \in \mathbb{R}^{|\mathcal{S}_{\underline{k}(i)}|}$; 2) their corresponding dictionary $\mathcal{S}_{\underline{k}(i)}$. As mentioned earlier, due to asynchronous communication, the statistics from different clients are based on different dictionaries, which means they have different dimensions and thus *cannot be directly aggregated* as in [100]. We propose to transform the statistics from each client $i \in [N]$ using the latest dictionary $\mathcal{S}_k$. This is based on the fact that $\mathbf{Z}_{\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_k} = \mathbf{\Phi}_{\mathcal{N}_{t_{\underline{k}(i)}}(i)}\mathbf{P}^{1/2}_{\mathcal{S}_k} = \mathbf{\Phi}_{\mathcal{N}_{t_{\underline{k}(i)}}(i)}\mathbf{P}^{1/2}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{P}^{-1/2}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{P}^{1/2}_{\mathcal{S}_k} = \mathbf{Z}_{\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_{\underline{k}(i)}}\mathcal{T}_{\underline{k}(i),k}$, where the linear transformation $\mathcal{T}_{\underline{k}(i),k} := \mathbf{P}^{-1/2}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{P}^{1/2}_{\mathcal{S}_k} = \mathbf{\Lambda}^{1/2}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{U}^{\top}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{\Phi}_{\mathcal{S}_{\underline{k}(i)}}\mathbf{\Phi}^{\top}_{\mathcal{S}_k}\mathbf{U}_{\mathcal{S}_k}\mathbf{\Lambda}^{-1/2}_{\mathcal{S}_k}$ serves the purpose. Hence, we have

$$
\begin{aligned}
\tilde{\mathbf{A}}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_k) &= \mathcal{T}^{\top}_{\underline{k}(i),k}\tilde{\mathbf{A}}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_{\underline{k}(i)})\mathcal{T}_{\underline{k}(i),k}, \\
\tilde{b}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_k) &= \mathcal{T}^{\top}_{\underline{k}(i),k}\tilde{b}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_{\underline{k}(i)}),
\end{aligned}
\tag{2.38}
$$



$\mathcal{D}_k$
$\tilde{A}\left(\mathcal{N}_{t_{\underline{k}(1)}}(1), \mathcal{S}_{\underline{k}(1)}\right)$
$\tilde{b}(\mathcal{N}_{t_{\underline{k}(1)}}(1), \mathcal{S}_{\underline{k}(1)})$
...
$\tilde{A}\left(\mathcal{N}_{t_{\underline{k}(i_t)}}(i_t), \mathcal{S}_{\underline{k}(i_t)}\right)$
$\tilde{b}(\mathcal{N}_{t_{\underline{k}(i_t)}}(i_t), \mathcal{S}_{\underline{k}(i_t)})$
...
$\tilde{A}\left(\mathcal{N}_{t_{\underline{k}(N)}}(N), \mathcal{S}_{\underline{k}(N)}\right)$
$\tilde{b}(\mathcal{N}_{t_{\underline{k}(N)}}(N), \mathcal{S}_{\underline{k}(N)})$
Server
$\sum_{i=1}^{N} \tilde{A}(\mathcal{D}_{k-1}, \mathcal{S}_{k-1})$
$\sum_{i=1}^{N} \tilde{b}(\mathcal{D}_{k-1}, \mathcal{S}_{k-1})$
$\mathcal{S}_{k-1}$
$\mathcal{S}_k = \mathcal{S}_{k-1} \cup \Delta\mathcal{S}_k$
$\sum_{i=1}^{N} \tilde{A}(\mathcal{D}_k, \mathcal{S}_k)$
$\sum_{i=1}^{N} \tilde{b}(\mathcal{D}_k, \mathcal{S}_k)$
$\Delta\mathcal{S}_k$
$\tilde{A}(\mathcal{N}_t(i_t), \mathcal{S}_k)$
$\tilde{b}(\mathcal{N}_t(i_t), \mathcal{S}_k)$
1
2
3
Client i_t

Figure 2.12: Illustration of asynchronous update of dictionary and embedded statistics

which makes the statistics received from all clients have the same dimension $|\mathcal{S}_k|$. Then we compute $\tilde{A}(\mathcal{D}_k; \mathcal{S}_k) = \sum_{i=1}^{N} \tilde{\mathbf{A}}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_k)$ and $\tilde{b}(\mathcal{D}_k; \mathcal{S}_k) = \sum_{i=1}^{N} \tilde{b}(\mathcal{N}_{t_{\underline{k}(i)}}(i); \mathcal{S}_k)$ (line 12), and send them to client i_t to update its UCB (line 13-14), which is illustrated as the green line in Figure 2.12.

2.3.9 Dictionary accuracy and size analysis

As mentioned earlier, the key to low regret and low communication cost, is to have a dictionary $\mathcal{S}_k$ that can accurately approximate the dataset $\mathcal{D}_k$, while having a compact size $|\mathcal{S}_k|$. In this section, we show that this is possible with our update procedure in Section 2.3.8. First, we need some additional notations. We denote the total number of times up to time T that communication is triggered, i.e., the number of times Eq (2.37) is true, as B, where $B \in [0, T]$. Following [75, 100], the approximation quality is formally defined using ϵ-accuracy: if the event

$$\left\{(1-\epsilon)(\mathbf{\Phi}_{\mathcal{D}_k}^{\top}\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I}) \preceq \mathbf{\Phi}_{\mathcal{D}_k}^{\top}\bar{\mathbf{S}}_k^{\top}\bar{\mathbf{S}}_k\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I} \preceq (1+\epsilon)(\mathbf{\Phi}_{\mathcal{D}_k}^{\top}\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I})\right\} \tag{2.39}$$

is true, then we say the dictionary $\mathcal{S}_k$ is ϵ-accurate w.r.t. dataset $\mathcal{D}_k$, for some $\epsilon \in (0,1)$, where $\bar{\mathbf{S}}_k \in \mathbb{R}^{|\mathcal{D}_k| \times |\mathcal{D}_k|}$ denotes a diagonal matrix, with s-th diagonal entry equal to $q_{k,s}/\sqrt{\tilde{p}_{k,s}}$, where $q_{k,s} = 1$ if $s \in \mathcal{S}_k$, and $q_{k,s} = 0$, otherwise. Based on this notion, we prove Lemma 2.3.4 below.

Lemma 2.3.4 (Dictionary Accuracy and Size)**.** *With $\bar{q} = 4\ln(2\sqrt{2}T/\delta)\beta(1+\epsilon/3)/\epsilon^2$, where $\beta := (1+\epsilon)/(1-\epsilon)$, and $\lambda \leq k(\mathbf{x}, \mathbf{x}), \forall \mathbf{x} \in \mathcal{A}$, we have with probability at least $1-\delta$ that dictionary $\mathcal{S}_k$ is ϵ-accurate w.r.t. dataset $\mathcal{D}_k$, and its size $|\mathcal{S}_k| \leq 12\beta(1+\beta D)\bar{q}\gamma_T$, $\forall k$, where $\delta \in (0,1)$.*

This shows that our incremental update procedure under asynchronous communication still matches the results in prior works that perform synchronous re-sampling over the whole dataset for dictionary update [100, 75]. We provide a proof sketch for Lemma 2.3.4 below to highlight our technical novelty and provide the detailed proof in Section 2.3.12.

Proof Sketch of Lemma 2.3.4. Let's define the unfavorable event $H_k = A_k \cup E_k$, where A_k is the event that the dictionary $\mathcal{S}_k$ is not ϵ-accurate w.r.t. $\mathcal{D}_k$, and E_k is the event that the size of dictionary $|\mathcal{S}_k| > 12\beta(1+\beta D)\bar{q}\gamma_T$. Therefore, the probability of $\cup_{k=0}^{B} H_k$ can be decomposed as

$$\mathbb{P}\big(\cup_{k=0}^{B} H_k\big) = \mathbb{P}\big(\cup_{k=0}^{B} A_k\big) + \mathbb{P}\big((\cup_{k=0}^{B} E_k) \cap (\cup_{k=0}^{B} A_k)^C\big).$$

Bounding the first term: In [73, 75, 100], the first term is further decomposed as $\mathbb{P}\big(\cup_{k=0}^{B} A_k\big) \leq \sum_{k=1}^{B} \mathbf{P}(A_k \cap A_{k-1}^C)$, because dictionary $\mathcal{S}_k$ is constructed by a fresh re-sampling over $\mathcal{D}_k$ using the latest approximated variance $\tilde{\sigma}_{k-1}^2(\cdot)$, and thus they only need to guarantee $\tilde{\sigma}_{k-1}^2(\cdot)$ is a good approximation to $\sigma_{k-1}^2(\cdot)$. In our case, $\mathcal{S}_k$ is incrementally updated in each communication, i.e., $\mathcal{S}_k = \cup_{k'=1}^{k} \Delta\mathcal{S}_{k'}$ where each $\Delta\mathcal{S}_{k'}$ is sampled using $\tilde{\sigma}_{k'-1}^2(\cdot)$.

The accuracy of $\mathcal{S}_k$ depends on the accuracy of every $\mathcal{S}_{k'}$, i.e., $\cap_{k'=1}^{k-1} A_{k'}^C$. Therefore, we decompose $\mathbb{P}\big(\cup_{k=0}^B A_k\big) = 1 - \mathbb{P}\big(\cap_{k=0}^B A_k^C\big) = 1 - \prod_{k=1}^B [1 - \mathbb{P}\big(A_k | \cap_{k'=0}^{k-1} A_{k'}^C\big)] \leq \sum_{k=1}^B \mathbb{P}\big(A_k | \cap_{k'=0}^{k-1} A_{k'}^C\big)$ using Bayes theorem and Weierstrass product inequality, and bound each conditional probability separately, which leads to Lemma 2.3.5.

Lemma 2.3.5 (Bounding $\sum_{k=1}^B \mathbb{P}\big(A_k | \cap_{k'=0}^{k-1} A_{k'}^C\big)$)**.** *By setting $\bar{q} = 4\ln(2\sqrt{2}T/\delta)\beta(1{+}\epsilon/3)/\epsilon^2$, we have $\sum_{k=0}^B \mathbb{P}\big(A_k | \cap_{k'}^{k-1} A_{k'}^C\big) \leq \delta/2$, for $\delta \in (0,1)$.*

Bounding the second term: The second term can be decomposed as $\mathbb{P}\big((\cup_{k=0}^B E_k) \cap (\cup_{k=0}^B A_k)^C\big) \leq \sum_{k=0}^B \mathbb{P}\big(E_k \cap (\cap_{k=0}^B A_k^C)\big)$. Note that the size of dictionary $|\mathcal{S}_k| = \sum_{s \in \mathcal{D}_k} q_{k,s}$ by the definition of $q_{k,s}$, and its analysis relies on upper bounding $\sum_{s \in \mathcal{D}_k} \tilde{p}_{k,s}$ [75]. Again, due to asynchronous communication, for data point s that was added during the k'-th communication, i.e., $s \in \Delta\mathcal{D}_{k'}$, we have $q_{k,s} = q_{k',s}$, $\tilde{p}_{k,s} = \tilde{p}_{k',s}$ and thus $\sum_{s \in \mathcal{D}_k} \tilde{p}_{k,s} = \sum_{k'=1}^k \sum_{s \in \Delta\mathcal{D}_{k'}} \tilde{p}_{k',s}$. Compared with [100, 75] that re-sample all $s \in \mathcal{D}_k$ using $\tilde{p}_{k,s} = \bar{q}\tilde{\sigma}_{k-1}^2(\mathbf{x}_s)$, we use a different approximated variance function for each $\Delta\mathcal{S}_{k'}$. Nevertheless, with our design in Section 2.3.8, i.e., $\tilde{p}_{k',s} = \bar{q}\tilde{\sigma}_{k'-1}^2(\mathbf{x}_s)$, we show in Lemma 2.3.6 that we can still ensure $|\mathcal{S}_k| = O(\gamma_T)$, as long as a proper threshold D is chosen to avoid any $\Delta\mathcal{D}_{k'}$ being too large.

Lemma 2.3.6 (Bounding $\sum_{k=0}^B \mathbb{P}\big(E_k \cap (\cap_{k=0}^B A_k^C)\big)$)**.** *By setting $\bar{q} = 4\ln(2\sqrt{2}T/\delta)\beta(1+\epsilon/3)/\epsilon^2$, and $\lambda \leq k(\mathbf{x},\mathbf{x}), \forall \mathbf{x} \in \mathcal{A}$, we have $\sum_{k=0}^B \mathbb{P}\big(E_k \cap (\cap_{k=0}^B A_k^C)\big) \leq \delta/2$, for $\delta \in (0,1)$.*

Putting everything together, we have $\mathbb{P}\big(\cup_{k=0}^B H_k\big) \leq \delta$, for $\delta \in (0,1)$, which finishes the proof. □

2.3.10 Regret and communication cost analysis

Lemma 2.3.4 guarantees a compact and accurate dictionary for Nyström approximation throughout the learning process. Based on it, we establish the upper bounds for the cumulative regret and communication cost of Async-KernelUCB. First, motivated by the confidence ellipsoid for asynchronous linear bandits [97], we construct the following confidence ellipsoid for our approximated estimator for kernel bandit defined in Section 2.3.7 (proof is provided in Section 2.3.12).

Lemma 2.3.7 (Confidence ellipsoid for approximated estimator)**.** *Under the same condition as Lemma 2.3.4, with probability at least $1-2\delta$, for $\delta \in (0,1)$, we have $\forall k$ that*

$$\|\tilde{\theta}_k - \theta_\star\|_{\tilde{\mathbf{V}}_k} \leq (1/\sqrt{1-\epsilon}+1)\sqrt{\lambda}S + 2R\big(\sqrt{1+ND\beta} + N\sqrt{2D\beta}\big)\sqrt{\ln(1/\delta)+\gamma_T} := \alpha,$$

where $\tilde{\mathbf{V}}_k := \mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\mathbf{P}_{\mathcal{S}_k} + \lambda\mathbf{I}$ and $\gamma_T := \max_{\mathcal{D}\subset\mathcal{A}:|\mathcal{D}|=T} \frac{1}{2}\log\det(\mathbf{K}_{\mathcal{D},\mathcal{D}}/(D\beta\lambda)+\mathbf{I})$ [7] is the maximum information gain after T interactions [72, 100].

Then based on Lemma 2.3.7, we establish Theorem 2.3.8 below.

Theorem 2.3.8. *Under the same condition as Lemma 2.3.4, we have*

$$\begin{aligned} R_T \leq\ & 4N\gamma_T LS + 4\sqrt{2}\Big[(1/\sqrt{1-\epsilon}+1)\sqrt{\lambda}S + 2R\big(\sqrt{1+ND\beta} + N\sqrt{2D\beta}\big)\sqrt{\ln(1/\delta)+\gamma_T}\Big] \\ & \cdot \sqrt{T\beta[1+N\beta(L^2/\lambda+D)]\gamma_T} \end{aligned}$$

with probability at least $1-2\delta$, and

$$C_T \leq 2\gamma_T(N+4\beta/D)\big[3(|\mathcal{S}_B|^2 + |\mathcal{S}_B|) + d|\mathcal{S}_B|\big].$$

where the dictionary size $|\mathcal{S}_B| \leq 12\beta(1+\beta D)\bar{q}\gamma_T$ due to Lemma 2.3.4. By setting $D = 1/N^2$, we have $R_T = O\big(N\gamma_T LS + \sqrt{T}(S\sqrt{\gamma_T}+\gamma_T)\big)$, and $C_T = \tilde{O}(N^2\gamma_T^3)$.

[7] As discussed in [100], γ_T is problem-dependent, showing how fast kernel's eigenvalues decay. For kernels with exponentially decaying eigenvalues, i.e., $\lambda_m = O(\exp(-m^{\beta_e}))$, for $\beta_e > 0$, $\gamma_T = O(\log^{1+1/\beta_e}(T))$, which includes Gaussian kernel that is widely used for GPs and SVMs. For kernels with polynomially decaying eigenvalues, i.e., $\lambda_m = O(m^{-\beta_p})$, for $\beta_p > 1$, $\gamma_T = O(T^{1/\beta_p}\log^{1-1/\beta_p}(T))$.

2.3.11 Experiment setup & results

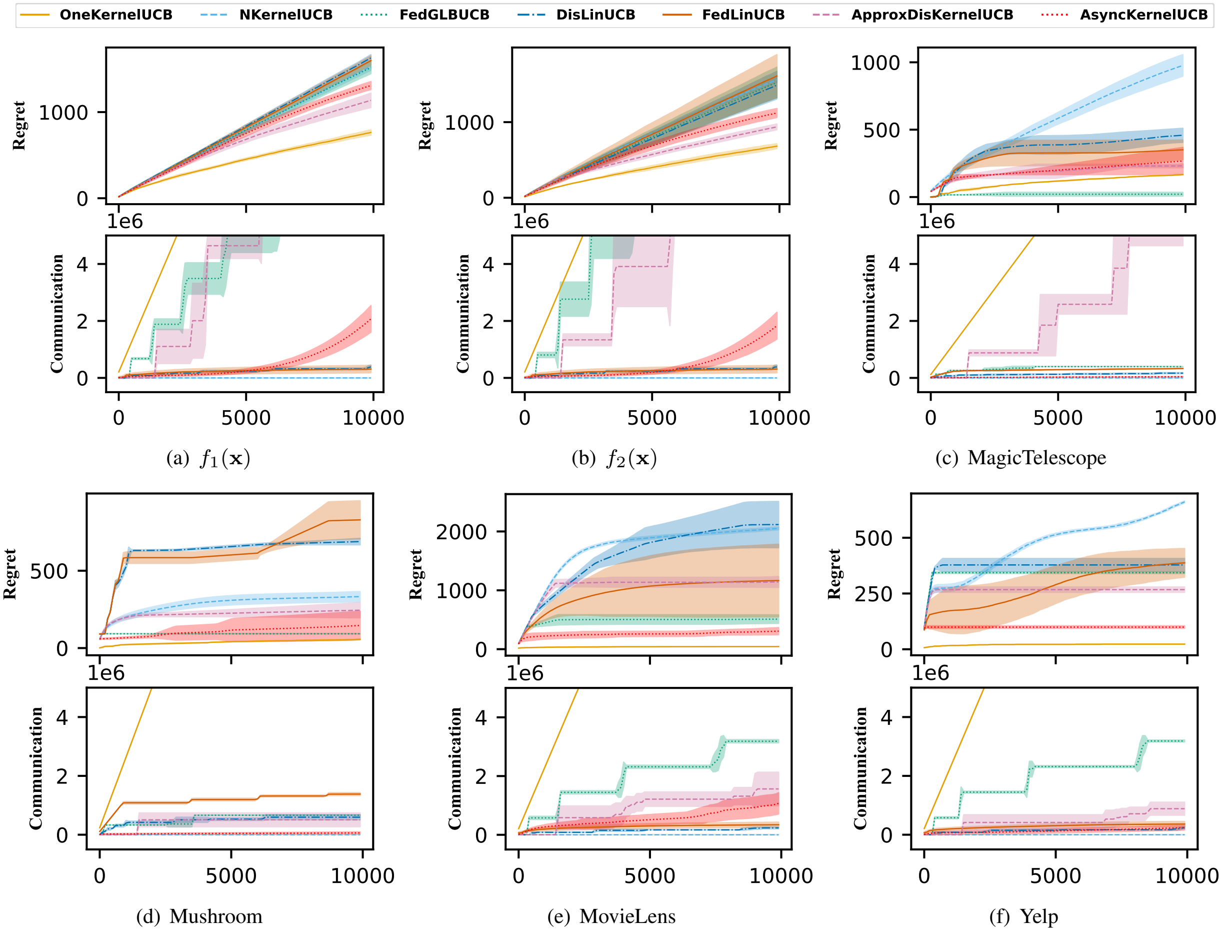
To validate Async-KernelUCB's effectiveness in reducing communication cost, we performed extensive empirical evaluations on both synthetic and real-world datasets, and reported the results (over 10 runs) in Figure 2.13. The baselines included in our comparisons are: 1) OneKernelUCB [26], it learns a shared kernel bandit model across all clients' aggregated data where data aggregation happens immediately after each new data point becomes available; 2) NKernelUCB, it learns a separate kernel bandit model for each client with no communication; 3) FedGLBUCB [84], it is a synchronous distributed GLB algorithm; 4) DisLinUCB [28], it is a synchronous distributed linear bandit algorithm; 5) FedLinUCB [97], it is an asynchronous distributed linear bandit algorithm; and 6) Approx-DisKernelUCB [100], it is a synchronous distributed kernel bandit algorithm. For all the kernel bandit algorithms, we used the Gaussian kernel $k(x, y) = \exp(-\gamma\|x - y\|^2)$, where we did a grid search of $\gamma \in \{0.1, 1, 4\}$, and for FedGLBUCB, we used Sigmoid function $\mu(z) = (1 + \exp(-z))^{-1}$ as link function. For all algorithms, instead of using their theoretically derived exploration coefficient α, we followed the convention [17, 94] to use grid search for α in $\{0.1, 1, 4\}$.

Synthetic dataset We simulated the distributed bandit setting in Section 2.3.7, with $d = 20, T = 10^4, N = 10^2$. At each time step $t \in [T]$, client $i_t \in [N]$ selects an arm from candidate set $\mathcal{A}_t$ (with $|\mathcal{A}_t| = 20$), which is uniformly sampled from a ℓ_2 unit ball. Then the reward is generated using one of the following reward functions: 1) $f_1(\mathbf{x}) = \cos(3\mathbf{x}^\top\theta_\star)$, and 2) $f_2(\mathbf{x}) = (\mathbf{x}^\top\theta_\star)^3 - 3(\mathbf{x}^\top\theta_\star)^2 - (\mathbf{x}^\top\theta_\star) + 3$, where the parameter $\theta_\star$ is uniformly sampled from a ℓ_2 unit ball and fixed.

UCI datasets We also performed experiments using MagicTelescope and Mushroom from the UCI Machine Learning Repository [89], which are converted to bandit problem following [25]. Specifically, we partitioned the dataset into 20 clusters using k-means, and used the centroid of each cluster as the context for the arms and used the averaged response as mean reward (the response is binarized by setting one class as 1, and all the others as 0). Then we simulated the distributed bandit setting in Section 2.3.7 with $|\mathcal{A}_t| = 20$, $T = 10^4$ and $N = 10^2$.

MovieLens and Yelp dataset Yelp dataset is released by the Yelp dataset challenge, and consists of 4.7 million rating entries for 157 thousand restaurants by 1.18 million users. MovieLens consists of 25 million ratings between 160 thousand users and 60 thousand movies [95]. Following the pre-processing steps in [96], we built the rating matrix by choosing the top 2,000 users and top 10,000 restaurants/movies and used singular-value decomposition to extract a 10-dimension feature vector for each user and restaurant/movie. We treated ratings greater than 2 as positive, and simulated the distributed bandit setting in Section 2.3.7 with $T = 10^4$ and $N = 10^2$. The candidate set $\mathcal{A}_t$ (with $|\mathcal{A}_t| = 20$) is constructed by sampling an arm with positive reward and nineteen arms with negative reward from the arm pool, and the concatenation of user and restaurant/movie feature vector is used as the context vector for the arm (thus $d = 20$).

Discussions OneKernelUCB and NKernelUCB correspond to the two extreme cases where the clients either communicate in every time step to learn a shared model, or they learn their own models independently with no communication. As shown in Figure 2.13, OneKernelUCB achieved the smallest cumulative regret in almost all experiments, but also incurred the highest communication cost, i.e., $O(TNd)$ due to sending each new data point to all clients in every round, which demonstrates the necessity of communication efficient bandit algorithms. On the other hand, distributed linear bandit algorithms, e.g., DisLinUCB and FedLinUCB, incurred very low communication cost as they directly communicate via the $d \times d$ statistics, but fail to capture the complicated reward mappings in most of these datasets, e.g., in Figure 2.13(d), they even had much worse regret than NKernelUCB that requries no communication. Equipped with logistic function, distributed GLB algorithm FedGLBUCB attained both low regret and low communication cost on the two classification datasets, i.e., Figure 2.13(c) and Figure 2.13(d), but required many iterations of distributed gradient updates to converge on the other four datasets where logistic function may not fit, and led to huge communication costs. In comparison, Approx-DisKernelUCB and our proposed Async-KernelUCB had consistently smaller regret than their linear counterparts, while requiring relatively lower communication cost for joint kernel estimation. It is also worth noting that despite having the same $\tilde{O}(N^2\gamma_T^3)$ theoretical scaling in communication cost, Async-KernelUCB incurs much smaller communication cost empirically, while having comparable or even better regret than Approx-DisKernelUCB.



(a) $f_1(\mathbf{x})$

(b) $f_2(\mathbf{x})$

(c) MagicTelescope

(d) Mushroom

(e) MovieLens

(f) Yelp

Figure 2.13: Experiment results on synthetic and real-world datasets.

2.3.12 Full proof of Async-KernelUCB algorithm

Omitted proof in Section 2.3.9

Let's define the unfavorable event $H_k = A_k \cup E_k$, where A_k is the event that the dictionary $\mathcal{S}_k$ is not ϵ-accurate w.r.t. $\mathcal{D}_k$, and E_k is the event that the size of dictionary $|\mathcal{S}_k|$ is large, i.e., $|\mathcal{S}_k| > 12\beta(1+\beta D)\bar{q}\gamma_T$. Therefore, we want to bound the probability of $\cup_{k=0}^{B} H_k$, which can be decomposed as

$$\begin{aligned}
\mathbb{P}\big(\cup_{k=0}^{B} H_k\big) &= \mathbb{P}\big(\cup_{k=0}^{B} (A_k \cup E_k)\big) = \mathbb{P}\big((\cup_{k=0}^{B} A_k) \cup (\cup_{k=0}^{B} E_k)\big) \\
&= \mathbb{P}\big(\cup_{k=0}^{B} A_k\big) + \mathbb{P}\big(\cup_{k=0}^{B} E_k\big) - \mathbb{P}\big((\cup_{k=0}^{B} A_k) \cap (\cup_{k=0}^{B} E_k)\big) \\
&= \mathbb{P}\big(\cup_{k=0}^{B} A_k\big) + \mathbb{P}\big((\cup_{k=0}^{B} E_k) \cap (\cup_{k=0}^{B} A_k)^{C}\big)
\end{aligned}$$

Note that, as in [92], we bound the second term as $\mathbb{P}\big((\cup_{k=0}^{B} E_k) \cap (\cup_{k=0}^{B} A_k)^{C}\big) = \mathbb{P}\big((\cup_{k=0}^{B} E_k) \cap (\cap_{k=0}^{B} A_k^{C})\big) = \mathbb{P}\big(\cup_{k=0}^{B} [E_k \cap (\cap_{k=0}^{B} A_k^{C})]\big) \leq \sum_{k=0}^{B} \mathbb{P}\big(E_k \cap (\cap_{k=0}^{B} A_k^{C})\big)$. For the first term $\mathbb{P}\big(\cup_{k=0}^{B} A_k\big)$, we need a decomposition different from prior works [92, 73], since our dictionary is *incrementally updated with a batch of samples at each communication round* (line 9 in Algorithm 14). Specifically, when bounding the probability of having an inaccurate dictionary at the k-th communication, i.e., event A_k, we need to condition on the event that dictionaries at all previous communications are ϵ-accurate, i.e., event $\cap_{k'=0}^{k-1} A_{k'}^{C}$. Hence, we decompose $\mathbb{P}\big(\cup_{k=0}^{B} A_k\big) = 1 - \mathbb{P}\big(\cap_{k=0}^{B} A_k^{C}\big) = 1 - \mathbb{P}(A_0^{C}) \prod_{k=1}^{B} \mathbb{P}\big(A_k^{C} | \cap_{k'=0}^{k-1} A_{k'}^{C}\big) = 1 - \prod_{k=1}^{B} [1 - \mathbb{P}\big(A_k | \cap_{k'=0}^{k-1} A_{k'}^{C}\big)] \leq \sum_{k=1}^{B} \mathbb{P}\big(A_k | \cap_{k'=0}^{k-1} A_{k'}^{C}\big)$, where the second equality is due to Bayes theorem, the third equality is because $\mathcal{D}_0 = \emptyset$ is well-approximated by $\mathcal{S}_0 = \emptyset$, and thus $\mathbb{P}\big(A_0^C\big) = 1$, and the inequality is due to Weierstrass product inequality. Putting everything together, we have

$$\mathbb{P}\big(\cup_{k=0}^{B} H_k\big) \le \sum_{k=1}^{B} \mathbb{P}\big(A_k | \cap_{k'}^{k-1} A_{k'}^C\big) + \sum_{k=1}^{B} \mathbb{P}\big(E_k \cap (\cap_{k=0}^{B} A_k^C)\big) \tag{2.40}$$

Then we can upper bound these two terms using Lemma 2.3.5 and Lemma 2.3.6 given in Section 2.3.9, which leads to $\mathbb{P}\big(\cup_{k=0}^{B} H_k\big) \le \delta$, for $\delta \in (0, 1)$, and thus finishes the proof of Lemma 2.3.4.

Proof of Lemma 2.3.5: bounding $\sum_{k=1}^{B} \mathbb{P}\big(A_k | \cap_{k'}^{k-1} A_{k'}^C\big)$. As [73], we can rewrite the event A_k, based on the definition of ϵ-accuracy given in (2.39), as

$$A_k = \{\| \sum_{s \in \mathcal{D}_k} G_{k,s} \| > \epsilon\}$$

where $G_{k,s} = (\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top$ and $\psi_{k,s} = (\mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k} + \lambda \mathbf{I})^{-1/2}\phi(\mathbf{x}_s)$. Then let's define $\mathcal{F}_k := \{q_{k,s}, \eta_s\}_{s \in \mathcal{D}_k}$ for $k \in [B]$, which contains all randomness in the construction of $\mathcal{S}_k$ during the k-th communication. With conditioning, we have

$$\begin{aligned}
&\mathbb{P}(A_k \mid \cap_{k'}^{k-1} A_{k'}^C) = \mathbb{P}\big(\| \sum_{s \in \mathcal{D}_k} G_{k,s} \| > \epsilon \mid \cap_{k'}^{k-1} A_{k'}^C\big) = \mathbb{E}_{\mathcal{F}_k}\big[\mathbf{1}\{\| \sum_{s \in \mathcal{D}_k} G_{k,s} \| > \epsilon\} \mid \cap_{k'}^{k-1} A_{k'}^C\big] \\
&= \mathbb{E}_{\mathcal{F}_{k-1}}\big[\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}\big[\mathbf{1}\{\| \sum_{s \in \mathcal{D}_k} G_{k,s} \| > \epsilon\} | \mathcal{F}_{k-1}\big] \mid \cap_{k'}^{k-1} A_{k'}^C\big] \\
&= \mathbb{E}_{\mathcal{F}_{k-1} : \cap_{k'}^{k-1} A_{k'}^C}\big[\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}\big[\mathbf{1}\{\| \sum_{s \in \mathcal{D}_k} G_{k,s} \| > \epsilon\} | \mathcal{F}_{k-1}\big]\big] \\
&= \mathbb{E}_{\mathcal{F}_{k-1} : \cap_{k'}^{k-1} A_{k'}^C}\big[\mathbb{P}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}\big(\| \sum_{s \in \mathcal{D}_k} (\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top \| > \epsilon \mid \mathcal{F}_{k-1}\big)\big].
\end{aligned}$$

where the third equality holds because when conditioned on the event $\cap_{k'}^{k-1} A_{k'}^C$, the outcomes associated with the complement of this event have zero probability, and thus we can restrict the expectation to the outcomes where the event $\cap_{k'}^{k-1} A_{k'}^C$ holds.

Consider the k-th communication for $k \in [B]$. We denote the client who triggers the k-th communication as $c_k \in [N]$, and the time step when the k-th communication happens as $t_k \in [T]$. In addition, recall that we denote the sequence of time steps in-between client c_k's last communication (whose index is denoted as $\underline{k}(c_k) \in [0, k-1]$) and the current (the k-th) communication when client c_k's is active as $\Delta\mathcal{D}_k := \mathcal{N}_{t_k}(c_k) \setminus \mathcal{N}_{t_{\underline{k}(c_k)}}(c_k) = \{t_{\underline{k}(c_k)} < s \le t_k : i_s = c_k\}$.

Note that due to our incremental update procedure, for some data point with time index s, that was added into $\mathcal{D}_k$ during the k'-th communication (sent to the server in the form of embedded statistics), i.e., $s \in \Delta\mathcal{D}_{k'}$, for $k' = 1, \dots, k$, we have $q_{k,s} = q_{k',s}$ and $\tilde{p}_{k,s} = \tilde{p}_{k',s}$. When conditioned on $\mathcal{F}_{k-1}$, $q_{k,s}$ for all $s \in \mathcal{D}_k$ are independent Bernoulli random variable with mean $\tilde{p}_{k,s}$, because they only correlate via the approximated variance function(s) that were used for arm selection and RLS sampling up to the k-th communication, which are deterministic conditioned on $\mathcal{F}_{k-1}$, and thus both $\tilde{p}_{k,s}$ and $\psi_{k,s}$ are deterministic as well.

Therefore, we can bound $\mathbb{P}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}\big(\|\sum_{s \in \mathcal{D}_k} (\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top\| > \epsilon | \mathcal{F}_{k-1}\big)$ using Lemma A.13. First, we need to show that each term in the summation has zero mean and bounded norm, i.e., $\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}[G_{k,s} | \mathcal{F}_{k-1}] = 0$ and $\|G_{k,s}\| \le R$ for some constant R:

$$\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}\big[(\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top | \mathcal{F}_{k-1}\big] = (\frac{\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}}[q_{k,s} | \mathcal{F}_{k-1}]}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top = 0,$$

and

$$\|G_{k,s}\| = \|(\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\psi_{k,s}\psi_{k,s}^\top\| \le (\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)\|\psi_{k,s}\psi_{k,s}^\top\| \le \frac{\sigma_k^2(\mathbf{x}_s)}{\tilde{p}_{k,s}},$$

where the last inequality is because $q_{k,s} \le 1$ and $\|\psi_{k,s}\psi_{k,s}^\top\| = \psi_{k,s}^\top\psi_{k,s} = \sigma_k^2(\mathbf{x}_s)$. As mentioned earlier, for $s \in \Delta\mathcal{D}_{k'}$, $k' = 1, \dots, k$, we have $\tilde{p}_{k,s} = \tilde{p}_{k',s} = \bar{q}\tilde{\sigma}_{k'-1}^2(\mathbf{x}_s)$, i.e., during the k'-th communication, client $c_{k'}$ first receives server's latest statistics to compute $\tilde{\sigma}_{k'-1}^2(\cdot)$ for RLS sampling. Conditioned on $\cap_{k'=0}^{k} A_{k'}^C$ and by Lemma A.11, we have $\tilde{\sigma}_{k'-1}^2(\mathbf{x}_s) \ge \sigma_{k'-1}^2(\mathbf{x}_s)/\beta$, where $\beta := (1+\epsilon)/(1-\epsilon)$. Hence,

$$\|G_{k,s}\| \le \frac{\sigma_k^2(\mathbf{x}_s)}{\tilde{p}_{k,s}} = \frac{\sigma_k^2(\mathbf{x}_s)}{\bar{q}\tilde{\sigma}_{k'-1}^2(\mathbf{x}_s)} \le \frac{\beta}{\bar{q}}\frac{\sigma_k^2(\mathbf{x}_s)}{\sigma_{k'-1}^2(\mathbf{x}_s)} \le \frac{\beta}{\bar{q}} := R.$$

where the last inequality is because the variance is non-increasing over time. Then by Lemma A.13,

$$\mathbb{P}_{\mathcal{F}_k \backslash \mathcal{F}_{k-1}}\big(\|\sum_{s\in\mathcal{D}_k} G_{k,s}\| > \epsilon|\mathcal{F}_{k-1}\big) \le 4|\mathcal{D}_k| \exp\big(-\frac{\epsilon^2/2}{\|\sum_{s\in\mathcal{D}_k}\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[G_{k,s}^2|\mathcal{F}_{k-1}]\| + R\epsilon/3}\big)$$

Now we need to further upper bound the term $\|\sum_{s\in\mathcal{D}_k}\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[G_{k,s}^2|\mathcal{F}_{k-1}]\|$. First, note that

$$\begin{aligned}\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[G_{k,s}^2|\mathcal{F}_{k-1}] &= \mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}\big[(\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)^2\psi_{k,s}\psi_{k,s}^\top\psi_{k,s}\psi_{k,s}^\top|\mathcal{F}_{k-1}\big] \\ &= \mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}\big[(\frac{q_{k,s}}{\tilde{p}_{k,s}} - 1)^2|\mathcal{F}_{k-1}\big]\psi_{k,s}\psi_{k,s}^\top\psi_{k,s}\psi_{k,s}^\top,\end{aligned}$$

and $\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[(\frac{q_{k,s}}{\tilde{p}_{k,s}}-1)^2|\mathcal{F}_{k-1}] = \mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[(\frac{q_{k,s}}{\tilde{p}_{k,s}})^2|\mathcal{F}_{k-1}] - 2\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[\frac{q_{k,s}}{\tilde{p}_{k,s}}|\mathcal{F}_{k-1}] + 1 = \mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[\frac{q_{k,s}}{\tilde{p}_{k,s}^2}|\mathcal{F}_{k-1}] - 1 = \frac{1}{\tilde{p}_{k,s}} - 1 \le \frac{1}{\tilde{p}_{k,s}}$. Substituting this to the RHS, we have

$$\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[G_{k,s}^2|\mathcal{F}_{k-1}] \preceq \frac{1}{\tilde{p}_{k,s}}\psi_{k,s}\psi_{k,s}^\top\psi_{k,s}\psi_{k,s}^\top \preceq \frac{1}{\tilde{p}_{k,s}}\|\psi_{k,s}\psi_{k,s}^\top\|\psi_{k,s}\psi_{k,s}^\top \preceq R\psi_{k,s}\psi_{k,s}^\top,$$

and thus,

$$\begin{aligned}&\|\sum_{s\in\mathcal{D}_k}\mathbb{E}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}[G_{k,s}^2|\mathcal{F}_{k-1}]\| \le R\|\sum_{s\in\mathcal{D}_k}\psi_{k,s}\psi_{k,s}^\top\| \\ &= R\|\sum_{s\in\mathcal{D}_k}(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I})^{-1/2}\phi_s\phi_s^\top(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I})^{-1/2}\| \\ &= R\|(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I})^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k} + \lambda\mathbf{I})^{-1/2}\| \le R,\end{aligned}$$

where the first equality is by definition of $\psi_{k,s}$. Putting everything together, we have

$$\mathbb{P}_{\mathcal{F}_k\backslash\mathcal{F}_{k-1}}\big(\|\sum_{s\in\mathcal{D}_k}G_{k,s}\| > \epsilon|\mathcal{F}_{k-1}\big) \le 4|\mathcal{D}_k|\exp\big(-\frac{\epsilon^2/2}{1+\epsilon/3}\cdot\frac{\bar{q}}{\beta}\big),$$

and thus $\mathbb{P}(A_k \mid \cap_{k'=0}^{k-1}A_{k'}^C) \le 4|\mathcal{D}_k|\exp(-\frac{\epsilon^2/2}{1+\epsilon/3}\cdot\frac{\bar{q}}{\beta})$. Summing over B terms, we have

$$\sum_{k=0}^{B}\mathbb{P}\big(A_k|\cap_{k'=0}^{k-1}A_{k'}^C\big) \le 4\exp\big(-\frac{\epsilon^2/2}{1+\epsilon/3}\cdot\frac{\bar{q}}{\beta}\big)\sum_{k=1}^{B}|\mathcal{D}_k| \le 4T^2\exp\big(-\frac{\epsilon^2/2}{1+\epsilon/3}\cdot\frac{\bar{q}}{\beta}\big)$$

In order to make sure $\sum_{k=0}^{B}\mathbb{P}\big(A_k|\cap_{k'=0}^{k-1}A_{k'}^C\big) \le \frac{\delta}{2}$, we need to set $\bar{q} = 4\beta\frac{1+\epsilon/3}{\epsilon^2}\ln(\frac{2\sqrt{2}T}{\delta})$.

□

Proof of Lemma 2.3.6: bounding $\sum_{k=0}^{B}\mathbb{P}\big(E_k\cap(\cap_{k=0}^{B}A_k^C)\big)$. First, note that $\mathbb{P}(E_0\cap(\cap_{k=0}^{B}A_k^C)) = 0$, because $\mathcal{S}_0 = \emptyset$, and by definition of $q_{k,s}$ for $s \in \mathcal{D}_k$, the size of dictionary $|\mathcal{S}_k| = \sum_{s\in\mathcal{D}_k}q_{k,s}$. We formally define unfavorable event E_k as

$$E_k = \big\{ \sum_{s \in \mathcal{D}_k} q_{k,s} > 12\beta(1+\beta D)\bar{q}\gamma_T \big\},$$

where $\beta = (1+\epsilon)/(1-\epsilon)$. Similar to [92, 73], we will use a stochastic dominance argument to upper bound the probability of event E_k. First, we use conditioning again to rewrite $\mathbb{P}\big(E_k \cap (\cap_{k=1}^B A_k^C)\big)$ as

$$\begin{aligned}
&\mathbb{P}(E_k \cap (\cap_{k=1}^B A_k^C)) = \mathbb{P}(E_k \mid \cap_{k=1}^B A_k^C)\mathbb{P}(\cap_{k=1}^B A_k^C) \le \mathbb{P}(E_k \mid \cap_{k=1}^B A_k^C) \\
&= \mathbb{P}\big(\sum_{s \in \mathcal{D}_k} q_{k,s} \ge 12\beta(1+\beta D)\bar{q}\gamma_T \mid \cap_{k=1}^B A_k^C \big) \\
&= \mathbb{E}_{\mathcal{F}_{k-1} : \cap_{k=1}^B A_k^C} \big[\mathbb{P}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}} \big(\sum_{s \in \mathcal{D}_k} q_{k,s} \ge 12\beta(1+\beta D)\bar{q}\gamma_T \mid \mathcal{F}_{k-1} \big) \big].
\end{aligned}$$

As discussed earlier, when conditioned on $\mathcal{F}_{k-1}$, $q_{k,s}$ for $s \in \mathcal{D}_k$ becomes independent Bernoulli random variable, with mean $\tilde{p}_{k,s}$. In addition, as a result of our incremental dictionary update (line 9 in Algorithm 14), the partition in $\mathcal{D}_k$ that were added during the k'-th communication for $k' \in 1, \dots, k$, which is denoted by $\Delta\mathcal{D}_{k'}$, is sampled using $\bar{q}\tilde{\sigma}^2_{k'-1}(\mathbf{x}_s)$ for $s \in \Delta\mathcal{D}_{k'}$. Hence,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}_k \setminus \mathcal{F}_{k-1}} \big[\sum_{s \in \mathcal{D}_k} q_{k,s} | \mathcal{F}_{k-1} \big] = \sum_{s \in \mathcal{D}_k} \tilde{p}_{k,s} \\
&= \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \tilde{p}_{k',s} = \bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \tilde{\sigma}^2_{k'-1}(\mathbf{x}_s) \\
&\le \beta\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1}(\mathbf{x}_s) = \beta\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1,s-1}(\mathbf{x}_s) \cdot \frac{\sigma^2_{k'-1}(\mathbf{x}_s)}{\sigma^2_{k'-1,s-1}(\mathbf{x}_s)} \\
&\le \beta\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1,s-1}(\mathbf{x}_s) \cdot [1 + \sum_{s' \in \Delta\mathcal{D}_{k'} : s' \le s-1} \sigma^2_{k'-1}(\mathbf{x}_{s'})] \\
&\le \beta\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1,s-1}(\mathbf{x}_s) \cdot [1 + \sum_{s' \in \Delta\mathcal{D}_{k'} : s' \le s-1} \sigma^2_{\underline{k'}(c_{k'})}(\mathbf{x}_{s'})] \\
&\le \beta\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1,s-1}(\mathbf{x}_s) \cdot [1 + \beta \sum_{s' \in \Delta\mathcal{D}_{k'} : s' \le s-1} \tilde{\sigma}^2_{\underline{k'}(c_{k'})}(\mathbf{x}_{s'})] \\
&\le \beta(1+\beta D)\bar{q} \sum_{k'=1}^{k} \sum_{s \in \Delta\mathcal{D}_{k'}} \sigma^2_{k'-1,s-1}(\mathbf{x}_s)
\end{aligned}$$

where the imaginary variance function $\sigma^2_{k'-1,s-1}(\cdot)$ is constructed using dataset $\big(\cup_{k=1}^{k'-1} \Delta\mathcal{D}_k\big) \cup \{s' \in \Delta\mathcal{D}_{k'} : s' \le s-1\}$ (not computed in the actual algorithm); the first and forth inequality is due to Lemma A.11 as we conditioned on $\cap_{k=0}^B A_k^C$; the second is due to Lemma A.10; the third is because $\underline{k'}(c_{k'}) \le k'-1$ and the variance is non-increasing over time; and the fifth is due to our event-trigger design in (2.7), i.e., $\sum_{s \in \Delta\mathcal{D}_{k'} : s \le t_{k'}-1} \tilde{\sigma}^2_{\underline{k'}(c_{k'})}(\mathbf{x}_s) < D$.

Now for each term in the summation on the RHS of the inequality above, we introduce an independent Bernoulli random variable $\hat{q}_{k,s} \sim \mathcal{B}\big(\beta(1+\beta D)\bar{q}\sigma^2_{k'-1,s-1}(\mathbf{x}_s)\big)$. Since $\hat{q}_{k,s}$ stochastically dominates $q_{k,s}$, i.e., $\mathbb{E}\big[q_{k,s} \mid \mathcal{F}_{k-1}\big] = \tilde{p}_{k,s} \le \beta(1+\beta D)\bar{q}\sigma^2_{k'-1,s-1}(\mathbf{x}_s) = \mathbb{E}\big[\hat{q}_{k,s}\big]$, we have

$$\mathbb{P}\big(\sum_{s \in \mathcal{D}_k} q_{k,s} > 12\beta(1+\beta D)\bar{q}\gamma_T \mid \mathcal{F}_{k-1} \big) \le \mathbb{P}\big(\sum_{s \in \mathcal{D}_k} \hat{q}_{k,s} > 12\beta(1+\beta D)\bar{q}\gamma_T \big).$$

Then we can further upper bound the RHS

$$\begin{aligned}
&\mathbb{P}\big(\sum_{s\in\mathcal{D}_k}\hat{q}_{k,s} > 12\beta(1+\beta D)\bar{q}\gamma_T\big)\\
&\le \mathbb{P}\big(\sum_{s\in\mathcal{D}_k}\hat{q}_{k,s} > 3\beta(1+\beta D)\bar{q}\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s)\big)\\
&\le \exp\big(-2\beta(1+\beta D)\bar{q}\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s)\big)
\end{aligned}$$

where the first inequality is because $\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s) \le 4\gamma_T$, and the second inequality is due to Lemma A.14. By substituting $\bar{q} = 4\beta\frac{1+\epsilon/3}{\epsilon^2}\ln(\frac{2\sqrt{2}T}{\delta})$ and under the condition that $\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s) \ge 1$, we have $\exp\big(-2\beta(1+\beta D)\bar{q}\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s)\big) \le \exp\big(-\ln(8T^2/\delta)\big)$. To ensure $\sum_{k'=1}^{k}\sum_{s\in\Delta\mathcal{D}_{k'}}\sigma^2_{k'-1,s-1}(\mathbf{x}_s) \ge 1$, we can set $\lambda \le k(\mathbf{x},\mathbf{x}), \forall \mathbf{x}\in\mathcal{A}$. Finally, by summing over B terms, we have

$$\sum_{k=0}^{B}\mathbb{P}\big(E_k\cap(\cap_{k=0}^{B}A_k^C)\big) \le T\exp\big(-\ln(8T^2/\delta)\big) \le T\cdot\frac{\delta}{8T^2} < \frac{\delta}{2}$$

where the last inequality is because $T \ge 1$.

□

Proof of Lemma 2.3.7 in Section 2.3.10

Recall from Section 2.3.7 that the approximated kernel Ridge regression estimator for $\theta_\star$ is defined as

$$\tilde{\theta}_k = \tilde{\mathbf{V}}_k^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{y}_{\mathcal{D}_k}$$

where $\tilde{\mathbf{V}}_k := \mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\mathbf{P}_{\mathcal{S}_k} + \lambda\mathbf{I}$. Then we can decompose

$$\begin{aligned}
&\|\tilde{\theta}_k-\theta_\star\|^2_{\tilde{\mathbf{V}}_k} = (\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k(\tilde{\theta}_k-\theta_\star)\\
=&(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k(\tilde{\mathbf{V}}_k^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{y}_{\mathcal{D}_k}-\theta_\star)\\
=&(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k[\tilde{\mathbf{V}}_k^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top(\mathbf{\Phi}_{\mathcal{D}_k}\theta_\star+\eta_{\mathcal{D}_k})-\theta_\star]\\
=&\underbrace{(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k(\tilde{\mathbf{V}}_k^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\theta_\star-\theta_\star)}_{A_1}+\underbrace{(\tilde{\theta}_k-\theta_\star)^\top\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}}_{A_2}
\end{aligned}$$

Since $\tilde{\mathbf{V}}_k(\tilde{\mathbf{V}}_k^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\theta_\star-\theta_\star) = \mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\theta_\star-\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\mathbf{P}_{\mathcal{S}_k}\theta_\star-\lambda\theta_\star = \mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\theta_\star-\lambda\theta_\star$, we have

$$\begin{aligned}
A_1 =&(\tilde{\theta}_k-\theta_\star)^\top\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\theta_\star-\lambda(\tilde{\theta}_k-\theta_\star)^\top\theta_\star\\
=&(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k^{1/2}\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\theta_\star-\lambda(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k^{1/2}\tilde{\mathbf{V}}_k^{-1/2}\theta_\star\\
\le&\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\big(\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\theta_\star\|+\lambda\|\theta_\star\|_{\tilde{\mathbf{V}}_k^{-1}}\big)\\
\le&\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\big(\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\|\|\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\|\|\theta_\star\|+\sqrt{\lambda}\|\theta_\star\|\big)\\
\le&\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\big(\|\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\|+\sqrt{\lambda}\big)\|\theta_\star\|
\end{aligned}$$

where the first inequality is due to Cauchy Schwartz, and the last inequality is because $\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\| = \sqrt{\mathbf{\Phi}_{\mathcal{D}_k}\mathbf{P}_{\mathcal{S}_k}(\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}\mathbf{P}_{\mathcal{S}_k}+\lambda\mathbf{I})^{-1}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top} \le 1$. Then by definition of the spectral norm $\|\cdot\|$, and the properties of the orthogonal projection matrix $\mathbf{P}_{\mathcal{S}_k}$, we have

$$\|\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\| = \sqrt{\lambda_{\max}\big(\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})^2\mathbf{\Phi}_{\mathcal{D}_k}^\top\big)} = \sqrt{\lambda_{\max}\big(\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\mathbf{\Phi}_{\mathcal{D}_k}^\top\big)}.$$

Moreover, due to Lemma 2.3.4, $\mathcal{S}_k$ is ϵ-accurate w.r.t. $\mathcal{D}_k$, for all k, so we have $\mathbf{I}-\mathbf{P}_{\mathcal{S}_k} \preceq \frac{\lambda}{1-\epsilon}(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}+\lambda\mathbf{I})^{-1}$ by the property of ϵ-accuracy (Proposition 10 of [73]). Substituting this to RHS of the equality above, we have

$$\|\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{I}-\mathbf{P}_{\mathcal{S}_k})\| \leq \sqrt{\frac{\lambda}{1-\epsilon}\lambda_{\max}\big(\mathbf{\Phi}_{\mathcal{D}_k}(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}+\lambda\mathbf{I})^{-1}\mathbf{\Phi}_{\mathcal{D}_k}^\top\big)} \leq \sqrt{\frac{\lambda}{1-\epsilon}}.$$

Therefore, $A_1 \leq \|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\big(\sqrt{\frac{1}{1-\epsilon}}+1\big)\sqrt{\lambda}\|\theta_\star\|$.

Similarly, by applying Cauchy-Schwartz inequality on term A_2, we have

$$\begin{aligned}A_2 =&(\tilde{\theta}_k-\theta_\star)^\top\tilde{\mathbf{V}}_k^{1/2}\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k} \leq \|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\| \\ =&\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{V}_k^{1/2}\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\| \\ \leq&\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{V}_k^{1/2}\|\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\|\end{aligned}$$

where $\mathbf{V}_k := \mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}+\lambda\mathbf{I}$. Note that $\mathbf{P}_{\mathcal{S}_k}\mathbf{V}_{k_k}\mathbf{P}_{\mathcal{S}_k} = \mathbf{P}_{\mathcal{S}_k}(\mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k}+\lambda\mathbf{I})\mathbf{P}_{\mathcal{S}_k} = \tilde{\mathbf{V}}_k+\lambda(\mathbf{P}_{\mathcal{S}_k}-\mathbf{I})$ and $\mathbf{P}_{\mathcal{S}_k} \preceq \mathbf{I}$, so we have

$$\begin{aligned}\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{V}_k^{1/2}\| =& \sqrt{\|\tilde{\mathbf{V}}_k^{-1/2}\mathbf{P}_{\mathcal{S}_k}\mathbf{V}_k^{1/2}\mathbf{V}_k^{1/2}\mathbf{P}_{\mathcal{S}_k}\tilde{\mathbf{V}}_k^{-1/2}\|} \leq \sqrt{\|\tilde{\mathbf{V}}_k^{-1/2}(\tilde{\mathbf{V}}_k+\lambda(\mathbf{P}_{\mathcal{S}_k}-\mathbf{I}))\tilde{\mathbf{V}}_k^{-1/2}\|} \\ =& \sqrt{\|\mathbf{I}+\lambda\tilde{\mathbf{V}}_k^{-1/2}(\mathbf{P}_{\mathcal{S}_k}-\mathbf{I}))\tilde{\mathbf{V}}_k^{-1/2}\|} \leq \sqrt{1+\lambda\|\tilde{\mathbf{V}}_k^{-1}\|\|\mathbf{P}_{\mathcal{S}_k}-\mathbf{I}\|} \\ \leq& \sqrt{1+\lambda\cdot\lambda^{-1}\cdot 1} = \sqrt{2},\end{aligned}$$

and thus $A_2 \leq \sqrt{2}\|\tilde{\theta}_k-\theta_\star\|_{\tilde{\mathbf{V}}_k}\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\|$.

As mentioned by [97], the standard self-normalized bound for vector-valued martingales cannot be directly applied to bound the term $\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\|$, since $\mathcal{D}_k$ is constructed by the data that each client has uploaded so far during the event-triggered communications. Therefore, in the following paragraphs, we bound this term by extending their results to the kernel bandit problem considered in our work.

We first need to establish the following lemma.

Lemma 2.3.9. *Let's denote $\mathbf{V}_k(i) = \sum_{s\in\mathcal{N}_{t_{\underline{k}(i)}}(i)}\phi(\mathbf{x}_s)\phi(\mathbf{x}_s)^\top$, such that $\mathbf{V}_k = \lambda\mathbf{I}+\sum_{i=1}^N\mathbf{V}_k(i)$, and then denote the covariance matrix for client i's data that hasn't been uploaded to server by time step t_k as $\Delta\mathbf{V}_k(i) = \sum_{s\in\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\phi(\mathbf{x}_s)\phi(\mathbf{x}_s)^\top$ for $i\in[N]$. Then we have*

$$\mathbf{V}_k \succeq \frac{1}{\beta D}\Delta\mathbf{V}_k(i), \tag{2.41}$$

and $\forall\mathbf{x}\in\mathbb{R}^d$,

$$\frac{\phi(\mathbf{x})^\top\mathbf{V}_k^{-1}\phi(\mathbf{x})}{\phi(\mathbf{x})^\top(\mathbf{\Phi}_{[t_k]}^\top\mathbf{\Phi}_{[t_k]}+\lambda\mathbf{I})^{-1}\phi(\mathbf{x})} \leq 1+N\beta D.$$

Bounding $\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top\eta_{\mathcal{D}_k}\|$: Recall that $\mathcal{D}_k$ contains data points that N clients have uploaded up to the k-th communication, i.e., $\mathcal{D}_k = \cup_{i=1}^N\mathcal{N}_{t_{\underline{k}(i)}}(i)$, where $t_{\underline{k}(i)}$ denotes the time step of client i's last communication with the server. Therefore, we have the following decomposition

$$
\begin{aligned}
\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top \eta_{\mathcal{D}_k} &= \sum_{i=1}^N \mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_{\underline{k}(i)}}(i)} \\
&= \sum_{i=1}^N \mathbf{V}_k^{-1/2}\big[\mathbf{\Phi}_{\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_{\underline{k}(i)}}(i)} + \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\big] \\
&\quad - \sum_{i=1}^N \mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)} \\
&= \mathbf{V}_k^{-1/2}\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]} - \sum_{i=1}^N \mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}.
\end{aligned}
$$

Then using triangle inequality, we have

$$
\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{D}_k}^\top \eta_{\mathcal{D}_k}\| \leq \|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]}\| + \sum_{i=1}^N \|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\|.
$$

We can bound $\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]}\|$ as

$$
\begin{aligned}
&\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]}\| = \|\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]}\|_{\mathbf{V}_k^{-1}} \leq \|\mathbf{\Phi}_{[t_k]}^\top \eta_{[t_k]}\|_{(\mathbf{\Phi}_{[t_k]}^\top \mathbf{\Phi}_{[t_k]}+\lambda\mathbf{I})^{-1}} \sqrt{1+ND\beta} \\
&\leq \sqrt{1+ND\beta} R \sqrt{2\ln(1/\delta) + \ln(\det(\mathbf{K}_{[T],[T]}/\lambda + \mathbf{I}))},
\end{aligned}
$$

with probability at least $1-\delta$, where the first inequality is due to Lemma 2.3.9, and the second inequality is due to the standard self-normalized bound for kernelized contextual bandit, e.g., Lemma B.3. of [100].

Then we can bound $\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\|$ as

$$
\begin{aligned}
&\|\mathbf{V}_k^{-1/2}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\| \\
&\leq \sqrt{2D\beta}\|\big[D\beta\lambda\mathbf{I} + \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\big]^{-1/2} \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\| \\
&= \sqrt{2D\beta}\|\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \eta_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\|_{\big[D\beta\lambda\mathbf{I}+\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\big]^{-1}} \\
&\leq \sqrt{2D\beta} R \sqrt{2\ln(1/\delta) + \ln(\det(\mathbf{K}_{[T],[T]}/(D\beta\lambda) + \mathbf{I}))}
\end{aligned}
$$

where the first inequality is because $\mathbf{V}_k = \lambda\mathbf{I} + \mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k} \succeq \frac{1}{D\beta}\mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}$ due to (2.41) in Lemma 2.3.9, so $\mathbf{V}_k = \lambda\mathbf{I} + \mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k} \succeq \frac{1}{2D\beta}(D\beta\lambda\mathbf{I} + \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}^\top \mathbf{\Phi}_{\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)})$, and the second inequality is again obtained using the standard self-normalized bound.

Putting everything together, we have

$$
\|\tilde{\theta}_k - \theta_\star\|_{\tilde{\mathbf{V}}_k} \leq (\sqrt{1/(1-\epsilon)} + 1)\sqrt{\lambda}\|\theta_\star\| + 2\big(\sqrt{1+ND\beta} + N\sqrt{2D\beta}\big) R \sqrt{\ln(1/\delta) + \gamma_T},
$$

where $\gamma_T := \max_{\mathcal{D}\subset\mathcal{A}:|\mathcal{D}|=T} \frac{1}{2}\log\det(\mathbf{K}_{\mathcal{D},\mathcal{D}}/(D\beta\lambda) + \mathbf{I})$.

Proof of Lemma 2.3.9. Note that by definition, $\Delta\mathbf{V}_k(c_k) = \mathbf{0}$, where $c_k \in [N]$ is the index of the client who triggers the k-th communication. In the following, we first show that

$$
\mathbf{V}_k \succeq \frac{1}{\beta D}\Delta\mathbf{V}_k(i)
$$

for all $i \in [N]$. For client c_k, $\mathbf{V}_k \succeq \mathbf{0} = \frac{1}{\beta D}\Delta\mathbf{V}_k(c_k)$. For client $i \neq c_k$, we have

$$\begin{aligned}
&\frac{\phi(\mathbf{x})^\top \mathbf{V}_{\underline{k}(i)}^{-1}\phi(\mathbf{x})}{\phi(\mathbf{x})^\top\big(\mathbf{V}_{\underline{k}(i)}+\Delta\mathbf{V}_k(i)\big)^{-1}\phi(\mathbf{x})}\\
&\leq 1+\sum_{s\in\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\phi(\mathbf{x}_s)^\top\mathbf{V}_{\underline{k}(i)}^{-1}\phi(\mathbf{x}_s) = 1+\sum_{s\in\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\sigma_{\underline{k}(i)}^2(\mathbf{x}_s)\\
&\leq 1+\beta\sum_{s\in\mathcal{N}_{t_k}(i)\setminus\mathcal{N}_{t_{\underline{k}(i)}}(i)}\tilde{\sigma}_{\underline{k}(i)}^2(\mathbf{x}_s)\leq 1+\beta D,
\end{aligned}$$

where the first inequality is due to Lemma A.10, the second is due to property of ϵ-accuracy in Lemma A.11, and the third is due to our event-trigger in (2.7).

This implies $\mathbf{V}_{\underline{k}(i)}^{-1} \preceq (1+\beta D)\big(\mathbf{V}_{\underline{k}(i)}+\Delta\mathbf{V}_k(i)\big)^{-1}$. Then due to Lemma A.5, we have $(1+\beta D)\mathbf{V}_{\underline{k}(i)} \succeq \mathbf{V}_{\underline{k}(i)} + \Delta\mathbf{V}_k(i)$, and thus $\mathbf{V}_{\underline{k}(i)} \succeq \frac{1}{\beta D}\Delta\mathbf{V}_k(i)$. In addition, since $\underline{k}(i) < k, \forall i \neq c_k$, we have $\mathbf{V}_k \succeq \mathbf{V}_{\underline{k}(i)} \succeq \frac{1}{\beta D}\Delta\mathbf{V}_k(i)$.

By averaging (2.41) over all N clients, we have

$$\mathbf{V}_k \succeq \frac{1}{N\beta D}\sum_{i=1}^N \Delta\mathbf{V}_k(i),$$

and thus, we have

$$\mathbf{\Phi}_{[t_k]}^\top\mathbf{\Phi}_{[t_k]}+\lambda\mathbf{I} = \mathbf{V}_k + \sum_{i=1}^N\Delta\mathbf{V}_k(i) \preceq (1+N\beta D)\mathbf{V}_k.$$

Using Lemma A.5 again finishes the proof. □

Proof of communication cost in Theorem 2.3.8 in Section 2.3.10

Recall from Section 2.3.8 that $\mathcal{D}_k$ is the set of time indices for the data points that are used to construct the embedded statistics on the server at the k-th communication round, for $k = 1, \dots, B$. We denote the corresponding (exact) covariance matrix as $\mathbf{V}_k = \lambda\mathbf{I} + \mathbf{\Phi}_{\mathcal{D}_k}^\top\mathbf{\Phi}_{\mathcal{D}_k} \in \mathbb{R}^{p\times p}$, with $\mathbf{V}_0 = \lambda\mathbf{I}$, and kernel matrix as $\mathbf{K}_{\mathcal{D}_k,\mathcal{D}_k} = \mathbf{\Phi}_{\mathcal{D}_k}\mathbf{\Phi}_{\mathcal{D}_k}^\top \in \mathbb{R}^{|\mathcal{D}_k|\times|\mathcal{D}_k|}$.

Similar to [97], by defining $k_p = \min\{k \in [B] \mid \det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_k,\mathcal{D}_k}) \geq 2^p)\}$, we have $\log\big(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}},\mathcal{D}_{k_{p+1}}})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p},\mathcal{D}_{k_p}})}\big) \geq 1$ for each $p \geq 0$. We call the sequence of time steps in-between t_{k_p} and $t_{k_{p+1}}$ an epoch, and denote the total number of epochs as P. Note that since

$$\begin{aligned}
&\log\Big(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_1},\mathcal{D}_{k_1}})}{\det(\mathbf{I})}\Big)+\log\Big(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_2},\mathcal{D}_{k_2}})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_1},\mathcal{D}_{k_1}})}\Big)+\cdots+\log\Big(\frac{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_P},\mathcal{D}_{k_P}})}{\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{P-1}},\mathcal{D}_{k_{P-1}}})}\Big)\\
&\leq \log\big(\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{[T],[T]})\big) \leq 2\gamma_T,
\end{aligned}$$

there can be at most $2\gamma_T$ terms, i.e., $P \leq 2\gamma_T$. Now that we have divided the time horizon $[T]$ into P epochs using $\{t_{k_p}\}_{p\in[P]}$, we prove the following lemma that upper bounds the total number of times communication is triggered in each epoch.

Lemma 2.3.10 (Number of communications per epoch)**.** *For each epoch, i.e., the sequence of time steps in-between t_{k_p} and $t_{k_{p+1}}$, the number of communications is upper bounded by $N + \frac{4\beta}{D}$.*

Since there are at most $2\gamma_T$ epochs, the total number of communications $B \leq 2\gamma_T(N + \frac{4\beta}{D})$. Moreover, by Lemma 2.3.4, we know that during each communication, the size of data being communicated is $O\big(\log^2(T)\gamma_T^2\big)$. Hence, with $D = \frac{1}{N^2}$, $C_T = O(N^2\gamma_T^3\log^2(T))$.

Proof of Lemma 2.3.10. Consider the epoch $[t_{k_p}, t_{k_{p+1}} - 1]$ for some $p = 0, 1, \dots, P$. We denote the total number of communications in this epoch as Q_p, and the total number of communications in this epoch that are triggered by client i as $Q_{p,i}$ for $i \in [N]$, i.e., $Q_p = \sum_{i=1}^{N} Q_{p,i}$.

Let's denote the indices associated with the communications triggered by some client i as $\kappa_1, \kappa_2, \dots, \kappa_{Q_{p,i}} \in [k_p, k_{p+1} - 1]$. Then for each $j = 2, 3, \dots, Q_{p,i}$, i.e., excluding client i's first communication in this epoch, due to our event-trigger design in (2.7), we have

$$\beta \sum_{s \in \Delta\mathcal{D}_{\kappa_j}} \sigma_{k_p}^2(\mathbf{x}_s) \geq \beta \sum_{s \in \Delta\mathcal{D}_{\kappa_j}} \sigma_{\kappa_{j-1}}^2(\mathbf{x}_s) \geq \sum_{s \in \Delta\mathcal{D}_{\kappa_j}} \tilde{\sigma}_{\kappa_{j-1}}^2(\mathbf{x}_s) > D,$$

where the first inequality is because by definition of κ_{j-1}, we have $\kappa_{j-1} \geq k_p$, so $\sigma_{\kappa_{j-1}}^2(\mathbf{x}) \leq \sigma_{k_p}^2(\mathbf{x}), \forall \mathbf{x}$, and the second inequality is due to Lemma A.11. Therefore, we have $\sum_{s \in \Delta\mathcal{D}_{k_j}} \sigma_{k_p}^2(\mathbf{x}_s) \geq D/\beta$. Since $\sigma_{k_p}^2(\mathbf{x}) = \|\phi(\mathbf{x})\|_{\mathbf{V}_{k_p}^{-1}}^2$, we have

$$\begin{aligned}
D/\beta &\leq \sum_{s \in \Delta\mathcal{D}_{k_j}} \|\phi(\mathbf{x}_s)\|_{\mathbf{V}_{k_p}^{-1}}^2 \leq 4 \log\Big(\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}, \mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}})}\Big) \\
&\leq -4 + 4 \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}, \mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}})}
\end{aligned}$$

where the second inequality is by definition of epoch, i.e., $\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}-1}, \mathcal{D}_{k_{p+1}-1}}) / \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}}) \leq 2$, combined with Lemma A.9, and the third is because $\log(x) \leq x - 1$ for $x > 0$. Hence, we have

$$\frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}, \mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}})} \geq 1 + \frac{D}{4\beta},$$

and thus, we have $\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}, \mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa_j}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}}) \geq \frac{D}{4\beta} \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}})$ for all $j = 2, 3, \dots, \mathcal{Q}_{p,i}$ and all client $i \in [N]$.

Denote the indices associated with the communications of all clients in this epoch as $\kappa'_1, \kappa'_2, \dots, \kappa'_{Q_p} \in \{k_p, k_{p+1} - 1\}$. Then for each $j \in [Q_p]$, if client $c_{\kappa'_j}$ has already communicated with the server ealier in this epoch, we have

$$\begin{aligned}
&\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_j}, \mathcal{D}_{\kappa'_j}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_{j-1}}, \mathcal{D}_{\kappa'_{j-1}}}) \\
&= \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_{j-1}} \cup \Delta\mathcal{D}_{\kappa'_j}, \mathcal{D}_{\kappa'_{j-1}} \cup \Delta\mathcal{D}_{\kappa'_j}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_{j-1}}, \mathcal{D}_{\kappa'_{j-1}}}) \\
&\geq \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa'_j}, \mathcal{D}_{k_p} \cup \Delta\mathcal{D}_{\kappa'_j}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}}) \\
&\geq \frac{D}{4\beta} \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}})
\end{aligned}$$

where the first inequality is obtained via matrix determinant lemma and Lemma A.6, and the second is due to the inequality we derived above. Summing over all communications in this epoch, we have

$$\begin{aligned}
&\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}-1}, \mathcal{D}_{k_{p+1}-1}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}}) \\
&= \sum_{j=1}^{Q_p} \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_j}, \mathcal{D}_{\kappa'_j}}) - \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{\kappa'_{j-1}}, \mathcal{D}_{\kappa'_{j-1}}}) \\
&\geq \sum_{i=1}^{N} (Q_{p,i} - 1) \frac{D}{4\beta} \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p}, \mathcal{D}_{k_p}}),
\end{aligned}$$

and since $\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}-1},\mathcal{D}_{k_{p+1}-1}})/\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p},\mathcal{D}_{k_p}}) \leq 2$ by our definition of epoch, we have

$$1+\frac{D}{4\beta}\sum_{i=1}^{N}(Q_{p,i}-1) \leq \det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}-1},\mathcal{D}_{k_{p+1}-1}})/\det(\mathbf{I}+\lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p},\mathcal{D}_{k_p}}) \leq 2,$$

so $Q_p=\sum_{i=1}^{N}Q_{p,i} \leq N+\frac{4\beta}{D}$, which finishes the proof. □

Proof of cumulative regret in Theorem 2.3.8 in Section 2.3.10

To facilitate regret analysis of Async-KernelUCB, we need to introduce some additional notations. First, let's denote the client who triggers the k-th communication as $c_k \in [N]$, the index of its next communication as $\bar{k}(c_k)$, and the time step when the $\bar{k}(c_k)$-th communication happens is $t_{\bar{k}(c_k)}$ ($t_{\bar{k}(c_k)}=T$ if k is client c_k's final communication with the server). Then we denote the set of time steps in-between (but not including) the current (the k-th) communication and client c_k's next communication when client c_k is active as $\mathcal{P}_k := \{t_k < s < t_{\bar{k}(c_k)} : i_s = c_k\}$, and thus by definition $\Delta\mathcal{D}_{\bar{k}(c_k)} = \mathcal{N}_{t_{\bar{k}(c_k)}}(c_k)\setminus\mathcal{N}_{t_k}(c_k) = \mathcal{P}_k \cup \{t_{\bar{k}(c_k)}\}$. We also define $\mathcal{P}_0$ as the union over the set of time steps before the first communication of each client $i \in [N]$. Therefore, we have $\big(\cup_{k=0}^{B}\mathcal{P}_k\big)\cup\{t_k\}_{k\in[B]}=[T]$. Since in Algorithm 14, the approximated mean and variance of each client only get updated when it triggers communication, and then remain fixed until after its next communication, we have that all the interactions in $\mathcal{P}_k \cup \{t_{\bar{k}(c_k)}\}$ are based on the same $\{\tilde{\mu}_k(\cdot),\tilde{\sigma}_k(\cdot)\}$, for $k=0,1,\dots,B$. In addition, an important observation is that, based on our event-trigger in (2.7), we have

$$\begin{aligned}&\sum_{s\in\mathcal{P}_k}\tilde{\sigma}_k^2(\mathbf{x}_s) \leq D,\\ &\Big[\sum_{s\in\mathcal{P}_k}\tilde{\sigma}_k^2(\mathbf{x}_s)\Big]+\tilde{\sigma}_k^2(\mathbf{x}_{t_{\bar{k}(c_k)}}) > D.\end{aligned} \tag{2.42}$$

Now we are ready to upper bound the cumulative regret. Consider some time step $t \in \mathcal{P}_k \cup \{t_{\bar{k}(c_k)}\}$. Due to our arm selection rule (line 5 of Algorithm 14), we have $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{A}_t}\tilde{\mu}_k(\mathbf{x})+\alpha\tilde{\sigma}_k(\mathbf{x})$. Combining this with Lemma 2.3.7, with probability at least $1-\delta$, we have

$$\begin{aligned}&f(\mathbf{x}_t^\star) \leq \tilde{\mu}_k(\mathbf{x}_t^\star)+\alpha\tilde{\sigma}_k(\mathbf{x}_t^\star) \leq \tilde{\mu}_k(\mathbf{x}_t)+\alpha\tilde{\sigma}_k(\mathbf{x}_t),\\ &f(\mathbf{x}_t) \geq \tilde{\mu}_k(\mathbf{x}_t)-\alpha\tilde{\sigma}_k(\mathbf{x}_t),\end{aligned}$$

where $\mathbf{x}_t^\star := \arg\max_{\mathbf{x}\in\mathcal{A}_t} f(\mathbf{x}) = \arg\max_{\mathbf{x}\in\mathcal{A}_t}\phi(\mathbf{x})^\top\theta_\star$ is the optimal arm at time step t, and thus $r_t = f(\mathbf{x}_t^\star)-f(\mathbf{x}_t) \leq 2\alpha\tilde{\sigma}_k(\mathbf{x}_t)$. The cumulative regret R_T can be rewritten as

$$R_T=\sum_{k=0}^{B}\sum_{s\in\mathcal{P}_k}r_s+\sum_{k=1}^{B}r_{t_k} \leq \sum_{k=0}^{B}\sum_{s\in\mathcal{P}_k}\min(2LS,2\alpha\tilde{\sigma}_k(\mathbf{x}_s))+\sum_{k=1}^{B}\min\{2LS,2\alpha\tilde{\sigma}_{\underline{k}(c_k)}(\mathbf{x}_{t_k})\}.$$

Bounding first term: To bound the first term, we introduce an imaginary variance function $\sigma_{k,s-1}^2(\cdot)$ (not computed in the actual algorithm) for $s \in \mathcal{P}_k$ and $k=0,1,\dots,B$, which is constructed using dataset $\big(\cup_{k'=0}^{k-1}\mathcal{P}_{k'}\big)\cup\{s' \in \mathcal{P}_k : s' \leq s-1\}$. In the following paragraph, we will bound the first term by showing that $\sum_{k=0}^{B}\sum_{s\in\mathcal{P}_k}\tilde{\sigma}_k^2(\mathbf{x}_s)$ is not too much larger than $\sum_{k=0}^{B}\sum_{s\in\mathcal{P}_k}\sigma_{k,s-1}^2(\mathbf{x}_s)$.

This requires us to bound the ratio $\frac{\sigma_k^2(\mathbf{x}_s)}{\sigma_{k,s-1}^2(\mathbf{x}_s)}$ for $s \in \mathcal{P}_k$ and $k=0,1,\dots,B$. Recall that $\sigma_k^2(\cdot)$ is constructed using data points that N clients have uploaded to the server up to the k-th communication, i.e., $\mathcal{D}_k=\cup_{i=1}^{N}\mathcal{N}_{t_{\underline{k}(i)}}(t_k)$, which is a subset of $\mathcal{D}_k \cup \big(\cup_{i=1}^{N}\Delta\mathcal{D}_{\bar{k}(i)}\big) = \mathcal{D}_k \cup \big(\cup_{i=1}^{N}\mathcal{P}_{\underline{k}(i)}\cup\{t_{\bar{k}(i)}\}\big)$. However, as shown in (2.42), the event-trigger cannot be directly used to upper bound the summation of approximated variances in $\mathcal{P}_{\underline{k}(i)}\cup\{t_{\bar{k}(i)}\}$, but can be used to upper bound that in $\mathcal{P}_{\underline{k}(i)}$, which is why we construct the imaginary variance function without using data points with time indices $\{t_k\}_{k\in[B]}$. Specifically, using the notations we just introduced, we can rewrite the variance as

$$\begin{aligned}
\sigma_k^2(\mathbf{x}) &= \phi(\mathbf{x})^\top \big(\mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k} + \lambda \mathbf{I}\big)^{-1} \phi(\mathbf{x}) \\
\sigma_{k,s-1}^2(\mathbf{x}) &= \phi(\mathbf{x})^\top \big(\mathbf{\Phi}_{\mathcal{D}_k \setminus \{t_{k'}\}_{k' \in [k]}}^\top \mathbf{\Phi}_{\mathcal{D}_k \setminus \{t_{k'}\}_{k' \in [k]}} + \lambda \mathbf{I} + \sum_{i \neq c_k} \mathbf{\Phi}_{\mathcal{P}_{\underline{k}(i)}}^\top \mathbf{\Phi}_{\mathcal{P}_{\underline{k}(i)}} \\
&\quad + \mathbf{\Phi}_{\{s' \in \mathcal{P}_k : s' \leq s-1\}}^\top \mathbf{\Phi}_{\{s' \in \mathcal{P}_k : s' \leq s-1\}}\big)^{-1} \phi(\mathbf{x}) \\
&\geq \phi(\mathbf{x})^\top \big(\mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k} + \lambda \mathbf{I} + \sum_{i \neq c_k} \mathbf{\Phi}_{\mathcal{P}_{\underline{k}(i)}}^\top \mathbf{\Phi}_{\mathcal{P}_{\underline{k}(i)}} + \mathbf{\Phi}_{\{s' \in \mathcal{P}_k : s' \leq s-1\}}^\top \mathbf{\Phi}_{\{s' \in \mathcal{P}_k : s' \leq s-1\}}\big)^{-1} \phi(\mathbf{x})
\end{aligned}$$

The following lemma provides a upper bound for this ratio.

Lemma 2.3.11 (Bounding $\sigma_k^2(\mathbf{x}_s)/\sigma_{k,s-1}^2(\mathbf{x}_s)$)**.** *Under the same condition as Lemma 2.3.4, with communication threshold D, we have $\forall k, s$ that*

$$\sigma_k^2(\mathbf{x}_s)/\sigma_{k,s-1}^2(\mathbf{x}_s) \leq 1 + N\beta D.$$

With Lemma 2.3.11, we can bound the first term as

$$\begin{aligned}
&\sum_{k=0}^{B} \sum_{s \in \mathcal{P}_k} \min(2LS, 2\alpha \tilde{\sigma}_k(\mathbf{x}_s)) \leq 2\alpha \sqrt{T \sum_{k=0}^{B} \sum_{s \in \mathcal{P}_k} \tilde{\sigma}_k^2(\mathbf{x}_s)} \leq 2\alpha \sqrt{T\beta \sum_{k=0}^{B} \sum_{s \in \mathcal{P}_k} \sigma_k^2(\mathbf{x}_s)} \\
&= 2\alpha \sqrt{T\beta \sum_{k=0}^{B} \sum_{s \in \mathcal{P}_k} \sigma_{k,s-1}^2(\mathbf{x}_s) \cdot \frac{\sigma_k^2(\mathbf{x}_s)}{\sigma_{k,s-1}^2(\mathbf{x}_s)}} \leq 2\alpha \sqrt{T\beta(1 + N\beta D) \sum_{k=0}^{B} \sum_{s \in \mathcal{P}_k} \sigma_{k,s-1}^2(\mathbf{x}_s)} \\
&\leq 4\alpha \sqrt{T\beta(1 + N\beta D)\gamma_T} \\
&\leq 4\Big[(1/\sqrt{1-\epsilon} + 1)\sqrt{\lambda} S + 2R\big(\sqrt{1 + ND\beta} + N\sqrt{2D\beta}\big)\sqrt{\ln(1/\delta) + \gamma_T}\Big] \sqrt{T\beta(1 + N\beta D)\gamma_T}
\end{aligned}$$

with probability at least $1 - 2\delta$, where the first inequality is due to Cauchy-Schwarz, and second is due to the property of ϵ-accuracy in Lemma A.11, the third is due to Lemma 2.3.11, the forth is by definition of maximum information gain γ_T, and the last is by substituting α defined in Lemma 2.3.7.

Bounding second term: For the second term $\sum_{k=1}^{B} \min\{2LS, 2\alpha \tilde{\sigma}_{\underline{k}(c_k)}(\mathbf{x}_{t_k})\}$, we should note that $\tilde{\sigma}_{\underline{k}(c_k)}(\cdot)$ is the approximated variance function that client c_k received during its last communication with the server, instead of $\sigma_{k-1}(\cdot)$ as in our proof of Lemma 2.3.6 when bounding the size of dictionary. Ideally, we want to relate each $\sigma_{\underline{k}(c_k)}(\cdot)$ to $\sigma_k(\cdot)$ and then apply the elliptical potential argument, but as we do not make any assumption on how frequent client arrives, it is possible that for clients who show up infrequently, these two functions are very different.

However, by using the epoch argument as in the proof for communication cost, we can show that this undesirable situation only occurs at most $2\gamma_T$ times. Specifically, recall that $\mathbf{V}_k = \lambda \mathbf{I} + \mathbf{\Phi}_{\mathcal{D}_k}^\top \mathbf{\Phi}_{\mathcal{D}_k}$, with $\mathbf{V}_0 = \lambda \mathbf{I}$, and kernel matrix as $\mathbf{K}_{\mathcal{D}_k,\mathcal{D}_k} = \mathbf{\Phi}_{\mathcal{D}_k} \mathbf{\Phi}_{\mathcal{D}_k}^\top \in \mathbb{R}^{|\mathcal{D}_k| \times |\mathcal{D}_k|}$. We define $k_p = \min\{k \in [B] \mid \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_k,\mathcal{D}_k}) \geq 2^p)\}$, such that $\log\big(\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}},\mathcal{D}_{k_{p+1}}})/\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p},\mathcal{D}_{k_p}})\big) \geq 1$ for each $p \geq 0$. We call the sequence of time steps in-between t_{k_p} and $t_{k_{p+1}}$ an epoch, and denote the total number of epochs as P. As shown in the proof for communication cost, we have $P \leq 2\gamma_T$.

Consider the epoch $[t_{k_p}, t_{k_{p+1}} - 1]$ for some $p = 0, 1, \dots, P$. We denote the total number of communications in this epoch that are triggered by client i as $Q_{p,i}$ for $i \in [N]$, and the indices associated with these communications triggered by client i as $\kappa_1, \kappa_2, \dots, \kappa_{Q_{p,i}} \in [k_p, k_{p+1} - 1]$.

As mentioned above, the approximated variance used during arm selection at t_{κ_1}, i.e, $\sigma_{\underline{\kappa_1}(c_{\kappa_1})}^2(\cdot)$ could be from a very long time ago. Therefore, we simply bound its regret by $2LS$, and in total, there can be at most $2\gamma_T N$ such terms for all N clients, leading to a upper bound of $4N\gamma_T LS$.

Now we only need to be concerned about the communications at $j = 2, 3, \dots, Q_{p,i}$, and show that $\sigma^2_{\underline{\kappa_j}(c_{\kappa_j})}(\mathbf{x})$ is close to $\sigma^2_{\kappa_j}(\mathbf{x})$ for all $\mathbf{x}$. Specifically, we have

$$\sigma^2_{\underline{\kappa_j}(c_{\kappa_j})}(\mathbf{x}) = \sigma^2_{\kappa_{j-1}}(\mathbf{x}) = \sigma^2_{\kappa_j}(\mathbf{x})\frac{\sigma^2_{\kappa_{j-1}}(\mathbf{x})}{\sigma^2_{\kappa_j}(\mathbf{x})} \le 2\sigma^2_{\kappa_j}(\mathbf{x}),$$

where the first equality is because by definition $\underline{\kappa_j}(c_{\kappa_j}) = \kappa_{j-1}$, the first inequality is because $\sigma^2_{\kappa_{j-1}}(\mathbf{x})/\sigma^2_{\kappa_j}(\mathbf{x}) \le \det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_{p+1}-1},\mathcal{D}_{k_{p+1}-1}})/\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{D}_{k_p},\mathcal{D}_{k_p}}) \le 2$ due to Lemma A.8, Lemma A.5 and the definition of epoch. Therefore, further applying Cauchy-Schwarz and the ϵ-accuracy property in Lemma A.11, the second term can be bounded by

$$\begin{aligned}
&\sum_{k=1}^{B} \min\{2LS, 2\alpha\tilde{\sigma}_{\underline{k}(c_k)}(\mathbf{x}_{t_k})\} \le 4N\gamma_T LS + 2\alpha\sqrt{2B\beta\sum_{k=1}^{B}\sigma^2_k(\mathbf{x}_{t_k})}\\
&\le 4N\gamma_T LS + 2\alpha\sqrt{2B\beta\sum_{k=1}^{B}\sigma^2_{k-1,t_k-1}(\mathbf{x}_{t_k})} < 4N\gamma_T LS + 2\alpha\sqrt{2B\beta\sum_{k=1}^{B}\sum_{s\in\Delta\mathcal{D}_k}\sigma^2_{k-1,s-1}(\mathbf{x}_s)}\\
&\le 4N\gamma_T LS + 4\alpha\sqrt{2T\beta\gamma_T}
\end{aligned}$$

where the imaginary variance function $\sigma^2_{k-1,s-1}(\cdot)$ is constructed using dataset $\big(\cup_{k'=1}^{k-1}\Delta\mathcal{D}_{k'}\big) \cup \{s' \in \Delta\mathcal{D}_k : s' \le s-1\}$, the second inequality is because variance is non-increasing over time, the third is because variances are positive, and the last is due to definition of maximum information gain γ_T and that $B \le T$.

Putting upper bounds for the first and second term together, we have $R_T \le 4N\gamma_T LS + 4\sqrt{2}\Big[(1/\sqrt{1-\epsilon}+1)\sqrt{\lambda}S + 2R\big(\sqrt{1+ND\beta} + N\sqrt{2D\beta}\big)\sqrt{\ln(1/\delta)+\gamma_T}\Big]\sqrt{T\beta(1+N\beta D)\gamma_T}$.

Proof of Lemma 2.3.11. We denote $\mathbf{V}_k = \lambda\mathbf{I} + \mathbf{\Phi}^\top_{\mathcal{D}_k}\mathbf{\Phi}_{\mathcal{D}_k}$, $\Delta\mathbf{V}_{k,s-1}(i) = \mathbf{\Phi}^\top_{\mathcal{P}_{\underline{k}(i)}}\mathbf{\Phi}_{\mathcal{P}_{\underline{k}(i)}}$ for $i \neq c_k$ and $\Delta\mathbf{V}_{k,s-1}(c_k) = \mathbf{\Phi}^\top_{\{s'\in\mathcal{P}_k:s'\le s-1\}}\mathbf{\Phi}_{\{s'\in\mathcal{P}_k:s'\le s-1\}}$.

In the following, we first show that

$$\mathbf{V}_k \succeq \frac{1}{\beta D}\Delta\mathbf{V}_{k,s-1}(i) \tag{2.43}$$

for all $i \in [N]$. Note that for any client $i \neq c_k$, we have

$$\begin{aligned}
&\frac{\mathbf{x}^\top\mathbf{V}^{-1}_{\underline{k}(i)}\mathbf{x}}{\mathbf{x}^\top\big(\mathbf{V}_{\underline{k}(i)} + \Delta\mathbf{V}_{k,s-1}(i)\big)^{-1}\mathbf{x}} \le 1 + \sum_{s\in\mathcal{P}_{\underline{k}(i)}}\mathbf{x}_s^\top\mathbf{V}^{-1}_{\underline{k}(i)}\mathbf{x}_s\\
&= 1 + \sum_{s\in\mathcal{P}_{\underline{k}(i)}}\sigma^2_{\underline{k}(i)}(\mathbf{x}_s) \le 1 + \beta\sum_{s\in\mathcal{P}_{\underline{k}(i)}}\tilde{\sigma}^2_{\underline{k}(i)}(\mathbf{x}_s)\\
&\le 1 + \beta D,
\end{aligned}$$

where the first inequality is due to Lemma A.10, the second inequality is due to Lemma 2.3.4 and Lemma A.11, and the last inequality is due to (2.42).

This implies $\mathbf{V}^{-1}_{\underline{k}(i)} \preceq (1+\beta D)\big(\mathbf{V}_{\underline{k}(i)} + \Delta\mathbf{V}_{k,s-1}(i)\big)^{-1}$. Then due to Lemma A.5, we have $(1+\beta D)\mathbf{V}_{\underline{k}(i)} \succeq \mathbf{V}_{\underline{k}(i)} + \Delta\mathbf{V}_{k,s-1}(i)$, and thus $\mathbf{V}_{\underline{k}(i)} \succeq \frac{1}{\beta D}\Delta\mathbf{V}_{k,s-1}(i)$. Moreover, since $\underline{k}(i) < k, \forall i \neq c_k$, we have $\mathbf{V}_k \succeq \mathbf{V}_{\underline{k}(i)} \succeq \frac{1}{\beta D}\Delta\mathbf{V}_{k,s-1}(i)$. Similarly for client c_k, we have

$$\frac{\mathbf{x}^\top\mathbf{V}_k^{-1}\mathbf{x}}{\mathbf{x}^\top\big(\mathbf{V}_k + \Delta\mathbf{V}_{k,s-1}(c_k)\big)^{-1}\mathbf{x}} \le 1 + \sum_{s'\in\mathcal{P}_k:s'\le s-1}\sigma^2_k(\mathbf{x}_{s'}) \le 1 + \beta D.$$

Again, this implies $\mathbf{V}_k \succeq \frac{1}{\beta D}\Delta\mathbf{V}_{k,s-1}(c_k)$, which finishes the proof of (2.43).

By averaging (2.43) over all N clients, we have

$$\mathbf{V}_k \succeq \frac{1}{N\beta D}\sum_{i=1}^{N}\Delta\mathbf{V}_{k,s-1}(i),$$

and thus, we have

$$\mathbf{V}_k + \sum_{i=1}^{N}\Delta\mathbf{V}_{k,s-1}(i) \preceq (1+N\beta D)\mathbf{V}_k.$$

Using Lemma A.5 again, we have $(1+N\beta D)(\mathbf{V}_k + \sum_{i=1}^{N}\Delta\mathbf{V}_{k,s-1}(i))^{-1} \succeq \mathbf{V}_k^{-1}$. Therefore, we have

$$\frac{\sigma_k^2(\mathbf{x})}{\sigma_{k,s-1}^2(\mathbf{x})} \leq \frac{\phi(\mathbf{x})^\top \mathbf{V}_k^{-1}\phi(\mathbf{x})}{\phi(\mathbf{x})^\top \big(\mathbf{V}_k + \sum_{i=1}^{N}\Delta\mathbf{V}_{k,s-1}(i)\big)^{-1}\phi(\mathbf{x})} \leq 1 + N\beta D$$

□

2.4 Conclusion

In conclusion, this chapter delved into the realm of cooperative decision-making algorithms for agents operating in heterogeneous and non-stationary environments, as well as in decentralized environments. The investigation aimed to understand the intricacies and challenges associated with cooperation in these complex scenarios and propose effective solutions to enhance the decision-making capabilities of cooperative agents.

We studied a new bandit problem formulation, named clustered and non-stationary bandit, where the reward function of each agent changes over time, which induces changes in the task similarities between agents. In this case, directly pooling agents' data together to learn a single model may have negative impacts on the performance. Instead, the decision making system needs to carefully decide *when and with whom the agents should collaborate*, in order to enjoy improved performance compared with learning a separate model on each agent. Existing cooperative decision making algorithms cannot cope with this situation, as they usually impose stationary assumptions about the environment, and in the meantime, existing algorithms designed for non-stationary environments cannot enjoy the benefit of cooperation, as they only focus on mitigating the influence of outdated data on model estimation for a single agent. In [99, 102], we proposed solutions that strictly generalize both lines of works, i.e., they work for non-stationary setting and cooperative multi-agent setting simultaneously, by equipping the agents with statistical hypothesis tests, such that they can readily adjust the group of agents to cooperate with, whenever change in the reward distribution is detected. With effective data sharing across agents, the proposed algorithms were proved to achieve optimal theoretical guarantee, and demonstrated superior performance on real-world recommendation datasets.

We then studied federated bandit problem, where data storage and computation are distributed to each agent. In this case, communication bandwidth becomes the main bottleneck for cooperative decision making, e.g., communication in a network of mobile devices can be slower than local computation by several orders of magnitude. This gives rise to the conflict between the need of timely data/model aggregation for *regret minimization* and the need of *communication efficiency*, and thus a well-designed communication strategy becomes vital to strike the balance. In [64, 84, 100, 98], we designed cooperative decision making algorithms that attain provably optimal regret, while effectively reducing the communication cost incurred during the online learning process.

Overall, the findings from our study highlight the importance of designing decision-making algorithms for cooperative agents in complex and diverse environments. The proposed approaches provide a solid foundation for future research in the field of cooperative multi-agent systems, offering insights into addressing challenges in heterogeneous and non-stationary environments, as well as decentralized scenarios. In conclusion, the advancements made in this chapter contribute to the broader goal of developing cooperative decision-making algorithms that can enable agents to effectively collaborate, adapt, and make optimal choices in heterogeneous, non-stationary, and decentralized environments. By fostering cooperation among agents, we can unlock tremendous potential for solving complex problems and achieving collective goals that surpass the capabilities of individual entities alone.

Chapter 3

Decision Making with Non-cooperative Agents

As we discussed in Chapter 1, in non-cooperative multi-agent systems, where agents pursue self-interest and compete for limited resources, the decision making problem takes on a more competitive nature, and collectively they may make sub-optimal decisions. In this case, it is essential to factor in their self-interested behaviors, i.e., by carefully modeling their various non-cooperative or even competitive behaviors, such as: 1) agents withhold the actual feedback, and the system can only learn from their revealed preference feedback; 2) agents refuse to participate in cooperation, unless the benefits of cooperation outweighs the risks, e.g., of privacy leakage; 3) agents are strictly competitive, i.e., the actions of one agent negatively affect the utilities of the others. Therefore, for decision making systems with such non-cooperative agents, we propose mechanisms/strategies on the system side to combat such non-cooperative behaviors for optimal decision making. This chapter mainly uses recommender systems as motivating examples for various strategic behaviors of the agents, but the proposed algorithms and theoretical results can be applied to much wider range of application scenarios where self-interest and individual objectives take precedence over collaborative effort.

3.1 Learn optimal action from revealed preference feedback

A recommender system (hereinafter referred to as *system*) is designed to predict users' preferences over items so as to maximize the utility of the recommended items [103, 104]. Driven by this principle, there has been a tremendous amount of research efforts and industry practices on developing various recommendation algorithms that predict item utility for each user based on the observed user-item interactions, including collaborative filtering [103, 105, 106], latent factor models [104, 107, 108], neural recommendation models [109, 110, 111], and sequential recommendation models [112, 113, 114].

Nevertheless, this paradigm is built on an overly simplified user model: users are omniscient about the (millions of) items and are willing to spend efforts to provide detailed feedback, so that the system can directly query their preferences. This assumption ceases to be true in real-world recommendation applications where the size of the item space could be formidably large. As a result, instead of being a static "classifier" [115, 17, 106], an ordinary user typically is also *learning* the item utility from her interactions with the system. For instance, a user might be new to a category of items; thus, her responses to such items can only be accurate after consuming the recommended items, possibly even after multiple times.

This "inaccuracy" in users' feedback cannot be simply modeled as random noise, since it naturally depend on the interaction history and thus could be biased by her previous choices. More specifically, any small bias (e.g., towards a particular item category) in the system's past recommendations will bias the user's learning, which consequently leads to biased user feedback, which then further bias the system's subsequent recommendations. This forms a vicious circle – even if an optimal item is recommended to the user, she might not take it due to her currently inaccurate utility estimation; but failing to consume the optimal item will stop the user from exploring that direction, and thus leading to repeated future rejections of the same optimal recommendations. This is similar to the explore-exploit dilemma in bandit problems, but is much worse because in bandit problems the noise of user feedback is independent from the interaction history, whereas here the bias will accumulate. Our problem setting also differs from reinforcement learning where the reward function is fixed by the environment and independent from the agent's actions.

To address the limitation caused by the previous omniscient user assumption, we propose to model a user as an autonomous agent who is learning to evaluate the utility of system's recommendations from her interaction history. We formulate the system-user interaction in a dueling bandit setup [116], such that the user does not explicitly disclose their estimated utility of a chosen item. This more challenging feedback assumption is motivated by the observation that an ordinary user will most often take action that fulfills her information needs with the least effort, and thus does not bother providing details, e.g., numerical ratings [15]. Specifically, we assume at each time step, the system proposes two items for the user and can only observe the user's choice between the two items, i.e., revealed preference feedback. The system aims at minimizing the cumulative regret from the interaction with the user in a given period T.

3.1.1 Related works

The first related direction is the dueling bandit problem. First proposed by [117], dueling bandit models an online learning problem where the feedback at each step is restricted to a noisy comparison between a pair of arms. In follow-up works, [118] developed solutions by proposing a black-box reduction from dueling bandit to classic multi-armed bandit (MAB), [119] studied the adversarial and contextual extensions of dueling bandit and generalized the solution concept. Our feedback assumption is fundamentally different from that in dueling bandit as the user's feedback evolves as she learns from the realized rewards. This coupled environment results in the failure of almost all existing dueling bandit algorithms, including those mentioned above, as we will demonstrate in our empirical study.

The ellipsoid method serves as a key building block in our algorithm design. First proposed by [120, 121], the ellipsoid method is used to prove linear programs are solvable in polynomial time. Such an elegant idea has found applications in preference elicitation [122], recommender systems design [123, 124], and feature-based dynamic pricing [125, 126]. The main challenge in applying the ellipsoid method to our problem is that due to the user's inaccurate feedback, the system cannot control the intersection of the cutting hyperplane and thus needs to determine when to shrink the uncertainty set adaptively.

3.1.2 Contemporaneous system-user learning problem

Our setup inherits from the celebrated contextual dueling bandit problem but considers intrinsically different user behaviors, i.e., a learning and thus dynamically evolving user. Let $\mathcal{A}$ be the set of candidate items (henceforth, the *arms*) that the system can recommend at each round $t \in [T]$. We are interested in scenarios where $\mathcal{A}$ is formidably large and diverse. Our results hold for arbitrary $\mathcal{A}$, continuous or discrete, so long as it has a non-trivial interior and is sufficiently "dense" (see formal definitions later). The user's expected utility of consuming any arm $\mathbf{a} \in \mathcal{A}$ is governed by a hidden preference parameter $\theta_* \in \mathbb{R}^d$ and, specifically, is realized by the linear reward function $\theta_*^\top \mathbf{a}$. At each round t, the system recommends a pair of arms $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$ and the user chooses one of them, i.e., the comparative feedback as in dueling bandits. We assume that the user does not know θ_* either and relies on her current estimation θ_t to make a choice between $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$. Since any non-zero scaling on θ_* does not affect the user's feedback, we assume $\|\theta_*\|_2 = 1$ without loss of generality.

The key conceptual contribution of our problem setup is a formal non-stationary user model that captures a wide range of user-system interactions yet still permits tractable analysis of online learning with non-trivial regret guarantees. We defer a formal description of this user model to the following paragraphs, and only summarize the interaction protocol at each round $t \in [T]$ as follows:

1. The system recommends $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t}) \in \mathcal{A}^2$ to the user.
2. The user uses θ_t, i.e., her estimation of θ^* at time t, to choose an arm from $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$, denoted as $\mathbf{a}_t$.
3. The user observes reward r_t and updates θ_{t+1} based on her observed history $\mathcal{H}_t = \{(\mathbf{a}_s, r_s)\}_{s=1}^t$.
4. The system observes the user's choice $\mathbf{a}_t$ and updates its recommendation policy.

The learning objective for the system is to minimize the regret defined as

$$R_T = \sum_{t=1}^{T} \theta_*^\top (2\mathbf{a}_* - \mathbf{a}_{0,t} - \mathbf{a}_{1,t}), \tag{3.1}$$

where $\mathbf{a}_* = \arg\max_{\mathbf{a}\in\mathcal{A}} \theta_*^\top \mathbf{a}$.

Next we introduce the remaining core components of the user behavior model by specifying: 1). her method for estimating θ_t; and 2). her strategy for selecting an arm based on θ_t. We refer to them as the *estimation rule* and the *decision rule* respectively.

Modeling a Learning User We consider a general model of a learning user as follows.

1. (Estimation Rule) The user collects the past observations $\mathcal{H}_{t-1}$ and calculate $\theta_t = F(\mathcal{H}_{t-1})$ using any learning algorithm F, such that
$$\|\theta_* - \theta_t\|_{V_t} \le c_1 t^{\gamma_1} g(\delta) \tag{3.2}$$
holds with probability $1-\delta$, where $V_t = V_0 + \sum_{s=1}^{t-1} \mathbf{a}_s \mathbf{a}_s^\top$, $\gamma_1 \in (0, \frac{1}{2})$ and c_1 are constants such that c_1 is independent of t. V_0 is assumed to be any Positive Semi-definite (PSD) matrix that summarizes the user's *prior knowledge* regarding the item space. One can interpret V_0 as $\sum_{i=1}^n \mathbf{a}_{-i}\mathbf{a}_{-i}^\top$, where $\mathbf{a}_{-i}$ is the user's consumed item before engaging with the system. The spectrum of V_0 thus reflects the estimation accuracy regarding different directions of the item space. For example, if V_0 has some small eigenvalues, the user's response can be inaccurate in the corresponding eigen-directions. Our algorithm does not depend on the exact knowledge about V_0, but only on a lower bound estimation of its smallest eigenvalue.

2. (Decision Rule) When facing recommendations $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$, the user makes the decision based on the following *index* which combines her estimated utility and an explorative bonus term
$$\hat{r}_i = \theta_t^\top \mathbf{a}_{i,t} + \beta_t^{(i)} \|\mathbf{a}_{i,t}\|_{V_t^{-1}}, i = \{0, 1\}, \tag{3.3}$$
where $\{\beta_t^{(0)}\}_{t\in[T]}$ and $\{\beta_t^{(1)}\}_{t\in[T]}$ are two *arbitrary* sequences satisfying $\beta_t^{(i)} \in [-c_2 t^{\gamma_2}, c_2 t^{\gamma_2}]$ for some constant c_2 and γ_2. Then, the user returns her choice $\mathbf{a}_t$ with the largest index $\hat{r}$ (breaking ties arbitrarily).

In essence, the estimation rule captures a crucial property of a learning user – the utility estimation for an item becomes more accurate only when the user has experienced more similar items before. This is reflected in the data-weighted matrix norm in (3.2). In other words, the user's response will not be reliable if the recommended item is barely related to her previously experienced items. A similar assumption is made to capture the user's explorative behaviors for previously unseen items, as described by (3.3). This is fundamentally different from classical recommendation settings, where the uncertainty in user feedback is modeled by homogeneous noise of the same scale throughout the course of user-system interactions.

Next we describe a learning user example, which is also the running example of our (more general) user behavior model. As the true underlying utility function is linear, i.e., $r_t = \theta_*^\top \mathbf{a}_t + \eta_t$, where η_t is sub-Gaussian noise, linear regression is a natural choice for a learning user's estimation rule and its estimation confidence bound satisfies $\|\theta_* - \theta_t\|_{V_t} \le O\left(\sqrt{d \log \frac{t}{\delta}}\right)$ with probability $1-\delta$ [127]. In this case, γ_1 can be any positive number and $g(\delta) = \sqrt{\log \frac{1}{\delta}}$. But our user model covers more general estimation methods than linear regression. For example, to capture the scenario where an ordinary user does not necessarily have the capacity to precisely execute such a sophisticate estimation method, we allow the user's estimation to have much larger error at the order of $O(t^{\gamma_1})$ as in (3.2), where the parameter γ_1 controls the convergence rate of user learning.

The decision rule accounts for a user's potential exploration behavior when facing uncertainty, which has been observed and supported in many studies in cognitive science [128, 129] and behavior science [130, 131]. One natural option is to follow the "optimism in the face of uncertainty" (OFUL) principle [20]. Specifically, if θ_t is the least square estimator, a learning user employing the celebrated LinUCB can be realized by setting $\beta_t^{(0)} = \beta_t^{(1)} = O(\sqrt{\log t})$ in (3.3). But our decision rule in (3.3) is, again, much more general. To capture cases where users use a much looser confidence bound estimation or even less rational arm choices, we allow $\beta_t^{(i)}$ to deviate in a much larger range with $O(t^{\gamma_2})$ (compared to $O(\sqrt{\log t})$ in LinUCB). Additionally, we allow $\{\beta_t^{(i)}\}_{t\in[T]}$ to be *arbitrary* and even consist of negative values. This enables us to model highly non-stationary user behaviors, e.g., being optimistic, pessimistic, purely myopic (when $\beta_t^1 = \beta_t^0 = 0$), or an arbitrary mixture of any of them.

Parameters $\{\gamma_1, \gamma_2\}$ depict the user learning's convergence rate and user's exploration strength, respectively. Notably, we are only interested in the regime $(\gamma_1, \gamma_2) \in [0, \frac{1}{2}) \times [0, \frac{1}{2})$, because trace$(V_t)$ is in the order of $O(t)$ by the

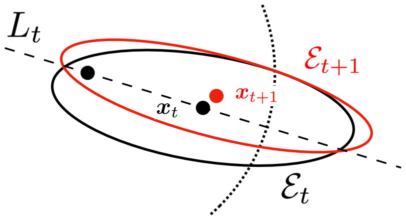


Figure 3.1: The unit ball (dashed line) is centered at the origin. L_t crosses the origin, cuts $\mathcal{E}_t$ through its center $\mathbf{x}_t$ and yields $\mathcal{E}_{t+1}$. In a high dimensional space, we have additional degree of freedom to pick L_t that shrinks $\mathcal{E}_t$ along all possible directions.

definition of V_t. Therefore, we must have $\|\theta_* - \theta_t\|_{V_t} = O(\sqrt{t})$ whenever θ_* is within a constant ℓ_2 distance to the user's estimated parameter θ_t. As a result, if $\gamma_1 \geq 1/2$, it must be that the estimated θ_t is at least a constant distance away from the true θ_*, and so is the estimated reward $\hat{r}_i$ from the expected true reward. This makes it impossible for the system to do no-regret learning. Similarly, $\|\mathbf{a}_{i,t}\|_{V_t^{-1}}$ will be $\Theta(\sqrt{t})$ for some $\mathbf{a}_{i,t}$ and a $\gamma_2 \geq 1/2$ will also make the estimated $\hat{r}_i$ arbitrarily bad. As we will demonstrate in later analysis, the estimation error of $\hat{r}$ turns out to be governed by $\max\{\gamma_1, \gamma_2\}$. Hence, for the ease of references, in the following analysis we conveniently refer to the above user behaviors as (c, γ)-*rationality*, formally defined as:

Definition 3.1.1. *[$(c, \gamma)-$rationality] Any user characterized by Estimation Rule* (3.2) *and Decision Rule* (3.3) *is said to be $(c, \gamma)-$rational if $\gamma \geq \max\{\gamma_1, \gamma_2\}, c \geq \max\{c_1, c_2\}$.*

As a concrete example, a user is (c, γ)-rational for an arbitrarily small γ if she runs LinUCB [1]. This is because under LinUCB we have $\|\theta_* - \theta_t\|_{V_t} = O(\sqrt{\log t})$ and $\{\beta_t^{(0)}, \beta_t^{(1)}\}$ are also both in the order $O(\sqrt{\log t})$. Therefore, γ here can be an arbitrarily small positive number since $\frac{\log t}{t^\gamma} \to 0$ as $t \to \infty$ for any $\gamma > 0$.

3.1.3 AES algrithm

In this section, we develop an efficient learning algorithm for the system to learn from *any* (c, γ)-rational user. The regret of our algorithm has an order of $\tilde{O}(cd^2 T^{\frac{1}{2}+\gamma})$. Recall that, a user using the LinUCB algorithm corresponds to an arbitrarily small γ. In this case, system learning essentially recovers the optimal $O(\sqrt{T})$ regret in bandit learning, despite that the system (1) only has limited comparative feedback about the user's utility estimation; and (2) faces non-stationary and non-stochastic user behaviors. More interestingly, our algorithm's regret deteriorates gracefully as $\gamma \in [0, \frac{1}{2})$ increases, i.e., as the user's learning converges at a slower rate or being more explorative as captured by γ. The key conceptual message from our theoretical findings is that it is possible for a system to learn from a learning user, and *the convergence rate of the system's learning deteriorates linearly in the convergence rate of the user's learning*.

The only caveat for our analysis is the $O(d^2)$ dependence in the regret upper bound, which is worse than the regret's linear dependence on d for standard no-regret learning problems. We believe this worse dependence is fundamentally due to the fact that the system has to learn from the users' binary feedback with diminishing yet *non-stochastic* noise. This more challenging setup invalidates classic linear contextual bandit algorithms that rely on rewards with stochastic noise. We thus develop an entirely different solution, which is a novel use of the celebrated *ellipsoid method* originally developed for solving linear programs (necessary technical details of the ellipsoid method are provided in Section 3.1.7 for curious readers) [132, 120]. Our idea is to maintain a sequence of confidence ellipsoid $\{\mathcal{E}_t\}$ for θ_* and reduce the volume of $\mathcal{E}_t$ via a carefully chosen cutting hyperplane. The user's binary comparative feedback then tells which side of the hyperplane contains the true parameter, which prepares the subsequent cuts.

Warm-up: fast learning from a perfect user To illustrate the main idea of our solution, we start with a stylized situation, where we make the following simplifications: 1). the user knows θ_* precisely and makes decisions by directly comparing $\theta_*^\top \mathbf{a}_{0,t}$ and $\theta_*^\top \mathbf{a}_{1,t}$; 2). the action set is simply the unit ball $\mathcal{A} = \{\mathbf{a} : \|\mathbf{a}\|_2 \leq 1\}$.

[1] This is also the reason for our terminology "rationality". That is, there exists (essentially) 0-rational learning users, so a γ-rational user for some $\gamma > 0$ must not be perfectly rational.

Algorithm 15 Active Ellipsoid Search on Unit Sphere

```
Input Dimension $d > 0$, number of iterations $T > 0$.
Initialization $\mathbf{x}_0 = \mathbf{0}, P_0 = I_d$
while $0 \le t \le T$ do
    Compute eigen-decomposition $P_t = \sum_{i=1}^d \sigma_i^{(t)} \mathbf{u}_i^{(t)} \mathbf{u}_i^{(t)\top}, \sigma_1^{(t)} \ge \cdots \ge \sigma_d^{(t)}$
    Compute any unit vector $\mathbf{g}_t \in \text{span}\{\mathbf{u}_1^{(t)}, \mathbf{u}_2^{(t)}\}$ that is orthogonal to $\mathbf{x}_t$;
    Pick $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t}) = (-\mathbf{g}_t, \mathbf{g}_t)$; and observe the user's choice $\mathbf{a}_{i,t}, i \in \{0, 1\}$.
    Set $\tilde{\mathbf{g}}_t = (2i-1)\mathbf{g}_t / \|\mathbf{g}_t\|_{P_t}$;
    Update $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{d+1} P_t \tilde{\mathbf{g}}_t$, and $P_{t+1} = \frac{d^2}{d^2-1}\left(P_t - \frac{2}{d+1} P_t \tilde{\mathbf{g}}_t \tilde{\mathbf{g}}_t^\top P_t\right)$.
Output: The estimation of $\theta_*$ :$\hat{\theta}_T = \mathbf{u}_1^{(T)}$.
```

Technical highlight I: novel use of the *ellipsoid method* Algorithm 15 describes our solution under this simplified problem setting. We should note Algorithm 15 differs from the classic ellipsoid method in two aspects. First, our algorithm has the freedom to *actively* choose the hyperplane L_t by picking $\{\mathbf{a}_{0,t}, \mathbf{a}_{1,t}\}$ (thus named "Active Ellipsoid Search"), while the classic ellipsoid method is always passively fed with an arbitrary separating hyperplane. Second, L_t has to cross the origin by construction. Therefore, to accelerate the shrinkage of the volume of $\mathcal{E}_t$ (i.e., Vol$(\mathcal{E}_t)$), we prefer a cutting direction $\mathbf{g}_t = \mathbf{a}_{0,t} - \mathbf{a}_{1,t}$ such that L_t goes through the center $\mathbf{x}_t$, i.e., $\mathbf{g}_t^\top \mathbf{x}_t = 0$, and Vol$(\mathcal{E}_t)$ is halved after each iteration, as illustrated in Figure 3.1.

Though given more freedom, we also face a strictly harder problem. Specifically, when solving LPs, it suffices to reach an ellipsoid $\mathcal{E}_t$ with a small volume where the LP *objective* is guaranteed to be approximately optimal. However, our goal here is to identify the *direction* of θ_* with small error, and thus a small Vol$(\mathcal{E}_t)$ is necessary but *not* sufficient. For instance, a zero-volume ellipsoid in $\mathbb{R}^d$ can still enclose a $d-1$ dimensional subspace and thus contains a very diverse set of directions that are far from θ_*.

To achieve this strictly harder objective, we need L_t to cut $\mathcal{E}_t$ along the direction in which $\mathcal{E}_t$ has the largest width, i.e., the most uncertain direction. This requires $\mathbf{g}_t$ to be aligned with the eigenvector corresponding to the largest eigenvalue of P_t, which is in general not compatible with $\mathbf{g}_t^\top \mathbf{x}_t = 0$. Here then comes the crux of our approach – we relax the second condition by picking $\mathbf{g}_t$ from a two-dimensional space spanned by the eigenvectors corresponding to the top-2 largest eigenvalues of P_t. Under this choice of $\mathbf{g}_t$, $\mathcal{E}_t$ is guaranteed to converge to a skinny-shaped ellipsoid with its longest axis converging to the direction of θ_* at an exponential rate. The detail is presented in Algorithm 15, and the convergence analysis of Algorithm 15 is formalized in the following theorem.

Theorem 3.1.2. *At each time step t in Algorithm 15, let the eigenvalues of P_t be $\sigma_1^{(t)} \ge \cdots \ge \sigma_d^{(t)}$. For any $d > 1, T > 0$, we have*

1. *for any $2 \le i \le d$,*

$$\sigma_i^{(T)} \le \exp\left(\frac{4}{d} - \frac{T}{d^2}\right), \tag{3.4}$$

2. *the ℓ_2 estimation error for θ_* is given by*

$$\left\|\theta_* - \hat{\theta}_T\right\|_2 \le 2\sqrt{d-1} \exp\left(\frac{2}{d} - \frac{T}{2d^2}\right). \tag{3.5}$$

We postpone the proof of Theorem 3.1.2 to Section 3.1.7. This theorem indicates that the ℓ_2 estimation error for θ_* converges to zero at the rate of $O\big(d^{\frac{1}{2}} \exp\left(-\frac{T}{2d^2}\right)\big)$. In other words, to guarantee $\|\theta_* - \hat{\theta}_T\|_2 < \epsilon$, at most $O(d^2 \log \frac{d}{\epsilon})$ iterations are needed.

3.1.4 RAES algrithm

The previous section illustrates our system learning principle, but under a greatly simplified setting with a perfect user. In this section, we extend the solution to account for a learning user who does not know θ_* and keeps refining her estimation θ_t. Here, the user's feedback still provides a linear inequality regarding θ^* and thus similarly serves as a cutting hyperplane. But since the user acts based on the *index* $\hat{r}_i = \theta_t^\top \mathbf{a}_{i,t} + \beta_t^{(i)} \|\mathbf{a}_{i,t}\|_{V_t^{-1}}$, the cutting hyperplane now has the form $L_t = \{\mathbf{z} : \mathbf{z}^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) = \beta_t^{(1)}\|\mathbf{a}_{1,t}\|_{V_t^{-1}} - \beta_t^{(0)}\|\mathbf{a}_{0,t}\|_{V_t^{-1}}\}$. Importantly, the intercept term now depends on $\{\beta_t^{(0)}, \beta_t^{(1)}\}$ which are arbitrary within the uncertainty region $[-ct^\gamma, ct^\gamma]$.

Technical highlight II: ellipsoid search with noise Due to the aforementioned noise in the users' binary feedback, we thus face an interesting challenge – how to perform the ellipsoid search under (non-stochastic) noisy feedback? Somewhat surprisingly, this basic question was not addressed in literature about ellipsoid method. We tackle this challenge by refining the ellipsoid method to tolerate carefully chosen scales of noise and decreasing the tolerance as the ellipsoid shrinks. In order to elicit more accurate feedback, our algorithm must ensure the diversity of the recommended items to prepare the user for improved precision of her responses in all directions. To this end, we improve Algorithm 15 by adaptively preparing the user until a desirable level of accuracy of her estimated θ_t is reached and then cut the ellipsoid. To our knowledge, this noise-robust version of ellipsoid method is novel by itself and may be of independent interest. We coin this new algorithm "Noise-robust Active Ellipsoid Search", or RAES in short.

Algorithm 16 Noise-robust Active Ellipsoid Search (RAES)

```
Input: Action set $\mathcal{A} \subset \mathbb{R}^d$ with constants $(D_1, D_0, L, \epsilon_0)$, time horizon $T_0$ and $T$, cutting threshold $k > 1$, and probability threshold $\delta > 0$
Initialization: A user who is $(c, \gamma)-$rational, $\lambda_0 > 0$ be any lower bound estimation of the minimum eigenvalue of $V_0$, set $V_0 = \lambda_0 I_d$, $\mathbf{x}_0 = \mathbf{0}$, $P_0 = I_d$.
while $0 \le t \le T$ do
    Compute eigen-decomposition
    $P_t = \sum_{i=1}^d \sigma_i^{(t)} \mathbf{u}_i^{(t)} \mathbf{u}_i^{(t)\top}, \sigma_1^{(t)} \ge \cdots \ge \sigma_d^{(t)}$.
    Compute a unit vector $\mathbf{g}_t \in \text{span}\{\mathbf{u}_1^{(t)}, \mathbf{u}_2^{(t)}\}$ that is orthogonal to $\mathbf{x}_t$;
    Pick any pair $(\bar{\mathbf{a}}_{0,t}, \bar{\mathbf{a}}_{1,t})$ such that $\bar{\mathbf{a}}_{1,t} - \bar{\mathbf{a}}_{0,t} = m\mathbf{g}_t$, $m \ge 2D_0$, and compute $\alpha_t$ according to (3.8);
    if $t \le T_0$ and $\alpha_t \ge -\frac{1}{kd}$ then
        Recommend $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$, observe the user's choice $\mathbf{a}_t = \mathbf{a}_{i,t}, i \in \{0, 1\}$;
        Set $\tilde{\mathbf{g}}_t = (2i - 1)\mathbf{g}_t / \|\mathbf{g}_t\|_{P_t}$;
        Update
            $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1 + d\alpha_t}{d + 1} P_t \tilde{\mathbf{g}}_t;$    (3.6)
            $P_{t+1} = \frac{d^2(1 - \alpha_t^2)}{d^2 - 1}\Big(P_t - \frac{2(1 + d\alpha_t)P_t\tilde{\mathbf{g}}_t\tilde{\mathbf{g}}_t^\top P_t}{(d + 1)(1 + \alpha_t)}\Big);$    (3.7)
    else if $t \le T_0$ then
        Compute $\mathbf{v}_1$ and $\mathbf{v}_d$, the two eigenvectors associated with the largest and smallest eigenvalues of $V_t$, and pick $(\bar{\mathbf{a}}_{0,t}, \bar{\mathbf{a}}_{1,t}) = D_0(\frac{4}{5}\mathbf{v}_1 \pm \frac{3}{5}\mathbf{v}_d)$;
        Recommend $(\mathbf{a}_{0,t}, \mathbf{a}_{1,t})$, observe user's choice $\mathbf{a}_t$;
        $(\mathbf{x}_{t+1}, P_{t+1}) = (\mathbf{x}_t, P_t)$;
    else
        Compute $\mathbf{a}_t = \arg\max_{\mathbf{a} \in \mathcal{A}} \mathbf{u}_1^{(t)\top}\mathbf{a}$;
        Recommend $(\mathbf{a}_t, \mathbf{a}_t)$;
        $(\mathbf{x}_{t+1}, P_{t+1}) = (\mathbf{x}_t, P_t)$;
    Update $V_{t+1} = V_t + \mathbf{a}_t\mathbf{a}_t^\top$.
```

Regularity assumptions on the action set Before introducing the RAES algorithm, we first pose several natural and technical assumptions regarding the action set $\mathcal{A} \subset \mathbb{R}^d$. Specifically, $\mathbb{B}_p^d(0, r)$ denotes the d-dimensional ℓ_p ball centered at the origin with radius r. Without loss of generality, we assume $\mathbf{0} \in \mathcal{A} \subset \mathbb{B}_2^d(0, D_1)$ since one can always shift all actions by the same amount and then re-scale the actions without changing the users' responses.

The first assumption is a familiar one, as also used in previous works such as [133].

Assumption 7 (L-Smooth Best Arm Response Condition, L-SRC)**.** *Let $\mathbf{x}_{\mathcal{A}}^* = \arg\max_{\mathbf{x}' \in \mathcal{A}} \mathbf{x}^\top\mathbf{x}', \forall \mathbf{x} \in \mathcal{A}$. There exists a constant $L > 0$ such that for any pair of non-zero unit vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have*

$$\|\mathbf{x}_{\mathcal{A}}^* - \mathbf{y}_{\mathcal{A}}^*\|_2 \le L \cdot \|\mathbf{x} - \mathbf{y}\|_2.$$

A compact set $\mathcal{A}$ satisfies L-SRC if and only if $\mathcal{A}$ can be represented as the intersection of closed balls of radius L. Intuitively, the L-SRC condition requires the boundary of $\mathcal{A}$ to have a curvature that is bounded below by a positive constant. For instance, the unit ball satisfies 1-SRC, and an ellipsoid of the form $\{\mathbf{u} \in \mathbb{R}^d : \mathbf{u}^\top P^{-1}\mathbf{u} \le 1\}$, where P is a PSD matrix, satisfies the $\frac{\lambda_{\max}(P)}{\sqrt{\lambda_{\min}(P)}}$-SRC.

Assumption 8 (ϵ-Dense Condition, ϵ-DC)**.** *$\mathcal{A}$ is an ϵ-cover of a continuous set $\bar{\mathcal{A}}$, i.e., $\bar{\mathcal{A}} \subset \cup_{\mathbf{x}\in\mathcal{A}}\mathbb{B}_2^d(\mathbf{x}, \epsilon)$. In addition, there exists constants $D_1 > D_0 > 0$ such that $\mathbb{B}_2^d(0, D_0) \subseteq \mathcal{A}, \bar{\mathcal{A}} \subseteq \mathbb{B}_2^d(0, D_1)$.*

This assumption suggests the action set $\mathcal{A}$ is sufficiently dense. A continuous $\mathcal{A}$ is 0-DC. However, ϵ-DC relaxes the continuity requirement on $\mathcal{A}$ by allowing $\mathcal{A}$ to take the form of an ϵ-net of a continuous set $\bar{\mathcal{A}}$. For convenience of references, we associate any element $\bar{\mathbf{a}} \in \bar{\mathcal{A}}$ with an element $\mathbf{a} \in \mathcal{A}$ such that $\|\mathbf{a} - \bar{\mathbf{a}}\|_2 \le \epsilon$. For our analysis, this relation does not need to be exclusive or reversible.

As indicated in the initialization of Algorithm 16, RAES does not rely on the exact values of (c, γ, V_0), which could be difficult to attain in reality. Instead, any reasonable upper bounds for c and γ, and a lower bound of $\lambda_{\min}(V_0)$ suffice. Similar to Algorithm 15, RAES also maintains a sequence of confidence ellipsoids $\{\mathcal{E}_t\}$. A hyper-parameter T_0 separates the time horizon T into two phases. At time step t, the system first proposes the most promising cutting direction $\mathbf{g}_t$. However, different from Algorithm 15 which always cuts $\mathcal{E}_t$ immediately, RAES needs to compute the cutting depth α_t (defined in (3.8)) and determine whether the user's feedback is precise enough for the system to yield an improved estimation. Intuitively, α_t measures the normalized signed distance between the center of $\mathcal{E}_t$ and the cutting hyperplane L_t: $\alpha_t \in (-\frac{1}{d}, 0)$ corresponds to a shallow-cut where L_t removes less than half of the volume of the ellipsoid; $\alpha_t \in (0, 1)$ corresponds to a deep-cut where more than half of the volume is reduced; and $\alpha_t = 0$ happens only when L_t cuts $\mathcal{E}_t$ through the center. Since we need to deal with the uncertainty in the user's response, we may only expect shallow-cuts. Depending on α_t and T_0, the system makes a decision among the following three options, which we refer to as *cut*, *exploration*, and *exploitation*:

1. (Cut) If $t \le T_0$ and $\alpha_t \ge -\frac{1}{kd}$, cut $\mathcal{E}_t$ and update $(\mathbf{x}_t, P_t)$.
2. (Exploration) If $t \le T_0$ and $\alpha_t < -\frac{1}{kd}$, make recommendations to ensure the user is exposed to the least explored directions in V_t.
3. (Exploitation) If $t > T_0$, recommend the empirically best arm to the user.

The purpose of an exploration step is to prepare the user such that a smaller α can be expected in the future. By the definition of α_t, the only way to decrease it is by increasing $\lambda_{\min}(V_t)$, which can be achieved by presenting the least exposed direction to the user [2]. Finally, when the system believes the user's estimation error of θ_* is acceptable to induce a small regret, it stops preparing the user and recommends the empirically best arm when no further cut is available. The algorithm can be understood as a phase of exploration of length T_0 followed by a phase of exploitation, with a sequence of cut steps scattered within. The sublinear regret can be guaranteed by carefully choosing T_0.

3.1.5 Regret analysis

Before analyzing RAES, we provide an intuitive explanation for it. First of all, the cutting direction $\mathbf{g}_t$ is the same as the choice in Algorithm 15, which ensures the separation hyperplane can intersect $\mathcal{E}_t$ along the most uncertain direction. Next, we translate the user's comparative feedback regarding θ_t into an inequality regarding θ_* with high probability, i.e., $\theta_*^\top \mathbf{g}_t \le (\text{or } \ge) b$, by pinning down the intersection term b. This can be realized by leveraging the property of the user's estimation and decision rules, resulting in the explicit form of α_t. To simplify the technical analysis, with a slight abuse of notation, we use the subscript t in $\{(\mathbf{x}_t, P_t)\}_{t=1}^N$ to describe the confidence ellipsoids after the t-th *cut* in RAES, and N is the total number of cuts in horizon T. Lemma 3.1.3 characterizes the effect from each cut, exploration, and exploitation step:

Lemma 3.1.3. *If we choose*

$$\alpha_t = -\frac{ct^\gamma\left(\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}} + g(\delta)\|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}}\right) + 2\epsilon_0}{\|\mathbf{g}_t\|_{P_t}} \tag{3.8}$$

[2] A straightforward way for increasing $\lambda_{\min}(V_t)$ is to feed the user with the eigenvector corresponding to $\lambda_{\min}(V_t)$. However, to avoid forcing a user to choose between two identical items (if they are not optimal), we let the system recommend two different items.

in Algorithm 16, we have

1. *After each cut,* $\text{Vol}(\mathcal{E}_{t+1}) \leq \exp\big(-\frac{(k-1)^2}{2k^2 d}\big)\text{Vol}(\mathcal{E}_t)$.
2. *If at least* d *exploration steps are taken starting from any time step* t_0 *to* t_0+n*, we have* $\lambda_{\min}(V_{n+t_0}) \geq \lambda_{\min}(V_{t_0}) + \frac{4D_0}{25} - 3\epsilon_0$.
3. *At any exploitation step* t*, the instantaneous regret is upper bounded by* $2L\|\theta_* - \mathbf{u}_1^{(t)}\|_2^2$.

Using Lemma 3.1.3, we can derive the convergence rate of $\sigma_i^{(t)}$ and the regret upper bound of RAES in the following Theorem 3.1.4, whose proof can be found in Section 3.1.7.

Theorem 3.1.4. *For any* $d > 1, n > 0$*, let* $\sigma_i^{(n)}$ *be the* i*-th largest eigenvalue of* P_n *after the* n*-th cut, we have*

1. *For any* $2 \leq i \leq d$*,*
$$\sigma_i^{(n)} \leq \exp\Big(\frac{4}{d} - \frac{(k-1)^2 n}{k^2 d^2}\Big). \tag{3.9}$$
2. *When* $T_0 = O\Big(cL^{\frac{1}{2}} D_1^{\frac{1}{2}} D_0^{-\frac{3}{2}} g(\delta) d^2 T^{\frac{1}{2}+\gamma}\Big)$ *and* $\epsilon_0 < O\big(cD_1 D_0^{-\frac{1}{2}} d^{-\frac{1}{2}} T^{-\frac{1}{4}+\frac{\gamma}{2}}\big)$*, the regret of RAES is upper bounded by* $O\Big(cL^{\frac{1}{2}} D_1^{\frac{3}{2}} D_0^{-\frac{3}{2}} g(\frac{\delta}{T_0}) d^2 T^{\frac{1}{2}+\gamma}\Big)$ *with probability* $1-\delta$.

Theorem 3.1.4 suggests when $\mathcal{A}$ is continuous or sufficiently dense, RAES achieves a regret upper bound $\tilde{O}(cd^2 T^{\frac{1}{2}+\gamma})$ when $g(\frac{\delta}{T_0})$ grows logarithmically in T_0. Recall that $\gamma \in [0, \frac{1}{2})$ denotes the rationality of the user: when γ is large, the system obtains less accurate responses from the user and thus suffers from a worse regret guarantee. When $\gamma = 0$, e.g., the user executes LinUCB, we get an upper bound of the order $\tilde{O}(\sqrt{T})$, which nearly matches the lower bound, as we will show in the following section.

We conclude this technical section by showing a regret lower bound for the system's learning. This lower bound applies for any $\gamma > 0$, and it nearly matches the above upper bound w.r.t. time horizon T when γ is close to zero. This result leaves an intriguing open question about how tight our Algorithm 16 is for general γ, i.e., for every $\gamma \in (0, 1/2)$, what is the best possible regret for the system? We remark that resolving this open question appears to require significantly different machinaries as used in current lower bound proofs for bandit algorithms since these arguments are primarily based on information theory and thus intrinsically rely on assumption of *random* noises [127, 133], whereas the user's feedback noise in our model is arbitrary (though also diminishing with more rounds). We thus leave this as an interesting future direction to explore.

Theorem 3.1.5. *For any* $\gamma > 0$*, there exists a function* $T_0(d) > 0$ *such that for any* $d \geq 1$*,* $T > T_0(d)$*, and any algorithm* $\mathcal{G}$ *that has merely access to the comparison feedback given by a rational user defined in Definition 3.1.1, there exists* $\theta_* \in \partial\mathbb{B}_1^d$ *such that the expected regret* R_T *defined in Eq (3.1) obtained by* $\mathcal{G}$ *satisfies*

$$R_T^{(s)}(\mathcal{G}, \theta_*) \geq \frac{\exp(-2)}{4}(d-1)\sqrt{T}. \tag{3.10}$$

Theorem 3.1.5 may appear not surprising since, intuitively, the system's learning task appears no easier than the standard stochastic linear bandit problems for which the lower bound is already $O(\sqrt{T})$ [133]. However, it turns out that delivering a rigorous proof is more subtle than this intuition, and for that we have to overcome two technical challenges: 1). adapting the current minimax lower bound proof for stochastic linear bandits to the setup where the norm of θ_* is bounded away from zero; 2). constructing a black-box reduction from the system's regret to the user's regret. Due to the space limit, we defer the proof details to Section 3.1.7.

3.1.6 Experiment setup & results

In this section, we study the empirical performance of RAES to validate our theoretical analysis by running simulations on synthetic datasets in comparison with several baselines.

There is no direct baseline for comparison since the learning environment we studied is new. Given the linear reward and the binary comparative feedback assumptions, we take several contextual dueling bandit algorithms for comparison, including Dueling Bandit Gradient Descent (DBGD) [117], Doubler [118], and Sparring [118, 134].

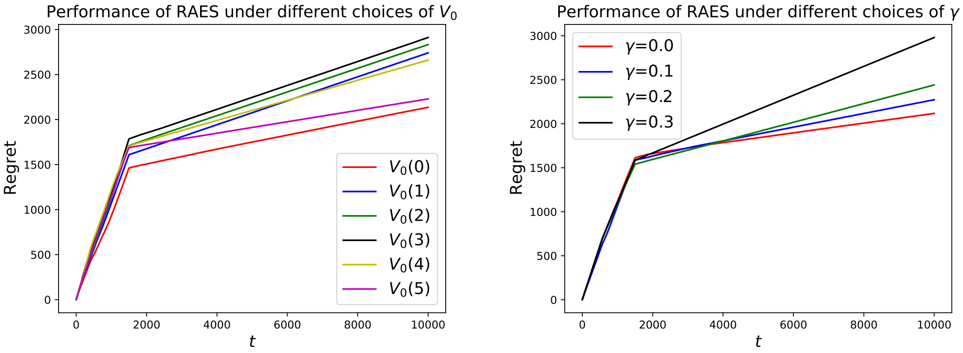


Figure 3.2: The regret of RAES against a learning user with different V_0 and γ over time. Left: Fix $\gamma = 0.1$, plot for different choices of V_0; Right: Fix $V_0 = I_d$, plot for different choices of γ.

Configuration of baseline algorithms DBGD [117] maintains the currently best candidate $\mathbf{a}_t$ and compares it with a neighboring point $\mathbf{a}_t + \eta \mathbf{u}_t$ along a random direction $\mathbf{u}_t$. An update is taken when the proposed point wins the comparison. DBGD works for continuous convex action set and has a regret guarantee of $O(T^{3/4})$. Although its theoretical guarantee only holds under a strictly concave utility function, it can be reasonably adapted to our problem setting empirically. DBGD's hyper-parameters include the starting point $\mathbf{w}_0$, and two learning rates δ, γ that control the step-lengths for proposing new points and update the current points, respectively. In the experiment, these hyper-parameters are set to $(\mathbf{w}_0, \delta, \gamma) = (\mathbf{0}, d^{-\frac{1}{2}} T^{-\frac{1}{4}}, T^{-\frac{1}{2}})$, as recommended in [117].

Doubler [118] is the first approach that converts a dueling bandit problem into a conventional multi-armed bandit (MAB) problem. Doubler proceeds in epochs of exponentially increasing size: in each epoch, the left arm is sampled from a fixed distribution, and the right arm is chosen using an MAB algorithm to minimize regret against the left arm. The feedback received by the MAB algorithm is the number of wins the right arm encounters when compared against the left arm. Doubler is proved to have $\tilde{O}(T^{1/2})$ regret for continuous action set under the linear reward assumption. The black-box MAB algorithm that is needed to initiate Doubler is set to the OFUL algorithm in [20].

Sparring [118, 134] is also a general reduction from dueling bandit to MAB. Like Doubler, it also requires black-box calls to an MAB algorithm and achieves regret of the same order as the MAB algorithm. Instead of comparing with a fixed distribution, Sparring initializes two MAB instances and lets them "spar" against each other. As a heuristic improvement of Doubler, Sparring does not have a regret upper bound guarantee but is reported to enjoy a better performance compared to Doubler [118]. The black-box MAB algorithm that is needed to initiate Sparring is set to the OFUL algorithm in [20].

Simulation environment and metrics In all experiments, we fix the action set $\mathcal{A} = \mathbb{B}_2^d(0,1)$, i.e., $D_0 = D_1 = 1$, and $\delta = 0.1, k = 1.05$. We consider a $(1,\gamma)$-rational user with $\gamma \in \{0, 0.2\}$ and prior knowledge matrix V_0. The user's decision sequence $\{\beta_t^{(0)}\}$ and $\{\beta_t^{(1)}\}$ are independently drawn from $[-t^\gamma, t^\gamma]$. The ground-truth parameter θ_* is sampled from $\partial\mathbb{B}_2^d(0,1)$ and the reported results are collected from the same problem instance and averaged over 10 independent runs.

Robustness of RAES against a learning user We first demonstrate the performance of RAES under $(T, T_0, d) = (10000, 1500, 5)$ against a $(1,\gamma)$-rational user with different γ and V_0 in Figure 3.2. The x-axis denotes time step t and y-axis denotes the accumulated regret up to the time step t. The left panel illustrates the performance of RAES when $\gamma = 0.1$ and $V_0 \in \{V_0(i) : 0 \leq i \leq 5\}$, where $V_0(i)$ is the diagonal matrix with i diagonal entries being 1 while other $5 - i$ entries being 100. Unsurprisingly, RAES achieves the best performance when the user has the most informative prior $V_0(0)$. When V_0 has small eigenvalues, RAES needs more exploration steps in the first T_0 rounds, but the resulting added regret is not significant. The right panel shows the result when $V_0 = I_d$ and $\gamma \in \{0, 0.1, 0.2, 0.3\}$ which confirms our theoretical analysis that the regret of RAES grows in order $O(T^{\frac{1}{2}+\gamma})$.

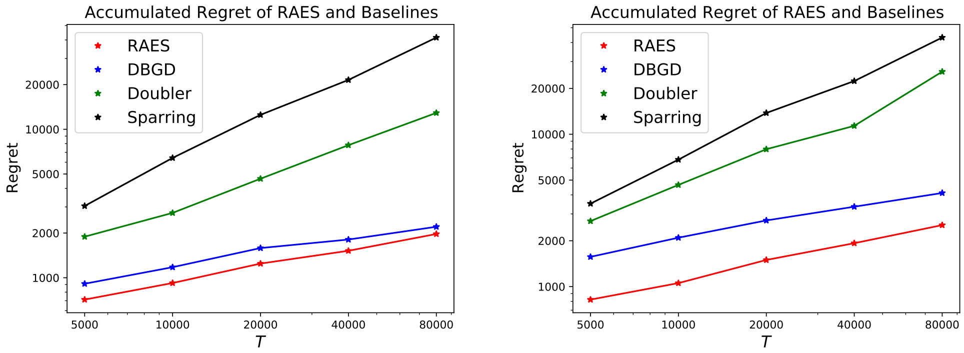


Figure 3.3: The accumulated regret of RAES and three baseline algorithms. Different colors specify different algorithms. Each star represents the accumulated regret (y-axis) of the algorithm given time horizon T (x-axis) with $\gamma = 0$. Left: $V_0 = 100I_d$; right: $V_0 = \text{diag}(100, 10, 5, 2, 1)$.

Comparison with baseline algorithms The comparison between RAES and the three baselines against learning users are shown in Figure 3.3, where the x-axis denotes different time horizons T, and the y-axis denotes the corresponding accumulated regret. $\{\gamma, T_0\}$ are set to 0 and $0.25 \times d^2\sqrt{T}$. The left panel shows the result with $V_0 = 100I_d$, i.e., each algorithm is facing a well-prepared user, while the right panel is plotted with $V_0 = \text{diag}(100, 20, 5, 2, 1)$. The result demonstrates that RAES enjoys the best performance and is robust against different types of learning users. Since Doubler and Sparring employ a black-box linear bandit algorithm as their subroutine, the violation of the stochastic reward assumption breaks down the linear bandit algorithm and thus the failure of the algorithms themselves. For DBGD, the left panel suggests that it can still enjoy a sub-linear regret under milder users' rationality assumptions. However, when the user's prior V_0 is ill-posed (i.e., $\lambda_{\min}(V_0)$ is small), the performance of DBGD deteriorates seriously. In particular, under an ill-posed V_0, the user's feedback can be misleading along certain directions, and the design of DBGD does not provide any mechanism to increase the accuracy of user feedback along these directions.

3.1.7 Full proof of AES and RAES algorithm

Preliminaries on ellipsoid method

A $d \times d$ matrix A is symmetric when $A = A^\top$, and any symmetric matrix A admits an eigenvalue decomposition $A = U\Sigma U^\top$, where U is a orthogonal matrix and $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_d)$ is a diagonal matrix with diagonal elements $\sigma_1 \geq \cdots \geq \sigma_d$. We refer to $\sigma_i(A)$ as the i-th largest eigenvalue of A. A symmetric matrix A is called positive definite (PD) if all its eigenvalues are strictly positive.

$$\{\mathbf{g}^\top(\mathbf{z}-\mathbf{x}) \leq b\} \cap \mathcal{E}'(\mathbf{x}', P')$$

An ellipsoid is a subset of $\mathbb{R}^d$ defined as

$$\mathcal{E}(\mathbf{x}, P) = \{\mathbf{z} | (\mathbf{z}-\mathbf{x})^\top P^{-1}(\mathbf{z}-\mathbf{x}) \leq 1\},$$

where $\mathbf{x} \in \mathbb{R}^d$ specifies its center and the PD matrix P specifies its geometric shape. Each of the d radii of $\mathcal{E}(\mathbf{x}, P)$ corresponds to the square root of an eigenvalue of P and the volume of the ellipsoid is given by

$$\text{Vol}(\mathcal{E}(\mathbf{x}, P)) = V_d\sqrt{\det P} = V_d\sqrt{\prod_{i=1}^{d}\sigma_d},$$

where V_d is a constant that represents the volume of the unit ball in $\mathbb{R}^d$. If a hyperplane $\mathbf{g}^\top(\mathbf{z}-\mathbf{x}) = b$ with normal direction $\mathbf{g}$ and intersection b cuts the ellipsoid $\mathcal{E}(\mathbf{x}, P)$ to two pieces, the smallest ellipsoid containing the area $\{\mathbf{g}^\top(\mathbf{z}-\mathbf{x}) \leq b\} \cap \mathcal{E}(\mathbf{x}, P)$ can be captured by $\mathcal{E}'(\mathbf{x}', P')$, where the new center $\mathbf{x}'$ and the shape matrix P' can be computed via the following closed form formula:

$$\mathbf{x}' = \mathbf{x} - \frac{1+d\alpha}{d+1} P\tilde{\mathbf{g}}, \tag{3.11}$$

$$P' = \frac{d^2(1-\alpha^2)}{d^2-1}\Big(P - \frac{2(1+d\alpha)}{(d+1)(1+\alpha)} P\tilde{\mathbf{g}}\tilde{\mathbf{g}}^\top P\Big), \tag{3.12}$$

$$\alpha = -\frac{b}{\sqrt{\mathbf{g}^\top P\mathbf{g}}}, \tag{3.13}$$

$$\tilde{\mathbf{g}} = \frac{1}{\sqrt{\mathbf{g}^\top P\mathbf{g}}}\mathbf{g}, \tag{3.14}$$

where α represents the cutting-depth which we will elaborate on later. To narrow down the feasible region of the target parameters, it is desirable to let Vol$(\mathcal{E}')$ as small as possible. At least, we need to ensure that Vol$(\mathcal{E}') <$ Vol$(\mathcal{E})$. Basic algebraic calculation shows that

$$\frac{\text{Vol}(\mathcal{E}^{'})}{\text{Vol}(\mathcal{E})} = \sqrt{\frac{\det P'}{\det P}} = \Big(\frac{d^2(1-\alpha^2)}{d^2-1}\Big)^{\frac{d}{2}}\Big(1 - \frac{2(1+d\alpha)}{(d+1)(1+\alpha)}\Big)^{\frac{1}{2}} \tag{3.15}$$

$$= \Big(1 + \frac{1+d\alpha}{d-1}\Big)^{\frac{d-1}{2}}\Big(1 - \frac{1+d\alpha}{d+1}\Big)^{\frac{d+1}{2}}$$

$$= \Big(\frac{d(1+\alpha)}{d-1}\Big)^{\frac{d-1}{2}}\Big(\frac{d(1-\alpha)}{d+1}\Big)^{\frac{d+1}{2}}, \tag{3.16}$$

where Eq (3.15) is from Eq (3.12) and the fact that $\det(P - \beta\mathbf{v}\mathbf{v}^\top) = (1-\beta\|\mathbf{v}\|_P^2)\det(P)$. Eq (3.16) indicates that Vol$(\mathcal{E}') <$ Vol$(\mathcal{E})$ if and only if $\alpha \in (-\frac{1}{d}, 1)$. The quantity α serves as an indicator of the "depth" of the cut: $\alpha \in (-\frac{1}{d}, 0)$ corresponds to a shallow-cut where the proposed cutting hyperplane removes less than half of the volume of the ellipsoid; $\alpha \in (0,1)$ corresponds to a deep-cut where more than half of the volume is removed. And $\alpha = 0$ happens only when $b = 0$, meaning the cutting hyperplane goes through the center $\mathbf{x}$ and exactly half of the volume is removed. In our problem setting, since we need to deal with the uncertainty in the user's response, we may only expect shallow-cuts. In addition, from Eq (3.16) we can show that for any $-\frac{1}{d} < \alpha < 1$,

$$\frac{\text{Vol}(\mathcal{E}^{'})}{\text{Vol}(\mathcal{E})} \leq \exp\Big(-\frac{(1+d\alpha)^2}{2d}\Big). \tag{3.17}$$

Omitted proof in Section 3.1.3

To prove Theorem 3.1.2, we need the following technical lemmas. Lemma 3.1.6 states that the product of the largest two eigenvalues of P_t must shrink w.r.t. a constant factor after each cut. Since $\det(P_t)$ approaches zero at an exponential rate (from Eq (3.17)), P_t can only have one potentially large eigenvalue while all other eigenvalues must approach zero. Lemma 3.1.7 implies that at any time step t, the "gap" between P_t's second-largest eigenvalue and the smallest eigenvalue can be upper bounded by a constant. Given that the determinant of P_t converges to 0 at an exponential rate, all the eigenvalues of P_t except the largest one must also converge to 0 exponentially fast.

Lemma 3.1.6. *In Algorithm 15, let the eigenvalues of P_t be $\sigma_1 \geq \cdots \geq \sigma_d$ and the eigenvalues of P_{t+1} be $\{\sigma'_1, \cdots, \sigma'_d\}$. Then we have*

1. *for any $3 \leq i \leq d$, we have equalities*

$$\sigma'_i = \frac{d^2}{d^2-1}\sigma_i.$$

2. *for σ'_1, σ'_2, we have $\frac{\sigma'_1\sigma'_2}{\sigma_1\sigma_2} = \frac{d^4}{(d+1)^3(d-1)} < 1$ and the following bound*

$$\max\{\sigma'_1, \sigma'_2\} \in [\frac{d^2}{(d+1)^2}\sigma_1, \frac{d^2}{d^2-1}\sigma_1], \tag{3.18}$$

$$\min\{\sigma_1', \sigma_2'\} \in [\frac{d^2}{(d+1)^2}\sigma_2, \frac{d^2}{d^2-1}\sigma_2]. \tag{3.19}$$

Proof. **Claim 1.** Suppose $P_t = U\Sigma U^\top$, where $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_d)$ and $U = [\mathbf{u}_1, \cdots, \mathbf{u}_d]$. From the update rule of P_{t+1}, for any $3 \le i \le d$ we have

$$\begin{aligned} P_{t+1}\mathbf{u}_i &= \frac{d^2}{d^2-1}\Big(P_t - \frac{2}{d+1}P_t\tilde{\mathbf{g}}_t\tilde{\mathbf{g}}_t^\top P_t\Big)\mathbf{u}_i \\ &= \frac{d^2}{d^2-1}\sigma_i\mathbf{u}_i - \frac{d^2}{d^2-1}\cdot\frac{2\sigma_i}{d+1}P_t\tilde{\mathbf{g}}_t(\tilde{\mathbf{g}}_t^\top\mathbf{u}_i) \\ &= \frac{d^2}{d^2-1}\sigma_i\mathbf{u}_i, \end{aligned} \tag{3.20}$$

where Eq (3.20) holds because $\tilde{\mathbf{g}}_t \in \mathrm{span}\{\mathbf{u}_1, \mathbf{u}_2\}$. Therefore, $\{\frac{d^2}{d^2-1}\sigma_i\}_{i=3}^d$ are $d-2$ eigenvalues of P_{t+1}.

Claim 2. By the choice of $\mathbf{g}_t$, the cutting hyper plane always goes through $\mathbf{x}_t$ (i.e., $\alpha = 0$). Therefore, by Eq (3.17) we obtain $\frac{\prod_{i=1}^d \sigma_i'}{\prod_{i=1}^d \sigma_i} = \frac{d^2}{(d+1)^2}\cdot\left(\frac{d^2}{d^2-1}\right)^{d-1}$. Consider Eq (3.20), we conclude that the remaining two eigenvalues of P_{t+1} satisfy

$$\frac{\sigma_1'\sigma_2'}{\sigma_1\sigma_2} = \frac{d^2}{(d+1)^2}\cdot\frac{d^2}{d^2-1} = \frac{d^4}{(d+1)^3(d-1)} < 1. \tag{3.21}$$

Next we derive the bound for σ_1', σ_2'. Let $\mathbf{g}_t = p\mathbf{u}_1 + q\mathbf{u}_2$, and

$$P_t\tilde{\mathbf{g}}_t = \frac{p\sigma_1}{\sqrt{p^2\sigma_1 + q^2\sigma_2}}\mathbf{u}_1 + \frac{q\sigma_2}{\sqrt{p^2\sigma_1 + q^2\sigma_2}}\mathbf{u}_2 \triangleq v_1\mathbf{u}_1 + v_2\mathbf{u}_2.$$

It is easy to see that $\frac{d^2-1}{d^2}\sigma_1'$, $\frac{d^2-1}{d^2}\sigma_2'$ are the two eigenvalues of the following 2×2 matrix

$$A = \begin{bmatrix}\sigma_1 & 0 \\ 0 & \sigma_2\end{bmatrix} - \frac{2}{d+1}\begin{bmatrix}v_1 \\ v_2\end{bmatrix}\cdot\begin{bmatrix}v_1 & v_2\end{bmatrix}. \tag{3.22}$$

Without loss of generality, we assume $\sigma_1' \ge \sigma_2'$. Applying Weyl's inequality in matrix theory [135, 136] to matrix A yields

$$\sigma_1 \ge \frac{d^2-1}{d^2}\sigma_1' \ge \sigma_2 \ge \frac{d^2-1}{d^2}\sigma_2'. \tag{3.23}$$

On the other hand, from Eq (3.21) we also have

$$\frac{\sigma_1'}{\sigma_1} = \frac{d^4}{(d+1)^3(d-1)}\frac{\sigma_2}{\sigma_2'} \ge \frac{d^4}{(d+1)^3(d-1)}\cdot\frac{d^2-1}{d^2} = \frac{d^2}{(d+1)^2}, \tag{3.24}$$

$$\frac{\sigma_2'}{\sigma_2} = \frac{d^4}{(d+1)^3(d-1)}\frac{\sigma_1}{\sigma_1'} \ge \frac{d^4}{(d+1)^3(d-1)}\cdot\frac{d^2-1}{d^2} = \frac{d^2}{(d+1)^2}. \tag{3.25}$$

From Eq (3.23), (3.24), (3.25), we obtain Eq (3.18), (3.19) and therefore complete the proof. □

Lemma 3.1.7. *At each time step t in Algorithm 15, let the eigenvalue of P_t be $\sigma_1^{(t)} \ge \cdots \ge \sigma_d^{(t)}$. Further let $D_t = \sigma_2^{(t)}/\sigma_d^{(t)}$, we claim*

1. *for any $t \ge 0$, $D_{t+1} \le \frac{d+1}{d-1}\cdot D_t$;*

2. *if $D_t > \frac{d+1}{d-1}$, $D_{t+1} \le D_t$.*

3. *for any $n \ge 0$,*

$$\max_{0\le t\le n} D_t \le \left(\frac{d+1}{d-1}\right)^2. \tag{3.26}$$

Proof. From Lemma 3.1.6, we know that the eigenvalues of P_{t+1} is $\{\sigma'_1, \sigma'_2, \frac{d^2}{d^2-1}\sigma_3^{(t)}, \cdots, \frac{d^2}{d^2-1}\sigma_d^{(t)}\}$, where $\sigma'_1 \geq \sigma'_2$ and

$$\frac{d^2}{(d+1)^2}\sigma_2^{(t)} \leq \sigma'_2 \leq \frac{d^2}{d^2-1}\sigma_2^{(t)} \tag{3.27}$$

Claim 1. Because $\sigma'_1 \geq \sigma'_2, \sigma_3^{(t)} \geq \cdots \geq \sigma_d^{(t)}$, and note that $\sigma_2^{(t+1)}$ and $\sigma_d^{(t+1)}$ are the second-largest element and the smallest element of $\{\sigma'_1, \sigma'_2, \frac{d^2}{d^2-1}\sigma_3^{(t)}, \cdots, \frac{d^2}{d^2-1}\sigma_d^{(t)}\}$, the value of $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)})$ must satisfy one of the following situation:

1. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\sigma'_2, \frac{d^2}{d^2-1}\sigma_d^{(t)})$, from Eq (3.27) we have

$$\frac{D_{t+1}}{D_t} = \frac{d^2-1}{d^2} \cdot \frac{\sigma'_2}{\sigma_2} \leq 1. \tag{3.28}$$

2. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\frac{d^2}{d^2-1}\sigma_i^{(t)}, \frac{d^2}{d^2-1}\sigma_d^{(t)})$ for some $3 \leq i \leq d-1$, we have

$$\frac{D_{t+1}}{D_t} = \frac{\sigma_i^{(t)}/\sigma_d^{(t)}}{\sigma_2^{(t)}/\sigma_d^{(t)}} \leq 1. \tag{3.29}$$

3. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\frac{d^2}{d^2-1}\sigma_i^{(t)}, \sigma'_2)$ for some $3 \leq i \leq d-1$, from Eq (3.27) we have

$$\frac{D_{t+1}}{D_t} = \frac{d^2}{d^2-1} \cdot \frac{\sigma_i^{(t)}}{\sigma'_2} \cdot \frac{\sigma_d^{(t)}}{\sigma_2^{(t)}} \leq \frac{d^2}{d^2-1} \cdot \frac{\sigma_2^{(t)}}{\sigma'_2} \leq \frac{d^2}{d^2-1} \cdot \frac{(d+1)^2}{d^2} = \frac{d+1}{d-1}. \tag{3.30}$$

By Eq (3.28), (3.29), (3.30), the first claim holds.
Claim 2. It suffices to show that the situation (3) cannot happen when $D_t > \frac{d+1}{d-1}$. In fact, when $D_t > \frac{d+1}{d-1}$, from Eq (3.27) we have

$$\sigma'_2 \geq \frac{d^2}{(d+1)^2}\sigma_2^{(t)} = \frac{d^2}{(d+1)^2}\sigma_d^{(t)} D_t > \frac{d^2}{(d+1)^2} \cdot \frac{d+1}{d-1} \cdot \sigma_d^{(t)} = \frac{d^2}{d^2-1}\sigma_d^{(t)},$$

meaning σ'_2 cannot be the smallest eigenvalue of P_{t+1}. As a result, the second claim holds by Eq (3.28), (3.29).
Claim 3. We prove Eq (3.26) by contradiction. Let n_0 be the smallest index in set $\arg\max_{0 \leq t \leq n} D_t$. If $n_0 = 0$, we have $\max_{0 \leq t \leq n} D_t = D_0 = 1 < \left(\frac{d+1}{d-1}\right)^2$. Now consider the case $n_0 \geq 1$ and suppose $D_{n_0} > \left(\frac{d+1}{d-1}\right)^2$. By Claim 1, we have $D_{n_0-1} \geq \frac{d-1}{d+1} D_{n_0} > \frac{d+1}{d-1}$. Apply Claim 2 to D_{n_0-1}, we obtain $D_{n_0} \leq D_{n_0-1}$, which contradicts the definition of n_0. Hence, Claim 3 holds. □

Now we are ready to present the proof of the convergence theorem for Algorithm 15:

Theorem 3.1.8. *At each time step t in Algorithm 15, let the eigenvalues of P_t be $\sigma_1^{(t)} \geq \cdots \geq \sigma_d^{(t)}$. For any $d > 1, T > 0$, we have*

1. *for any $2 \leq i \leq d$,*

$$\sigma_i^{(T)} \leq \exp\left(\frac{4}{d} - \frac{T}{d^2}\right). \tag{3.31}$$

2. *the ℓ_2 estimation error for θ_* is given by*

$$\left\|\theta_* - \hat{\theta}_T\right\|_2 \leq 2\sqrt{d-1}\exp\left(\frac{2}{d} - \frac{T}{2d^2}\right), \tag{3.32}$$

Proof. Since the depth of the cut $\alpha = 0$ through out the execution of Algorithm 15, from Eq (3.17) we have

$$\prod_{i=1}^{d} \sigma_i^{(T)} = \frac{\det P_n}{\det P_0} \le \exp\Big(-\frac{T}{d}\Big). \tag{3.33}$$

From Lemma 3.1.7, we have $\sigma_i^{(T)} \ge \sigma_d^{(T)} \ge \big(\frac{d-1}{d+1}\big)^2 \cdot \sigma_2^{(n)}, \forall 3 \le i \le d$. Therefore,

$$\begin{aligned}\exp\Big(-\frac{T}{d}\Big) &\ge \prod_{i=1}^{d} \sigma_i^{(T)} \\ &\ge \sigma_2^{(T)} \cdot \sigma_2^{(T)} \cdot \Big[\big(\frac{d-1}{d+1}\big)^2 \cdot \sigma_2^{(T)}\Big]^{d-2} \\ &= [\sigma_2^{(T)}]^d \cdot \Big(1 - \frac{2}{d+1}\Big)^{2d-4} \\ &\ge \exp(-4) \cdot [\sigma_2^{(T)}]^d.\end{aligned}$$

Rearranging terms yields $\sigma_2^{(T)} \le \exp\Big(\frac{4}{d} - \frac{T}{d^2}\Big)$, and thus $\sigma_i^{(T)} \le \exp\Big(\frac{4}{d} - \frac{T}{d^2}\Big), \forall 2 \le i \le d$.

Let $\langle \mathbf{x}, \mathbf{y}\rangle = \arccos\big(\frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\cdot\|\mathbf{y}\|}\big)$ denote the included angle between vector $\mathbf{x}$ and $\mathbf{y}$, now we are prepared to upper bound the directional estimation error $\sin\langle\hat{\theta}_T, \theta_*\rangle$. First of all, note that $\theta_*, \mathbf{0} \in \mathcal{E}_T$ for any $n \ge 0$, meaning there exists $\{(p_i, q_i)\}_{i=1}^d$ such that

$$\theta_* = \mathbf{x}_T + \sum_{i=1}^{d} p_i \mathbf{u}_i^{(T)}, \sum_{i=1}^{d} \frac{p_i^2}{\sigma_i^{(T)}} \le 1. \tag{3.34}$$

$$\mathbf{0} = \mathbf{x}_T + \sum_{i=1}^{d} q_i \mathbf{u}_i^{(T)}, \sum_{i=1}^{d} \frac{q_i^2}{\sigma_i^{(T)}} \le 1. \tag{3.35}$$

As a result, $\theta_* = \sum_{i=1}^d (p_i - q_i)\mathbf{u}_i^{(T)}$, and $p_i, q_i \le \sqrt{\sigma_i^{(T)}}, 2 \le i \le d$. Therefore,

$$\begin{aligned}\sin\langle\theta_*, \hat{\theta}_T\rangle = \sqrt{1 - \cos^2\langle\theta_*, \hat{\theta}_T\rangle} = \sqrt{1 - \frac{(\theta_*^\top \mathbf{u}_1)^2}{\|\theta_*\|_2^2}} = \frac{1}{\|\theta_*\|_2} \cdot \sqrt{\sum_{i=2}^{d} (p_i - q_i)^2} \\ \le \frac{2}{\|\theta_*\|_2} \cdot \sqrt{\sum_{i=2}^{d} \sigma_i^{(T)}},\end{aligned}$$

Now we know that the directional inference error for θ_* converges to zero at rate $O\big(d^{\frac{1}{2}} \exp(-\frac{T}{2d^2})\big)$. When the system knows $\|\theta_*\|_2 = 1$, the ℓ_2 estimation error for θ_* can be obtained from

$$\begin{aligned}\Big\|\theta_* - \|\theta_*\|_2 \cdot \frac{\hat{\theta}_T}{\|\hat{\theta}_T\|_2}\Big\|_2 &\le 2\|\hat{\theta}_T\|_2 \sin(\langle\theta_*, \hat{\theta}_T\rangle/2) \\ &\le 2\sqrt{\sum_{i=2}^{d} \sigma_i^{(T)}}\end{aligned} \tag{3.36}$$

where the last inequality holds because $\sin x \le x, \forall x > 0$. In particular, plugin Eq (3.4) into the R.H.S. of Eq (3.36), we obtain Eq (3.5).

□

Proof of Theorem 3.1.4 in Section 3.1.5

The following Lemma 3.1.9 and 3.1.10 are used in the proof of Theorem 3.1.4. Lemma 3.1.9 and 3.1.10 are generalizations of Lemma 3.1.6 and 3.1.7 under arbitrary cutting depth α_t.

Lemma 3.1.9. *In Algorithm 16, suppose a valid cut is executed at step t with depth $-\frac{1}{kd} \le \alpha_t \le 0$. Let the eigenvalues of P_t be $\sigma_1 \ge \cdots \ge \sigma_d$ and the eigenvalues of P_{t+1} be $\{\sigma_1', \cdots, \sigma_d'\}$. Then we have*

1. *for any $3 \le i \le d$, we have equalities*
$$\sigma_i' = \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i.$$

2. *for σ_1', σ_2', we have $\frac{\sigma_1'\sigma_2'}{\sigma_1\sigma_2} = \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)} < 1$ and the following bound*

$$\max\{\sigma_1', \sigma_2'\} \in [\frac{d^2(1-\alpha_t)^2}{(d+1)^2}\sigma_1, \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_1], \tag{3.37}$$

$$\min\{\sigma_1', \sigma_2'\} \in [\frac{d^2(1-\alpha_t)^2}{(d+1)^2}\sigma_2, \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_2]. \tag{3.38}$$

Proof. **Claim 1.** Suppose $P_t = U\Sigma U^\top$, where $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_d)$ and $U = [\mathbf{u}_1, \cdots, \mathbf{u}_d]$. From the update rule of P_{t+1}, for any $3 \le i \le d$ we have

$$\begin{aligned} P_{t+1}\mathbf{u}_i &= \frac{d^2(1-\alpha_t^2)}{d^2-1}\Big(P_t - \frac{2(1+d\alpha_t)}{(d+1)(1+\alpha_t)}P_t\tilde{\mathbf{g}}_t\tilde{\mathbf{g}}_t^\top P_t\Big)\mathbf{u}_i \\ &= \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i\mathbf{u}_i - \frac{d^2(1-\alpha_t^2)}{d^2-1} \cdot \frac{2(1+d\alpha_t)\sigma_i}{(d+1)(1+\alpha_t)}P_t\tilde{\mathbf{g}}_t(\tilde{\mathbf{g}}_t^\top\mathbf{u}_i) \\ &= \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i\mathbf{u}_i, \end{aligned} \tag{3.39}$$

where Eq (3.39) holds because $\tilde{\mathbf{g}}_t \in \mathrm{span}\{\mathbf{u}_1, \mathbf{u}_2\}$. Therefore, $\{\frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i\}_{i=3}^d$ constitute $d-2$ eigenvalues of P_{t+1}.

Claim 2. From Eq (3.17) we have $\frac{\prod_{i=1}^d \sigma_i'}{\prod_{i=1}^d \sigma_i} = \frac{d^2(1-\alpha_t)^2}{(d+1)^2} \cdot \left(\frac{d^2(1-\alpha_t^2)}{d^2-1}\right)^{d-1}$. Consider Eq (3.39), we conclude that the remaining two eigenvalues of P_{t+1} satisfy

$$\frac{\sigma_1'\sigma_2'}{\sigma_1\sigma_2} = \frac{d^2(1-\alpha_t)^2}{(d+1)^2} \cdot \frac{d^2(1-\alpha_t^2)}{d^2-1} = \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)} < 1. \tag{3.40}$$

Next we derive the bound for σ_1', σ_2'. Let $\mathbf{g}_t = p\mathbf{u}_1 + q\mathbf{u}_2$, and

$$P_t\tilde{\mathbf{g}}_t = \frac{p\sigma_1}{\sqrt{p^2\sigma_1 + q^2\sigma_2}}\mathbf{u}_1 + \frac{q\sigma_2}{\sqrt{p^2\sigma_1 + q^2\sigma_2}}\mathbf{u}_2 \triangleq v_1\mathbf{u}_1 + v_2\mathbf{u}_2.$$

It is easy to see that $\frac{d^2-1}{d^2(1-\alpha_t^2)}\sigma_1'$, $\frac{d^2-1}{d^2(1-\alpha_t^2)}\sigma_2'$ are the two eigenvalues of the following 2×2 matrix

$$A = \begin{bmatrix}\sigma_1 & 0 \\ 0 & \sigma_2\end{bmatrix} - \frac{2(1+d\alpha_t)}{(d+1)(1+\alpha_t)}\begin{bmatrix}v_1 \\ v_2\end{bmatrix} \cdot \begin{bmatrix}v_1 & v_2\end{bmatrix}. \tag{3.41}$$

Without loss of generality, we assume $\sigma_1' \ge \sigma_2'$. Applying Weyl's inequality in matrix theory [135, 136] to matrix A yields

$$\sigma_1 \ge \frac{d^2-1}{d^2(1-\alpha_t^2)}\sigma_1' \ge \sigma_2 \ge \frac{d^2-1}{d^2(1-\alpha_t^2)}\sigma_2'. \tag{3.42}$$

On the other hand, from Eq (3.40) we also have

$$\frac{\sigma_1'}{\sigma_1} = \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)}\frac{\sigma_2}{\sigma_2'} \geq \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)} \cdot \frac{d^2-1}{d^2(1-\alpha_t^2)} = \frac{d^2(1-\alpha_t)^2}{(d+1)^2}, \tag{3.43}$$

$$\frac{\sigma_2'}{\sigma_2} = \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)}\frac{\sigma_1}{\sigma_1'} \geq \frac{d^4(1-\alpha_t)^3(1+\alpha_t)}{(d+1)^3(d-1)} \cdot \frac{d^2-1}{d^2(1-\alpha_t^2)} = \frac{d^2(1-\alpha_t)^2}{(d+1)^2}. \tag{3.44}$$

From Eq (3.42), (3.43), (3.44), we obtain Eq (3.18), (3.19) and therefore complete the proof. □

Lemma 3.1.9 characterizes the convergence of P_t: the product of the largest two eigenvalues shrinks by a constant factor after each step. Since $\det(P_t)$ approaches zero at an exponential rate (from Eq (3.17)), P_t can only have one potentially large eigenvalue while all other eigenvalues must approach zero. We formalize the claim in the following Lemma 3.1.10.

Lemma 3.1.10. *Suppose a valid cut is executed at step t with depth $-\frac{1}{kd} \leq \alpha_t \leq 0$ in Algorithm 16. Let the eigenvalue of P_t be $\sigma_1^{(t)} \geq \cdots \geq \sigma_d^{(t)}$. Further let $D_t = \sigma_2^{(t)}/\sigma_d^{(t)}$, we claim*

1. *for any $t \geq 0$, $D_{t+1} \leq \frac{(d+1)(1+\alpha_t)}{(d-1)(1-\alpha_t)} \cdot D_t$;*

2. *if $D_t > \frac{(d+1)(1+\alpha_t)}{(d-1)(1-\alpha_t)}$, $D_{t+1} \leq D_t$.*

3. *for any $n \geq 0$,*

$$\max_{0 \leq t \leq n} D_t \leq \left(\frac{d+1}{d-1}\right)^2. \tag{3.45}$$

Proof. From Lemma 3.1.9, we know that the eigenvalues of P_{t+1} is $\{\sigma_1', \sigma_2', \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_3^{(t)}, \cdots, \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_d^{(t)}\}$, where $\sigma_1' \geq \sigma_2'$ and

$$\frac{d^2(1-\alpha_t)^2}{(d+1)^2}\sigma_2^{(t)} \leq \sigma_2' \leq \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_2^{(t)} \tag{3.46}$$

Claim 1. Because $\sigma_1' \geq \sigma_2'$, $\sigma_3^{(t)} \geq \cdots \geq \sigma_d^{(t)}$, and note that $\sigma_2^{(t+1)}$ and $\sigma_d^{(t+1)}$ are the second-largest element and the smallest element of $\{\sigma_1', \sigma_2', \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_3^{(t)}, \cdots, \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_d^{(t)}\}$, the value of $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)})$ must satisfy one of the following situation:

1. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\sigma_2', \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_d^{(t)})$, from Eq (3.46) we have

$$\frac{D_{t+1}}{D_t} = \frac{d^2-1}{d^2(1-\alpha_t^2)} \cdot \frac{\sigma_2'}{\sigma_2^{(t)}} \leq 1. \tag{3.47}$$

2. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i^{(t)}, \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_d^{(t)})$ for some $3 \leq i \leq d-1$, we have

$$\frac{D_{t+1}}{D_t} = \frac{\sigma_i^{(t)}/\sigma_d^{(t)}}{\sigma_2^{(t)}/\sigma_d^{(t)}} \leq 1. \tag{3.48}$$

3. if $(\sigma_2^{(t+1)}, \sigma_d^{(t+1)}) = (\frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_i^{(t)}, \sigma_2')$ for some $3 \leq i \leq d-1$, from Eq (3.46) we have

$$\frac{D_{t+1}}{D_t} = \frac{d^2(1-\alpha_t^2)}{d^2-1} \cdot \frac{\sigma_i^{(t)}}{\sigma_2'} \cdot \frac{\sigma_d^{(t)}}{\sigma_2^{(t)}} \leq \frac{d^2(1-\alpha_t^2)}{d^2-1} \cdot \frac{\sigma_2^{(t)}}{\sigma_2'} \leq \frac{d^2(1-\alpha_t^2)}{d^2-1} \cdot \frac{(d+1)^2}{d^2(1-\alpha_t)^2} = \frac{(d+1)(1+\alpha_t)}{(d-1)(1-\alpha_t)}. \tag{3.49}$$

By Eq (3.47), (3.48), (3.49), the first claim holds.

Claim 2. It suffices to show that the situation (3) cannot happen when $D_t > \frac{d+1}{d-1}$. In fact, when $D_t > \frac{d+1}{d-1}$, from Eq (3.46) we have

$$\sigma_2' \geq \frac{d^2(1-\alpha_t)^2}{(d+1)^2}\sigma_2^{(t)} = \frac{d^2(1-\alpha_t)^2}{(d+1)^2}\sigma_d^{(t)} D_t > \frac{d^2(1-\alpha_t)^2}{(d+1)^2} \cdot \frac{(d+1)(1+\alpha_t)}{(d-1)(1-\alpha_t)} \cdot \sigma_d^{(t)} = \frac{d^2(1-\alpha_t^2)}{d^2-1}\sigma_d^{(t)},$$

meaning σ_2' cannot be the smallest eigenvalue of P_{t+1}. As a result, the second claim holds by Eq (3.47), (3.48).
Claim 3. We prove Eq (3.45) by contradiction. Let n_0 be the smallest index in set $\arg\max_{0\leq t\leq n} D_t$. If $n_0 = 0$, we have $\max_{0\leq t\leq n} D_t = D_0 = 1 < \left(\frac{d+1}{d-1}\right)^2$. Now consider the case $n_0 \geq 1$ and suppose $D_{n_0} > \left(\frac{d+1}{d-1}\right)^2$. By Claim 1 and the fact that $-\frac{1}{2d} \leq \alpha_{n_0-1} \leq 0$, we have $D_{n_0-1} \geq \frac{(d-1)(1-\alpha_{n_0-1})}{(d+1)(1+\alpha_{n_0-1})} D_{n_0} > \frac{(d+1)(1+\alpha_{n_0-1})}{(d-1)(1-\alpha_{n_0-1})}$. Apply Claim 2 to D_{n_0-1}, we obtain $D_{n_0} \leq D_{n_0-1}$, which contradicts the definition of n_0. Hence, Claim 3 holds.

□

Lemma 3.1.11. *With the choice of α_t given in Eq (3.8), we conclude that*

1. *After each cut step, Vol$(\mathcal{E}_{t+1}) \leq \exp\big(-\frac{(k-1)^2}{2k^2 d}\big)Vol(\mathcal{E}_t)$.*
2. *If at least d exploration steps are taken during $t_0 \leq t < t_0 + n$, we have $\lambda_{\min}(V_{n+t_0}) \geq \lambda_{\min}(V_{t_0}) + \frac{4D_0}{25} - 3\epsilon_0$.*
3. *At any exploitation step t, the instantaneous regret is upper bounded by $2L\|\theta_* - \mathbf{u}_1^{(t)}\|_2^2$.*

Proof. **First Claim:** We first justify our choice of α_t. With out loss of generality, assume $\mathbf{a}_{1,t}$ is preferred over $\mathbf{a}_{0,t}$, then according to the user's decision rule (3.3) we have

$$\theta_t^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) \leq |\beta_t| \cdot (\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}}) \leq c_2 t^{\gamma_2} \cdot (\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}}). \tag{3.50}$$

Next we translate Eq (3.50) into the estimation with respect to θ_*. According to the Estimation rule (3.2), with probability $1-\delta$,

$$\begin{aligned}(\theta_* - \theta_t)^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) &\leq \|\theta_* - \theta_t\|_{V_t} \cdot \|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \\ &\leq c_1 g(\delta) t^{\gamma_1}\|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}},\end{aligned}$$

and therefore according to the $(c,\gamma)-$rational assumption, we obtain

$$(\mathbf{a}_{0,1} - \mathbf{a}_{1,t})^\top(\theta_* - \mathbf{x}) \leq 0$$

$$\begin{aligned}\theta_*^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) &\leq \theta_t^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) + (\theta_* - \theta_t)^\top(\mathbf{a}_{0,t} - \mathbf{a}_{1,t}) \\ &\leq c_2 t^{\gamma_2} \cdot (\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}}) + c_1 g(\delta) t^{\gamma_1}\|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \\ &\leq ct^\gamma\Big(\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}} + g(\delta)\cdot\|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}}\Big).\end{aligned} \tag{3.51}$$

According to ϵ_0-DC and the definition of $\mathbf{g}_t$, we have

$$\|\mathbf{g}_t - (\mathbf{a}_{0,t} - \mathbf{a}_{1,t})\|_2 \leq \|\mathbf{a}_{0,t} - \bar{\mathbf{a}}_{0,t}\|_2 + \|\mathbf{a}_{1,t} - \bar{\mathbf{a}}_{1,t}\|_2 \leq 2\epsilon_0. \tag{3.52}$$

Using Eq (3.52), we may relax Eq (3.51) by replacing $\mathbf{a}_{0,t} - \mathbf{a}_{1,t}$ with $\mathbf{g}_t = \bar{\mathbf{a}}_{0,t} - \bar{\mathbf{a}}_{1,t}$, accounting for the error introduced by the inaccuracy of the exploration direction as below:

$$
\begin{aligned}
&\mathbf{g}_t^\top(\theta_* - \mathbf{x}_t) = \mathbf{g}_t^\top \theta_* \\
=&(\mathbf{a}_{0,t} - \mathbf{a}_{1,t})^\top \theta_* + (\mathbf{g}_t - (\mathbf{a}_{0,t} - \mathbf{a}_{1,t}))^\top \theta_* \\
\leq& ct^\gamma \Big(\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}} + g(\delta) \cdot \|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \Big) + \|\mathbf{g}_t - a_{0,t} + a_{1,t}\|_2 \cdot \|\theta_*\|_2 \\
\leq& ct^\gamma \Big(\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}} + g(\delta) \cdot \|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \Big) + 2\epsilon_0, \qquad (3.53)
\end{aligned}
$$

where Eq (3.53) holds because we assume $\|\theta_*\|_2 = 1$. Hence, by equation (3.13), the cutting depth

$$
\alpha_t = -\frac{ct^\gamma \Big(\|\mathbf{a}_{0,t}\|_{V_t^{-1}} + \|\mathbf{a}_{1,t}\|_{V_t^{-1}} + g(\delta) \cdot \|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \Big) + 2\epsilon_0}{\|\mathbf{g}_t\|_{P_t}}. \qquad (3.54)
$$

Therefore, we may leverage Eq (3.54) to evaluate the cutting depth α_t and perform a cut whenever $\alpha_t \geq -\frac{1}{kd} > -\frac{1}{d}$ is satisfied. From Eq (3.17), we therefore conclude $\text{Vol}(\mathcal{E}_{t+1}) \leq \exp\big(-\frac{(k-1)^2}{2k^2 d}\big)\text{Vol}(\mathcal{E}_t)$.

Second Claim: To prove the second claim, we need the following auxiliary lemma:

Lemma 3.1.12. *A is a $d \times d$ PSD matrix with eigendecomposition $A = U\mathrm{diag}(\sigma_1, \cdots, \sigma_d)U^T$, where $\sigma_1 \leq \cdots \leq \sigma_d$ and $U = [\mathbf{u}_1, \cdots, \mathbf{u}_d]$. For any $\mathbf{v} \in \mathbb{R}^d$, let the eigenvalues of $A + \mathbf{v}\mathbf{v}^T$ be $\sigma'_1 \leq \cdots \leq \sigma'_d$. Then we have*

1. $\sigma_1 \leq \sigma'_1 \leq \sigma_2 \leq \sigma'_2 \leq \cdots \leq \sigma_d \leq \sigma'_d \leq \sigma_d + \mathbf{v}^T\mathbf{v}$.
2. *if $\mathbf{v} = p\mathbf{u}_1 + q\mathbf{u}_d + \boldsymbol{\epsilon}$ for some $p^2 + q^2 = 1$, $\|\boldsymbol{\epsilon}\|_2 = \epsilon < 1$, $\{\sigma_i\}_{i=1}^d$ and $\{\sigma'_i\}_{i=1}^d$ have at least $d-2$ common values. Furthermore, conditioned on $\sigma_d > \sigma_1 + p^2 - q^2$, at least one of the following claims is true:*
   a) $\sigma'_1 \geq \sigma_1 + p^2 - |pq| - 3\epsilon$.
   b) $\sigma'_1 = \sigma_2$, and $\sigma'_i \geq \sigma_1 + p^2 - |pq| - 3\epsilon$ for some $2 \leq i \leq d$.

Proof. The first claim is a direct corollary of Weyl's inequality in matrix theory [135, 136]. Now we prove the second claim for the special case $\boldsymbol{\epsilon} = \mathbf{0}$. From Secular Equations, we know that σ'_1 is the smallest root of the following equation

$$
\begin{aligned}
f(\lambda) &= \prod_{i=1}^{d}(\sigma_i - \lambda) + p^2 \prod_{j \neq 1}^{d}(\sigma_j - \lambda) + q^2 \prod_{j \neq d}^{d}(\sigma_j - \lambda) \\
&= \Big[(\sigma_1 - \lambda)(\sigma_d - \lambda) + p^2(\sigma_d - \lambda) + q^2(\sigma_1 - \lambda)\Big] \prod_{j \neq 1,d}^{d}(\sigma_j - \lambda) \\
&= \Big[\lambda^2 - (1 + \sigma_1 + \sigma_d)\lambda + q^2\sigma_1 + p^2\sigma_d + \sigma_1\sigma_d\Big] \prod_{j \neq 1,d}^{d}(\sigma_j - \lambda).
\end{aligned}
$$

Therefore, σ'_1 is the smaller one between σ_2 and the smallest root of the quadratic equation $\lambda^2 - (1 + \sigma_1 + \sigma_d) + q^2\sigma_1 + p^2\sigma_d + \sigma_1\sigma_d = 0$, i.e.,

$$
\sigma'_1 = \min\{\sigma_2, \frac{1 + \sigma_1 + \sigma_d - \sqrt{(1 + \sigma_1 + \sigma_d)^2 - 4(q^2\sigma_1 + p^2\sigma_d + \sigma_1\sigma_d)}}{2}\}. \qquad (3.55)
$$

Note that when $\sigma_d > \sigma_1 + p^2 - q^2$, we have

$$
\begin{aligned}
&\frac{1+\sigma_1+\sigma_d-\sqrt{(1+\sigma_1+\sigma_d)^2-4(q^2\sigma_1+p^2\sigma_d+\sigma_1\sigma_d)}}{2}\\
=&\frac{1+\sigma_1+\sigma_d-\sqrt{(p^2-q^2+\sigma_1-\sigma_d)^2+4p^2q^2}}{2}\\
\geq&\frac{1}{2}(1+\sigma_1+\sigma_d-|p^2-q^2+\sigma_1-\sigma_d|-2|pq|) \qquad (3.56)\\
=&\sigma_1+p^2-|pq|, \qquad (3.57)
\end{aligned}
$$

where Eq (3.56) holds because $\sqrt{a^2+b^2} \leq |a|+|b|$. From Eq (3.55) and Eq (3.57) we conclude the proof.

Next it remains to show that with a small perturbation $\boldsymbol{\epsilon}$ on $\mathbf{v}$, the change of the smallest eigenvalue will only deviate at most 3ϵ. From Weyl's eigenvalue perturbation inequality, for any Hermitian matrices M, Δ, we have $|\lambda_k(M+\Delta)-\lambda_k(M)| \leq \|\Delta\|_2$, where $\lambda_k(\cdot)$ denotes the $k-$th largest eigenvalue of a given matrix. Using this tool, we can upper bound the difference between the smallest eigenvalues of matrix $A+\mathbf{v}\mathbf{v}^\top$ and $A+(\mathbf{v}+\boldsymbol{\epsilon})(\mathbf{v}+\boldsymbol{\epsilon})^\top$ as below:

$$
\begin{aligned}
&\lambda_1(A+(\mathbf{v}+\boldsymbol{\epsilon})(\mathbf{v}+\boldsymbol{\epsilon})^\top)-\lambda_1(A+\mathbf{v}\mathbf{v}^\top)\\
\leq&\|\boldsymbol{\epsilon}\mathbf{v}^\top+\mathbf{v}\boldsymbol{\epsilon}^\top+\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\|_2 \leq \|\boldsymbol{\epsilon}\mathbf{v}^\top+\mathbf{v}\boldsymbol{\epsilon}^\top\|_2+\|\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\|_2\\
\leq&2\epsilon+\epsilon^2<3\epsilon, \qquad (3.58)
\end{aligned}
$$

where Eq (3.58) holds because for any $\|\mathbf{x}\|_2=1$, $\mathbf{x}^\top(\boldsymbol{\epsilon}\mathbf{v}^\top+\mathbf{v}\boldsymbol{\epsilon}^\top)\mathbf{x} \leq 2\|\boldsymbol{\epsilon}\|_2$ and $\mathbf{x}^\top(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top)\mathbf{x} \leq \|\boldsymbol{\epsilon}\|_2^2$. □

Now we are ready to prove the second claim. Without loss of generality, we consider the case $D_0=1$. Suppose Algorithm 16 had executed d exploration steps from $t=t_0$ to $t=t_0+n$. By the first claim of Lemma 3.1.12, we know $\{\sigma_1^{(\tau)}\}_{\tau=1}^t$ is always non-decreasing. Therefore, it suffices to prove that after d consecutive exploration steps, $\sigma_1^{(t_0+d)} \geq \sigma_1^{(t_0)}+p^2-|pq|-3\epsilon_0$.

From the second claim in Lemma 3.1.12:

1. if situation a) happens at least once during the d exploration steps, we already obtain $\sigma_1^{(t_0+d)} \geq \sigma_1^{(t_0)}+p^2-|pq|-3\epsilon_0$.

2. if we always observe situation b), consider the set $C_t=\{i:\sigma_i^{(t_0+t)}<\sigma_1^{(t_0)}+p^2-|pq|-3\epsilon_0\}$. From Lemma 3.1.12, we can prove $|C_{t+1}| \leq |C_t|-1$. Since $\sigma_d^{(t_0)}>\sigma_1^{(t_0)}+p^2-|pq|-3\epsilon_0$, we have $|C_1| \leq d-1$. Therefore, there must exists $1 \leq k \leq d$ such that $|C_k|=0$, meaning $\sigma_1^{(t_0+d)} \geq \sigma_1^{(k)} \geq \sigma_1^{(t_0)}+p^2-|pq|-3\epsilon_0$.

By taking $(p,q)=(\frac{4}{5},\frac{3}{5})$, we obtain the desirable result.

Thrid Claim: Given $\|\theta_*\|_2=1$, denote $\hat{\theta}=\mathbf{u}_1^{(t)}$ and $\|\theta_*-\hat{\theta}\|_2=\epsilon$. Let $x_*=\arg\max_{x\in\mathcal{A}} x^T\theta_*$ and $\hat{x}=\arg\max_{x\in\mathcal{A}} x^T\hat{\theta}$. We have

$$
\begin{aligned}
\theta_*^T(x_*-\hat{x}) &= (\theta_*-\hat{\theta})^Tx_*+(x_*-\hat{x})^T\hat{\theta}+(\hat{\theta}-\theta_*)^T\hat{x} \\
&\leq (\theta_*-\hat{\theta})^Tx_*+(\hat{\theta}-\theta_*)^T\hat{x} && \text{by definition of } \hat{x}\\
&= (\hat{\theta}-\theta_*)^T(\hat{x}-x_*)\\
&\leq \|\hat{\theta}-\theta_*\|_2\cdot\|\hat{x}-x_*\|_2 && \text{by Cauchy-Schwarz}\\
&\leq L\cdot\|\hat{\theta}-\theta_*\|_2^2. && \text{by L-SRC}
\end{aligned}
$$

As a result, the instantaneous regret is upper bounded by $2L\|\mathbf{u}_1^{(t)}-\theta_*\|_2^2$. □

Now we are ready to analyze the regret of Algorithm 16:

Theorem 3.1.13. *For any $d > 1, n > 0$, let $\sigma_i^{(n)}$ be the i-th largest eigenvalue of P_n after the n-th cut, we have*

1. *For any $2 \le i \le d$,*
$$\sigma_i^{(n)} \le \exp\Big(\frac{4}{d} - \frac{(k-1)^2 n}{k^2 d^2}\Big). \tag{3.59}$$

2. *When $T_0 = O\Big(cL^{\frac{1}{2}} D_1^{\frac{1}{2}} D_0^{-\frac{3}{2}} g(\delta) d^2 T^{\frac{1}{2}+\gamma}\Big)$ and $\epsilon_0 < O\big(cD_1 D_0^{-\frac{1}{2}} d^{-\frac{1}{2}} T^{-\frac{1}{4}+\frac{\gamma}{2}}\big)$, the regret of RAES is upper bounded by $O\Big(cL^{\frac{1}{2}} D_1^{\frac{3}{2}} D_0^{-\frac{3}{2}} g(\frac{\delta}{T_0}) d^2 T^{\frac{1}{2}+\gamma}\Big)$ with probability $1-\delta$.*

Proof. Since the depth of the cut $\alpha_t \ge -\frac{1}{kd}$ through out the execution of Algorithm 16, from Lemma 3.1.3 and Eq (3.17) we have
$$\prod_{i=1}^{d} \sigma_i^{(n)} = \prod_{i=0}^{n-1} \frac{\det P_{i+1}}{\det P_i} \le \prod_{i=0}^{n-1} \frac{\text{Vol}(\mathcal{E}_{i+1})}{\text{Vol}(\mathcal{E}_i)} = \exp\Big(-\frac{(k-1)^2 n}{k^2 d}\Big). \tag{3.60}$$
From Lemma 3.1.10, we have $\sigma_i^{(n)} \ge \sigma_d^{(n)} \ge \big(\frac{d-1}{d+1}\big)^2 \cdot \sigma_2^{(n)}, \forall 3 \le i \le d$. Therefore,
$$\begin{aligned}
\exp\Big(-\frac{(k-1)^2 n}{k^2 d}\Big) &\ge \prod_{i=1}^{d} \sigma_i^{(n)} \\
&\ge \sigma_2^{(n)} \cdot \sigma_2^{(n)} \cdot \Big[\big(\frac{d-1}{d+1}\big)^2 \cdot \sigma_2^{(n)}\Big]^{d-2} \\
&= [\sigma_2^{(n)}]^d \cdot \Big(1 - \frac{2}{d+1}\Big)^{2d-4} \\
&\ge \exp(-4) \cdot [\sigma_2^{(n)}]^d.
\end{aligned}$$
Rearranging terms yields $\sigma_2^{(n)} \le \exp\Big(\frac{4}{d} - \frac{(k-1)^2 n}{k^2 d^2}\Big)$, and thus $\sigma_i^{(n)} \le \exp\Big(\frac{4}{d} - \frac{(k-1)^2 n}{k^2 d^2}\Big), \forall 2 \le i \le d$.

Next we show the second claim. Suppose the total number of cut during the first $T_0/2$ step is N_0.

1. if $N_0 \ge \frac{d^2k^2}{(k-1)^2} \log T_0 + \frac{4dk^2}{(k-1)^2}$, from Eq (3.59) we have $\sigma_i^{(N_0)} \le \frac{1}{T_0}$.

2. if $N_0 < \frac{d^2k^2}{(k-1)^2} \log T_0 + \frac{4dk^2}{(k-1)^2}$, for sufficiently large T, there are at least $T_0/2 - N \ge T_0/2 - \frac{d^2k^2}{(k-1)^2} \log T_0 - \frac{4dk^2}{(k-1)^2} > \frac{T_0}{3}$ exploration steps during the first $T_0/2$ iterations. From the second claim of Lemma 3.1.3, $\lambda_{\min}(V_{T_0}) \ge \frac{\beta T_0}{d}$, where $\beta = \frac{1}{3}(\frac{4D_0}{25} - 3\epsilon_0)$ is a positive constant. Using the definition of matrix norm, we have for any t, $\|\mathbf{a}_{0,t}\|_{V_t^{-1}}, \|\mathbf{a}_{1,t}\|_{V_t^{-1}} \le D_1\sqrt{\lambda_{\max}(V_t^{-1})}, \|\mathbf{a}_{0,t} - \mathbf{a}_{1,t}\|_{V_t^{-1}} \le 2D_1\sqrt{\lambda_{\max}(V_t^{-1})}$, and $\|\mathbf{g}_t\|_{P_t} \ge D_0(\sigma_2^{(t)})^{-\frac{1}{2}}$. Therefore, we have
$$\alpha_t \ge -2\Big[ct^\gamma D_1 D_0^{-1}\Big(1+g(\delta)\Big) \cdot \sqrt{\lambda_{\max}(V_t^{-1})} + \epsilon_0 D_0^{-1}\Big] \cdot (\sigma_2^{(t)})^{-\frac{1}{2}}.$$
According to Algorithm 16, as long as we have $-2\Big[ct^\gamma D_1 D_0^{-1}\Big(1+g(\delta)\Big) \cdot \sqrt{\lambda_{\max}(V_t^{-1})} + \epsilon_0 D_0^{-1}\Big] \cdot (\sigma_2^{(t)})^{-\frac{1}{2}} \ge -\frac{1}{kd}$, a cut will happen at step t and we can shrink $\sqrt{\sigma_2^{(t)}}$ with probability $1-\delta$. In other words, after the last time Algorithm 16 choose to cut during the first T_0 round, we have
$$\sqrt{\sigma_2^{(t)}} \le \frac{2D_1ckd^{1.5}t^\gamma(1+g(\delta))}{D_0\sqrt{\beta T_0}} + \frac{2kd\epsilon_0}{D_0} < \frac{3D_1ckd^{1.5}T_0^\gamma(1+g(\delta))}{D_0\sqrt{\beta T_0}}, \tag{3.61}$$
where the last inequality holds because $\epsilon_0 < \frac{cD_1}{2\sqrt{\beta}} d^{-\frac{1}{2}} T^{-\frac{1}{4}+\frac{\gamma}{2}}$. On the other hand, the total number of cuts n such that Eq (3.61) is satisfied is upper bounded by $O(\log T_0)$ since $\sigma_2^{(t)}$ shrinks exponentially w.r.t. the cut

number t. Therefore, when T is reasonably large, we can guarantee $n < T_0/2$ and conclude that Eq (3.61) holds for all $t > T_0$.

According to Eq (3.36) and the third claim in Lemma 3.1.3, when algorithm 16 enters the exploitation phase when $t > T_0$, with probability $1 - T_0\delta$, the instantaneous regret is upper bounded by

$$\theta_*^\top[(a_* - a_{0,t}) + (a_* - a_{1,t})] \leq 8(d-1) \cdot L \cdot \Big(\frac{3D_1ckd^{1.5}T_0^\gamma(1+g(\delta))}{D_0\sqrt{\beta T_0}}\Big)^2 \tag{3.62}$$

$$\leq \frac{72D_1^2Lc^2k^2d^4(1+g(\delta))^2}{\beta D_0^2T_0^{1-2\gamma}} \tag{3.63}$$

For each cut or exploration step in the first T_0 rounds, the incurred instantaneous regret is at most T_0D_1. For each following exploitation step, the regret is upper bounded by $\frac{72D_1^2Lc^2k^2d^4(1+g(\delta))^2}{D_0^2\beta T_0^{1-2\gamma}}$. Hence, we can upper bound the accumulated regret by

$$R_T \leq D_1T_0 + \frac{72D_1^2Lc^2k^2d^4(1+g(\delta))^2}{D_0^2\beta T_0} \cdot T^{1+2\gamma}$$

$$\leq \frac{12D_1}{D_0}\sqrt{\frac{2LD_1}{\beta}}ck(1+g(\delta)) \cdot d^2T^{\frac{1}{2}+\gamma}, \tag{3.64}$$

where the optimal regret is achieved when $T_0 = \frac{6ck}{D_0}\sqrt{\frac{6LD_1}{\frac{4D_0}{25}-3\epsilon_0}}\big(1+g(\delta)\big)d^2T^{\frac{1}{2}+\gamma}$, we have $R_T \leq \frac{12ckD_1}{D_0}\sqrt{\frac{6LD_1}{\frac{4D_0}{25}-3\epsilon_0}}\big(1+g(\delta)\big)d^2T^{\frac{1}{2}+\gamma}$. By applying the union bound to the first T_0 rounds, we thus conclude that with probability $1-\delta$,

$$R_T \leq \frac{60D_1}{D_0}\sqrt{\frac{6LD_1}{4D_0-75\epsilon_0}}ck\big(1+g(\frac{\delta}{T_0})\big) \cdot d^2T^{\frac{1}{2}+\gamma}.$$

□

Proof of Theorem 3.1.5 in Section 3.1.5

To derive our lower bound result, we need to leverage the minimax lower bound result for stochastic linear bandits (adapted from Theorem 24.1 in [127]). For convenience, we use $\theta_{i:j}$ to denote the slice of vector θ from the $i-$th element to the $j-$th element.

Theorem 3.1.14. *There exists a function $T_0(d) > 0$ such that for any $d \geq 1$, $T > T_0(d)$, and any algorithm $\mathcal{G}$ that has merely access to the comparison feedback given by a rational user defined in Definition 3.1.1, there exists $\theta \in \partial\mathbb{B}_1^d$ such that the expected regret R_T given by Eq (3.1) obtained by $\mathcal{G}$ satisfies*

$$R_T^{(s)}(\mathcal{G},\theta) \geq \frac{\exp(-2)}{4}(d-1)\sqrt{T}. \tag{3.65}$$

Proof. We prove our claim by contradiction using Theorem 3.1.15. Essentially, we show that if the system has a powerful algorithm to achieve an expected regret lower than the RHS of Eq. (3.10), then we can leverage this algorithm for the linear bandit problem in Theorem 3.1.15 with an expected regret even lower than the lower bound and thus draw the contradiction.

Suppose for any $d > 0$, there exists sufficiently large T and an algorithm $\mathcal{G}$ such that for any parameter $\theta_* \in \partial\mathbb{B}_1^d$, we have

$$\mathbb{E}\Big[\sum_{t=1}^{T}\theta_*^\top(2a_* - a_{0,t} - a_{1,t})\Big] = R_T^{(s)}(\mathcal{G},\theta_*) < \frac{\exp(-2)}{4}(d-1)\sqrt{T}.$$

As a result, the following inequalities must hold simultaneously:

$$\begin{aligned}\mathbb{E}\Big[\sum_{t=1}^{T}\theta_*^\top(a_* - a_{0,t})\Big] &< \frac{\exp(-2)}{4}(d-1)\sqrt{T},\\ \mathbb{E}\Big[\sum_{t=1}^{T}\theta_*^\top(a_* - a_{1,t})\Big] &< \frac{\exp(-2)}{4}(d-1)\sqrt{T}.\end{aligned} \tag{3.66}$$

Now suppose a principal can observe the interaction between a user and a system equipped with algorithm $\mathcal{G}$, then he can construct two algorithms $\mathcal{G}_0, \mathcal{G}_1$ for linear bandit as follows:

```
Algorithm $\mathcal{G}_i$ :
Input: the time horizon $T$.
For $t \in [T]$:
    1. Call algorithm $\mathcal{G}$ to generate two candidates $(a_{0,t}, a_{1,t})$.
    2. Present $(a_{0,t}, a_{1,t})$ to the user and and let her decide the winner $a_{*,t}$ using decision rule 3.3.
    3. Return the feedback $a_{*,t}$ to algorithm $\mathcal{G}$ and update the internal state of $\mathcal{G}$ accordingly.
Output: the sequential decisions $\{a_{i,t}\}_{t=1}^{T}$.
```

From Eq. (3.66), we know that both $\mathcal{G}_0$ and $\mathcal{G}_1$ achieve an expected regret no greater than $\frac{\exp(-2)}{4}(d-1)\sqrt{T}$, which draws a contradiction to Theorem 3.1.15.

□

To prove Theorem 3.1.14, we need the following technical lemma:

Lemma 3.1.15. *Let $d \geq 2$ and $T \geq d^2$, the action set $\mathcal{A} = [-1,1]^d$ be a hypercube in $\mathbb{R}^d$, and*

$$\Theta = \Big\{\theta \in \mathbb{R}^d : \|\theta\|_1 = 1, \theta_{1:d-1} \in \{-\frac{1}{\sqrt{T}}, \frac{1}{\sqrt{T}}\}^{d-1}\Big\}.$$

Let the expected regret for a linear bandit problem induced by any fixed algorithm $\mathcal{G}$ and parameter θ be

$$R_T(\mathcal{G}, \theta) = T\max_{a\in\mathcal{A}}\langle a, \theta\rangle - \mathbb{E}[\sum_{t=1}^{T}\langle a_t, \theta\rangle], \tag{3.67}$$

where the expectation is taken with respect to the randomness generated by the standard Gaussian noise $\mathcal{N}(0,1)$ in the reward. Then there must exist a parameter vector $\theta \in \Theta$ such that

$$R_T(\mathcal{G}, \theta) \geq \frac{\exp(-2)}{8}(d-1)\sqrt{T}. \tag{3.68}$$

Proof. Fix an algorithm $\mathcal{G}$ and a time horizon T. For any $\theta \in \Theta$, let $\mathbb{P}_\theta$ be the probability measure on the probability space induced by the T-round interconnection of policy $\mathcal{G}$ and the problem instance given by θ. Let $D(\cdot,\cdot)$ denote the relative entropy, from the general form of divergence decomposition lemma (Lemma 15.1 in [127]), we have

$$\begin{aligned}D(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) &= \mathbb{E}_\theta\Big[\sum_{t=1}^{T} D(\mathcal{N}(\langle a_t, \theta\rangle, 1), \mathcal{N}(\langle a_t, \theta'\rangle, 1))\Big]\\ &= \frac{1}{2}\sum_{t=1}^{T}\mathbb{E}_\theta[\langle a_t, \theta - \theta'\rangle^2].\end{aligned} \tag{3.69}$$

For any $i \in [d-1]$ and $\theta \in \Theta$, let $a_{t,i}$ and θ_i be the i-th element of a_t and θ and define

$$p_{\theta_i} = \mathbb{P}_\theta\Big(\sum_{t=1}^T \mathbb{I}\{\text{sign}(a_{t,i}) \neq \text{sign}(\theta_i)\} \geq \frac{T}{2}\Big).$$

Let θ, θ' be any pair of elements in Θ such that they only differ in the $i-$th element. Therefore, by the Bretagnolle-Huber inequality (Theorem 14.2 in [127]) and Eq. (3.69),

$$\begin{aligned}
p_{\theta_i} + p_{\theta'_i} &\geq \frac{1}{2}\exp\Big(-D(\mathbb{P}_\theta, \mathbb{P}_{\theta'})\Big) \\
&= \frac{1}{2}\exp\Big(-\frac{1}{2}\sum_{t=1}^T \mathbb{E}_\theta[\langle a_t, \theta - \theta'\rangle^2]\Big) \\
&\geq \frac{1}{2}\exp\Big(-\frac{1}{2}\cdot T(\frac{2}{\sqrt{T}})^2\Big) = \frac{1}{2}\exp(-2).
\end{aligned}$$

Fix $i \in [d-1]$, there are $|\Theta| = 2^{d-1}$ such pairs (θ, θ'). Take summation over i and all such pairs, we obtain

$$\begin{aligned}
\sum_{\theta\in\Theta}\frac{1}{|\Theta|}\sum_{i=1}^d p_{\theta_i} &\geq \frac{1}{|\Theta|}\sum_{i=1}^{d-1}\sum_{\theta\in\Theta} p_{\theta_i} \\
&= \frac{1}{|\Theta|}\sum_{i=1}^{d-1}\frac{1}{2}\sum_{(\theta,\theta')}(p_{\theta_i} + p_{\theta'_i}) \\
&\geq \frac{d-1}{4}\exp(-2),
\end{aligned}$$

which implies that there exists a $\theta \in \Theta$ such that $\sum_{i=1}^d p_{\theta_i} \geq \frac{d-1}{4}\exp(-2)$. By the definition of p_{θ_i}, the regret of $\mathcal{G}$ for this problem instance with parameter θ is at least

$$\begin{aligned}
R_T(\mathcal{A}, \theta) &= \mathbb{E}_\theta\Big[\sum_{t=1}^T\sum_{i=1}^d(\text{sign}(\theta_i) - a_{t,i})\theta_i\Big] \\
&\geq \sqrt{\frac{1}{T}}\sum_{i=1}^d \mathbb{E}_\theta\Big[\sum_{t=1}^T \mathbb{I}\{\text{sign}(a_{t,i}) \neq \text{sign}(\theta_i)\}\Big] \\
&\geq \frac{\sqrt{T}}{2}\sum_{i=1}^d \mathbb{P}_\theta\Big(\sum_{t=1}^T \mathbb{I}\{\text{sign}(a_{ti}) \neq \text{sign}(\theta_i)\} \geq \frac{T}{2}\Big) \\
&= \frac{\sqrt{T}}{2}\sum_{i=1}^d p_{\theta_i} \geq \frac{\exp(-2)}{8}(d-1)\sqrt{T},
\end{aligned}$$

where the first line follows since the optimal action satisfies $a_i^* = \text{sign}(\theta_i)$ and for $i \in [d]$, the first inequality follows from a simple case-based analysis showing that $(\text{sign}(\theta_i) - a_{ti})\theta_i \geq |\theta_i|\mathbb{I}\{\text{sign}(a_{ti}) \neq \text{sign}(\theta_i)\}$, the second inequality is from Markov's inequality, and the last inequality follows from the choice of θ.

□

3.2 Incentivize communication in federated bandit

Despite our extensive exploration of various settings of federated bandit learning in Section 2.2 and Section 2.3, all these proposed algorithms rely on the assumption that every client in the system is willing to share their local data/model with the server, regardless of the communication protocol design. For instance, synchronous protocols [28, 84, 100] require all clients to simultaneously engage in data exchange with the server in every communication round. Similarly, asynchronous protocols [64, 97, 98] also assume clients must participate in communication as long as the individualized upload or download event is triggered, albeit allowing interruptions by external factors (e.g., network failure).

In contrast, this work is motivated by the practical observation that many clients in a federated system are inherently self-interested and thus reluctant to share data without receiving explicit benefits from the server [16]. For instance, consider the following scenario: a recommendation platform (server) wants its mobile app users (clients) to opt in its new recommendation service, which switches previous on-device local bandit algorithm to a federated bandit algorithm. Although the new service is expected to improve the overall recommendation quality for all clients, particular clients may not be willing to participate in this collaborative learning, as the expected gain for them might not compensate their locally increased cost (e.g., communication bandwidth, added computation, lost control of their data, and etc). In this case, additional actions have to be taken by the server to encourage participation, as it has no power to force clients. This exemplifies the most critical concern in the real-world application of federated learning [16]. And a typical solution is known as *incentive mechanism*, which motivates individuals to contribute to the social welfare goal by offering incentives such as monetary compensation.

While recent studies have explored incentivized data sharing in federated learning [137, 138], most of which only focused on the supervised offline learning setting [16]. In this section, we propose the first work that studies incentive design for federated bandit learning, which inherently imposes new challenges.

First, there is a lack of well-defined metric to measure the utility of data sharing, which rationalizes a client's participation. Under the context of bandit learning, we measure data utility by the expected regret reduction from the exchanged data for each client. As a result, each client values data (e.g., sufficient statistics) from server differently, depending on how such data aligns with their local data (e.g., the more similar the less valuable). Second, the server is set to minimize regret across all clients through data exchange. But as the server does not generate data, it can be easily trapped by the situation where its collected data cannot pass the critical mass to ensure every participating client's regret is close to optimal (e.g., the data under server's possession cannot motivate the clients who have more valuable data to participate). To break the deadlock, we give the server the ability to provide monetary incentives. Subsequently, the server needs to minimize its cumulative monetary payments, in addition to the regret and communication minimization objectives as required by federated bandit learning.

3.2.1 Related works

In this work, we situate the incentivized federated bandit learning problem under linear bandits with time-varying arm sets, which is a popular setting in many recent works [28, 61, 64, 97]. But we do not assume the clients will always participate in data sharing: they will choose not to share its data with the server if the resulting benefit of data sharing is not deemed to outweigh the cost. Here we need to differentiate this new setting from the one considered in our prior work about asynchronous communication [64], which still assumes all clients are willing to share, though sometimes the communication can be interrupted by some external factors (e.g., network failure). Here we do not assume communication failures and leave it as our future work. Instead, we assume the clients need to be motivated to participate in federated learning, and our focus is to devise the minimum incentives to obtain the desired regret and communication cost for all participating clients.

Incentivized Federated Learning Data sharing is essential to the success of federated learning [137], where client participation plays a crucial role. However, participation involves costs, such as the need for additional computing and communication resources, and the risk of potential privacy breaches, which can lead to opt-outs [139, 140]. In light of this, recent research has focused on investigating incentive mechanisms that motivate clients to contribute, rather than assuming their willingness to participate. Most of the existing research involves multiple decentralized clients solving the same task, typically with different copies of IID datasets, where the focus is on designing data valuation methods that ensure fairness or achieve a specific accuracy objective [141, 142, 143]. On the other hand, Donahue et al. [144] study voluntary participation in model-sharing games, where clients may opt out due to biased global models caused by the aggregated non-IID datasets. More recently, Karimireddy et al. [16] investigated incentive mechanism design for data maximization while avoiding free riders. For a detailed discussion of this topic, we refer readers to recent surveys on incentive mechanism design in federated learning [145, 138].

However, most works on incentivized federated learning only focus on better model estimation among fixed offline datasets, which does not apply to the bandit learning problem, where the exploration of growing data is also part of the objective. More importantly, in our incentivized federated bandit problem, the server is obligated to improve the overall performance of the learning system, i.e., minimizing regret among all clients, which is essentially different from previous studies where the server only selectively incentivizes clients to achieve a certain accuracy [141] or to investigate how much accuracy the system can achieve without payment [16].

3.2.2 Incentivized federated bandit problem

Here we adopt the federated bandit problem formulation from Section 2.3.1. Specifically, at each time step $t \in [T]$, an arbitrary client $i_t \in [N]$ becomes active and chooses an arm $\mathbf{x}_t$ from a candidate set $\mathcal{A}_t \subseteq \mathbb{R}^d$, and then receives the corresponding reward feedback $y_t = f(\mathbf{x}_t) + \eta_t \in \mathbb{R}$. Note that $\mathcal{A}_t$ is time-varying, f denotes the unknown reward function shared by all clients, and η_t denotes zero mean sub-Gaussian noise with known variance σ^2.

Different from the works discussed in Section 2.2 and Section 2.3, where all clients altruistically share their data with the server whenever a communication round is triggered, we are intrigued in a more realistic setting where clients are *self-interested* and thus reluctant to share data with the server if not well motivated. Formally, each client in the federated system inherently experiences a cost[3] of data sharing, denoted by $\widetilde{D}_i^p \in \mathbb{R}$, due to their individual consumption of computing resources in local updates or concerns about potential privacy breaches caused by communication with the server. Moreover, as the client has nothing to lose when there is no local update to share in a communication round at time step t, in this case we assume the cost is 0, i.e., $D_i^p = \widetilde{D}_i^p \cdot \mathbb{I}(\Delta V_{i,t} \neq \mathbf{0})$. As a result, the server needs to motivate clients to participate in data sharing via the incentive mechanism $\mathcal{M} : \mathbb{R}^N \times \mathbb{R}^{d\times d} \rightarrow \mathbb{R}^N$, which takes as inputs a collection of client local updates $\Delta V_{i,t} \in \mathbb{R}^{d\times d}$ and a vector of cost values $D^p = \{D_1^p, \cdots, D_N^p\} \in \mathbb{R}^N$, and outputs the incentive $\mathcal{I} = \{\mathcal{I}_{1,t}, \cdots, \mathcal{I}_{N,t}\} \in \mathbb{R}^N$ to be distributed among the clients. Specifically, to make it possible to measure gains and losses of utility in terms of real-valued incentives (e.g., monetary payment), we adopt the standard quasi-linear utility function assumption, as is standard in economic analysis [146, 147].

At each communication round, a client decides whether to share its local update with the server based on the potential utility gained from participation, i.e., the difference between the incentive and the cost of data sharing. This requires the incentive mechanism to be *individually rational*:

Definition 3.2.1 (Individual Rationality [148])**.** *An incentive mechanism $\mathcal{M} : \mathbb{R}^N \times \mathbb{R}^{d\times d} \rightarrow \mathbb{R}^N$ is individually rational if for any i in the participant set S_t at time step t, we have*

$$\mathcal{I}_{i,t} \geq D_i^p \tag{3.70}$$

In other words, each participant must be guaranteed non-negative utility by participating in data sharing under $\mathcal{M}$.

The server coordinates with all clients and incentivizes them to participate in the communication to realize its own objective (e.g., collective regret minimization). This requires $\mathcal{M}$ to be *sufficient*:

Definition 3.2.2 (Sufficiency)**.** *An incentive mechanism $\mathcal{M} : \mathbb{R}^N \times \mathbb{R}^{d\times d} \rightarrow \mathbb{R}^N$ is sufficient if the resulting outcome satisfies the server's objective.*

Typically, under different application scenarios, the server may have different objectives, such as regret minimization or best arm identification. In this work, we set the objective of the server to minimize the regret across all clients; and ideally the server aims to attain the optimal $\tilde{O}(d\sqrt{T})$ regret in centralized setting via the incentivized communication. Therefore, we consider an incentive mechanism is sufficient if it ensures that the resulting accumulated regret is bounded by $\tilde{O}(d\sqrt{T})$.

3.2.3 INC-FEDUCB algorithm

The communication backbone of our solution derives from DisLinUCB [28], which is a widely adopted paradigm for federated linear bandits. We adopt their strategy for arm selection and communication trigger, so as to focus on the incentive mechanism design. We name the resulting algorithm INC-FEDUCB, and present it in Algorithm 17. Note that the two incentive mechanisms to be presented in Section 3.2.4 and 3.2.5 are not specific to any federated bandit learning algorithms, and each of them can be easily extended to alternative workarounds as a plug-in to accommodate the incentivized federated learning setting.

Our framework comprises three main steps: 1) client's local update; 2) communication trigger; and 3) incentivized data exchange among the server and clients. Specifically, after initialization, an active client performs a local update

[3] Note that if the costs are trivially set to zero, then clients have no reason to opt-out of data sharing and our problem essentially reduces to the standard federated bandits problem [28].

in each time step and checks the communication trigger. If a communication round is triggered, the system performs incentivized data exchange between clients and the server. Otherwise, no communication is needed.

Algorithm 17 INC-FEDUCB Algorithm

```
1:  Input $D_c \geq 0$, $D^p = \{D^p_1, \cdots, D^p_N\}$, $\sigma, \lambda > 0$, $\delta \in (0,1)$
2:  Initialize: [Server] $V_{g,0} = \mathbf{0}_{d\times d} \in \mathbb{R}^{d\times d}$, $b_{g,0} = \mathbf{0}_d \in \mathbb{R}^d$
                        $\Delta V_{-j,0} = \mathbf{0}_{d\times d}, \Delta b_{-j,0} = \mathbf{0}_d, \forall j \in [N]$
              [All clients] $V_{i,0} = \mathbf{0}_{d\times d}$, $b_{i,0} = \mathbf{0}_d$, $\Delta V_{i,0} = \mathbf{0}_{d\times d}$, $\Delta b_{i,0} = \mathbf{0}_d$, $\Delta t_{i,0} = 0, \forall i \in [N]$
3:  for $t = 1, 2, \dots, T$ do
4:    [Client $i_t$] Observe arm set $\mathcal{A}_t$
5:    [Client $i_t$] Select arm $\mathbf{x}_t \in \mathcal{A}_t$ by Eq. (3.71) and observe reward $y_t$
6:    [Client $i_t$] Update: $V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^\top$, $b_{i_t,t} \mathrel{+}= \mathbf{x}_t y_t$
                          $\Delta V_{i_t,t} \mathrel{+}= \mathbf{x}_t\mathbf{x}_t^\top$, $\Delta b_{i_t,t} \mathrel{+}= \mathbf{x}_t y_t$, $\Delta t_{i_t,t} \mathrel{+}= 1$
7:    if $\Delta t_{i_t,t} \log \frac{\det(V_{i_t,t}+\lambda I)}{\det(V_{i_t,t}-\Delta V_{i_t,t}+\lambda I)} > D_c$ then
8:      [All clients → Server] Upload $\Delta V_{i,t}$ such that $\tilde{S}_t = \{\Delta V_{i,t} | \forall i \in [N]\}$
9:      [Server] Select incentivized participants $S_t = \mathcal{M}(\tilde{S}_t)$          {//Incentive Mechanism }
10:     for $i : \Delta V_{i,t} \in S_t$ do
11:       [Participant $i$ → Server] Upload $\Delta b_{i,t}$
12:       [Server] Update: $V_{g,t} \mathrel{+}= \Delta V_{i,t}$, $b_{g,t} \mathrel{+}= \Delta b_{i,t}$
                           $\Delta V_{-j,t} \mathrel{+}= \Delta V_{i,t}$, $\Delta b_{-j,t} \mathrel{+}= \Delta b_{i,t}, \forall j \neq i$
13:       [Participant $i$] Update: $\Delta V_{i,t} = 0$, $\Delta b_{i,t} = 0$, $\Delta t_{i,t} = 0$
14:     for $\forall i \in [N]$ do
15:       [Server → All Clients] Download $\Delta V_{-i,t}$, $\Delta b_{-i,t}$
16:       [Client $i$] Update: $V_{i,t} \mathrel{+}= \Delta V_{-i,t}$, $b_{i,t} \mathrel{+}= \Delta b_{-i,t}$
17:       [Server] Update: $\Delta V_{-i,t} = 0$, $\Delta b_{-i,t} = 0$
```

Formally, at each time step $t = 1, \dots, T$, an arbitrary client i_t becomes active and interacts with its environment using observed arm set $\mathcal{A}_t$ (Line 5). Specifically, it selects an arm $\mathbf{x}_t \in \mathcal{A}_t$ that maximizes the UCB score as follows (Line 6):

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{A}_t} \mathbf{x}^\top \hat{\theta}_{i_t,t-1}(\lambda) + \alpha_{i_t,t-1} ||\mathbf{x}||_{V^{-1}_{i_t,t-1}(\lambda)} \tag{3.71}$$

where $\hat{\theta}_{i_t,t-1}(\lambda) = V^{-1}_{i_t,t-1}(\lambda) b_{i_t,t-1}$ is the ridge regression estimator of $\theta_\star$ with regularization parameter $\lambda > 0$, $V_{i_t,t-1}(\lambda) = V_{i_t,t-1} + \lambda I$, and $\alpha_{i_t,t-1} = \sigma\sqrt{\log \frac{\det(V_{i_t,t-1}(\lambda))}{\det(\lambda I)} + 2\log 1/\delta} + \sqrt{\lambda}$. $V_{i_t,t}(\lambda)$ denotes the covariance matrix constructed using data available to client i_t up to time t. After obtaining a new data point $(\mathbf{x}_t, y_t)$ from the environment, client i_t checks the communication event trigger $\Delta t_{i_t,t} \cdot \log \frac{\det(V_{i_t,t}(\lambda))}{\det(V_{i_t,t_{\text{last}}}(\lambda))} > D_c$ (Line 9), where $\Delta t_{i_t,t}$ denotes the time elapsed since the last time t_{last} it communicated with the server and $D_c \geq 0$ denotes the specified threshold.

Incentivized Data Exchange With the above event trigger, communication rounds only occur if (1) a substantial amount of new data has been accumulated locally at client i_t, and/or (2) significant time has elapsed since last communication. However, in our incentivized setting, triggering a communication round does not necessarily lead to data exchange at time step t, as the participant set S_t may be empty (Line 11). This characterizes the fundamental difference between INC-FEDUCB and DisLinUCB [28]: we no longer assume all N clients will share their data with the server in an altruistic manner; instead, a rational client only shares its local update with the server if the condition in Eq. (3.70) is met. In light of this, to evaluate the potential benefit of data sharing, all clients must first reveal the value of their data to the server before the server determines the incentive. Hence, after a communication round is triggered, all clients upload their latest sufficient statistics update $\Delta V_{i,t}$ to the server (Line 10) to facilitate data valuation and participant selection in the incentive mechanism (Line 11). Note that this disclosure does not compromise clients' privacy, as the clients' secret lies in $\Delta b_{i,t}$ that is constructed by the rewards. Only participating clients will upload their $\Delta b_{i,t}$ to the server (Line 13). After collecting data from all participants, the server download the aggregated

Algorithm 18 Payment-free Incentive Mechanism

```
Input $D^p = \{D_i^p | i \in [N]\}$, $\tilde{S}_t = \{\Delta V_{i,t} | i \in [N]\}$
Initialize participant set $S_t = \tilde{S}_t$
while $S_t \neq \emptyset$ do
    StableFlag $=$ True                                  {//iteratively update $S_t$ until it becomes stable}
    for $i : \Delta V_{i,t} \in S_t$ do
        if $\mathcal{I}_{i,t} < D_i^p$ (Eq. 3.73) then
            Update participant set $S_t = S_t \setminus \{\Delta V_{i,t}\}$          {//remove client $j$ from $S_t$}
            StableFlag $=$ False
            break
    if StableFlag $=$ True then
        break
Return $S_t \subseteq \tilde{S}_t$
```

updates $\Delta V_{-i,t}$ and $\Delta b_{-i,t}$ to every client i (Line 17-20). Follow the convention in federated bandit learning [28], the communication cost is defined as the total number of scalars transferred during this data exchange process.

3.2.4 Payment-free incentive mechanism

As showed in Section 2.2 and Section 2.3, in federated bandit learning, clients can reduce their regret by using models constructed via shared data. Denote $\widetilde{V}_t$ as the covariance matrix constructed by all available data in the system at time step t. As discussed in Section 2.3.2, the instantaneous regret of client i_t is upper bounded by:

$$r_t \leq 2\alpha_{i_t,t-1}\sqrt{\mathbf{x}_t^\top \widetilde{V}_{t-1}^{-1}\mathbf{x}_t} \cdot \sqrt{\frac{\det(\widetilde{V}_{t-1})}{\det(V_{i_t,t-1})}} = O\left(\sqrt{d\log\frac{T}{\delta}}\right) \cdot \|\mathbf{x}_t\|_{\widetilde{V}_{t-1}^{-1}} \cdot \sqrt{\frac{\det(\widetilde{V}_{t-1})}{\det(V_{i_t,t-1})}} \tag{3.72}$$

where the determinant ratio reflects the additional regret due to the delayed synchronization between client i_t's local sufficient statistics and the global optimal oracle. Therefore, *minimizing this ratio directly corresponds to reducing client i_t's regret*. For example, full communication keeps the ratio at 1, which recovers the $\tilde{O}(\sqrt{T})$ regret of the centralized setting.

Therefore, given the client's desire in regret minimization, data itself can be used as a form of incentive by the server. And the star-shaped communication network also gives the server an information advantage over any single client in the system: a client can only communicate with the server, while the server can communicate with every client. Therefore, the server should utilize this advantage to create incentives (i.e., the LHS of Eq (3.70)), and a natural design to evaluate this data incentive is:

$$\mathcal{I}_{i,t} := \mathcal{I}_{i,t}^d = \frac{\det\left(D_{i,t}(S_t) + V_{i,t}\right)}{\det(V_{i,t})} - 1. \tag{3.73}$$

where $D_{i,t}(S_t) = \sum_{j:\{\Delta V_{j,t} \in S_t\} \wedge \{j \neq i\}} \Delta V_{j,t} + \Delta V_{-i,t}$ denotes the data that the server can offer to client i during the communication at time t (i.e., current local updates from other participants that have not been shared with the server) and $\Delta V_{-i,t}$ is the historically aggregated updates stored in server that has not been shared with client i. Eq. (3.73) suggests a substantial increase in the determinant of the client's local data is desired by the client, which results in regret reduction.

With the above data valuation in Eq. (3.73), we propose the *payment-free* incentive mechanism that motivates clients to share data by redistributing data collected from participating clients. We present this mechanism in Algorithm 18, and briefly sketch it below. First, we initiate the participant set $S_t = \widetilde{S}_t$, assuming all clients agree to participate. Then, we iteratively update S_t by checking the willingness of each client i in S_t according to Eq. (3.70). If S_t is empty or all clients in it are participating, then terminate; otherwise, remove client i from S_t and repeat the process.

While this payment-free incentive mechanism is neat and intuitive, it has no guarantee on the amount of data that can be collected. To see this, we provide a theoretical negative result with rigorous regret analysis in Theorem 3.2.3 (see proof in Section 3.2.7).

Algorithm 19 Payment-efficient Incentive Mechanism

```
1: Input $\widetilde{S}_t = \{\Delta V_{i,t} | i \in [N]\}$, data-incentivized participant set $\widehat{S}_t \subseteq \widetilde{S}_t$, threshold $\beta$
2: for client $i : \Delta V_{i,t} \in \widetilde{S}_t \setminus \widehat{S}_t$ do
3:     Compute client's potential contribution to the server (i.e., marginal gain in determinant):
           $$c_{i,t}(\widehat{S}_t) = \det(\Delta V_{i,t} + V_{g,t}(\widehat{S}_t)) / \det(V_{g,t}(\widehat{S}_t)), \;\; V_{g,t}(S_t) = V_{g,t-1} + \Sigma(S_t) \tag{3.75}$$
4: Rank clients $\{i_1, \dots, i_m\}$ by their potential contribution, where $m = |\widetilde{S}_t \setminus \widehat{S}_t|$
5: Segment the list by finding $\alpha = \min\{j \mid \frac{\det(V_{g,t}(\widehat{S}_t) + \Delta V_{i_j,t})}{\det(V_{g,t}(\widetilde{S}_t))} \geq \beta, \; \forall j \in [m]\}$
6: $k = \alpha - 1, \mathcal{I}^m_{\text{last}} = D^p_{i_\alpha} - \mathcal{I}^d_{i_\alpha,t}$
7: Return participant set $S_t = \textit{Heuristic Search}(k, \mathcal{I}^m_{\text{last}})$        {// Algorithm 20}
```

Theorem 3.2.3 (Sub-optimal Regret)**.** *When there are at most $\frac{c}{2C}\frac{N}{\log(T/N)}$ number of clients (for some constant $C, c > 0$), whose cost value $D^p_i \leq (1 + \frac{L^2}{\lambda})^T$, there exists a linear bandit instance with $\sigma = L = S = 1$ such that for $T \geq Nd$, the expected regret for INC-FEDUCB algorithm with payment-free incentive mechanism is at least $\Omega(d\sqrt{NT})$.*

Note that when there is no communication R_T is upper bounded by $O(d\sqrt{NT})$. Hence, in the worst case scenario, the payment-free incentive mechanism might not motivate any client to participate. It is thus not a sufficient mechanism.

3.2.5 Payment-efficient incentive mechanism

To address the insufficiency issue, we further devise a *payment-efficient* incentive mechanism that introduces additional monetary incentives to motivate clients' participation:

$$\mathcal{I}_{i,t} := \mathcal{I}^d_{i,t} + \mathcal{I}^m_{i,t} \tag{3.74}$$

where $\mathcal{I}^d_{i,t}$ is the data incentive defined in Eq (3.73), and $\mathcal{I}^m_{i,t}$ is the real-valued monetary incentive, i.e., the payment assigned to client for its participation. Specifically, we are intrigued by the question: rather than trivially paying unlimited amounts to ensure everyone's participation, can we devise an incentive mechanism that guarantees a certain level of client participation such that the overall regret is still nearly optimal but under acceptable monetary incentive cost?

Inspired by the determinant ratio principle discussed in Eq. (3.72), we propose to control the overall regret by ensuring that every client closely approximates the oracle after each communication round, which can be formalized as $\det(V_{g,t})/\det(\widetilde{V}_t) \geq \beta$, where $V_{g,t} = V_{g,t-1} + \Sigma(S_t)$ is to be shared with all clients and $\Sigma(S_t) = \sum_{j:\{\Delta V_{j,t} \in S_t\}} \Delta V_{j,t}$. The parameter $\beta \in [0, 1]$ characterizes the chosen gap between the practical and optimal regrets that the server commits to. Denote the set of clients motivated by $\mathcal{I}^d_{i,t}$ at time t as S^d_t and those motivated by $\mathcal{I}^m_{i,t}$ as S^m_t, and thus $S_t = S^m_t \cup S^d_t$. At each communication round, the server needs to find the minimum $\mathcal{I}^m_{i,t}$ such that pooling local updates from S_t satisfies the required regret reduction for the entire system.

Algorithm 18 maximizes $\mathcal{I}^d_{i,t}$, and thus the servers should compute $\mathcal{I}^m_{i,t}$ on top of optimal $\mathcal{I}^d_{i,t}$ and resulting S^d_t, which however is still combinatorially hard. First, a brute-force search can yield a time complexity up to $O(2^N)$. Second, different from typical optimal subset selection problems [149], the dynamic interplay among clients in our specific context brings a unique challenge: once a client is incentivized to share data, the other uninvolved clients may change their willingness due to the increased data incentive, making the problem even more intricate.

To solve the above problem, we propose a heuristic ranking-based method, as outlined in Algorithm 19. We rank clients by the marginal gain they bring to the server's determinant, as formally defined in Eq (3.75). This helps minimize the number of clients requiring monetary incentives, while empowering the participation of other clients motivated by the aggregated data. This forms an iterative search process. First, we rank all m non-participating clients (Line 2-3) by their potential contribution to the server (with participant set S_t committed). Then, we segment the list by β, anyone whose participation satisfies the overall β gap constraint is an immediately valid choice (Line 4). The first client i_α in the valid list and its payment $\mathcal{I}^m_{\text{last}}$ (∞ if not available) will be our *last resort* (Line 5). Lastly, we check if there exist potentially more favorable solutions from the invalid list (Line 6). Specifically, we try to elicit up to $k = \alpha - 1$ ($k = m$ if i_α is not available) clients from the invalid list in $n \leq k$ rounds, where only one client will be chosen using the same heuristic in each round. If having n clients from the invalid list also satisfies the β constraint and results in a reduced monetary incentive cost compared to $\mathcal{I}^m_{\text{last}}$, then we opt for this alternative solution. Otherwise, we will adhere to the *last resort*. This *Heuristic Search* is detailed in the following paragraph, and it has a time complexity of only $O(N)$ in the worst-case scenarios, i.e., $n = m = N$.

Algorithm 20 Heuristic Search

```
1:  Input invalid client list $\{i_1, i_2, \cdots, i_k\}$, data-incentivized participant set $\widehat{S}_t$, and the last resort cost $\mathcal{I}^m_{\text{last}}$
2:  Initialization: $S_t = \widehat{S}_t$
3:  for $n \in [k]$ do
4:    Rank clients $\{i_1, \ldots, i_{k-n+1}\}$ (in new order) by Eq (3.75)
5:    $S_t = S_t \cup \{i_{k-n+1}\}$                {// add the client with the largest contribution}
6:    for client $j \in \{i_1, \ldots, i_{k-n}\}$ do
7:      Compute data incentive $\mathcal{I}^d_{j,t}$ for client $j$ by Eq (3.73)      {// find extra data-incentivized participants}
8:      if $\mathcal{I}^d_{j,t} > D^p_j$ then
9:        $S_t = S_t \cup \{\Delta V_{j,t}\}$
10:   Compute total payment $\mathcal{I}^m_{n,t} = \sum_{i \in \widetilde{S}_t \setminus S_t} \mathcal{I}^m_{i,t}$ by Eq (3.74)
11:   if $\mathcal{I}^m_{n,t} \leq \mathcal{I}^m_{\text{last}}$ then
12:     Return $S_t = \widehat{S}_t \cup \{\Delta V_{i_\alpha,t}\}$                {// return last resort}
13:   else
14:     if $\det(\Sigma(S_t) + V_{g,t-1}) / \det(\Sigma(\widetilde{S}_t) + V_{g,t-1}) > \beta$ then
15:       Return $S_t$                {// return search result}
```

Heuristic Search Algorithm As sketched in Section 3.2.5, we devised an iterative search method based on the following ranking heuristic (formally defined in Eq (3.75)): the more one client assists in increasing the server's determinant, the more valuable its contribution is, and thus we should motivate the most valuable clients to participate. Denote $n \leq k$ (initialized as 1) as the number of clients to be selected from the invalid list $\{i_1, \ldots, i_k\}$, and initialize the participant set $S_t = \widehat{S}_t$. In each round n, we rank the remaining $k - n + 1$ clients based on their potential contribution to the server by Eq (3.75), and add the most valuable one to S_t (Line 3-4). With the latest S_t committed, we then proceed to determine additional data-incentivized participants by Eq (3.73) (Line 5-8), and compute the total payment by Eq 3.74 (Line 9). If having n clients results in the total cost $\mathcal{I}^m_{n,t} > \mathcal{I}^m_{\text{last}}$, then we terminate the search and resort to our *last resort* (Line 10-11). Otherwise, if the resulting S_t enables the server to satisfy the β gap requirement, then we successfully find a better solution than *last resort* and can terminate the search. However, if having n client is insufficient for the server to pass the β gap requirement, we increase $n = n + 1$ and repeat the search process (Line 12-14). In particular, if the above process fails to terminate (i.e., having all m clients still not suffices, we will still use the *last resort*. Note that, by utilizing matrix computation to calculate the contribution list in each round, this method only incurs a linear time complexity of $O(N)$, when $n = m = N$.

Theorem 3.2.4 guarantees the sufficiency of this mechanism *w.r.t* communication and payment bounds (proof given in Section 3.2.7).

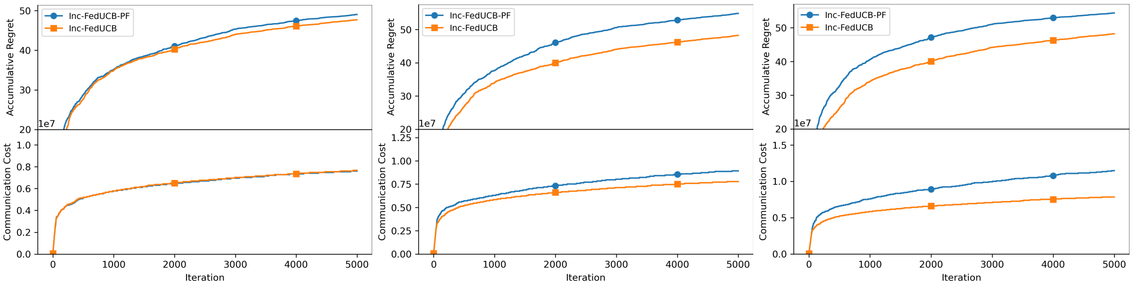
Theorem 3.2.4. *Under threshold β and clients' committed data sharing cost $D^p = \{D^p_1, \cdots, D^p_N\}$, with high probability the monetary incentive cost of INC-FEDUCB satisfies*

$$M_T = O\left(\max D^p \cdot P \cdot N - \sum_{i=1}^{N} P_i \cdot \left(\frac{\det(\lambda I)}{\det(V_T)}\right)^{\frac{1}{P_i}}\right).$$

where P_i is the number of epochs client i participated throughout time horizon T, P is the total number of epochs, which is bounded $P = O(Nd\log T)$ by setting communication threshold $D_c = \frac{T}{N^2 d \log T} - \sqrt{\frac{T^2}{N^2 dR \log T}} \log \beta$, *where* $R = \lceil d \log(1 + \frac{T}{\lambda d}) \rceil$. *Henceforth, the communication cost satisfies* $C_T = O(Nd^2) \cdot P = O(N^2 d^3 \log T)$. *Furthermore, by setting* $\beta \geq e^{-\frac{1}{N}}$, *the cumulative regret is* $R_T = O\left(d\sqrt{T}\log T\right)$.

3.2.6 Experiment setup & results

We simulate the incentivized federated bandit problem under various environment settings. Specifically, we create an environment of $N = 50$ clients with cost of data sharing $D^p = \{D_1^p, \cdots, D_N^p\}$, total number of iterations $T = 5,000$, feature dimension $d = 25$, and time-varing arm pool size $K = 25$. By default, we set $D_i^p = D_\star^p \in \mathbb{R}, \forall i \in [N]$.



(a) $D_\star^p = 1$

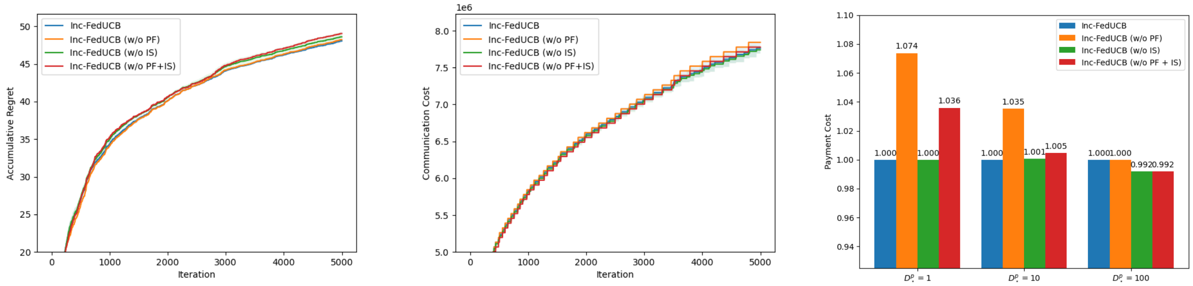
(b) $D_\star^p = 10$

(c) $D_\star^p = 100$

Figure 3.4: Comparison between payment-free vs. payment-efficient incentive designs.

Payment-free vs. Payment-efficient We first empirically compared the performance of the payment-free mechanism (named as INC-FEDUCB-PF) and the payment-efficient mechanism INC-FEDUCB in Figure 3.4. It is clear that the added monetary incentives lead to lower regret and communication costs, particularly with increased $D_\star^p$. Lower regret is expected as more data can be collected and shared; while the reduced communication cost is contributed by reduced communication frequency. When less clients can be motivated in one communication round, more communication rounds will be triggered as the clients tend to have outdated local statistics.

Ablation Study on Heuristic Search To investigate the impact of different components in our heuristic search, we compare the full-fledged model INC-FEDUCB with following variants on various environments: (1) INC-FEDUCB (w/o PF): without payment-free incentive mechanism, where the server only use money to incentivize clients; (2) INC-FEDUCB (w/o IS): without iterative search, where the server only rank the clients once. (3) INC-FEDUCB (w/o PF + IS): without both above strategies.



(a) Accumulative Regret

(b) Communication Cost

(c) Payment Cost

Figure 3.5: Ablation Study on Heuristic Search (w.r.t $D_\star^p \in [1, 10, 100]$).

In Figure 3.5, we present the averaged learning trajectories of regret and communication cost, along with the final payment costs (normalized) under different $D_\star^p$. The results indicate that the full-fledged INC-FEDUCB consistently outperforms all other variants in various environments. Additionally, there is a substantial gap between the variants with and without the PF strategy, emphasizing the significance of leverage server's information advantage to motivate participation.

Environment & hyper-parameter study We further explored diverse β hyperparameter settings for INC-FEDUCB in various environments with varying $D^p_\star$, along with the comparion with DisLinUCB [28] (only comparable when $D^p_\star = 0$). As shown in Table 3.1, when all clients are incentivized to share data, our INC-FEDUCB essentially recover the performance of DisLinUCB, while overcoming its limitation in incentivized settings when clients are not willing to share by default. Moreover, by reducing the threshold β, we can substantially save payment costs while still maintaining highly competitive regret, albeit at the expense of increased communication costs. And the reason for this increased communication cost has been explained before: more communication rounds will be triggered, as clients become more outdated.

$d = 25, K = 25$		DisLinUCB	INC-FEDUCB ($\beta = 1$)	INC-FEDUCB ($\beta = 0.7$)	INC-FEDUCB ($\beta = 0.3$)
$T = 5,000, N = 50, D^p_\star = 0$	Regret (Acc.)	48.46	48.46	48.46 ($\Delta = 0\%$)	48.46 ($\Delta = 0\%$)
	Commu. Cost	7,605,000	7,605,000	7,605,000 ($\Delta = 0\%$)	7,605,000 ($\Delta = 0\%$)
	Pay. Cost	\	0	0 ($\Delta = 0\%$)	0 ($\Delta = 0\%$)
$T = 5,000, N = 50, D^p_\star = 1$	Regret (Acc.)	\	48.46	47.70 ($\Delta - 1.6\%$)	48.38 ($\Delta - 0.2\%$)
	Commu. Cost	\	7,605,000	7,668,825 ($\Delta + 0.8\%$)	7,733,575 ($\Delta + 1.7\%$)
	Pay. Cost	\	75.12	60.94 ($\Delta - 18.9\%$)	22.34 ($\Delta - 70.3\%$)
$T = 5,000, N = 50, D^p_\star = 10$	Regret (Acc.)	\	48.46	48.21 ($\Delta - 0.5\%$)	47.55 ($\Delta - 1.9\%$)
	Commu. Cost	\	7,605,000	7,779,425 ($\Delta + 2.3\%$)	8,599,950 ($\Delta + 13\%$)
	Pay. Cost	\	12,819.61	9,050.61 ($\Delta - 29.4\%$)	4,859.17 ($\Delta - 62.1\%$)
$T = 5,000, N = 50, D^p_\star = 100$	Regret (Acc.)	\	48.46	48.22 ($\Delta - 0.5\%$)	48.44 ($\Delta - 0.1\%$)
	Commu. Cost	\	7,605,000	7,842,775 ($\Delta + 3.1\%$)	8,718,425 ($\Delta + 14.6\%$)
	Pay. Cost	\	190,882.45	133,426.01 ($\Delta - 30.1\%$)	88,893.78 ($\Delta - 53.4\%$)

Table 3.1: Study on Hyper-Parameter of INC-FEDUCB and Environment

3.2.7 Full proof of INC-FEDUCB algorithm

Proof of Theorem 3.2.3

Our proof relies on the following lower bound result for federated linear bandits established in [97].

Lemma 3.2.5 (Theorem 5.3 of [97]). *Let p_i denote the probability that an agent $i \in [N]$ will communicate with the server at least once over time horizon T. Then for any algorithm with*

$$\sum_{i=1}^{N} p_i \leq \frac{c}{2C} \cdot \frac{N}{\log(T/N)} \tag{3.76}$$

there always exists a linear bandit instance with $\sigma = L = S = 1$, such that for $T \geq Nd$, the expected regret of this algorithm is at least $\Omega(d\sqrt{NT})$.

In the following, we will create a situation, where Eq (3.76) always holds true for payment-free incentive mechanism. Specifically, recall that the payment-free incentive mechanism (Section 3.2.4) motivates clients to participate using only data, i.e., the determinant ratio defined in Eq (3.73) that indicates how much client i's confidence ellipsoid can shrink using the data offered by the server. Based on matrix determinant lemma [150], we know that $\mathcal{I}_{i,t} \leq (1 + \frac{L^2}{\lambda})^T$. Additionally, by applying the determinant-trace inequality (Lemma 10 of [20]), we have $\mathcal{I}_{i,t} \leq (1 + \frac{TL^2}{\lambda d})^d$. Therefore, as long as $D^p_i > \min\{(1 + \frac{L^2}{\lambda})^T, (1 + \frac{TL^2}{\lambda d})^d\}$, where the tighter choice between the two upper bounds depends on the specific problem instance (i.e., either d or T being larger), it becomes impossible for the server to incentivize client i to participate in the communication. Now based on Lemma 3.2.5, if the number of clients that satisfy $\mathcal{I}_{i,t} \leq (1+\frac{L^2}{\lambda})^T$ is smaller than $\frac{c}{2C} \cdot \frac{N}{\log(T/N)}$, a sub-optimal regret of the order $\Omega(d\sqrt{NT})$ is inevitable for payment-free incentive mechanism, which finishes the proof. □

Proof of Theorem 3.2.4

To prove this theorem, we first need the following lemma.

Lemma 3.2.6 (Communication Frequency Bound)**.** *By setting the communication threshold* $D_c = \frac{T}{N^2 d \log T} - \sqrt{\frac{T^2}{N^2 dR \log T}} \log \beta$*, the total number of epochs defined by the communication rounds satisfies,*

$$P = O(d \log T)$$

where $R = \lceil d \log(1 + \frac{T}{\lambda d}) \rceil = O(d \log T)$.

Proof of Lemma 3.2.6. Denote P as the total number of epochs divided by communication rounds throughout the time horizon T, and V_{g,t_p} as the aggregated covariance matrix at the p-th epoch. Specifically, $V_{g,t_0} = \lambda I$, $\widetilde{V}_T$ is the covariance matrix constructed by all data points available in the system at time step T.

Note that according to the incentivized communication scheme in INC-FEDUCB, not all clients will necessarily share their data in the last epoch, hence $\det(V_{g,t_P}) \leq \det(\widetilde{V}_T) \leq \left(\frac{tr(\widetilde{V}_T)}{d}\right) \leq (\lambda + T/d)^d$. Therefore,

$$\log \frac{\det(V_{g,t_P})}{\det(V_{g,t_{P-1}})} + \log \frac{\det(V_{g,t_{P-1}})}{\det(V_{g,t_{P-2}})} + \cdots + \log \frac{\det(V_{g,t_1})}{\det(V_{g,t_0})} = \log \frac{\det(V_{g,t_P})}{\det(V_{g,t_0})} \leq \left\lceil d \log(1 + \frac{T}{\lambda d}) \right\rceil$$

Let $\alpha \in \mathbb{R}^+$ be an arbitrary positive value, for epochs with length greater than α, there are at most $\lceil \frac{T}{\alpha} \rceil$ of them. For epochs with length less than α, say the p-th epoch triggered by client i, we have

$$\Delta t_{i,t_p} \cdot \log \frac{\det(V_{i,t_p})}{\det(V_{i,t_{\text{last}}})} > D_c$$

Combining the β gap constraint defined in Section 3.2.5 and the fact that the server always downloads to all clients at every communication round, we have $\Delta t_{i,t_p} \leq \alpha$ and hence

$$\log \frac{\det(g, V_{t_p})}{\det(V_{g,t_{p-1}})} \geq \log \frac{\beta \cdot \det(\widetilde{V}_{t_p})}{\det(V_{g,t_{p-1}})} \geq \log \frac{\beta \cdot \det(V_{i,t_p})}{\det(V_{g,t_{p-1}})} \geq \log \frac{\beta \cdot \det(V_{i,t_p})}{\det(V_{i,t_{\text{last}}})} \geq \frac{D_c}{\alpha} + \log \beta$$

Let $R = \lceil d \log(1 + \frac{T}{\lambda d}) \rceil = O(d \log T)$, therefore, there are at most $\lceil \frac{R}{\frac{D_c}{\alpha} + \log \beta} \rceil$ epochs with length less than α time steps. As a result, the total number of epochs $P \leq \lceil \frac{T}{\alpha} \rceil + \lceil \frac{R}{\frac{D_c}{\alpha} + \log \beta} \rceil$. Note that $\lceil \frac{T}{\alpha} \rceil + \lceil \frac{R}{\frac{D_c}{\alpha} + \log \beta} \rceil \geq 2\sqrt{\frac{TR}{D_c + \alpha \log \beta}}$, where the equality holds when $\alpha = \sqrt{\frac{T(D_c + \alpha \log \beta)}{R}}$.

Furthermore, let $D_c = \frac{T}{N^2 d \log T} - \alpha \log \beta$, then $\alpha = \sqrt{\frac{T^2}{N^2 dR \log T}}$, we have

$$P = O\left(\sqrt{\frac{TR}{D_c + \alpha \log \beta}}\right) = O(N\sqrt{dR \log T}) = O(Nd \log T) \tag{3.77}$$

This concludes the proof of Lemma 3.2.6. □

Communication Cost The proof of communication cost upper bound directly follows Lemma 3.2.6. In each epoch, all clients first upload $O(d^2)$ scalars to the server and then download $O(d^2)$ scalars. Therefore, the total communication cost is $C_T = P \cdot O(Nd^2) = O(N^2 d^3 \log T)$ □

Monetary incentive cost Under the clients' committed data sharing cost $D^p = \{D^p_1, \cdots, D^p_N\}$, during each communication round at time step t_p, we only pay clients in the participant set S_{t_p}. Specifically, the payment (i.e., monetary incentive cost) $\mathcal{I}^m_{i,t_p} = 0$ if the data incentive is already sufficient to motivate the client to participate, i.e., when $\mathcal{I}^d_{i,t_p} \geq D^p_i$. Otherwise, we only need to pay the minimum amount of monetary incentive such that Eq (3.70) is satisfied, i.e., $\mathcal{I}^m_{i,t_p} = D^p_i - \mathcal{I}^d_{i,t_p}$. Therefore, the accumulative monetary incentive cost is

$$\begin{aligned}
M_T = \sum_{p=1}^{P}\sum_{i=1}^{N}\mathcal{I}^m_{i,t_p} &= \sum_{p=1}^{P}\sum_{i=1}^{N}\max\left\{0, D^p_i - \mathcal{I}^d_{i,t_p}\right\}\cdot\mathbb{I}(\Delta V_{i,t_p}\in S_{t_p}) \\
&\leq \sum_{p=1}^{P}\sum_{i=1}^{N}\max\left\{0, \max_{i\in[N]}\{D^p_i\} - \mathcal{I}^d_{i,t_p}\right\}\cdot\mathbb{I}(\Delta V_{i,t_p}\in S_{t_p}) \\
&\leq \sum_{p=1}^{P}\sum_{i\in\bar{\mathcal{N}}_p}(\max_{i\in[N]}\{D^p_i\} - \mathcal{I}^d_{i,t_p})\cdot\mathbb{I}(\Delta V_{i,t_p}\in S_{t_p}) \\
&\leq \max_{i\in[N]}\{D^p_i\}\sum_{p=1}^{P}\sum_{i=1}^{N}\mathbb{I}(\Delta V_{i,t_p}\in S_{t_p}) - \sum_{p=1}^{P}\sum_{i\in\bar{\mathcal{N}}_p}\mathcal{I}^d_{i,t_p}\cdot\mathbb{I}(\Delta V_{i,t_p}\in S_{t_p}) \\
&= \max_{i\in[N]}\{D^p_i\}\sum_{p=1}^{P}N_p - \sum_{i=1}^{N}\sum_{p\in\bar{\mathcal{P}}_i}\mathcal{I}^d_{i,t_p}
\end{aligned}$$

where P and N represent the number of epochs and clients, N_p is the number of participants in p-th epoch, $\bar{\mathcal{N}}_p$ is the set of money-incentivized participants in the p-th epoch, $\bar{\mathcal{P}}_i$ is the set of epochs where client i gets monetary incentive, whose size is denoted as $P_i = |\bar{\mathcal{P}}_i|$. Denote $D^p_{\max} = \max_{i\in[N]}\{D^p_i\}$ to simplify our later discussion.

Recall the definition of data incentive and $D_{i,t_p}(S_{t_p}) = \sum_{j:\{\Delta V_{j,t_p}\in S_{t_p}\}\wedge\{j\neq i\}}\Delta V_{j,t_p} + \Delta V_{-i,t_p}$ introduced in Eq (3.73), we can show that

$$\begin{aligned}
\mathcal{I}^d_{i,t_p} &= \frac{\det\left(D_{i,t_p}(S_{t_p}) + V_{i,t_p}\right)}{\det(V_{i,t_p})} - 1 \\
&\geq \frac{\det(V_{g,t_p})}{\det(V_{i,t_p})} - 1
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
M_T &\leq D^p_{\max}\cdot\sum_{p=1}^{P}N_p + \sum_{i=1}^{N}\sum_{p\in\bar{\mathcal{P}}_i}1 - \sum_{i=1}^{N}\sum_{p\in\bar{\mathcal{P}}_i}\frac{\det(V_{g,t_p})}{\det(V_{i,t_p})} \\
&\leq D^p_{\max}\cdot\sum_{p=1}^{P}N_p + \sum_{i=1}^{N}P_i - \sum_{i=1}^{N}P_i\cdot\left(\frac{\det(V_{g,t_1})}{\det(V_{i,t_1})}\cdot\frac{\det(V_{g,t_2})}{\det(V_{i,t_2})}\cdots\frac{\det(V_{g,t_{P_i}})}{\det(V_{i,t_{P_i}})}\right)^{\frac{1}{P_i}} \\
&\leq D^p_{\max}\cdot\sum_{p=1}^{P}N_p + \sum_{i=1}^{N}P_i - \sum_{i=1}^{N}P_i\cdot\left(\frac{\det(V_{g,t_1})}{\det(V_{i,t_1})}\cdot\frac{\det(V_{i,t_1})}{\det(V_{i,t_2})}\cdots\frac{\det(V_{i,t_{P_i-1}})}{\det(V_{i,t_{P_i}})}\right)^{\frac{1}{P_i}} \\
&= D^p_{\max}\cdot\sum_{p=1}^{P}N_p + \sum_{i=1}^{N}P_i - \sum_{i=1}^{N}P_i\cdot\left(\frac{\det(V_{g,t_1})}{\det(V_{i,t_{P_i}})}\right)^{\frac{1}{P_i}} \\
&\leq (1 + D^p_{\max})\cdot P\cdot N - \sum_{i=1}^{N}P_i\cdot\left(\frac{\det(\lambda I)}{\det(V_T)}\right)^{\frac{1}{P_i}}
\end{aligned}$$

where the second step holds by Cauchy-Schwarz inequality and the last step follows the facts that $P_i \leq P$, $N_p \leq N$, $\det(V_{g,t_1}) \geq \det(\lambda I)$, and $\det(V_{i,t_{P_i}}) \leq \det(V_T)$.

Specifically, by setting the communication threshold $D_c = \frac{T}{N^2 d \log T} - \sqrt{\frac{T^2}{N^2 dR \log T}} \log \beta$, where $R = \lceil d \log(1 + \frac{T}{\lambda d}) \rceil$, we have the total number of epochs $P = O(N d \log T)$ (Lemma 3.2.6). Therefore,

$$\begin{aligned} M_T &\leq (1 + D_{\max}^p) \cdot O(N^2 d \log T) - \sum_{i=1}^{N} P_i \cdot \left(\frac{\det(\lambda I)}{\det(V_T)} \right)^{\frac{1}{P_i}} \\ &= O(N^2 d \log T) \end{aligned}$$

which finishes the proof. □

Cumulative regret To prove the regret upper bound, we first need the following lemma.

Lemma 3.2.7 (Instantaneous Regret Bound). *Under threshold β, with probability $1 - \delta$, the instantaneous pseudo-regret $r_t = \langle \theta^*, \mathbf{x}^* - \mathbf{x}_t \rangle$ in j-th epoch is bounded by*

$$r_t = O\left(\sqrt{d \log \frac{T}{\delta}} \right) \cdot \|\mathbf{x}_t\|_{\widetilde{V}_{t-1}^{-1}} \cdot \sqrt{\frac{1}{\beta} \cdot \frac{\det(V_{g,t_j})}{\det(V_{g,t_{j-1}})}}$$

Proof of Lemma 3.2.7. Denote $\widetilde{V}_t$ as the covariance matrix constructed by all available data in the system at time step t. As introduced in Eq (3.72), the instantaneous regret of client i is upper bounded by

$$r_t \leq 2\alpha_{i_t, t-1} \sqrt{\mathbf{x}_t^\top \widetilde{V}_{t-1}^{-1} \mathbf{x}_t} \cdot \sqrt{\frac{\det(\widetilde{V}_{t-1})}{\det(V_{i_t, t-1})}} = O\left(\sqrt{d \log \frac{T}{\delta}} \right) \cdot \|\mathbf{x}_t\|_{\widetilde{V}_{t-1}^{-1}} \cdot \sqrt{\frac{\det(\widetilde{V}_{t-1})}{\det(V_{i_t, t-1})}}$$

Suppose the client i_t appears at the j-th epoch, i.e., $t_{j-1} \leq t \leq t_j$. As the server always downloads the aggregated data to every client in each communication round, we have

$$\frac{\det(\widetilde{V}_t)}{\det(V_{i_t, t})} \leq \frac{\det(\widetilde{V}_t)}{\det(V_{i_t, t_{j-1}})} \leq \frac{\det(\widetilde{V}_t)}{\det(V_{g, t_{j-1}})}$$

Combining the β gap constraint defined in Section 3.2.5, we can show that

$$\frac{\det(\widetilde{V}_t)}{\det(V_{i_t, t})} \leq \frac{\det(\widetilde{V}_t)}{\det(V_{g, t_{j-1}})} \leq \frac{\det(V_{g,t_j})/\beta}{\det(V_{g, t_{j-1}})} = \frac{1}{\beta} \cdot \frac{\det(V_{g,t_j})}{\det(V_{g, t_{j-1}})}$$

Lastly, plugging the above inequality into Eq (3.72), we have

$$r_t = O\left(\sqrt{d \log \frac{T}{\delta}} \right) \cdot \|\mathbf{x}_t\|_{\widetilde{V}_{t-1}^{-1}} \cdot \sqrt{\frac{1}{\beta} \cdot \frac{\det(V_{g,t_j})}{\det(V_{g,t_{j-1}})}}$$

which finishes the proof of Lemma 3.2.7. □

Now, we are ready to prove the accumulative regret upper bound. Similar to DisLinUCB [28], we group the communication epochs into *good epochs* and *bad epochs*.

Good epochs Note that for good epochs, we have $1 \leq \frac{\det(V_{g,t_j})}{\det(V_{g,t_{j-1}})} \leq 2$. Therefore, based on Lemma 3.2.7, the instantaneous regret in good epochs is

$$r_t = O\left(\sqrt{d \log \frac{T}{\delta}} \right) \cdot \|\mathbf{x}_t\|_{\widetilde{V}_{t-1}^{-1}} \cdot \sqrt{\frac{2}{\beta}}$$

Denote the accumulative regret among all good epochs as REG_{good}, then using the Cauchy–Schwarz inequality we can see that

$$
\begin{aligned}
REG_{good} &= \sum_{p \in P_{good}} \sum_{t \in \mathcal{B}_p} r_t \\
&\leq \sqrt{T \cdot \sum_{p \in P_{good}} \sum_{t \in \mathcal{B}_p} r_t^2} \\
&\leq O\left(\sqrt{T \cdot d \log \frac{T}{\delta} \cdot \frac{2}{\beta} \sum_{p \in P_{good}} \sum_{t \in \mathcal{B}_p} \|\mathbf{x}_t\|^2_{\widetilde{V}_{t-1}^{-1}}}\right)
\end{aligned}
$$

Combining the fact $x \leq 2\log(1+x), \forall x \in [0,1]$ and Lemma A.2, we have

$$
\begin{aligned}
REG_{good} &\leq O\left(\sqrt{T \cdot \frac{d}{\beta} \log \frac{T}{\delta} \sum_{p \in P_{good}} \sum_{t \in \mathcal{B}_p} 2\log\left(1 + \|\mathbf{x}_t\|^2_{\widetilde{V}_{t-1}^{-1}}\right)}\right) \\
&\leq O\left(\sqrt{T \cdot \frac{d}{\beta} \log \frac{T}{\delta} \cdot \sum_{p \in P_{good}} \log \frac{\det(\widetilde{V}_{t_p})}{\det(\widetilde{V}_{t_{p-1}})}}\right) \\
&\leq O\left(\sqrt{T \cdot \frac{d}{\beta} \log \frac{T}{\delta} \sum_{p \in P_{All}} \log \frac{\det(\widetilde{V}_{t_p})}{\det(\widetilde{V}_{t_{p-1}})}}\right) \\
&= O\left(\sqrt{T \cdot \frac{d}{\beta} \log \frac{T}{\delta} \cdot \log \frac{\det(\widetilde{V}_{t_P})}{\det(\widetilde{V}_{t_0})}}\right) \\
&\leq O\left(\sqrt{T \cdot \frac{d}{\beta} \log \frac{T}{\delta} \cdot d \log\left(1 + \frac{T}{\lambda d}\right)}\right) \\
&= O\left(\frac{d}{\sqrt{\beta}} \cdot \sqrt{T} \cdot \sqrt{\log \frac{T}{\delta} \cdot logT}\right)
\end{aligned}
$$

Bad epochs Now moving on to the bad epoch. For any bad epoch starting from time step t_s to time step t_e, the regret in this epoch is

$$
REG = \sum_{t=t_s}^{t_e} r_t = \sum_{i=1}^{N} \sum_{\tau \in \mathcal{N}_i(t_e) \setminus \mathcal{N}_i(t_s)} r_\tau
$$

where $\mathcal{N}_i(t) = \{1 \leq \tau \leq t : i_\tau = i\}$ denotes the set of time steps when client i interacts with the environment up to t. By standard optimism argument for linear bandit [20, 28], we have

$$
r_\tau \leq \min\{2, 2\alpha_{i_\tau,\tau-1}\sqrt{\mathbf{x}_\tau^\top V_{i_\tau,\tau-1}^{-1}\mathbf{x}_\tau}\} = O\left(\sqrt{d \log \frac{T}{\delta}}\right) \min\{1, \|\mathbf{x}_\tau\|_{V_{i_\tau,\tau-1}^{-1}}\}
$$

Therefore,

$$
\begin{aligned}
REG &\leq O\left(\sqrt{d\log\frac{T}{\delta}}\right)\sum_{i=1}^{N}\sum_{\tau\in\mathcal{N}_i(t_e)\setminus\mathcal{N}_i(t_s)}\min\{1,\|\mathbf{x}_\tau\|_{V_{i,\tau-1}^{-1}}\} \\
&\leq O\left(\sqrt{d\log\frac{T}{\delta}}\right)\sum_{i=1}^{N}\sqrt{\Delta t_{i,t_e}\sum_{\tau\in\mathcal{N}_i(t_e)\setminus\mathcal{N}_i(t_s)}\min\{1,\|\mathbf{x}_\tau\|^2_{V_{i,\tau-1}^{-1}}\}} \\
&\leq O\left(\sqrt{d\log\frac{T}{\delta}}\right)\sum_{i=1}^{N}\sqrt{\Delta t_{i,t_e}\sum_{\tau\in\mathcal{N}_i(t_e)\setminus\mathcal{N}_i(t_s)}\log\left(1+\|\mathbf{x}_\tau\|^2_{V_{i,\tau-1}^{-1}}\right)} \\
&= O\left(\sqrt{d\log\frac{T}{\delta}}\right)\sum_{i=1}^{N}\sqrt{\Delta t_{i,t_e}\sum_{\tau\in\mathcal{N}_i(t_e)\setminus\mathcal{N}_i(t_s)}\log\left(\frac{\det(V_{i,\tau})}{\det(V_{i,\tau-1})}\right)} \\
&\leq O\left(\sqrt{d\log\frac{T}{\delta}}\right)\sum_{i=1}^{N}\sqrt{\Delta t_{i,t_e}\cdot\log\frac{\det(V_{i,t_e})}{\det(V_{i,t_{\text{last}}})}} \\
&\leq O\left(\sqrt{d\log\frac{T}{\delta}}\right)N\cdot\sqrt{D_c}.
\end{aligned}
$$

where the second step holds by the Cauchy-Schwarz inequality, the third step follows from $x \leq 2\log(1+x), \forall x \in [0,1]$, the fourth step utilizes the elementary algebra, and the last two steps follow the fact that no client triggers the communication before t_e.

Recall that, as introduced in Lemma 3.2.6, the number of bad epochs is less than $R = \lceil d\log(1+\frac{T}{\delta})\rceil = O(d\log T)$, therefore the regret across all bad epochs is

$$
REG_{bad} = O\left(\sqrt{d\log\frac{T}{\delta}}\right)N\cdot\sqrt{D_c}\cdot O(d\log T) = O\left(Nd^{1.5}\sqrt{D_c\cdot\log\frac{T}{\delta}}\log T\right)
$$

Combining the regret for all good and bad epochs, we have accumulative regret

$$
R_T = REG_{good} + REG_{bad} = O\left(\frac{d}{\sqrt{\beta}}\cdot\sqrt{T}\cdot\sqrt{\log\frac{T}{\delta}\cdot\log T}\right) + O\left(Nd^{1.5}\sqrt{D_c\cdot\log\frac{T}{\delta}}\log T\right)
$$

According to Lemma 3.2.7, the above regret bound holds with high probability $1-\delta$. For completeness, we also present the regret when it fails to hold, which is bounded by $\delta\cdot\sum r_t \leq 2T\cdot\delta$ in expectation. And this can be trivially set to $O(1)$ by selecting $\delta = 1/T$. In this way, we can primarily focus on analyzing the following regret when the bound holds.

$$
R_T = O\left(\frac{d}{\sqrt{\beta}}\sqrt{T}\log T\right) + O\left(Nd^{1.5}\log^{1.5}T\cdot\sqrt{D_c}\right)
$$

With our choice of $D_c = \frac{T}{N^2 d\log T} - \sqrt{\frac{T^2}{N^2 dR\log T}}\log\beta$ in Lemma 3.2.6, we have

$$
R_T = O\left(\frac{d}{\sqrt{\beta}}\sqrt{T}\log T\right) + O\left(Nd^{1.5}\log^{1.5}T\cdot\sqrt{\frac{T}{N^2 d\log T} - \sqrt{\frac{T^2}{N^2 dR\log T}}\log\beta}\right)
$$

Plugging in $R = \lceil d \log(1 + \frac{T}{\lambda d}) \rceil = O(d \log T)$, we get

$$R_T = O\left(\frac{d}{\sqrt{\beta}}\sqrt{T}\log T\right) + O\left(Nd^{1.5}\log^{1.5} T \cdot \sqrt{\frac{T}{N^2 d \log T} + \frac{T}{Nd\log T}\log\frac{1}{\beta}}\right)$$

Furthermore, by setting $\beta > e^{-\frac{1}{N}}$, we can show that $\frac{T}{N^2 d \log T} > \frac{T}{Nd\log T}\log\frac{1}{\beta}$, and therefore

$$R_T = O\left(\frac{d}{\sqrt{\beta}}\sqrt{T}\log T\right) + O\left(d\sqrt{T}\log T\right) = O\left(d\sqrt{T}\log T\right)$$

This concludes the proof. □

3.3 Mediate content creator competition for social welfare

Online recommendation platforms such as Instagram and YouTube have become prevalent in our daily life [151]. At the core of those platforms is a recommender system (RS) designed to match each user with the most relevant content based on predicted relevance. Such a practice, often referred to as the top-K recommendation, is believed to improve user satisfaction and has served as a rule-of-thumb principle in both academia and industry for decades [105, 104, 109].

Until recently, the community came to realize that users' utilities cannot be maximized unilaterally due to the potential strategic behaviors of content creators [4]. Because the content creators' utilities are directly tied to their content's exposure, they are motivated to adaptively maximize their own utilities. This leads to competition that may potentially harm the social welfare (defined as the total user satisfaction/engagement) [152]. For example, consider a scenario where the user population contains a large group of sports fans and a small group of travel enthusiasts. Social welfare is maximized when the available content for recommendation covers both topics. However, one possible equilibrium of the competition is that all creators post homogeneous sports content when the gain from creating niche content cannot compensate for the utility loss caused by abandoning the exposure from the majority of users. It is thus urgent to understand in the long run how bad the social welfare loss could be under strategic content creators driven by a top-K RS.

In this work, we propose the *competing content creation game* to model the impact of the creators' competition on user engagement in a top-K RS. We measure the social welfare guarantee through the lens of Price of Anarchy (PoA) [153], which quantifies the inefficiency of selfish behavior by the ratio between the worst-case welfare value of the game's equilibrium and that of an optimal outcome. Some previous works touched upon this question under different competition models, and their answers are all pessimistic. For example, [154] noticed that the PoA of social welfare under the RS implemented by a Shapley mediator is unbounded. [155] studied a competition model in 1-dimensional space and showed that the PoA under the top-1 matching principle could be as bad as a constant 2. These negative results are either based on a deterministic user choice model or assume creators compete for the shares of *content exposure*. We overturn these pessimistic conclusions by showing that the PoA induced by a top-K RS is at most $1 + O(\frac{1}{\log K})$ when (1) $K > 1$, (2) user choices have mild stochastic noises, and (3) creators are incentivized to compete for *user engagement* instead of content exposure. We also prove its tightness by analyzing a lower-bound instance. Thus an RS under these assumptions will approach the optimal efficiency (i.e., PoA ratio approaches 1) when K grows, though at a relatively slow rate of $1/\log K$. Notably, our PoA upper bound also holds in dynamic settings where creators gradually learn to improve their strategies in an online fashion. Extensive synthetic and real-world data based simulations also support these theoretical findings. Overall, our results robustly demonstrate that content creation competitions are efficient under properly set incentives.

Our results rely on three key assumptions, all of which find their roots in recommendation literature and practice. First, on the platform side, we assume the top-K RS is based on a relevance function that best predicts user satisfaction if recommended content is consumed. To simplify our setting, we assume the true relevance function is known to the RS, since a tremendous amount of research has been spent on this aspect [151, 105, 104, 109] and the goal of our study is not to improve its estimation. Second, on the user side, we employ the well-established *Random Utility (RU) model* [156] to specify the distributional structure of a user's choices and resulting utility when presented with a list of recommendations. The RU model has been widely adopted and found its success in marketing research to model consumer choices [157]. Third, on the creator side, we assume that their utilities collected from matching their content with a user are proportional to the user's utility, as it is a common practice by platforms to set revenue sharing with content creators proportional to the user's satisfaction or engagement [158, 159, 160, 161]. When we move on to the dynamic setting where the creators do not have oracle access to their utility functions, we allow creators to adopt arbitrary no-regret learning algorithms, which cover a variety of rational learning behaviors.

3.3.1 Related works

The theoretical studies of content creators' strategic behavior under the mediation of an RS date back to the seminal works from [154, 162], where they extended the game setting in search and ranking systems [163, 164] and proposed an RS based on Shapley value that leads to the unique PNE and several fairness-related requirements. However, they showed that the social welfare under the proposed Shapley mediator could be arbitrarily bad.

Another line of work studies the RS with strategic content creators under the Hotelling's spatial competition framework [165]. First introduced by [165], Hotelling's model studied two restaurants trying to determine their locations to attract customers who are evenly distributed on the segment $[0, 1]$. The Nash equilibrium (NE) of the resulting game is that both restaurants locate at the center, known as the "principle of minimum differentiation". Recently, [166] proposed a variant of Hotelling's competition in which each player has its attraction region, and they showed that the PoA is 2 in the worse case. We show that their game settings are special cases of our proposed competing content creation game in Section 3.3.7, and thus our main result directly implies their PoA bound. A more closely related work is from [155], where they introduce the RS into the competition as a mediator who directs users to facilities. They studied mediators with different levels of intervention and proposed a limited intervention mediator with a good trade-off between social welfare and intervention cost. Interestingly, their game setting under a no-intervention mediator also turns out to be a special case of ours. We also note that the problem settings and theoretical discussions in both [166] and [155] are limited to pure strategy in 1-dimensional cases with a distance-induced user utility function, while our model and result apply to arbitrary dimensions and a generic form of user utility functions.

Two recent works [167, 168] studied the supply-side competition where the creators' strategy space is high dimensional. Their models assume creators directly compete for user exposure without considering the role of an RS. They focused on the characterization of NE and the identification of conditions under which specialization among creators' strategies may occur. In contrast, we study the social welfare under the impact of a top-K RS without being limited to the existence of NE, and our result applies to general user utility functions.

Our user decision model (see Section 3.3.2) stems from the RU model [157] in econometrics, which explains how an individual makes choices among a discrete set of alternatives. In the RU model, the utility that a decision maker could obtain from alternative j is decomposed into $U_j = V_j + \epsilon_j$, where V_j is the known parameterized part, and ϵ_j is the unknown stochastic part. The observed choice is then given by the alternative with the maximum utility. It is shown that if the unobserved stochastic utility follows the extreme value distribution (i.e., Gumbel distribution), then the choice probability is given by the logit formula, i.e., $P_j \propto \exp(V_j)$ [169]. In our work, we apply the RU model to explain how a typical user allocates her attention across the recommended list.

To analyze the equilibrium efficiency of the competing content creation game, we employed the standard framework of the price of anarchy (PoA). This originates from the seminal work of [153] and has since led to an extensive literature on understanding the efficiency of numerous strategic games. Our discussion by no means can do justice to this rich literature; here, we only mention the few works that are closely related to ours. Since Nash equilibrium (NE) is not guaranteed to exist in our problem with non-continuous agent utilities [167], it is thus crucial for us to consider a solution concept that is weaker than NE and thus to prove a stronger PoA bound. Specifically, we consider coarse correlated equilibrium (CCE). The PoA for CCE is first studied by [170], who considered the efficiency of a dynamic setup with no-regret learners and coined the new notion of the price of total anarchy, which turns out to be equivalent to the PoA bound for CCE. This is precisely the question we want to address, but the structure of our new competing content creation game is significantly different from the games they studied, such as Hotelling's game on a graph and the valid utility game of [171]. Thus their techniques are not readily applicable to our problem. We instead employed a recent framework of [172] using the smoothness argument. It is well-known that this framework can yield strong PoA bound applicable to CCE. However, the bounds obtained by this powerful framework are usually not tight; so far, it is only known that it yields tight PoA bounds for linear cost congestion games [173], second price auctions [174], and the valid utility games [171]. Interestingly, We show that the smoothness argument also yields a tight PoA bound for our competing content creation game and thus register an additional member to this important list of games.

3.3.2 Competing content creation game

In this section, we formalize the competing content creation game. The game $\mathcal{G}$ is defined by a tuple $(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ with the following ingredients:

1. A finite set of users $\mathcal{X} = \{\mathbf{x}_j \in \mathbb{R}^d\}_{j=1}^m$, and a set of players (i.e., content creators[4]) denoted by $[n] = \{1, \cdots, n\}$. Each player i can take an action $\boldsymbol{s}_i$, often referred to as a *pure strategy* in game-theoretic literature, from an action set[5] $\mathcal{S}_i \subset \mathbb{R}^d$. $\boldsymbol{s}_i$ can be understood as the embedding for the *type* of content that creator i can produce. Let $\mathcal{S} = \prod_{i=1}^n \mathcal{S}_i$ denote the space of joint strategies. As a convention, for any $\boldsymbol{s} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \in \mathcal{S}$, we use $\boldsymbol{s}_{-i}$ to denote the joint strategy $\boldsymbol{s}$ excluding $\boldsymbol{s}_i$. Moreover, we use $\boldsymbol{\alpha}_i \in \Delta(\mathcal{S}_i)$ to denote a mixed strategy of player i, which is a probability measure with support $\mathcal{S}_i$. Similarly, $\boldsymbol{\alpha} \in \Delta(\mathcal{S})$ is used to represent a (possibly correlated) joint strategy profile distribution over all players.

2. A relevance function $\sigma(\boldsymbol{s}, \mathbf{x}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ which measures the *relevance* between a user $\mathbf{x} \in \mathcal{X}$ and content $\boldsymbol{s}$. Without loss of generality, we normalize σ to $[0, 1]$, where 1 suggests perfect matching. We focus on modeling the strategic behavior of creators and thus abstract away the estimation of σ.

3. Recommendation policy: given a joint strategy $\boldsymbol{s} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \in \mathcal{S}$ for all players, for each user $\mathbf{x}_j$, the RS first calculates the relevance scores $\{\sigma(\boldsymbol{s}_i, \mathbf{x}_j)\}_{i=1}^n$ over all available content and then generates $\mathcal{T}_j(\boldsymbol{s}; K)$, the subset of $\boldsymbol{s}$ containing the top-K recommendations for user j. Formally,

$$\mathcal{T}_j(\boldsymbol{s}; K) = \{\boldsymbol{s}_{l_i} | i = 1, \cdots, K\}, \tag{3.78}$$

where $(l_i)_{i=1}^n$ is a permutation of $[n]$ such that $\sigma(\boldsymbol{s}_{l_1}, \mathbf{x}_j) \geq \sigma(\boldsymbol{s}_{l_2}, \mathbf{x}_j) \geq \cdots \geq \sigma(\boldsymbol{s}_{l_n}, \mathbf{x}_j)$.[6]

4. User utility and choice model: we employ the widely adopted random utility (RU) model to capture users' utility and choices of recommendations. Formally, the RU model assumes that the utility for user $\mathbf{x}_j$ to consume content $\boldsymbol{s}_i$ is $\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i$, where ε_i is a noise term containing any additional uncertainty that cannot be captured by the RS's prediction $\sigma(\boldsymbol{s}_i, \mathbf{x}_j)$ (e.g., user's mood at that moment). The RU model assumes that $\{\varepsilon_i\}$ are i.i.d. random, which are often assumed to follow the *Gumbel distribution* with cumulative distribution function $\text{Gumbel}(\mu, \beta) = e^{-e^{-\frac{x-\mu}{\beta}}}$.[7] We further assume ε_i is zero mean, thus implying $\mu = -\beta\gamma$ where $\gamma \approx 0.577$ is the Euler–Mascheroni constant. The variance of $\text{Gumbel}(-\beta\gamma, \beta)$ is $\frac{\pi\beta}{\sqrt{6}}$ and the parameter β measures the noise level.

   Upon receiving the recommended list $\mathcal{T}_j(\boldsymbol{s}; K)$, user j chooses $i_j^* \in \mathcal{T}_j(\boldsymbol{s}; K)$ that maximizes her utility:

$$i_j^* = \arg \max_{\boldsymbol{s}_i \in \mathcal{T}_j(\boldsymbol{s}; K)} \{\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i\}. \tag{3.79}$$

   Note that i_j^* is random, with randomness inherited from $\{\varepsilon_i\}$. Consequently, user j derives the following expected utility π_j from consuming the selected content

$$\pi_j(\boldsymbol{s}) \triangleq \mathbb{E}_{\{\varepsilon_i\}} \left[\max_{\boldsymbol{s}_i \in \mathcal{T}_j(\boldsymbol{s}; K)} \{\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i\} \right]. \tag{3.80}$$

5. Player utilities: following the convention, we assume that each player i's expected utility is the sum of the utilities from users that i served, i.e.,

$$u_i(\boldsymbol{s}) = \sum_{j=1}^m \mathbb{E}[\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i | \mathbf{x}_j \to \boldsymbol{s}_i] \cdot \mathbf{Pr}[\mathbf{x}_j \to \boldsymbol{s}_i], \tag{3.81}$$

[4] We use these two terms interchangeably when there is no ambiguity.

[5] The action sets are not assumed to be finite and thus can be continuous.

[6] When $(l_i)_{i=1}^n$ is not unique, $\mathcal{T}_j(\boldsymbol{s}; K)$ can be the top-K truncation of any such permutation with equal probability.

[7] There are many natural reasons to use the Gumbel noise model. This noise model is nearly indistinguishable from a Gaussian distribution empirically, but has slightly thicker tails, allowing for more aberrant user behavior. The RU model with Gumbel noise is also known as the multinomial logit model [175]. It deeply connects to the discrete choice model [176], quantal response equilibrium to capture bounded rational behaviors [177], and entropy regularizer for optimizing randomized strategies [178].

where "$\mathbf{x}_j \to \boldsymbol{s}_i$" denotes the event i_j^* defined in (3.79) equals i. Elegantly, $\mathbf{Pr}[\mathbf{x}_j \to \boldsymbol{s}_i] \propto e^{\beta\sigma(\mathbf{x}_j, \boldsymbol{s}_i)}$ for any $i \in \mathcal{T}_j(\boldsymbol{s}; K)$ [175] and $\mathbf{Pr}[\mathbf{x}_j \to \boldsymbol{s}_i] = 0$ if $i \notin \mathcal{T}_j(\boldsymbol{s}; K)$.

6. Social welfare: the social welfare function is defined as the total utilities from all the users:

$$W(\boldsymbol{s}) = \sum_{j=1}^{m} \pi_j(\boldsymbol{s}). \tag{3.82}$$

   Note that under the player utility function (3.81), we have $W(\boldsymbol{s}) = \sum_{i=1}^{n} u_i(\boldsymbol{s})$. That is, the social welfare is also the total utility of players.

We remark that the player i's utility defined in (3.81) depends on not only the proportion of users matched with i, but also the user's engagement reflected in the term $\mathbb{E}[\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i | \mathbf{x}_j \to \boldsymbol{s}_i]$. This differs crucially from the settings in Hotelling's competition [165] and its recent applications to recommender systems [166, 155, 167, 168], where players' utilities are set to the total *user exposure*, i.e., total number or proportion of user visits (regardless of how satisfied the users are with the recommendations). Both metrics have been widely used in current industry practice to reward creators [158, 159]. In this work, we primarily consider user engagement (i.e., the previously less studied case) as the creator's utility, and in Section 3.3.4 we will compare it with the *user exposure* metric to highlight their different impact.

Our research question and equilibrium concept. We are particularly interested in quantifying the average social welfare when creators learn to update their strategies adaptively. Specifically, we consider the repeated form of a competing content creation game played by n creators over a period of time T. At each time t, each creator chooses an action, observes the utility induced by all creators' strategies at that round, and uses the feedback to adjust their subsequent actions. Naturally, creators aim to optimize their accumulated utility over the course of interactions. However, in real-world online recommendation platforms, creators can only evaluate the utility of their chosen actions and have to gradually learn their optimal strategies through trial and error with such limited information (i.e., bandit feedback). A natural notion for capturing the "reasonable" learning behavior under such an environment is *no regret*. The (external) regret $R_i(T)$ for player i is defined as the difference between her optimal utility in hindsight and the realized accumulated utilities, i.e.,

$$R_i(T) = \max_{\boldsymbol{s}_i'} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s}_{-i} \sim \boldsymbol{\alpha}_{-i}^t}[u_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i})] - \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}^t}[u_i(\boldsymbol{s})] \tag{3.83}$$

where $\boldsymbol{\alpha}^t = \prod_{i=1}^{n} \boldsymbol{\alpha}_i^t$ denotes the joint-strategy distribution at time t. Player i's learning has no regret if $R_i(T) = o(T)$, or equivalently, the average regret $R_i(T)/T \to 0$ as T goes to infinity. Note that such no-regret algorithms exist since any no-regret adversarial online learning algorithm (e.g., Exp3 in bandit literature [179]) guarantees no regret in such a multi-agent learning setup.

To characterize the outcome under no-regret learning players, we focus on an equilibrium concept termed coarse correlated equilibrium (CCE), as it is well known that the empirical action distribution of any no-regret playing sequence in a repeated game converges to its set of CCEs [170]. The formal definition of CCE is as follows:

Definition 3.3.1. *A* coarse correlated equilibrium *(CCE) is a distribution $\boldsymbol{\alpha}$ over the space of joint-strategy profile $\mathcal{S}$ such that for every player i and every action $\boldsymbol{s}_i' \in \mathcal{S}_i$,*

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}}[u_i(\boldsymbol{s})] \geq \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}}[u_i(\boldsymbol{s}_i', \boldsymbol{s}_{-i})]. \tag{3.84}$$

Thanks to the nice connection between no-regret dynamics and CCE, we first establish the welfare guarantee for CCE in Section 3.3.3 and then extend it to account for the accumulated welfare induced by repeated plays in Section 3.3.4.

We also note that the concept of CCE is particularly useful for two additional reasons. First, CCE always exists in any finite games (thus in our game), hence eliminating the necessity to address the existence of Nash equilibrium (NE), perhaps the most celebrated solution concept, as in previous research [167]. In fact, when the action sets are continuous, the existence of NE (either pure or mixed) cannot be guaranteed in our game as the player utility function defined in (3.81) is not continuous. This is an inherent challenge of the problem, as any change in $\sigma(\boldsymbol{s}, \mathbf{x})$ may result in a different top-K recommendation list $\mathcal{T}_j(\boldsymbol{s}; K)$, leading to dramatically different player utilities. Similar challenges and the non-existence of mixed NE have also been observed by [167], though their utility model and research questions differ from ours. Second, even in situations where NE exists, it is more realistic to assume that players eventually achieve some CCE rather than NE due to various criticisms about NE, including the computational concerns [180] of NE.

3.3.3 The price of anarchy analysis

We analyze the social welfare of any top-K RS under any possible CCE; or more specifically, *how bad can the welfare possibly be due to the competition among self-interested content creators* – compared to the *idealized* non-strategic situation in which the platform can "dictate" all creators' content choices and thus globally optimize the welfare function (3.82). This can be captured by the celebrated concept of the Price of Anarchy (PoA) [153]. As its name indicates, PoA captures the welfare inefficacy due to players' selfish behavior. Our main result in this section is a comprehensive characterization of the PoA of competing content creation games.

Definition 3.3.2 (PoA under CCE)**.** *Define the price of anarchy of a game $\mathcal{G}$ as*

$$PoA(\mathcal{G}) = \frac{\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})}{\min_{\boldsymbol{\alpha}\in CCE(\mathcal{G})} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]}, \tag{3.85}$$

where $CCE(\mathcal{G})$ is the set of CCEs of $\mathcal{G}$.

By definition, PoA$(\mathcal{G}) \geq 1$ always holds and larger values indicate worse welfare. Our choice of the CCE concept leads to the strongest possible welfare guarantee in the sense that any upper bound of PoA under CCE also trivially holds for the PoA under refined solution concepts such as correlated equilibrium (CE), PNE or mixed NE (if they exist), since these are all CCEs as well. Unless otherwise emphasized, any PoA in this work refers to the PoA under CCE.

Matching PoA Upper and Lower Bounds Our main theoretical findings are an upper and lower bound for the PoA, which match with each other and thus demonstrates the tightness of our analysis. We first present the upper bound as follows.

Theorem 3.3.3. *The PoA of any competing content creation game instance $\mathcal{G}$ with parameter $\beta \geq 0$ and $K \geq 1$ satisfies*

$$PoA(\mathcal{G}) < 1 + \frac{1}{c(\beta, K)}, \tag{3.86}$$

where $c(\beta, K)$ is defined as

$$c(\beta, K) = \frac{(b+1)\log(b+K)}{(b+K)(\log(b+K) - \log K)}, b = e^{\frac{1}{\beta}} - 1. \tag{3.87}$$

The proof of Theorem 3.3.3 is intricate and thus the detailed arguments are relegated to Section 3.3.6. The primary challenge in the proof is to analyze various smoothness properties of the welfare and players' utility functions, especially how the welfare function changes after excluding any player i's participation. In Section 3.3.6, we highlight some of the noteworthy properties of the welfare function, including its submodularity, which we develop en route to proving Theorem 3.3.3 but is also of independent merit towards understanding the competing content creation game.

The format of $c(\beta, K)$ may not be intuitive enough for the readers to appreciate the derived PoA upper bound. We thus provide the following observations, which reveal various properties of $c(\beta, K)$ aiding the interpretation of Eq (3.86):

1. For any $\beta > 0$ and $K \geq 1$, we have $c(\beta, K) \geq 1$ and thus PoA$(\mathcal{G}) < 2$ always holds.
2. $c(\beta, K) = 1$ if and only if $K = 1$ or $\beta \to 0$.
3. Fix any $\beta > 0$, $c(\beta, K)$ monotonically increases in K; similarly, fix any $K \geq 1$, $c(\beta, K)$ monotonically increases in β.
4. For sufficiently large β and K, $c(\beta, K) \approx (1+\beta)\log K$ asymptotically, and therefore

$$PoA(\mathcal{G}) < 1 + \frac{1}{(1+\beta)\log K}. \tag{3.88}$$

Based on these observations, Theorem 3.3.3 has multiple interesting and immediate implications. First, the welfare loss under any CCE is at most half in any situation, as the PoA is always upper bounded by 2. The second and third facts above show that such worst-case PoA occurs and only occurs when users' choices are made in a "hard" manner: either the RS dictates the user's choice by setting $K = 1$ or the randomness in users' choices is extremely low (i.e., $\beta \to 0$). Note that in the latter case, the user will only consume the most relevant content (i.e., the top-ranked content) due to small decision randomness.

Second, the welfare guarantee improves as either K increases (i.e., more items are recommended) or β increases (i.e., users' choices have more randomness). Welfare improvement in the latter situation is intuitive because when supplied with multiple items, the user can pick the content with large ε_i (i.e., the reward component that is not predictable by the RS) to gain utility. These together reveal an interesting operational insight that when the RS cannot perfectly predict user utility (i.e., $\beta > 0$), providing more items can help improve social welfare. This justifies top-K recommendation and the necessity of diversity in recommendation [181].

Our following second main result shows that this PoA upper bound is tight, up to negligible constants.

Theorem 3.3.4. *Given any $0 \le \beta \le 1$, $n > 2$ and any $1 \le K \le \min\{n-1, e^{\frac{1}{5\beta}}\}$, there exists a competing content creation game instance $\mathcal{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ such that*

$$PoA(\mathcal{G}) > \frac{n-1}{n} + \frac{1}{1+5\beta\log K}. \tag{3.89}$$

This theorem also implies that the argument we employed for Theorem 3.3.3, which is based on the *smoothness* proof developed by [172], yields a tight PoA bound for our proposed game. The tightness of the smoothness argument is itself an intriguing research question. Only three classes of games are known to enjoy a tight PoA bound derived from the smoothness argument: congestion games with affine cost [173], second price auctions [174], and the valid utility game [171], which are all fundamental classes of games. Theorem 3.3.4 suggests that our competing content creation game subscribes to this list. The proof of Theorem 3.3.4 is to explicitly construct a game instance which provably yields the stated PoA lower bound (see Section 3.3.6 for more details).

3.3.4 Implications of the PoA bounds

We have discussed some direct implications of Theorem 3.3.3. Now we develop new results which are either derived from or can be compared to Theorem 3.3.3 and 3.3.4. They will reveal additional insights from our main theoretical results.

Welfare implications to *learning* content creators. The PoA bounds presented in Theorem 3.3.3 and 3.3.4 are based on the assumption that creators are aware of the game parameters and play some CCEs of the game. While CCE is a reasonable equilibrium concept, one potential critique is that to find the CCE, it is assumed that each creator has knowledge about the system parameters (e.g., all other creators' strategies and the σ function), which can be unrealistic. Fortunately, in real-world scenarios where creators utilize no-regret algorithms to play a repeated competing content creation game with bandit feedback, we can still establish a slightly worse PoA upper bound leveraging the fact that the average strategy history of no-regret players converges to a CCE, as shown in the following Corollary 3.3.4.1.

Corollary 3.3.4.1. *[**Dynamic Version of Theorem 3.3.3**] Suppose each player in a repeated competing content creation game $\mathcal{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ independently executes some no-regret learning algorithm, with worst regret $R(T) = \max_i R_i(T)$ as defined in (3.83). Then we have*

$$\frac{\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})}{\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}^t}[W(\boldsymbol{s})]} < 1 + \left(1 + \frac{n}{\beta\log K}\cdot\frac{R(T)}{T}\right)\cdot\frac{1}{c(\beta,K)}, \tag{3.90}$$

where $\boldsymbol{\alpha}^t$ denotes the joint-strategy distribution at step t and $c(\beta, K)$ is the constant defined in (3.87).

In other words, the average welfare across all rounds $\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}^t}[W(\boldsymbol{s})]$ is close to the maximum possible welfare $\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})$, up to a constant factor. The quantity in the LHS of (3.90) is also known as the "price of total anarchy" [170]. It is a substitute for PoA when we want to characterize the welfare of an outcome from repeated play which does not necessarily fall into any equilibrium concept. The proof of Corollary 3.3.4.1 is presented in Section 3.3.6. Because $R(T)/T \to 0$ as $T \to \infty$ for any no-regret algorithm (most no-regret algorithms have $R(T) = O(\sqrt{T})$), the RHS of (3.90) is still strictly less than 2 for any fixed constants (n, β, K).

Theorem 3.3.3 relies on two crucial platform features: 1. the player's utility in (3.81) is defined as the total user engagement that accounts for the user utility $\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_j$, as opposed to just "user exposure" (i.e., the expected total number of matches); 2. the platform uses the top-K recommendation policy. Next, we illustrate the insights revealed from Theorem 3.3.3 with respect to these two key features.

The importance of rewarding user engagement rather than solely exposure. A key reason for the nice PoA guarantee in our competing content creation game is each player i's utility is chosen as the user engagement $\sum_j \mathbb{E}[\sigma(\boldsymbol{s}_i, \mathbf{x}_j) + \varepsilon_i | \mathbf{x}_j \to \boldsymbol{s}_i] \, \mathbf{Pr}(\mathbf{x}_j \to \boldsymbol{s}_i)$ in (3.81), while not the following user exposure metric:

$$\text{User exposure for player } i: \quad \sum_{j=1}^{m} \mathbf{Pr}(\mathbf{x}_j \to \boldsymbol{s}_i). \tag{3.91}$$

Our next result shows that incentivizing creators to maximize user exposure can lead to significantly worse welfare.

Proposition 1. *Let $\tilde{\mathcal{G}}$ denote the variant of the competing content creation game $\mathcal{G} = (\{\mathcal{S}_i\}_{i=1}^n, \{\mathbf{x}_j\}_{j=1}^m, \sigma, \beta, K)$ by substituting player utility function in* (3.81) *by the above user exposure in* (3.91)*. Then for any $K \geq 1, 0 \leq \beta \leq \min\{0.14, \frac{1}{5 \log K}\}$, there exist $\mathcal{G}$ and $\tilde{\mathcal{G}}$ such that*

$$PoA(\tilde{\mathcal{G}}) > 2 > PoA(\mathcal{G}). \tag{3.92}$$

Moreover, when $K = 1$ or β approaches 0, PoA$(\tilde{\mathcal{G}})$ can be arbitrarily large.

In stark contrast to Theorem 3.3.3 guaranteeing PoA$(\mathcal{G}) < 2$, Proposition 1 implies the deterioration of user welfare when content creators are incentivized to compete for the expected exposure of their content. However, we find in practice both metrics are used: for example, user engagement has been used more often as a reward metric for established creators, whereas user exposure is used more for new creators [158, 159]. Our result serves as a theoretical defense for rewarding creators by user engagement if the system aims to improve overall welfare of the users.

To prove Proposition 1, we construct a game instance in which the user welfare at NE is arbitrarily close to zero. Our construction also reveals interesting insights about situations where user welfare can be very bad. Hence, we briefly explain our construction here and leave our formal arguments in Section 3.3.6. Our constructed game has two groups of users: one *dispersed* group that is fine with any content but is never very happy with it (i.e., a low relevant score for all content) and one *focused* group who looks for a specific type of high-quality content (a high relevance score on such content); but only a small group of specialized creators can produce such high-quality content. However, if players are incentivized to compete for exposure, even creators from the small group tend to produce low-quality content that appeals to the dispersed group rather than high-quality content that benefits the focused group. This, in the worse case, can lead to arbitrarily worse welfare for the platform.

The welfare efficiency of top-K recommendation policy. One may wonder whether the top-K recommendation is indeed a good policy for securing the platform's welfare, i.e., is it possible that other recommendation policies (e.g., a probabilistic policy based on Plackett-Luce model [182, 183]) may even lead to better equilibrium outcomes? Our following analysis, as a corollary of Theorem 3.3.3, shows that the answer is to some extent *no* since *any* recommendation policy cannot be better than the top-K rule by more than a tiny fraction of the theoretical optimality. We believe this finding also serves as a theoretical justification for the wide adoption of the top-K principle in practice.

Corollary 3.3.4.2. *Consider an arbitrary recommendation policy providing at most K recommendations, which induces a different competing content creation game $\mathcal{G}'$. Let $CCE(\mathcal{G}')$ denote the corresponding CCE set of $\mathcal{G}'$ and $W(\mathcal{G}') = \min_{\boldsymbol{\alpha} \in CCE(\mathcal{G}')} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}}[W(\boldsymbol{s})]$ be its worst-case CCE welfare. Then we have*

$$W(\mathcal{G}') \leq W(\mathcal{G}) + W_K^* / \Big(1 + \frac{K \log(K+b)}{K+b}\Big), \tag{3.93}$$

where W_K^* is the best possible social welfare achieved via any centralized recommendation policy with K slots.

As indicated by (3.93), the fraction of the loss of welfare is approximately $O(\frac{1}{\log K})$ as $\frac{K}{K+b} \sim O(1)$ when K is large. The proof is straightforward based on Theorem 3.3.3 and can be found in Section 3.3.6.

3.3.5 Experiment setup & results

To confirm our theoretical findings and also to empirically measure the social welfare induced by creators' competition, we conduct simulations on game instances $\mathcal{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ constructed from two synthetic datasets and the

Table 3.2: PoA under $\beta = 0.1$. Results reflect the worst cases obtained from 10 independently sampled game instances.

n K	$*$	2	3	4	5
1	2.00	1.33	1.54	1.66	1.72
2	1.93	1.28	1.46	1.56	1.60
3	1.89		1.42	1.47	1.51
4	1.86			1.43	1.42
5	1.84				1.42

$*$ denotes the theoretical upper bound.

MovieLens-1m dataset [95]. Before presenting our results, we provide a detailed overview of the simulation environment, including the characteristics of the datasets utilized and the metrics employed for evaluation.

Synthetic dataset-1 Dataset-1 simulates the situation where content creators compete over an unbalanced user interest distribution. We construct n user clusters with the largest cluster containing half of the population, and let each strategy from a creator's action set generate content that only appeals to a specific user group.

Specifically, the user population is given by disjoint clusters $\mathcal{X} = \cup_{i=1}^{n} \mathcal{X}_i$ such that $|\mathcal{X}_1| = \frac{m}{2}$, and the sizes of smaller clusters $|\mathcal{X}_l|$ are sampled uniformly at random such that $\sum_{l=2}^{n} |\mathcal{X}_l| = \frac{m}{2}$. Players share the same action set $\mathcal{S}_i = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$, and the σ function satisfies that for any $i \in [n]$,

$$\sigma(\boldsymbol{s}_i, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{X}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3.94}$$

Dataset-1 depends on the randomness of the partition $\cup_{i=1}^{n} \mathcal{X}_i$.

Synthetic dataset-2 Dataset-2 simulates the situation where content creators can either "chase the trend" by generating mediocre content or cater to a specific user interest group with high-quality content. Similar to the construction of dataset-1, we let the user population comprise of n clusters and allow each player to take actions targeting at any specific user group. But, in addition, we also allow each player to take a "safe" action $\boldsymbol{s}_0$ by producing some popular content that can satisfy all users to a certain extent δ.

Specifically, the user population is also given by disjoint clusters $\mathcal{X} = \cup_{i=1}^{n} \mathcal{X}_i$, where the sizes of all clusters $|\mathcal{X}_l|$ are sampled uniformly at random such that $\sum_{l=i}^{n} |\mathcal{X}_l| = m$. Players share the same action set $\mathcal{S}_i = \{\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$, and the σ function satisfies that for any $i \in [n]$,

$$\sigma(\boldsymbol{s}_i, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{X}_i, i \geq 1 \\ \delta, & \text{if } i = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.95}$$

Dataset-2 depends on the randomness of the partition $\cup_{i=1}^{n} \mathcal{X}_i$ and the parameter $\delta \in [0, 1]$.

MovieLens-1m dataset We use deep matrix factorization [184] to train user and movie embeddings targeted at movie ratings from 1 to 5. The total number of users $m = 6040$, the number of movies $k = 3883$, and the embedding dimension is set to $d = 32$. To validate the quality of the trained representations, we first performed a 5-fold cross-validation and obtain an averaged RMSE =0.883 on the test sets, and then train the user/item embeddings with the complete dataset. The resulting user embeddings $\mathcal{X} = \{\mathbf{x}_j\}_{j \in [m]}$ are used as the user population. To construct each player-i's action set $\mathcal{S}_i$, we randomly sample 500 vectors from the trained movie embedding set $\mathcal{M}$ ($|\mathcal{M}| = 3883$) independently. To normalize the relevance score to $[0, 1]$, we let $\sigma(\boldsymbol{s}, \mathbf{x}) = 1$ when the predicted rating of movie $\boldsymbol{s}$ to user $\mathbf{x}$ is at least 4, i.e., $\sigma(\boldsymbol{s}, \mathbf{x}) = \mathbb{I}[\langle \boldsymbol{s}, \mathbf{x} \rangle \geq 4]$.

Evaluation Metrics We use both PoA and PotA in our experiments. The evaluation of PoA requires solving two optimization problems, which are both intractable in general due to the non-concavity of $W(\cdot)$ and the undetermined structure of CCE($\mathcal{G}$). As a result, we use simulated annealing to approach $\max_{\boldsymbol{s} \in \mathcal{S}} W(\boldsymbol{s})$ when the exact computation

Table 3.3: PotA under $\beta = 0.1$. Results reflect the worst cases obtained from 10 independently sampled game instances.

K \ n	$*$	5	10	15	20	40
1	2.00	1.59	1.59	1.60	1.50	1.38
3	1.89	1.37	1.39	1.42	1.41	1.32
5	1.84	1.35	1.34	1.33	1.36	1.31
7	1.80		1.30	1.31	1.30	1.29

$*$ denotes the theoretical upper bound.

is intractable. To compute $\min_{\boldsymbol{\alpha} \in CCE(\mathcal{G})} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}}[W(\boldsymbol{s})]$, we compute its exact solution by solving a linear program with k^n variables and kn constraints [185] for small n and a moderate size of action set k. To deal with larger problems, we let each player run Exp3 [179] over $T = 5000$ rounds and compute the price of total anarchy PotA$(\mathcal{G}) = \frac{\max_{\boldsymbol{s} \in \mathcal{S}} W(\boldsymbol{s})}{\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\alpha}^t}[W(\boldsymbol{s})]}$.

Empirical PoA from simulations We first demonstrate the empirical welfare under different game parameter (n, K, β) for dataset-1. We fix $\beta = 0.1$ and report PoA and PotA under varying n and K. Results are reported in Table 3.2 and 3.3. We observe that for fixed n, both PoA and PotA decrease w.r.t. K and β, as revealed in Theorem 3.3.3. Furthermore, under fixed (β, K), PoA approaches its theoretical upper bound as n increases. However, PotA follows this trend for values of n less than 15, but begins to decrease as n increases further. This discrepancy can be attributed to the fact that for larger values of n (i.e., in Table 3.3), the approximated optimal welfare becomes less accurate and as such, the PotA tends to be underestimated.

Comparison between user engagement/exposure metrics Next we investigate the consequence of utilizing two different incentive metrics, namely *user engagement* vs., *user exposure*. However, Dataset-1 is no longer a good benchmark for this purpose, as the utility functions derived under a simple binary valued $\sigma(\cdot, \cdot)$ are almost indistinguishable under these two metrics. To this end, we use dataset-2, which has a more complex $\sigma(\cdot, \cdot)$ function that models the situation in which creators could focus on chasing the trends other than paying attention to the content quality.

We fix $(\beta, K) = (0.1, 2)$ and report PotA under different n and δ. The results, shown in Figure 3.6, demonstrate the advantage of using the user engagement metric, which consistently leads to a smaller PotA across different values of n and δ. For n larger than 10, PotA with user-exposure can exceed 2 as revealed by Proposition 1[8]. The performance gap between the two metrics is more distinct when δ gets smaller, which can be understood as when creators can produce popular content with lower effort, simply using exposure to reward creators can be catastrophic to the total user welfare.

Social welfare under different levels of rationality In this experiment, we aim to investigate the competition outcomes when players utilize online-learning algorithms with varying levels of rationality. To better simulate what happens in practice, we employed the dataset generated from MovieLens-1m [95]. In our simulation, we model the scenario in which each player runs Exp3 under different exploration rates ϵ (i.e., with probability ϵ, each player will take a random action in each round). We use this simulation setup to examine a practical situation, i.e., creators try to optimize their accumulated regret but with bounded rationality: since Exp3 is known to enjoy a sub-linear regret when $\epsilon \sim O(\sqrt{\frac{k \log k}{T}})$ [179], it would be less rational for creators to set ϵ to be too large or too small as it would incur a larger regret $R(T)$. We fix $(\beta, K, T) = (0.1, 5, 1000)$ and report the averaged social welfare over T rounds, i.e., $\bar{W} = \frac{1}{mT}\sum_{t=1}^{T} W(\boldsymbol{s}^{(t)})$ under different n and ϵ, as illustrated in Figure 3.7.

Our results indicate that the optimal exploration rate associated with the maximum welfare is around $\epsilon = 0.1$ across different values of n. When ϵ is set to be either too small or too large, the average welfare decreases, thereby confirming our claim in Corollary 3.3.4.1 that the welfare guarantee deteriorates as the accumulated regret of each player's learning algorithm increases. Additionally, we observed that the average welfare increases when more creators are involved, which is not unexpected given that users will have a higher chance of receiving a satisfactory recommendation when there is a larger pool of content on the platform. Furthermore, when the number of players is sufficiently large ($n = 100$), the welfare is fairly good even when players adopt nearly randomized strategies ($\epsilon = 0.9$).

[8] Again, due to the approximation error in computing optimal W, the PotA could be underestimated as n gets larger.

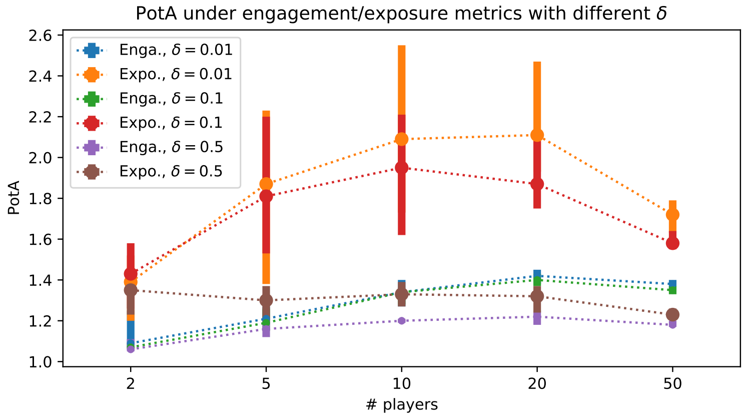


Figure 3.6: PotA under exposure/engagement metrics with $(\beta, K) = (0.1, 2)$. δ is the relevance score obtained from creators' "safe" action. The error bars indicate the largest/smallest values from 10 independent trials and the dots correspond to the mean values.

3.3.6 Full proof of the PoA bounds

Proof highlights of Theorem 3.3.3

Our first step is to derive clean characterizations for the game primitives by utilizing properties of Gumbel distribution. The form of the user utility π_j and welfare W are corollaries of RU models [157], however, the closed-form of the creator utility u_i is a new property we derive.

The main proof of Theorem 3.3.3 is based on a smoothness argument framework developed in the seminal work by [172]. For any strategy profile $\boldsymbol{s}$, $W(\boldsymbol{s}) = \sum_i u_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i})$ is its total welfare function. A game is (λ, μ)-*smooth* if $\lambda W(\boldsymbol{s}') - \mu W(\boldsymbol{s}) \leq \sum_i u_i(\boldsymbol{s}'_i, \boldsymbol{s}_{-i})$ for any $(\boldsymbol{s}, \boldsymbol{s}') \in \mathcal{S}$. [172] observes that the PoA of any (λ, μ)-smooth game can be upper bounded by $\frac{1+\mu}{\lambda}$. After plugging in the expression of $W(\boldsymbol{s})$, the (λ, μ)-smoothness condition can be re-written as

$$\sum_i [\lambda u_i(\boldsymbol{s}'_i, \boldsymbol{s}'_{-i}) - u_i(\boldsymbol{s}'_i, \boldsymbol{s}_{-i})] \leq \sum_i \mu u_i(\boldsymbol{s}).$$

Intuitively, the smoothness parameters bound how much *externality* other players' actions (i.e., $\boldsymbol{s}'_{-i}$ or $\boldsymbol{s}_{-i}$) impose on any player i's utility. Moreover, the tighter this bound is, the smoother the game is and the smaller the PoA is. To gain some intuition and also as a sanity check, consider the extreme situation in which each player's utility is not affected by other players' actions at all (i.e., the *no externality* situation), we have $\lambda = 1$ and $\mu = 0$ implying PoA=1. That is, if any player's utility is not affected by others, then self-interested utility-maximizing behaviors also maximize social welfare, which is a straightforward observation. Certainly, we cannot hope for such a nice property to hold in general, but fortunately, many well-known games have been shown to be smooth. For example, second-price auctions are $(1, 1)$-smooth as shown by [174], congestion games are $(\frac{5}{3}, \frac{1}{3})$-smooth as shown by [172], and all-pay auctions are $(1/2, 1)$-smooth as shown by [186].

Hence, the key challenge in proving Theorem 3.3.3 is to pin down the tightest possible (λ, μ) parameters for our competing content creation game. This boils down to a fundamental question in top-K RS – i.e., *to what extent does the existence of other competing content creators affect a creator's utility*? To answer this question, we discover multiple interesting properties of the welfare and creator utility functions formulated as follows. Besides proving our main result in Theorem 3.3.3, we believe these properties are also of interest for us to understand recommender systems.

Our Lemma 3.3.5 demonstrates the *sub-modularity* of $W(\boldsymbol{s})$. That is, the marginal gain of welfare from adding a new player decreases as the total number of creators increases. Lemma 3.3.6 further relates this marginal welfare increase with the added player's own utility. It shows that the increased welfare after introducing a new player i with strategy $\boldsymbol{s}_i$ is at most i's utility under $\boldsymbol{s}_i$, multiplied by a shrinkage factor $c^{-1}(\beta, K) \in (0, 1]$. These two lemmas

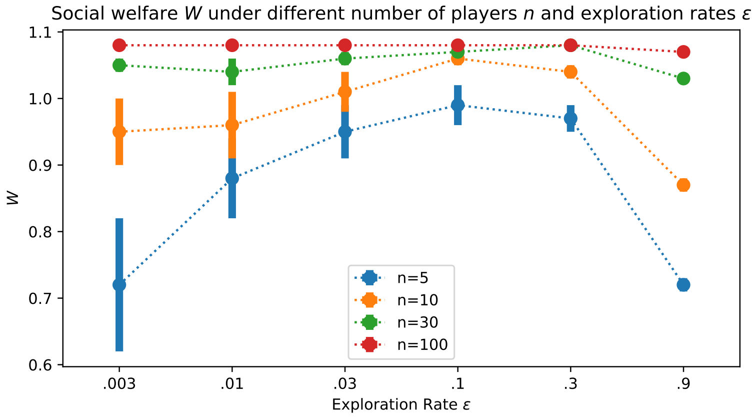


Figure 3.7: The averaged welfare $\bar{W}$ over $T = 1000$ rounds under different exploration rate ϵ and number of players n. Results are averaged over 10 independent runs under $(\beta, K) = (0.1, 5)$.

together allow us to prove that the competing content creation game is $(c^{-1}(\beta, K), c^{-1}(\beta, K))$-smooth, yielding Theorem 3.3.3.

Lemma 3.3.5. *[Submodularity of Welfare] For any $\boldsymbol{s} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \in \mathcal{S}$, let $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$. The social welfare function defined in Eq (3.82) is submodular as a set function, i.e., for any $S, \boldsymbol{s}_x, \boldsymbol{s}_y$ it holds that*

$$W(S \cup \{\boldsymbol{s}_x\}) - W(S) \geq W(S \cup \{\boldsymbol{s}_x, \boldsymbol{s}_y\}) - W(S \cup \{\boldsymbol{s}_y\}).$$

Lemma 3.3.6. *[Smoothness of Welfare] For any $\boldsymbol{s} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \in \mathcal{S}$, $i \in [n]$ and $c(\beta, K)$ defined in Eq (3.87), player-i's utility function $u_i(\boldsymbol{s})$ defined in Eq (3.81) satisfies*

$$W(\boldsymbol{s}) - W(\boldsymbol{s}_{-i}) \leq c^{-1}(\beta, K) \cdot u_i(\boldsymbol{s}_i; \boldsymbol{s}_{-i}).$$

Full proof of Theorem 3.3.3

First we derive the closed-forms of the utility and welfare functions of competing content creation game.

Lemma 3.3.7 (Closed forms of utility and welfare functions)**.** *Given $\{\varepsilon_i\}$ are drawn i.i.d. from zero-mean Gumbel$(-\beta\gamma, \beta)$, the utility and welfare functions defined in (3.80), (3.81) and (3.82) have the following closed forms*

$$\pi_j(\boldsymbol{s}) = \beta \log \Big[\sum_{\boldsymbol{s}_k \in \mathcal{T}_j(\boldsymbol{s}; K)} \exp\left(\beta^{-1}\sigma(\boldsymbol{s}_k, \mathbf{x}_j)\right)\Big], \tag{3.96}$$

$$u_i(\boldsymbol{s}) = \sum_{j=1}^{m} \pi_j(\boldsymbol{s}) \frac{\mathbb{I}[\boldsymbol{s}_i \in \mathcal{T}_j(\boldsymbol{s}; K)] \exp(\beta^{-1}\sigma(\boldsymbol{s}_i, \mathbf{x}_j))}{\sum_{\boldsymbol{s}_k \in \mathcal{T}_j(\boldsymbol{s}; K)} \exp(\beta^{-1}\sigma(\boldsymbol{s}_k, \mathbf{x}_j))}, \tag{3.97}$$

$$W(\boldsymbol{s}) = \beta \sum_{j=1}^{m} \log \Big[\sum_{\boldsymbol{s}_k \in \mathcal{T}_j(\boldsymbol{s}; K)} \exp\left(\beta^{-1}\sigma(\boldsymbol{s}_k, \mathbf{x}_j)\right)\Big]. \tag{3.98}$$

Proof of Lemma 3.3.7. We start with a few known and useful properties of Gumbel distributions.

Lemma 3.3.8. *[e.g., [187]] Let $(v_1, \cdots, v_n) \in \mathbb{R}^n$ be any real-valued vector and $\varepsilon_1, \cdots, \varepsilon_n$ be independent samples from Gumbel(μ, β). Then*

$$\arg\max_i (v_i + \varepsilon_i) \sim \textit{Categorical}\Big(\frac{\exp(\beta^{-1}v_i)}{\sum_{j=1}^{n} \exp(\beta^{-1}v_j)}\Big), \tag{3.99}$$

and

$$\max_i(v_i+\varepsilon_i) \sim \textit{Gumbel}\Big(\mu+\beta\log\Big(\sum_{j=1}^{n}\exp(\beta^{-1}v_j)\Big),\beta\Big). \tag{3.100}$$

Derivation of user utility and welfare. These derivations follow easily from Lemma 3.3.8. Since we assumed that ε_i ∼Gumbel$(-\beta\gamma,\beta)$, leveraging properties in Lemma 3.3.8 we conclude that $\mathbf{x}_j$'s choice distribution over K alternatives $\{\boldsymbol{s}_1,\cdots,\boldsymbol{s}_K\}=\mathcal{T}_j(\boldsymbol{s};K)$ follows the soft-max rule

$$\mathbf{Pr}[\mathbf{x}_j\to\boldsymbol{s}_i]=\frac{\exp(\beta^{-1}\sigma(\boldsymbol{s}_i,\mathbf{x}_j))}{\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp(\beta^{-1}\sigma(\boldsymbol{s}_k,\mathbf{x}_j))}, \tag{3.101}$$

and the expected user utility after making choices has the following form

$$\pi_j(\mathbf{x}_j)=\mathbb{E}\left[\max_{i\in[K]}\{\sigma(\boldsymbol{s}_i,\mathbf{x}_j)+\varepsilon_i\}\right]=\beta\log\left[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp(\beta^{-1}\sigma(\boldsymbol{s}_k,\mathbf{x}_j))\right]. \tag{3.102}$$

Taking expectation over all users, we obtain the following welfare function

$$W(\boldsymbol{s})=\sum_{j=1}^{m}\mathbb{E}\left[\max_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\{\sigma(\boldsymbol{s}_k,\mathbf{x}_j)+\varepsilon_i\}\right]=\beta\sum_{j=1}^{m}\log\left[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\beta^{-1}\sigma(\boldsymbol{s}_k,\mathbf{x}_j)\right)\right]. \tag{3.103}$$

By setting $\tilde{W}(\boldsymbol{s})=\beta W(\boldsymbol{s}),\tilde{\sigma}(\boldsymbol{s},\mathbf{x})=\beta^{-1}\sigma(\boldsymbol{s},\mathbf{x})$, we have $\tilde{W}(\boldsymbol{s})=\sum_{j=1}^{m}\log[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\tilde{\sigma}(\boldsymbol{s}_k,\mathbf{x}_j)\right)]$. Therefore, under a rescaling of constant β it is with out loss of generality to consider a scoring function $\sigma\in[0,\frac{1}{\beta}]$, the user utility function and the social welfare function in the following form

$$\pi_j(\boldsymbol{s})=\log\Big[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\sigma(\boldsymbol{s}_k,\mathbf{x}_j)\right)\Big], \tag{3.104}$$

$$W(\boldsymbol{s})=\sum_{j=1}^{m}\log\Big[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\sigma(\boldsymbol{s}_k,\mathbf{x}_j)\right)\Big]. \tag{3.105}$$

Derivation of creator utility. This turns out to be a new result which requires non-trivial arguments. The players' utility is given by

$$u_i(\boldsymbol{s})=\sum_{j=1}^{m}\mathbb{E}[\sigma(\boldsymbol{s}_i,\mathbf{x}_j)+\varepsilon_i|\mathbf{x}_j\to\boldsymbol{s}_i]\cdot\mathbf{Pr}[\mathbf{x}_j\to\boldsymbol{s}_i] \tag{3.106}$$

$$=\sum_{j=1}^{m}\mathbb{E}[\sigma(\boldsymbol{s}_i,\mathbf{x}_j)+\varepsilon_i|\mathbf{x}_j\to\boldsymbol{s}_i]\cdot\frac{\exp(\sigma(\boldsymbol{s}_i,\mathbf{x}_j))}{\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp(\sigma(\boldsymbol{s}_k,\mathbf{x}_j))}, \tag{3.107}$$

According to the definition in (3.107), what we need to show is that for i.i.d. random variables $\{\varepsilon_i\}_{i=1}^{K}$ sampled from Gumbel$(-\beta\gamma,\beta)$,

$$\mathbb{E}[\sigma(\boldsymbol{s}_i,\mathbf{x}_j)+\varepsilon_i|\mathbf{x}_j\to\boldsymbol{s}_i]=\mathbb{E}[\max_{k\in[K]}\{\sigma(\boldsymbol{s}_k,\mathbf{x}_j)+\varepsilon_i\}]=\log\Big[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\sigma(\boldsymbol{s}_k,\mathbf{x}_j)\right)\Big], \tag{3.108}$$

i.e., for any $(v_1,\cdots,v_K)\in\mathbb{R}^K$ and i.i.d. random variables $\{\varepsilon_i\}_{i=1}^{K}$ sampled from Gumbel$(0,1)$,

$$\mathbb{E}[v_i+\varepsilon_i|i=\arg\max_{k\in[K]}(v_k+\varepsilon_k)]=\gamma+\log\Big(\sum_{k=1}^{K}\exp(v_k)\Big). \tag{3.109}$$

Let $Y_i = \max_{k\in[K],k\neq i}(v_k+\varepsilon_k) \sim \text{Gumbel}(\log(\sum_{k\neq i}\exp(v_k)),1)$ and $X_i = v_i+\varepsilon_i \sim \text{Gumbel}(v_i,1)$. Then X_i has the probability density function

$$f_i(x) = \exp(-((x-v_i)+e^{-(x-v_i)})), \tag{3.110}$$

and Y has the cumulative distribution function

$$F_i(y) = \exp(-e^{-(y-\log(\sum_{k\neq i}\exp(v_k)))}). \tag{3.111}$$

Therefore we can explicitly compute the conditional expectation of X_i as follows:

$$\begin{aligned}
&\mathbb{E}[v_i+\varepsilon_i \mid i = \arg\max_{k\in[K]}(v_k+\varepsilon_k)]\\
=&\mathbb{E}[v_i+\varepsilon_i \mid v_i+\varepsilon_i \geq \max_{k\in[K],k\neq i}(v_k+\varepsilon_k)]\\
=&\mathbb{E}[X \mid X\geq Y, X\sim\text{Gumbel}(v_i,1), Y\sim\text{Gumbel}(\log(\sum_{k\neq i}\exp(v_k)),1)] && (3.112)\\
=&\frac{\int_{\mathbb{R}} x f_i(x)F_i(x)dx}{\int_{\mathbb{R}} f_i(x)F_i(x)dx}\\
=&\frac{\int_{\mathbb{R}} x\exp(-((x-v_i)+e^{-(x-v_i)}))\exp(-e^{-(x-\log(\sum_{k\neq i}\exp(v_k)))})dx}{\int_{\mathbb{R}} \exp(-((x-v_i)+e^{-(x-v_i)}))\exp(-e^{-(x-\log(\sum_{k\neq i}\exp(v_k)))})dx}\\
=&\frac{\int_{\mathbb{R}_{\geq 0}} -\ln t\cdot\exp(-t\sum_{k=1}^K\exp(v_k))dt}{\int_{\mathbb{R}_{\geq 0}}\exp(-t\sum_{k=1}^K\exp(v_k))dt} && (3.113)\\
=&\ln\Big(\sum_{k=1}^K\exp(v_k))\Big)+\frac{\int_{\mathbb{R}_{\geq 0}}-\ln s\cdot\exp(-s)ds}{\int_{\mathbb{R}_{\geq 0}}\exp(-s)ds} && (3.114)\\
=&\ln\Big(\sum_{k=1}^K\exp(v_k))\Big)-\frac{d}{d\alpha}\int_{\mathbb{R}_{\geq 0}} s^\alpha e^{-s}ds\\
=&\ln\Big(\sum_{k=1}^K\exp(v_k))\Big)-\frac{d}{d\alpha}\Gamma(\alpha+1)\Big|_{\alpha=0}\\
=&\ln\Big(\sum_{k=1}^K\exp(v_k))\Big)+\gamma. && (3.115)
\end{aligned}$$

where (3.112) holds because of Lemma 3.3.8, (3.113) and (3.114) hold by change of variables $t = e^{-x}$ and $s = t\sum_{k=1}^K\exp(v_k))$, and (3.115) is from the definition of Euler-Mascheroni constant. Therefore we show (3.109) and the players' utility function has the following form

$$u_i(\boldsymbol{s}) = \sum_{j=1}^m\Big(\log\Big[\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp\left(\sigma(\boldsymbol{s}_k,\mathbf{x}_j)\right)\Big]\Big)\frac{\mathbb{I}[\boldsymbol{s}_i\in\mathcal{T}_j(\boldsymbol{s};K)]\exp(\sigma(\boldsymbol{s}_i,\mathbf{x}_j))}{\sum_{\boldsymbol{s}_k\in\mathcal{T}_j(\boldsymbol{s};K)}\exp(\sigma(\boldsymbol{s}_k,\mathbf{x}_j))}. \tag{3.116}$$

This finishes the proof of Lemma 3.3.7. □

We consider the utility and welfare functions given in (3.104), (3.116) and (3.105) under the re-scaling of constant β with the new assumption that $\sigma(\boldsymbol{s},\mathbf{x})\in[0,\frac{1}{\beta}], \forall\boldsymbol{s}\in\cup_{i=1}^n\mathcal{S}_i, \mathbf{x}\in\mathcal{X}$. To simplify the subsequent analysis, we first specify some useful notations and conventions. For any joint strategy profile $\boldsymbol{s}=(\boldsymbol{s}_1,\cdots,\boldsymbol{s}_n)$, we use capital letter S to denote its set representation, i.e., $S=\{\boldsymbol{s}_1,\cdots,\boldsymbol{s}_n\}$. In this way we can view $\mathcal{T}_j(\boldsymbol{s};K),\pi_j(\boldsymbol{s}),u_i(\boldsymbol{s}),W(\boldsymbol{s})$ defined in (3.78), (3.80), (3.81), (3.82) as set functions $\mathcal{T}_j(S;K),\pi_j(S),u_i(S),W(S)$. From now on, we will use the set notation S and the vector notation $\boldsymbol{s}$ interchangeably, depending on the context. Similarly, we use S_{-i} to denote the set $\{\boldsymbol{s}_1,\cdots,\boldsymbol{s}_n\}$ excluding element $\boldsymbol{s}_i$. Moreover, we extend the definition of $\mathcal{T}_j(S;K)$ by allowing $|S|=K-1$ in the following sense: when $|S| = K - 1$, we let $\mathcal{T}_j(S; K) = S \cup \{\bar{\boldsymbol{s}}\}$, where $\bar{\boldsymbol{s}}$ is a default external choice such that $\sigma(\bar{\boldsymbol{s}}, \mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$. This extension captures the situation when the system does not have enough active content creators to allocate to the users. When such a situation happens, the system will put a default choice $\bar{\boldsymbol{s}}$ in the top-K list without any utility guarantee. We remark that this extended definition is introduced merely for the convenience of presentation and does not affect the implication of our main result.

Prior to the proofs for Lemma 3.3.5 and 3.3.6, we present two intermediate results in Proposition 2 and Lemma 3.3.9. Proposition 2 reveals a rather basic property of social welfare W which is useful in the proof of Theorem 3.3.3, and Lemma 3.3.9 is useful in the proof of Lemma 3.3.6.

Proposition 2. *Fix a joint strategy $S = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\}$ in any $n-$player competing content creation game $\mathcal{G}$. If we add an additional player indexed by $n + 1$ with pure strategy $\boldsymbol{s}_{n+1}$ to the game and let $S' = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n, \boldsymbol{s}_{n+1}\}$, the social welfare W will strictly increase, i.e.,*

$$W(S') > W(S). \tag{3.117}$$

Proof. By definition,

$$W(S') = \sum_{j=1}^{m} \Big(\log \Big[\sum_{\boldsymbol{s} \in \mathcal{T}_j(S';K)} \exp\left(\sigma(\boldsymbol{s}, \mathbf{x}_j)\right) \Big] \Big), \tag{3.118}$$

$$W(S) = \sum_{j=1}^{m} \Big(\log \Big[\sum_{\boldsymbol{s} \in \mathcal{T}_j(S;K)} \exp\left(\sigma(\boldsymbol{s}, \mathbf{x}_j)\right) \Big] \Big), \tag{3.119}$$

It is obvious that for any fixed user j, the sum of exponential scores of top-K choices from S' is better than that from $\mathcal{S}$, i.e.,

$$\sum_{\boldsymbol{s} \in \mathcal{T}_j(S';K)} \exp\left(\sigma(\boldsymbol{s}, \mathbf{x}_j)\right) \geq \sum_{\boldsymbol{s} \in \mathcal{T}_j(S';K)} \exp\left(\sigma(\boldsymbol{s}, \mathbf{x}_j)\right).$$

Therefore, (3.117) holds immediately by the monotonicity of the logarithmic function. □

Proposition 2 reveals an important yet natural property of real-world content provider competitions: when there are more competitors in the market, users are facing more alternatives and thus their welfare will always increase.

Lemma 3.3.9. *The following function*

$$f(x, y) = \frac{(x+1)\log(x+y)}{(x+y)(\log(x+y) - \log y)}, (x, y) \in \mathbb{R}_+ \times \mathbb{N}_+, \tag{3.120}$$

is monotonically increasing in y for any $x \in \mathbb{R}_+$, and is monotonically decreasing in x for any integer $y \in \mathbb{N}_+$.

Proof. We first demonstrate the monotonicity of $f(\cdot, y)$ by directly calculating its partial derivatives. Note that $t \geq \log(1 + t)$ holds for any $t \geq 0$, we have

$$\frac{1}{x+1} \frac{\partial f(x, y)}{\partial y} = \frac{\log(1 + \frac{x}{y}) + \log(x+y)[\frac{x}{y} - \log(1 + \frac{x}{y})]}{[(x+y)(\log(x+y) - \log y)]^2} > 0, \tag{3.121}$$

which implies that $f(x, y)$ is increasing in y. Now it remains to show the monotonicity w.r.t. x when fixing $y = K$, which is slightly more intricate. The derivative of $f(x, K)$ w.r.t. x now writes

$$f'(x, K) = \frac{(K-1)\log(x+K)\log(1 + \frac{x}{K}) - (x+1)\log K}{[(x+K)(\log(x+K) - \log K)]^2} \triangleq \frac{-g(x, K)}{[(x+K)(\log(x+K) - \log K)]^2}, \tag{3.122}$$

and

$$
\begin{aligned}
g'(x,K) &= \frac{1}{x+K}\Big[(2K+x-1)\log K - 2(K-1)\log(x+K)\Big] &(3.123)\\
&= \frac{2(K-1)}{x+K}\Big[\frac{2K+x-1}{2(K-1)}\log K - \log(x+K)\Big]\\
&= \frac{2(K-1)}{x+K}\Big[\frac{x+1}{2(K-1)}\log K - \log(1+\frac{x}{K})\Big] &(3.124)\\
&\geq \frac{2(K-1)}{x+K}\Big[\frac{x+1}{2(K-1)}\log K - \frac{x}{K}\Big]\\
&\geq \frac{x}{x+K}\Big[\log K - \frac{2(K-1)}{K}\Big]. &(3.125)
\end{aligned}
$$

We claim $g'(x,K) \geq 0, \forall K \in \mathbb{N}^+$, and this is because

1. if $K=1$, from (3.123) we have $g'(x,K)=0$.
2. for $K \geq 5$, we can verify $\log K - \frac{2(K-1)}{K} > 0$. From (3.125) we have $g'(x,K) > 0$.
3. for $K \in \{2,3,4\}$, we can verify $\frac{x+1}{2(K-1)}\log K - \log(1+\frac{x}{K}) > 0$ for any $x \geq 0$. Therefore (3.124) we have $g'(x,K) > 0$.

Now since $g'(x,K) \geq 0$, we conclude that $g(x,K) \geq g(0,K) = \log K \geq 0$, which implies $f'(x,K) \leq 0, \forall K \in \mathbb{N}^+$. Hence, $f(x,K)$ is decreasing in x. □

Now we are ready to prove Lemma 3.3.5 and 3.3.6.

Proof of Lemma 3.3.5. By the definition we only need to show the submodularity of $\pi_j(S)$ for any $j \in [m]$, i.e.,

$$\pi_j(\mathcal{T}_j(S \cup \{\boldsymbol{s}_x\}; K)) - \pi_j(\mathcal{T}_j(S;K)) \geq \pi_j(\mathcal{T}_j(S \cup \{\boldsymbol{s}_x, \boldsymbol{s}_y\}; K)) - \pi_j(\mathcal{T}_j(S \cup \{\boldsymbol{s}_y\}; K)). \tag{3.126}$$

With out loss of generality we assume $\sigma(\boldsymbol{s}_x, \mathbf{x}_j) \geq \sigma(\boldsymbol{s}_y, \mathbf{x}_j)$, and let

$$\{v_1, \cdots, v_K\} = \{\exp(\sigma(\boldsymbol{s}, \mathbf{x}_j)) | \boldsymbol{s} \in \mathcal{T}_j(S;K)\},$$

where $v_1 \leq \cdots \leq v_K$. Then depending on the values of $v_x = \exp(\sigma(\boldsymbol{s}_x, \mathbf{x}_j)), v_y = \exp(\sigma(\boldsymbol{s}_y, \mathbf{x}_j))$ and K, there are three situations :

1. $v_x \leq v_1$: (3.126) holds because its LHS and RHS are both equal to 0.
2. $v_x > v_1, K=1$: The LHS of (3.126) is equal to $\log \frac{v_x}{v_1} > 0$, the RHS of (3.126) is equal to 0.
3. $v_x > v_1, K \geq 2$: The LHS of (3.126) is equal to $\log \frac{v_x+v_2+a}{v_1+v_2+a}$, the RHS of (3.126) is equal to $\log \frac{v_x+v_y+a}{v_y+v_2+a}$, where $a = \sum_{k=3}^K v_k$ if $K \geq 3$ and $a=0$ if $K=2$. We can verify

$$
\begin{aligned}
&(v_x+v_2+a)(v_y+v_2+a) - (v_1+v_2+a)(v_x+v_y+a)\\
=&(v_2-v_1)(a+v_1+v_2) + (v_x-v_1)(v_y-v_1) \geq 0.
\end{aligned}
$$

Therefore, (3.126) holds and Lemma 3.3.5 follows by summing (3.126) over all $j \in [m]$.

□

Proof of Lemma 3.3.6. By definition,

$$u_i(\boldsymbol{s}_i; \boldsymbol{s}_{-i}) = \sum_{j=1}^m \Big(\log\Big[\sum_{\boldsymbol{s}' \in \mathcal{T}_j(S;K)} \exp\left(\sigma(\boldsymbol{s}', \mathbf{x}_j)\right)\Big]\Big) \frac{\mathbb{I}[\boldsymbol{s}_i \in \mathcal{T}_j(S;K)] \exp(\sigma(\boldsymbol{s}_i, \mathbf{x}_j))}{\sum_{\boldsymbol{s}' \in \mathcal{T}_j(S;K)} \exp(\sigma(\boldsymbol{s}', \mathbf{x}_j))}, \tag{3.127}$$

and

$$W(S) = \sum_{j=1}^{m} \pi_j(\mathcal{T}_j(S;K)). \tag{3.128}$$

It is sufficient to prove that for any user j,

$$\Big(\log\Big[\sum_{\boldsymbol{s}'\in\mathcal{T}_j(S;K)} \exp\left(\sigma(\boldsymbol{s}',\mathbf{x}_j)\right)\Big]\Big)\frac{\mathbb{I}[\boldsymbol{s}_i\in\mathcal{T}_j(S;K)]\exp(\sigma(\boldsymbol{s}_i,\mathbf{x}_j))}{\sum_{\boldsymbol{s}'\in\mathcal{T}_j(S;K)}\exp(\sigma(\boldsymbol{s}',\mathbf{x}_j))} \ge c(\beta,K)\cdot\big[\pi_j(\mathcal{T}_j(S;K)) - \pi_j(\mathcal{T}_j(S_{-i};K))\big]. \tag{3.129}$$

Note that when $\boldsymbol{s}_i \notin \mathcal{T}_j(S;K)$, (3.129) is trivial as its LHS=RHS=0. Now we suppose $\boldsymbol{s}_i \in \mathcal{T}_j(S;K)$ and thus $\mathcal{T}_j(S_{-i};K)$ and $\mathcal{T}_j(;K)$ only differ in one element. Without loss of generality we let

$$\{\exp(\sigma(\boldsymbol{s},\mathbf{x}_j))|\boldsymbol{s}\in\mathcal{T}_j(S_{-i};K)\} = \{v_1', v_2, \cdots, v_K\},$$

and

$$\{\exp(\sigma(\boldsymbol{s},\mathbf{x}_j))|\boldsymbol{s}\in\mathcal{T}_j(S;K)\} = \{v_1, v_2, \cdots, v_K\}, v_1 \ge v_1'.$$

Because of our extended definition of $\mathcal{T}_j(S;K)$, $\mathcal{T}_j(S_{-i},K)$ is well defined when $K=n$, under which case we have $v_1' = \exp(\sigma(\bar{\boldsymbol{s}},\mathbf{x}_j)) = 1$. Now we let $z = v_2 + \cdots + v_K$, (3.129) is equivalent to

$$\frac{v_1}{v_1+z}\log(v_1+z) \ge c(\beta,K)\cdot\log\Big[\frac{v_1+z}{v_1'+z}\Big]. \tag{3.130}$$

Since $v_1' = \exp(\sigma(\cdot,\mathbf{x}_j)) \ge 1$, a sufficient condition for (3.130) to hold is

$$\frac{v_1}{v_1+z}\cdot\frac{\log(v_1+z)}{\log(v_1+z)-\log(1+z)} \ge c(\beta,K). \tag{3.131}$$

Note that $x = v_1 - 1 \in [0, e^{1/\beta}-1], y = z+1 \in [K, (K-1)e^{1/\beta}+1]$, the LHS of (3.131) becomes a function of (x,y) which has the following form

$$f(x,y) = \frac{(x+1)\log(x+y)}{(x+y)(\log(x+y)-\log y)}. \tag{3.132}$$

From Lemma 3.3.9 we know $f(x,y)$ is monotonically increasing in y for any $x>0$ and is monotonically decreasing in x any integer $K\ge 1$. Therefore, it holds that

$$f(x,y) \ge f(x,K) \ge f(e^{1/\beta}-1,K) = c(\beta,K). \tag{3.133}$$

Hence, (3.129) holds and we complete the proof of Lemma 3.3.6.

□

With the help of Proposition 2, Lemma 3.3.6 and 3.3.5, now we are ready to prove our claim in Theorem 3.3.3. We will demonstrate that any competing content creation game instance $\mathcal{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ is a smooth game with parameter $(\lambda,\mu) = (c(\beta,K), c(\beta,K))$ so that its PoA can be upper bounded by $\frac{1+\mu}{\lambda} = 1 + \frac{1}{c(\beta,K)}$.

Let $\boldsymbol{s} = (\boldsymbol{s}_1, ..., \boldsymbol{s}_n)$ and $\boldsymbol{s}^* = (\boldsymbol{s}_1^*, ..., \boldsymbol{s}_n^*)$ be two different strategy profiles. First, due to function W's sub-modular property disclosed in Lemma 3.3.5, for every $i\in[n]$ we have

$$W([\boldsymbol{s}_i^*, \boldsymbol{s}_{-i}]) - W(\boldsymbol{s}_{-i}) \ge W([\boldsymbol{s}_1^*, \cdots, \boldsymbol{s}_{i-1}^*, \boldsymbol{s}_i^*, \boldsymbol{s}]) - W([\boldsymbol{s}_1^*, \cdots, \boldsymbol{s}_{i-1}^*, \boldsymbol{s}]). \tag{3.134}$$

Summing over all player i we obtain

$$\begin{aligned}\sum_{i=1}^{n}(W([\boldsymbol{s}_i^*, \boldsymbol{s}_{-i}]) - W(\boldsymbol{s}_{-i})) &\geq \sum_{i=1}^{n}(W([\boldsymbol{s}_1^*, \cdots, \boldsymbol{s}_{i-1}^*, \boldsymbol{s}_i^*, \boldsymbol{s}]) - W([\boldsymbol{s}_1^*, \cdots, \boldsymbol{s}_{i-1}^*, \boldsymbol{s}]))\\ &= W([\boldsymbol{s}^*, \boldsymbol{s}]) - W(\boldsymbol{s})\\ &> W(\boldsymbol{s}^*) - W(\boldsymbol{s}), \end{aligned} \tag{3.135}$$

where the last inequality holds because of Proposition 2. On the other hand, from Lemma 3.3.6 it also holds that

$$u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i}) \geq c(\beta, K) \cdot \left[W([\boldsymbol{s}_i^*, \boldsymbol{s}_{-i}]) - W(\boldsymbol{s}_{-i})\right], \tag{3.136}$$

And therefore

$$\sum_{i=1}^{n} u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i}) \geq c(\beta, K) \cdot \sum_{i=1}^{n} \left[W([\boldsymbol{s}_i^*, \boldsymbol{s}_{-i}]) - W(\boldsymbol{s}_{-i})\right] \tag{3.137}$$

$$> c(\beta, K)[W(\boldsymbol{s}^*) - W(\boldsymbol{s})]. \tag{3.138}$$

where inequality (3.137) holds by (3.136), and inequality (3.138) holds by (3.135).

Since (3.138) holds for any $\boldsymbol{s} \in \mathcal{S}$, for any $\boldsymbol{\alpha} \in CCE(\mathcal{G})$ we can take expectation over $\boldsymbol{s} \sim \boldsymbol{\alpha}$ and obtain

$$\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i})] > c(\beta, K)[W(\boldsymbol{s}^*) - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]]. \tag{3.139}$$

Therefore,

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[\sum_{i=1}^{n} u_i(\boldsymbol{s})]$$

$$\geq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[\sum_{i=1}^{n} u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i})] \tag{3.140}$$

$$\geq c(\beta, K) \cdot \sum_{i=1}^{n} \left[\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W([\boldsymbol{s}_i^*, \boldsymbol{s}_{-i}])] - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s}_{-i})]\right]$$

$$> c(\beta, K)[W(\boldsymbol{s}^*) - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]]. \tag{3.141}$$

where inequality (3.140) follows by the definition of CCE and inequality (3.141) holds by (3.138). Rearranging terms we obtain

$$PoA(\mathcal{G}) = \frac{\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})}{\min_{\boldsymbol{\alpha}\in CCE(\mathcal{G})} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]} < 1 + \frac{1}{c(\beta, K)}. \tag{3.142}$$

Proof of the Property of $c(\beta, K)$ The $c(\beta, K)$ function has the following form:

$$c(\beta, K) = \frac{(b+1)\log(b+K)}{(b+K)(\log(b+K) - \log K)}, b = e^{\frac{1}{\beta}} - 1. \tag{3.143}$$

We prove the following facts one by one.

1. Fix any $\beta > 0$, $c(\beta, K)$ is monotonically increasing in K; similarly, fix any $K \geq 1$, $c(\beta, K)$ is monotonically increasing in β.

   Note that $e^{\frac{1}{\beta}} - 1$ is decreasing in β, from Lemma 3.3.9 the claim holds.

2. $c(\beta, K) = 1$ if and only if $K = 1$ or $\beta \to 0$.

When $K = 1$, $c(\beta, K) = 1$ directly holds. When $\beta \to 0$, $b \to +\infty$ and $c(\beta, K) \to 1$. The "only if" direction follows from the monotonicity property of $c(\beta, K)$.

3. For any $\beta > 0$ and $K \geq 1$, we have $c(\beta, K) \geq 1$ and thus $PoA(\mathcal{G}) < 2$ always holds.
   By the monotonicity of c, $c(\beta, K) \geq c(\beta, 1) = 1$. Hence, $PoA(\mathcal{G}) < 1 + \frac{1}{c(\beta,K)} \leq 2$.

4. For sufficiently large β and K, $c(\beta, K) \approx (1 + \beta) \log K$ asymptotically, and therefore

$$PoA(\mathcal{G}) < 1 + \frac{1}{(1+\beta)\log K}. \tag{3.144}$$

When β is sufficiently large, $b = e^{\frac{1}{\beta}} - 1 \approx \frac{1}{\beta} \to 0$. Therefore,

$$\begin{aligned} c(\beta, K) &= \frac{(b+1)\log(b+K)}{(b+K)\log(1+\frac{b}{K})} \\ &\approx \frac{(b+1)K\log(b+K)}{(b+K)b} && \text{since } \log(1+x) \approx x \text{ as } x \to 0 \\ &\approx \frac{(b+1)\log K}{b} && \text{since } K >> b \\ &\approx (1+\beta)\log K. && \text{since } b \approx \tfrac{1}{\beta} \end{aligned}$$

Proof of Corollary 3.3.4.1

Proof. Let $\epsilon(T) = \frac{R(T)}{T}$, and $\boldsymbol{s}^* = (\boldsymbol{s}_1^*, ..., \boldsymbol{s}_n^*)$ be a global maximizer of $W(\boldsymbol{s})$. By definition,

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s})] \geq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s}_i^*, \boldsymbol{s}_{-i})] - \epsilon(T). \tag{3.145}$$

Summing over all player $i \in [n]$ we obtain

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})] = \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s})] \geq \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s}_i^*, \boldsymbol{s}_{-i})] - n\epsilon(T). \tag{3.146}$$

On the other hand, by (3.138) from the proof of Theorem 3.3.3, we have

$$\sum_{i=1}^{n} u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i}) > c(\beta, K)[W(\boldsymbol{s}^*) - W(\boldsymbol{s})], \forall \boldsymbol{s} \in S. \tag{3.147}$$

Taking the expectation of $\boldsymbol{s}$ over distribution $\boldsymbol{\alpha}$ we obtain

$$\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[u_i(\boldsymbol{s}_i^*; \boldsymbol{s}_{-i})] > c(\beta, K)\big(W(\boldsymbol{s}^*) - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]\big). \tag{3.148}$$

(3.146) and (3.148) together imply that

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})] + n\epsilon(T) > c(\beta, K)\big(W(\boldsymbol{s}^*) - \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]\big). \tag{3.149}$$

Note that for any $\boldsymbol{s} \in \mathcal{S}$, we have $W(\boldsymbol{s}) = \sum_{j=1}^{m} \left(\log \left[\sum_{\boldsymbol{s} \in \mathcal{T}_j(\boldsymbol{s};K)} \exp\left(\sigma(\boldsymbol{s}, \mathbf{x}_j)\right) \right] \right) \geq \beta \log K$ and therefore, $n\epsilon(T) \leq \frac{n\epsilon(T)}{\beta \log K} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]$. Substituting it into (3.149), we obtain (3.90).

□

Proof of Corollary 3.3.4.2

Proof. Note that fix any players' strategy profile $\boldsymbol{s}$, the top-K matching mechanism maximizes the social welfare W. Therefore, $W(\mathcal{G}') \leq W_K^*$. On the other hand, from the PoA bound in Theorem 3.3.3 it holds that

$$\begin{aligned}\frac{W_K^*}{W(\mathcal{G})} &< 1 + \frac{(b+K)(\log(1+b/K))}{(b+1)\log(b+K)} \\ &< 1 + \frac{(b+K)(b/K)}{(b+1)\log(b+K)} \\ &= 1 + \frac{b+K}{K\log(b+K)}.\end{aligned}$$

Rearranging term yields $W(\mathcal{G}') - W(\mathcal{G}) \leq W_K^* - W(\mathcal{G}) \leq W_K^*/(1 + \frac{K\log(b+K)}{b+K})$. □

Proof of Theorem 3.3.4

Proof. Let $b = \exp(1/\beta) - 1$. Consider an n-player game where each player-i has the same action set $\mathcal{S}_i = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$. Let the user population $\mathcal{X}$ be a set with size $m = n + (n-1)a$, in which n users have profile $\mathbf{x}_1$ and a users have profile $\mathbf{x}_i$ for $i = 2, \cdots, n$. Here $a = \beta \log K + 1$ is a constant whose choice will become clear later. Let the scoring function σ be the indicator function defined as follow:

$$\sigma(\boldsymbol{s}, \mathbf{x}) = \begin{cases} 1, & \text{if } \boldsymbol{s} = \mathbf{x}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.150}$$

First we lower bound the optimal welfare $\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})$. Consider the joint-strategy profile $\boldsymbol{s}^* = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, under which each user gets one player with σ score 1 and $K-1$ player with σ score 0. In this case, each user-j's utility $\pi_j(\boldsymbol{s}) = \beta\log(b+K)$ and the social welfare $W(\boldsymbol{s}) = m\beta\log(b+K)$. Therefore, the optimal social welfare

$$\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s}) \geq W(\boldsymbol{s}^*) = m\beta\log(b+K). \tag{3.151}$$

Next we show that $\boldsymbol{s} = (\mathbf{x}_1, \mathbf{x}_1, \cdots, \mathbf{x}_1)$ is a pure NE of $\mathcal{G}$ and thus $\boldsymbol{s} \in CCE(\mathcal{G})$. Given players' joint-strategy $\boldsymbol{s}$, n users will be assigned with K players with σ score 1 and $(n-1)a$ users will be assigned with K players with σ score 0. Therefore, the utility for an arbitrary player-i is given by

$$\begin{aligned} u_i(\boldsymbol{s}) &= \Big[n \cdot (\beta\log K + 1) + a(n-1)\cdot\beta\log K\Big]/n \\ &= \beta\log K + 1 + \frac{a(n-1)\beta\log K}{n}. \end{aligned} \tag{3.152}$$

If player-i switches from strategy $\boldsymbol{s}_1$ to $\boldsymbol{s}_j$, n users still get K players with score 1, $(n-2)a$ users get players with score 0, and a users get K players with scores $(1, 0, \cdots, 0)$. Therefore, player-i's utility after the deviation is

$$\begin{aligned} u_i(\boldsymbol{s}_j, \boldsymbol{s}_{-i}) &= n\cdot 0 + a(n-2)\cdot\beta\log K \cdot \frac{1}{n} + a\cdot\beta\log(b+K)\cdot\frac{e^{\frac{1}{\beta}}}{e^{\frac{1}{\beta}} + K - 1} \\ &= \frac{a(n-2)\beta\log K}{n} + \frac{b+1}{b+K}\cdot a\beta\log(b+K). \end{aligned}$$

We can verify that $u_i(\boldsymbol{s}) \geq u_i(\boldsymbol{s}_j, \boldsymbol{s}_{-i})$ for any $2 \leq j \leq n$ if we take

$$a = \beta\log K + 1 \leq \frac{\beta\log K + 1}{\beta\Big(\frac{b+1}{b+K}\log(b+K) - \frac{1}{n}\log K\Big)}. \tag{3.153}$$

This is because: 1. The inequality in (3.153) always holds as $\frac{b+1}{b+K}\log(b+K) < \log(b+1) = \frac{1}{\beta}$ when $\beta \in [0,1]$ (this is due to the monotonicity of $\log x/x$); 2. $u_i(\boldsymbol{s}) = u_i(\boldsymbol{s}_j, \boldsymbol{s}_{-i})$ when $a = \frac{\beta \log K + 1}{\beta\left(\frac{b+1}{b+K}\log(b+K) - \frac{1}{n}\log K\right)}$. Hence, $\boldsymbol{s}$ is an NE of $\mathcal{G}$. Putting (3.151), (3.152), and (3.153) together, we have

$$
\begin{aligned}
PoA(\mathcal{G}) &= \frac{\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})}{\min_{\boldsymbol{\alpha}\in CCE(\mathcal{G})} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]} \geq \frac{W(\boldsymbol{s}^*)}{W(\boldsymbol{s})} \\
&= \frac{m\beta\log(b+K)}{nu_i(\boldsymbol{s})} \geq \frac{m}{nu_i(\boldsymbol{s})} \qquad (3.154) \\
&= \frac{n+(n-1)a}{n(\beta\log K+1)+a(n-1)\beta\log K} \\
&= \frac{n-1}{n} + \frac{1-t^2a(a-1)}{a+ta(a-1)} \quad (t = \frac{n-1}{n}) \\
&> \frac{n-1}{n} + \frac{1}{5a-4} \qquad (3.155) \\
&= \frac{n-1}{n} + \frac{1}{1+5\beta\log K}.
\end{aligned}
$$

where inequality (3.154) holds because $\beta\log(b+K) \geq \beta\log(b+1) = 1$, and (3.155) holds because when $a = 1+\beta\log K \in [1, 1.2]$ and $t = \frac{n-1}{n} \in [0.5, 1)$, it is easy to verify that $\frac{1-t^2a(a-1)}{a+ta(a-1)} > \frac{1}{5a-4}$. It is equivalent to $4t^2a + 4 > 5t^2a^2 + ta$, which is true because $t^2a(5a-4) + ta < a(5a-4) + a < 4$. □

Proof of Proposition 1

Proof. Consider a population of two users $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$ and n players in which one player has two pure strategies and the other $n-1$ players only have access to a single strategy, i.e, $\mathcal{S}_i = \{\boldsymbol{s}_0\}, i = 2, \ldots, n$ and $\mathcal{S}_1 = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$. Let the scoring function σ be

$$
\sigma(\boldsymbol{s}, \mathbf{x}) = \begin{cases} 1, & \text{if } \boldsymbol{s} = \boldsymbol{s}_1, \mathbf{x} = \mathbf{x}_1, \\ \delta, & \text{if } \boldsymbol{s} = \boldsymbol{s}_2, \\ 0, & \text{otherwise.} \end{cases} \qquad (3.156)
$$

We will show that for any given $K \geq 1, 0 \leq \beta \leq \min\{0.14, \frac{1}{5\log K}\}$ there exists $\delta \in (0,1)$ such that the PoA of game $\tilde{\mathcal{G}}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ is always strictly greater than 2.

From the proof of Lemma 3.3.7, the user utility and welfare functions of $\tilde{\mathcal{G}}$ share the same form as in (3.96), (3.98), while the player utility functions of $\tilde{\mathcal{G}}$ have the following form

$$
u_i(\boldsymbol{s}) = \sum_{j=1}^{m} \frac{\mathbb{I}[\boldsymbol{s}_i \in \mathcal{T}_j(\boldsymbol{s}; K)] \exp(\beta^{-1}\sigma(\boldsymbol{s}_i, \mathbf{x}_j))}{\sum_{\boldsymbol{s}_k \in \mathcal{T}_j(\boldsymbol{s}; K)} \exp(\beta^{-1}\sigma(\boldsymbol{s}_k, \mathbf{x}_j))}. \qquad (3.157)
$$

Let $b = \exp(1/\beta) - 1$ and we choose any $\delta \in [\delta_0, 1)$ such that $\exp(\delta_0/\beta) + K - 1 = \frac{2}{\frac{1}{K} + \frac{1}{b+K}}$. Such $\delta_0 \in (0,1)$ must exist because function $f(\delta) = \exp(\delta/\beta) + K - 1$ is monotonically increasing in $[0,1]$ with range $[K, b+K] \supset \frac{2}{\frac{1}{K} + \frac{1}{b+K}}$.

Given such choice of δ, we can verify that

$$\begin{aligned}
2u_1(\boldsymbol{s}_2, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0) &= \frac{2\exp(\delta/\beta)}{\exp(\delta/\beta) + K - 1} \\
&\geq \frac{2\exp(\delta_0/\beta)}{\exp(\delta_0/\beta) + K - 1} \\
&= \frac{\exp(1/\beta)}{\exp(1/\beta) + K - 1} + \frac{1}{K} \\
&\geq \frac{\exp(1/\beta)}{\exp(1/\beta) + K - 1} + \frac{1}{n} = 2u_1(\boldsymbol{s}_1, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0),
\end{aligned}$$

which indicates that $(\boldsymbol{s}_2, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0)$ is a PNE of $\tilde{\mathcal{G}}$. Therefore, by picking $\delta = \delta_0$ we have

$$\begin{aligned}
PoA(\tilde{\mathcal{G}}) &= \frac{\max_{\boldsymbol{s}\in\mathcal{S}} W(\boldsymbol{s})}{\min_{\boldsymbol{\alpha}\in CCE(\tilde{\mathcal{G}})} \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\alpha}}[W(\boldsymbol{s})]} \geq \frac{W(\boldsymbol{s}_1, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0)}{W(\boldsymbol{s}_2, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0)} \\
&= \frac{\log(\exp(1/\beta) + K - 1) + \log K}{2\log(\exp(\delta_0/\beta) + K - 1)} && (3.158) \\
&= \frac{\log(b + K) + \log K}{2\log[2K(b + K)] - 2\log(b + 2K)} && \text{by the choice of } \delta_0 \quad (3.159) \\
&> 2, && (3.160)
\end{aligned}$$

where (3.160) holds because (3.160) is equivalent to

$$(b + 2K)^4 > 16K^3(b + K)^3. \tag{3.161}$$

And we show the correctness of (3.161) by verifying the following situations:

1. when $K \in \{2, 3\}$, (3.161) holds for all $\beta \in [0, 0.14]$, $b = \exp(1/\beta) - 1$.
2. when $K \geq 4$, from $\beta \leq \frac{1}{5\log K}$ we know $b+K = \exp(1/\beta)+K-1 > K^5$ and thus $\frac{(b+2K)^4}{(b+K)^3} > b > K^5 \geq 16K^3$. Therefore, (3.161) holds.

Finally we show that when $K = 1$ or $\beta \to 0$, $PoA(\tilde{\mathcal{G}})$ can be arbitrarily large.

1. when $\beta \to 0$, we have $b \to \infty$. From (3.159) we have for any fixed K,
$$\lim_{\beta\to 0} PoA(\tilde{\mathcal{G}}) = \lim_{b\to+\infty} \left\{ \frac{\log(b + K) + \log K}{2\log[2K(b + K)] - 2\log(b + 2K)} \right\} = \lim_{b\to+\infty} \log(b + K) \to +\infty.$$
2. when $K = 1$, the user's choice is deterministic and thus any $\delta \in (0, 1)$ makes $(\boldsymbol{s}_2, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_0)$ a PNE of $\tilde{\mathcal{G}}$. Let $\delta \to 0$ and from (3.158) we have for any fixed β,
$$\lim_{\delta\to 0} PoA(\tilde{\mathcal{G}}) = \lim_{\delta\to 0} \left\{ \frac{\log(\exp(1/\beta) + K - 1) + \log K}{2\log(\exp(\delta/\beta) + K - 1)} \right\} = \lim_{\delta\to 0} \frac{1}{2\delta} \to +\infty.$$

□

3.3.7 Connections to existing models

As an extended discussion to the related work, we show how our competing content creation games connect to the following three previously proposed competition models for content creators. All the following models do not consider the presence of an RS and match each user with all content creators (players), which corresponds to the case $K = n$ in our setting. Interestingly, we found that each of them turns out to be a special case of our competing content creation game.

Facility location games under the no intervention mediator [155]

Consider the following competing content creation game instance:

1. the user population $\mathcal{X} \subseteq [0,1]$ is a finite set of size m,
2. each player $i \in [n]$ shares the same action set $\mathcal{S}_i = [0,1]$,
3. the scoring function is given by $\sigma(s,x) = |s-x|$,
4. $(\beta, K) = (0, n)$,
5. utility function is the user exposure metric, i.e., $u_i(\boldsymbol{s}) = \sum_{x\in\mathcal{X}} \mathbf{Pr}(x \to s_i)$.

If we let $m \to \infty$ so that $\mathcal{X}$ becomes a continuum with density function g over the unit interval $[0,1]$, the game instance $\tilde{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ [9] defined above is equivalent to the facility location game under the no intervention mediator proposed by [155].

Hotelling-Downs model with limited attraction under support utility functions [166]

Consider the following competing content creation game instance:

1. the user population $\mathcal{X} = \{x_1, \cdots, x_m\} \subseteq [0,1]$ is a finite set of size m,
2. each player $i \in [n]$ shares the same action set $\mathcal{S}_i = [0,1] \times [0,1]$. For each action $\boldsymbol{s}_i = (s_i, w_i)$ taken by player-i, it is associated with an attraction region $R_i = [s_i - \frac{w_i}{2}, s_i + \frac{w_i}{2}] \cap [0,1]$.
3. for each $i \in [n]$, the scoring function is given by $\sigma(\boldsymbol{s}_i, x) = \mathbb{I}[x \in R_i]$,
4. $(\beta, K) = (0, n)$,
5. the utility function is induced by the user engagement metric, i.e., $u_i(\boldsymbol{s}) = \sum_{j=1}^m \pi_j(\boldsymbol{s}) \mathbf{Pr}(x_j \to s_i)$.

In fact, given $\beta = 0$ and the above definition of σ, we can see the utility functions under both exposure and engagement metrics are identical, because it holds that $\pi_j(\boldsymbol{s}) \in \{0,1\}$ and $\pi_j(\boldsymbol{s}) = 1$ if and only if $\mathbf{Pr}(x_j \to s_i) > 0$. We can verify that the game instance $G(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ defined above is equivalent to the Hotelling-Downs model with limited attraction under support utility functions proposed by [166].

Exposure games [167]

Consider the following competing content creation game instance:

1. the user population $\mathcal{X} \subseteq \mathbb{R}^d$ is a finite set of size m,
2. each player $i \in [n]$ is associated with an action set $\mathcal{S}_i$ on the unit sphere in $\mathbb{R}^d$, i.e., $\mathcal{S}_i \in \mathbb{S}^{d-1}$,
3. the scoring function is given by the inner product, i.e., $\sigma(\boldsymbol{s}, \mathbf{x}) = \langle \boldsymbol{s}, \mathbf{x} \rangle$,
4. $(\beta, K) = (\tau, n)$,
5. the utility function is induced by the user exposure metric, i.e., $u_i(\boldsymbol{s}) = \sum_{x\in\mathcal{X}} \mathbf{Pr}(x \to s_i)$.

Note that in exposure games the parameter β no longer represents the user decision noise but becomes a temperature parameter τ controlling the spread of exposure probabilities over items. The game instance $\tilde{G}(\{\mathcal{S}_i\}_{i=1}^n, \mathcal{X}, \sigma, \beta, K)$ defined above is equivalent to the exposure games proposed by [167].

[9] Note that we use $\tilde{G}$ to refer to the variant of G that utilizes the user exposure metric instead of the user engagement metric in player utility functions.

3.4 Conclusion

In conclusion, the investigation of decision-making algorithms for non-cooperative agents has shed light on the complexities and nuances involved in navigating interactions among agents with diverse preferences and motivations. Throughout this chapter, we explored three distinct scenarios: decision making in the face of learning agents that provides revealed preference feedback, agents who require incentives to participate in federated learning, and agents driven by competitiveness. These scenarios provide valuable insights into the dynamics of non-cooperative environments and offer opportunities for designing effective strategies and algorithms.

Motivated by the observation that users' feedback can be coupled with their interaction history with a recommender system, in [188, 189], we proposed a new problem setting where the system learns from reveal preference feedback of a learning user. We formulate the problem of "learning from a learner" and establish efficient learning algorithms for the system, i.e., the system can help the user identify the globally optimal item in sub-linear time if the user is a no-regret learner. Besides the new algorithms, our user learning model also provides a new perspective to studying the feedback loop in recommender systems. A key insight of our proposed solutions is that a healthy recommender system needs to expose a diversified spectrum of items to its users and thus "foster" them to respond with informed feedback. This leads to the win-win outcome for both users and the system in exploring the item space.

Motivated by the application scenarios where multiple companies/organizations cooperate in large scale model training and decision making, we proposed a new problem of "incentivized federated bandit". Compared with our works in Section 2.2 and Section 2.3, it factored in some additional practical concerns of these self-interested agents: cooperation may help their competitors, or cause privacy and security issues, and thus an agent will only cooperate when the benefits outweigh such risks. To minimize the overall regret incurred by all agents, we proposed incentive mechanisms, in the form of data exchange and monetary payment, to enable cooperation among self-interested agents.

Motivated by the observation that content creators may strategically generate contents to maximize their own utilities in online recommendation platform, we proposed the "competing content creation game", a game-theoretical framework for analyzing the strategic behaviors of content creators. Our primary contribution is a comprehensive characterization of social welfare as the outcome of competition among creators, which suggests that the traditional top-K recommendation principle is effective when the platform utilizes user engagement as an incentive metric and offers a sufficient number of choices to users, resonating with the well-known "invisible hand" argument posited by Adam Smith.

In conclusion, the study of decision-making algorithms for non-cooperative agents offers valuable insights and techniques for addressing challenges in cooperative and competitive environments. By considering the preferences, incentives, and competitiveness of agents, we can design algorithms that effectively navigate interactions among diverse agents. These findings contribute to the broader field of multi-agent systems, paving the way for the development of intelligent and adaptable decision-making frameworks that facilitate cooperation and optimize outcomes in complex environments.

Chapter 4

Conclusion and Future Work

In this dissertation, we explored decision-making in multi-agent systems for both cooperative and non-cooperative agents, investigating various scenarios and environments. This research aimed to provide a comprehensive understanding of decision-making dynamics and strategies in different agent settings.

For cooperative agents studied in Chapter 2, we focused on two main aspects: cooperation in heterogeneous and non-stationary environments (Section 2.1), and cooperation in decentralized environments (Section 2.2 and Section 2.3). By studying these contexts, we gained valuable insights into the challenges and opportunities associated with cooperative decision making. Our findings highlighted the importance of adaptive and flexible algorithms that can account for diverse agent capabilities and changing environments. Additionally, we uncovered the significance of communication and coordination mechanisms to facilitate effective cooperation among agents.

For non-cooperative agents studied in Chapter 3, our investigation encompassed decision making when we can only observe revealed preference feedback from another learning agent (Section 3.1), when agents require incentives to participate in federated optimization (Section 3.2), and when agents engage in competitive behavior under the context of content creation in recommender systems (Section 3.3). These scenarios posed unique challenges that required us to explore novel decision-making algorithms. By delving into these areas, we shed light on the intricacies of decision making when individual agents prioritize their own interests. We identified the need for sophisticated incentive mechanisms, and strategic reasoning models to encourage cooperation and navigate competitive landscapes.

Overall, this dissertation contributes to the growing field of multi-agent systems and decision making by providing comprehensive insights into cooperative, non-cooperative and competitive behaviors. By addressing heterogeneous and non-stationary environments, decentralized settings, and various forms of non-cooperation, we have expanded the understanding of decision-making dynamics in complex agent interactions. Our findings have practical implications for real-world applications, such as autonomous systems, distributed networks, and economic markets, where multiple agents make decisions in a cooperative or competitive manner. There are several avenues for future research that can further extend the current knowledge and contribute to practical applications.

Privacy and security For many decision making systems, it is important to consider: potential privacy breaches especially when people' personal information is involved, e.g., purchasing history, or medical records; and adversarial attacks that target model estimation and decision making pipelines, e.g., data, model, and action poisoning attacks that aim to degrade the performance of the decision making systems. The multi-agent setting brings in several new dimension that complicates the privacy and security aspects of decision making algorithms. For example, the decentralized nature of many multi-agent system poses bigger challenge in ensuring privacy and security compared with centralized setting, as the agents need to share information with each other to make informed decisions. Moreover, the complex decision space introduces challenges in understanding and analyzing the privacy and security implications, and the interactions between agents can lead to behaviors that may have unforeseen privacy or security risks, e.g., malicious agents collude with each other to manipulate the system or exploit vulnerabilities for personal gains.

Hybrid agent behaviors In complex real-world scenarios, agents often encounter situations where they need to switch between cooperative and competitive behaviors based on the context or external factors. This flexibility is crucial for agents to adapt to dynamic environments and optimize their decision-making processes. In this case, the decision making algorithm needs to assess the current context and adapt the behavior of agents accordingly. For instance, an agent may exhibit cooperative behavior when resources are abundant and cooperation is beneficial, but switch to a competitive strategy when resources become scarce or when faced with a rival agent. Pursuit in this direction can further enhance the efficiency, adaptability, and robustness of multi-agent systems, enabling agents to achieve optimal outcomes in a wide range of cooperative and competitive environments.

Bibliography

[1] Naim Kapucu and Vener Garayev. Collaborative decision-making in emergency and disaster management. *International Journal of Public Administration*, 34(6):366–375, 2011.

[2] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.

[3] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.

[4] Kun Qian and Sanjay Jain. Digital content creation: An analysis of the impact of recommendation systems. *Available at SSRN 4311562*, 2022.

[5] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[6] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232, 2014.

[7] Ambuj Tewari and Susan A Murphy. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, pages 495–517. Springer, 2017.

[8] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[9] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

[10] Daniel J Lizotte, Tao Wang, Michael H Bowling, Dale Schuurmans, et al. Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pages 944–949, 2007.

[11] Xinbin Li, Jiajia Liu, Lei Yan, Song Han, and Xinping Guan. Relay selection in underwater acoustic cooperative networks: A contextual bandit approach. *IEEE Communications Letters*, 21(2):382–385, 2016.

[12] Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 495–504. ACM, 2018.

[13] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An in-depth study of lte: Effect of network protocol and application behavior on performance. *ACM SIGCOMM Computer Communication Review*, 43(4):363–374, 2013.

[14] Marcel K Richter. Revealed preference theory. *Econometrica: Journal of the Econometric Society*, pages 635–645, 1966.

[15] Franck Tétard and Mikael Collan. Lazy user theory: A dynamic model to understand user selection of products and services. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–9. IEEE, 2009.

[16] Sai Praneeth Karimireddy, Wenshuo Guo, and Michael Jordan. Mechanisms that incentivize data sharing in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[17] 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. ACM, 2010.

[18] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. Exploitation and exploration in a performance based contextual advertising system. In *16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 27–36. ACM, 2010.

[19] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine learning for healthcare conference*, pages 67–82. PMLR, 2018.

[20] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

[21] Andreas Krause and Cheng Soon Ong. Contextual gaussian process bandit optimization. In *Nips*, pages 2447–2455, 2011.

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

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

[24] Yifei Min, Jiafan He, Tianhao Wang, and Quanquan Gu. Multi-agent reinforcement learning: Asynchronous communication and linear function approximation. *arXiv preprint arXiv:2305.06446*, 2023.

[25] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

[26] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[27] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *31st International Conference on Machine Learning (ICML-14)*, pages 757–765, 2014.

[28] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309*, 2019.

[29] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[30] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538. ACM, 2016.

[31] Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144, 2014.

[32] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

[33] Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT*, pages 343–354, 2008.

[34] Yang Cao, Wen Zheng, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure for piecewise-stationary bandit: a change-point detection approach. *AISTATS,(Okinawa, Japan)*, 2019.

[35] Lilian Besson and Emilie Kaufmann. The generalized likelihood ratio test meets klucb: an improved algorithm for piece-wise non-stationary bandits. *arXiv preprint arXiv:1902.01575*, 2019.

[36] Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026, 2019.

[37] Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. *arXiv preprint arXiv:1902.00980*, 2019.

[38] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548. ACM, 2016.

[39] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1253–1262. JMLR. org, 2017.

[40] Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[41] Jia Yuan Yu and Shie Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1177–1184. ACM, 2009.

[42] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, ALT'11, pages 174–188, Berlin, Heidelberg, 2011. Springer-Verlag.

[43] Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michéle Sebag. Multi-armed bandit, dynamic environments and meta-bandits. 2006.

[44] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

[45] David Siegmund. *Sequential analysis: tests and confidence intervals*. Springer Science & Business Media, 2013.

[46] Othmane Mazhar, Cristian Rojas, Carlo Fischione, and Mohammad Reza Hesamzadeh. Bayesian model selection for change point detection and clustering. In *International Conference on Machine Learning*, pages 3433–3442. PMLR, 2018.

[47] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration–exploitation in a multi-armed bandit problem with non-stationary rewards. *Stochastic Systems*, 9(4):319–337, 2019.

[48] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1079–1087, 2019.

[49] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

[50] Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158, 2019.

[51] Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[52] Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 2020, 2020.

[53] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

[54] Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*, pages 1739–1776, 2018.

[55] Shuai Li. *The art of clustering bandits.* PhD thesis, Università degli Studi dell'Insubria, 2016.

[56] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Multi-armed bandits in the presence of side observations in social networks. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7309–7314. IEEE, 2013.

[57] Kaige Yang, Laura Toni, and Xiaowen Dong. Laplacian-regularized graph bandits: Algorithms and theoretical analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 3133–3143, 2020.

[58] Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, pages 591–605, 1960.

[59] R Stephen Cantrell, Peter M Burrows, and Quang H Vuong. Interpretation and use of generalized chow tests. *International Economic Review*, pages 725–741, 1991.

[60] AL Wilson. When is the chow test ump? *The American Statistician*, 32(2):66–68, 1978.

[61] Abhimanyu Dubey and Alex Pentland. Differentially-private federated linear bandits. *arXiv preprint arXiv:2010.11425*, 2020.

[62] 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.

[63] Ruiquan Huang, Weiqiang Wu, Jing Yang, and Cong Shen. Federated linear contextual bandits. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[64] Chuanhao Li and Hongning Wang. Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6529–6553. PMLR, 2022.

[65] 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.

[66] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, and Rachel Cummings. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[67] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, pages 19–36. JMLR Workshop and Conference Proceedings, 2012.

[68] 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.

[69] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2012.

[70] Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. *arXiv preprint arXiv:1706.00136*, 2017.

[71] Qin Ding, Cho-Jui Hsieh, and James Sharpnack. An efficient algorithm for generalized linear bandit: Online stochastic gradient descent and thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 1585–1593. PMLR, 2021.

[72] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

[73] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory*, pages 533–557. PMLR, 2019.

[74] Houssam Zenati, Alberto Bietti, Eustache Diemert, Julien Mairal, Matthieu Martin, and Pierre Gaillard. Efficient kernel ucb for contextual bandits. *arXiv preprint arXiv:2202.05638*, 2022.

[75] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Near-linear time gaussian process optimization with adaptive batching and resparsification. In *International Conference on Machine Learning*, pages 1295–1305. PMLR, 2020.

[76] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, 26, 2013.

[77] Chao Tao, Qin Zhang, and Yuan Zhou. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 126–146. IEEE, 2019.

[78] Yihan Du, Wei Chen, Yuko Yuroki, and Longbo Huang. Collaborative pure exploration in kernel bandit. *arXiv preprint arXiv:2110.15771*, 2021.

[79] Nathan Korda, Balazs Szorenyi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*, pages 1301–1309. PMLR, 2016.

[80] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[81] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[82] Aritra Mitra, Rayana Jaafar, George J Pappas, and Hamed Hassani. Achieving linear convergence in federated learning under objective and systems heterogeneity. *arXiv preprint arXiv:2102.07053*, 2021.

[83] Reese Pathak and Martin J Wainwright. Fedsplit: an algorithmic framework for fast federated optimization. *Advances in Neural Information Processing Systems*, 33:7057–7066, 2020.

[84] Chuanhao Li and Hongning Wang. Communication efficient federated learning for generalized linear bandits. In *Advances in Neural Information Processing Systems*, 2022.

[85] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[86] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[87] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *arXiv preprint arXiv:1506.01900*, 2015.

[88] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[89] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[90] Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401. PMLR, 2016.

[91] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[92] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning*, pages 645–653. PMLR, 2017.

[93] Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.

[94] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[95] 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.

[96] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. Ee-net: Exploitation-exploration neural networks in contextual bandits. *arXiv preprint arXiv:2110.03177*, 2021.

[97] Jiafan He, Tianhao Wang, Yifei Min, and Quanquan Gu. A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. *arXiv preprint arXiv:2207.03106*, 2022.

[98] Chuanhao Li, Huazheng Wang, Mengdi Wang, and Hongning Wang. Learning kernelized contextual bandits in a distributed and asynchronous environment. In *The Eleventh International Conference on Learning Representations*, 2023.

[99] Chuanhao Li, Qingyun Wu, and Hongning Wang. Unifying clustered and non-stationary bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1063–1071. PMLR, 2021.

[100] Chuanhao Li, Huazheng Wang, Mengdi Wang, and Hongning Wang. Communication efficient distributed learning for kernelized contextual bandits. In *Advances in Neural Information Processing Systems*, 2022.

[101] Evert J Nyström. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica*, 54:185–204, 1930.

[102] Chuanhao Li, Qingyun Wu, and Hongning Wang. When and whom to collaborate with in a changing environment: A collaborative dynamic bandit solution. SIGIR '21.

[103] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

[104] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[105] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[106] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[107] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[108] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90, 2010.

[109] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[110] Travis Ebesu, Bin Shen, and Yi Fang. Collaborative memory network for recommendation systems. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 515–524, 2018.

[111] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[112] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[113] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.

[114] Jibang Wu, Renqin Cai, and Hongning Wang. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*, pages 2199–2209, 2020.

[115] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280, 2007.

[116] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[117] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26thAnnual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.

[118] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pages 856–864. PMLR, 2014.

[119] Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Conference on Learning Theory*, pages 563–587. PMLR, 2015.

[120] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[121] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[122] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.

[123] Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 101–108, 2009.

[124] Sreenivas Gollapudi, Guru Guruganesh, Kostas Kollias, Pasin Manurangsi, Renato Paes Leme, and Jon Schneider. Contextual recommendations and low-regret cutting-plane algorithms. *arXiv preprint arXiv:2106.04819*, 2021.

[125] Maxime C Cohen, Ilan Lobel, and Renato Paes Leme. Feature-based dynamic pricing. *Management Science*, 66(11):4921–4943, 2020.

[126] Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. *Operations Research*, 66(5):1346–1361, 2018.

[127] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[128] Jonathan D Cohen, Samuel M McClure, and Angela J Yu. Should I stay or should I go? how the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942, 2007.

[129] Nathaniel D Daw, John P O'doherty, Peter Dayan, Ben Seymour, and Raymond J Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879, 2006.

[130] Samuel J Gershman. Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42, 2018.

[131] Robert C Wilson, Andra Geana, John M White, Elliot A Ludvig, and Jonathan D Cohen. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074, 2014.

[132] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method. In *Geometric Algorithms and Combinatorial Optimization*, pages 64–101. Springer, 1993.

[133] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[134] Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. *arXiv preprint arXiv:1705.00253*, 2017.

[135] Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652, 1949.

[136] James R Bunch, Christopher P Nielsen, and Danny C Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978.

[137] Jian Pei. A survey on data pricing: from economics to data science. *IEEE Transactions on knowledge and Data Engineering*, 34(10):4586–4608, 2020.

[138] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. Incentive mechanisms for federated learning: From economic and game theoretic perspective. *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[139] Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. To federate or not to federate: Incentivizing client participation in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[140] Rui Hu and Yanmin Gong. Trading data for learning: Incentive mechanism for on-device federated learning. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[141] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. Collaborative machine learning with incentive-aware model rewards. In *International conference on machine learning*, pages 8927–8936. PMLR, 2020.

[142] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Advances in Neural Information Processing Systems*, 34:16104–16117, 2021.

[143] Kate Donahue and Jon Kleinberg. Fairness in model-sharing games. In *Proceedings of the ACM Web Conference 2023*, pages 3775–3783, 2023.

[144] Kate Donahue and Jon Kleinberg. Model-sharing games: Analyzing federated learning under voluntary participation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5303–5311, 2021.

[145] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 10(2):1035–1044, 2021.

[146] Maurice Allais. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica: Journal of the econometric society*, pages 503–546, 1953.

[147] Malcolm Pemberton and Nicholas Rau. *Mathematics for economists: an introductory textbook*. Manchester University Press, 2007.

[148] Roger B Myerson and Mark A Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of economic theory*, 29(2):265–281, 1983.

[149] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[150] Jiu Ding and Aihui Zhou. Eigenvalues of rank-one updated matrices with some applications. *Applied Mathematics Letters*, 20(12):1223–1226, 2007.

[151] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[152] Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712, 2009.

[153] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pages 404–413. Springer, 1999.

[154] Omer Ben-Porat and Moshe Tennenholtz. A game-theoretic approach to recommendation systems with strategic content providers. *Advances in Neural Information Processing Systems*, 31, 2018.

[155] Omer Ben-Porat, Gregory Goren, Itay Rosenberg, and Moshe Tennenholtz. From recommendation systems to facility location games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1772–1779, 2019.

[156] Charles F Manski. The structure of random utility models. *Theory and decision*, 8(3):229, 1977.

[157] George Baltas and Peter Doyle. Random utility models in marketing research: a survey. *Journal of Business Research*, 51(2):115–125, 2001.

[158] Meta. Meta is experimenting with new monetization options for creators, 2022. https://www.digitalinformationworld.com/2022/03/meta-is-experimenting-with-new.html.

[159] Savy. Will the new youtube algorithm impact your content?, 2019. https://savyagency.com/new-youtube-algorithm/.

[160] Youtube. Youtube partner program overview & eligibility, 2023. https://support.google.com/youtube/answer/72851.

[161] TikTok. What is the tiktok creator fund? here's how to join + start making money, 2022. https://www.backstage.com/magazine/article/tiktok-creator-fund-explained-how-to-join-75090/.

[162] Omer Ben-Porat and Moshe Tennenholtz. Shapley facility location games. In *International Conference on Web and Internet Economics*, pages 58–73. Springer, 2017.

[163] Ran Ben Basat, Moshe Tennenholtz, and Oren Kurland. The probability ranking principle is not optimal in adversarial retrieval settings. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 51–60, 2015.

[164] Nimrod Raifer, Fiana Raiber, Moshe Tennenholtz, and Oren Kurland. Information retrieval meets game theory: The ranking competition between documents' authors. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 465–474, 2017.

[165] Harold Hotelling. (1929): Stability in competition. *Economic Journal*, 39(4):57, 1929.

[166] Weiran Shen and Zihe Wang. Hotelling-downs model with limited attraction. *arXiv preprint arXiv:1611.05959*, 2016.

[167] Jiri Hron, Karl Krauth, Michael I Jordan, Niki Kilbertus, and Sarah Dean. Modeling content creator incentives on algorithm-curated platforms. *arXiv preprint arXiv:2206.13102*, 2022.

[168] Meena Jagadeesan, Nikhil Garg, and Jacob Steinhardt. Supply-side equilibria in recommender systems. *arXiv preprint arXiv:2206.13489*, 2022.

[169] Robert Duncan Luce, Patrick Suppes, et al. Preference, utility, and subjective probability. 1965.

[170] Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 373–382, 2008.

[171] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 416–425. IEEE, 2002.

[172] Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5):1–42, 2015.

[173] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73, 2005.

[174] George Christodoulou, Annamária Kovács, and Michael Schapira. Bayesian combinatorial auctions. In *International Colloquium on Automata, Languages, and Programming*, pages 820–832. Springer, 2008.

[175] Daniel McFadden. The measurement of urban travel demand. *Journal of public economics*, 3(4):303–328, 1974.

[176] Daniel L McFadden. Econometric analysis of qualitative response models. *Handbook of econometrics*, 2:1395–1457, 1984.

[177] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

[178] Chun Kai Ling, Fei Fang, and J Zico Kolter. What game are we playing? end-to-end learning in normal and extensive form games. *arXiv preprint arXiv:1805.02777*, 2018.

[179] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[180] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[181] Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*, 10(4):1–30, 2011.

[182] Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

[183] Robert D Luce. Individual choice behavior: A theoretical analysis, new york, ny: John willey and sons, 1959.

[184] Jicong Fan and Jieyu Cheng. Matrix completion by deep matrix factorization. *Neural Networks*, 98:34–41, 2018.

[185] Christos H Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *SODA*, volume 5, pages 82–91. Citeseer, 2005.

[186] Tim Roughgarden, Vasilis Syrgkanis, and Eva Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017.

[187] Matej Balog, Nilesh Tripuraneni, Zoubin Ghahramani, and Adrian Weller. Lost relatives of the gumbel trick. In *International Conference on Machine Learning*, pages 371–379. PMLR, 2017.

[188] Fan Yao, Chuanhao Li, Denis Nekipelov, Hongning Wang, and Haifeng Xu. Learning the optimal recommendation from explorative users. AAAI ’22.

[189] Fan Yao, Chuanhao Li, Denis Nekipelov, Hongning Wang, and Haifeng Xu. Learning from a learning user for optimal recommendations. ICML ’22.

[190] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[191] Lin Tie, Kai-Yuan Cai, and Yan Lin. Rearrangement inequalities for hermitian matrices. *Linear algebra and its applications*, 434(2):443–456, 2011.

[192] Joel Tropp et al. Freedman’s inequality for matrix martingales. *Electronic Communications in Probability*, 16:262–270, 2011.

Appendix

A Technical lemmas

Here are the technical lemmas needed for the proofs in this dissertation.

Lemma A.1. *For a symmetric positive definite matrix $A \in \mathbb{R}^{d\times d}$ and any vector $\mathbf{x} \in \mathbb{R}^d$, we have the following inequality*

$$\mathbf{x}^\top \mathbf{x} \le \mathbf{x}^\top A\mathbf{x} \cdot \mathbf{x}^\top A^{-1}\mathbf{x} \le \frac{||\mathbf{x}||_2^4 \lambda_{max}(A)}{\lambda_{min}(A)}$$

Lemma A.2 (Lemma 11 of [20])**.** *Let $\{X_t\}_{t=1}^\infty$ be a sequence in $\mathbb{R}^d$, V is a $d \times d$ positive definite matrix and define $V_t = V + \sum_{s=1}^t X_s X_s^\top$. Then we have that*

$$\log\left(\frac{\det(V_n)}{\det(V)}\right) \le \sum_{t=1}^n \|X_t\|^2_{V_{t-1}^{-1}} .$$

Further, if $\|X_t\|_2 \le L$ for all t, then

$$\sum_{t=1}^n \min\left\{1, \|X_t\|^2_{V_{t-1}^{-1}}\right\} \le 2\left(\log\det(V_n) - \log\det V\right) \le 2\left(d\log\left(\left(\text{trace}(V) + nL^2\right)/d\right) - \log\det V\right) .$$

Lemma A.3 (Lemma 12 of [20])**.** *Let A, B and C be positive semi-definite matrices such that $A = B + C$. Then, we have that:*

$$\sup_{\mathbf{x}\neq \mathbf{0}} \frac{\mathbf{x}^\top A\mathbf{x}}{\mathbf{x}^\top B\mathbf{x}} \le \frac{\det(A)}{\det(B)}$$

Lemma A.4. *Let A be symmetric positive-definite matrix, and B, C be symmetric positive semi-definite matrices, we have*

$$\frac{\det(A+B+C)}{\det(A+C)} \le \frac{\det(A+B)}{\det(A)}$$

Lemma A.5 (Corollary 7.7.4. (a) of [190])**.** *Let A, B be positive definite matrices, such that $A \succeq B$, then we have*

$$A^{-1} \preceq B^{-1}.$$

Lemma A.6 (Lemma 2.2 of [191])**.** *For any positive semi-definite matrices A, B and C, it holds that* $\det(A+B+C) + \det(A) \ge \det(A+B) + \det(A+C)$.

Lemma A.7 (Matrix Weighted Cauchy-Schwarz)**.** *If $A \in \mathbb{R}^{d\times d}$ is a PSD matrix, then $x^T A y \le \sqrt{x^T A x \cdot y^T A y}$ holds for any vectors $x, y \in \mathbb{R}^d$.*

Proof. Consider a quadratic function $(x+ty)^T A(x+ty) = x^T Ax + 2(x^T Ay)t + (y^T Ay)t^2$ for some variable $t \in \mathbb{R}$, where $x, y \in \mathbb{R}^d$ are arbitrary vectors. Since A is PSD, the value of this quadratic function $(x+ty)^T A(x+ty) =$

$x^T Ax + 2(x^T Ay)t + (y^T Ay)t^2 \geq 0, \forall t$, which means there can be at most one root. This is equivalent to saying the discriminant of this quadratic function $4(x^T Ay)^2 - 4x^T Ax \cdot y^T Ay \leq 0$, which finishes the proof. □

Lemma A.8 (Extension of Lemma A.3 to kernel matrix)**.** *Define positive definite matrices $A = \lambda \mathbf{I} + \mathbf{\Phi}_1^\top \mathbf{\Phi}_1 + \mathbf{\Phi}_2^\top \mathbf{\Phi}_2$ and $B = \lambda \mathbf{I} + \mathbf{\Phi}_1^\top \mathbf{\Phi}_1$, where $\mathbf{\Phi}_1^\top \mathbf{\Phi}_1, \mathbf{\Phi}_2^\top \mathbf{\Phi}_2 \in \mathbb{R}^{p \times p}$ and p is possibly infinite. Then, we have that:*

$$\sup_{\phi \neq \mathbf{0}} \frac{\phi^\top A \phi}{\phi^\top B \phi} \leq \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_A)}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_B)}$$

where $\mathbf{K}_A = \begin{bmatrix} \mathbf{\Phi}_1 \\ \mathbf{\Phi}_2 \end{bmatrix} \left[\mathbf{\Phi}_1^\top, \mathbf{\Phi}_2^\top\right]$ and $\mathbf{K}_B = \mathbf{\Phi}_1 \mathbf{\Phi}_1^\top$.

Proof of Lemma A.8. Similar to the proof of Lemma 12 of [20], we start from the simple case when $\mathbf{\Phi}_2^\top \mathbf{\Phi}_2 = mm^\top$, where $m \in \mathbb{R}^p$. Using Cauchy-Schwartz inequality, we have

$$(\phi^\top m)^2 = (\phi^\top B^{1/2} B^{-1/2} m)^2 \leq \|B^{1/2}\phi\|^2 \|B^{-1/2} m\|^2 = \|\phi\|_B^2 \|m\|_{B^{-1}}^2,$$

and thus,

$$\phi^\top (B + mm^\top)\phi \leq \phi^\top B \phi + \|\phi\|_B^2 \|m\|_{B^{-1}}^2 = (1 + \|m\|_{B^{-1}}^2)\|\phi\|_B^2,$$

so we have

$$\frac{\phi^\top A \phi}{\phi^\top B \phi} \leq 1 + \|m\|_{B^{-1}}^2$$

for any ϕ. Then using the kernel trick, e.g., see the derivation of Eq (27) in [74], we have

$$1 + \|m\|_{B^{-1}}^2 = \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_A)}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_B)},$$

which finishes the proof of this simple case. Now consider the general case where $\mathbf{\Phi}_2^\top \mathbf{\Phi}_2 = m_1 m_1^\top + m_2 m_2^\top + \cdots + m_{t-1} m_{t-1}^\top$. Let's define $V_s = B + m_1 m_1^\top + m_2 m_2^\top + \cdots + m_{s-1} m_{s-1}^\top$ and the corresponding kernel matrix $\mathbf{K}_{V_s} = \begin{bmatrix} \mathbf{\Phi}_1 \\ m_1^\top \\ \cdots \\ m_{s-1}^\top \end{bmatrix} \left[\mathbf{\Phi}_1^\top, m_1, \ldots, m_{s-1}\right]$, and note that $\frac{\phi^\top A \phi}{\phi^\top B \phi} = \frac{\phi^\top V_t \phi}{\phi^\top V_{t-1} \phi} \frac{\phi^\top V_{t-1} \phi}{\phi^\top V_{t-2} \phi} \cdots \frac{\phi^\top V_2 \phi}{\phi^\top B \phi}$. Then we can apply the result for the simple case on each term in the product above, which gives us

$$\begin{aligned} \frac{\phi^\top A \phi}{\phi^\top B \phi} &\leq \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_t})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_{t-1}})} \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_{t-1}})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_{t-2}})} \cdots \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_2})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_B)} \\ &= \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_t})}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_B)} = \frac{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_A)}{\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_B)}, \end{aligned}$$

which finishes the proof.

□

Lemma A.9 (Eq (26) and Eq (27) of [74])**.** *Let $\{\phi_t\}_{t=1}^\infty$ be a sequence in $\mathbb{R}^p$, $V \in \mathbb{R}^{p \times p}$ a positive definite matrix, where p is possibly infinite, and define $V_t = V + \sum_{s=1}^t \phi_s \phi_s^\top$. Then we have that*

$$\sum_{t=1}^n \min\left(\|\phi_t\|_{V_{t-1}^{-1}}^2, 1\right) \leq 2 \ln\left(\det(\mathbf{I} + \lambda^{-1}\mathbf{K}_{V_t})\right),$$

where $\mathbf{K}_{V_t}$ is the kernel matrix corresponding to V_t as defined in Lemma A.8.

Lemma A.10 (Lemma 4 of [75]). *For $t > t_{last}$, we have for any $\mathbf{x} \in \mathbb{R}^d$*

$$\hat{\sigma}_t^2(\mathbf{x}) \leq \hat{\sigma}_{t_{last}}^2(\mathbf{x}) \leq \big(1 + \sum_{s=t_{last}+1}^{t} \hat{\sigma}_{t_{last}}^2(\mathbf{x}_s)\big)\hat{\sigma}_t^2(\mathbf{x})$$

Lemma A.11 (Lemma 6 of [73]). *If $\mathcal{S}_k$ is ϵ-accurate w.r.t. $\mathcal{D}_k$, then*

$$\frac{1-\epsilon}{1+\epsilon}\sigma^2(\mathbf{x}) \leq \min(\tilde{\sigma}_k^2(\mathbf{x}), 1) \leq \frac{1+\epsilon}{1-\epsilon}\sigma^2(\mathbf{x})$$

for all $\mathbf{x} \in \mathbb{R}^d$.

Lemma A.12 (Hoeffding inequality). *Suppose that we have independent variables $x_i, i = 1, \dots, n$, and x_i has mean μ_i and sub-Gaussian parameter σ_i. Then for all $h \geq 0$, we have*

$$P\big(\sum_{i=1}^{n}(x_i - \mu_i) \geq h\big) \leq \exp\left(-\frac{h^2}{2\sum_{i=1}^{n}\sigma_i^2}\right)$$

Lemma A.13 (Proposition 7 of [73]). *Let $G_1, \dots, G_n$ be a sequence of independent self-adjoint random operators such that $\mathbb{E}[G_i] = 0$ and $\|G_i\| \leq R$. Then for any $\epsilon \geq 0$, we have*

$$\mathbb{P}\big(\|\sum_{i=1}^{t} G_i\| \geq \epsilon\big) \leq 4t\exp\big(-\frac{\epsilon^2/2}{\|\sum_{i=1}^{t}\mathbb{E}[G_i^2]\| + R\epsilon/3}\big).$$

Lemma A.14 (Proposition 8 of [73]). *Let $\{q_s\}_{s=1}^{t}$ be independent Bernoulli random variables, each with success probability p_s. Then we have*

$$\mathbb{P}\big(\sum_{s=1}^{t} q_s \geq 3\sum_{s=1}^{t} p_s\big) \leq \exp(-2\sum_{s=1}^{t} p_s).$$

Lemma A.15 (Lemma 1 of [39]). *Under Assumption 3 that, at each time t, arm set C_t is generated i.i.d. from a sub-Gaussian random vector $X \in \mathbb{R}^d$, such that $\mathbb{E}[XX^\top]$ is full-rank with minimum eigenvalue $\lambda' > 0$; and the variance ς^2 of the random vector satisfies $\varsigma^2 \leq \frac{\lambda'^2}{8\ln 4K}$. Then we have the following lower bound on minimum eigenvalue of the correlation matrix of observation history $\mathcal{H}$:*

$$\lambda_{\min}\Big(\sum_{(\mathbf{x}_k, y_k)\in\mathcal{H}} \mathbf{x}_k\mathbf{x}_k^\top\Big) \geq \frac{\lambda^{'}}{4}|\mathcal{H}| - 8\Big(\log\frac{d|\mathcal{H}|}{\delta'} + \sqrt{|\mathcal{H}|\log\frac{d|\mathcal{H}|}{\delta'}}\Big)$$

with probability at least $1 - \delta^{'}$.

Lemma A.16 (Matrix Freedman's inequality [192]). *Consider a matrix martingale $\{Y_s\}_{s=1,2,\dots}$ whose values are matrices with dimension $d_1 \times d_2$, and let $\{Z_s\}_{s=1,2,\dots}$ be the corresponding martingale difference sequence. Assume that the difference sequence is almost surely uniformly bounded, i.e., $||Z_s||_{op} \leq R$, for $s = 1, 2, \dots$.*

Define two predictable quadratic variation processes of the martingale:

$$W_{col,t} := \sum_{s=1}^{t}\mathbb{E}_{s-1}[Z_sZ_s^\top] \quad \textit{and}$$

$$W_{row,t} := \sum_{s=1}^{t}\mathbb{E}_{s-1}[Z_s^\top Z_s] \quad \textit{for } t = 1, 2, \dots$$

Then for all $u \geq 0$ and $\omega^2 \geq 0$, we have

$$P(\exists t \geq 0 : ||Y_t||_{op} \geq u, \text{ and } \max\{||W_{col,t}||_{op}, ||W_{row,t}||_{op}\} \leq \omega^2) \leq (d_1 + d_2) \exp\left(-\frac{u^2/2}{\omega^2 + Ru/3}\right)$$

Lemma A.17 (Vector-valued self-normalized bound (Theorem 1 of [20]))**.** *Let $\{\mathcal{F}_t\}_{t=1}^{\infty}$ be a filtration. Let $\{\eta_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that η_t is $\mathcal{F}_{t+1}$-measurable, and η_t is conditionally zero mean R-sub-Gaussian for some $R \geq 0$. Let $\{X_t\}_{t=1}^{\infty}$ be a $\mathbb{R}^d$-valued stochastic process such that X_t is $\mathcal{F}_t$-measurable. Assume that V is a $d \times d$ positive definite matrix. For any $t > 0$, define*

$$V_t = V + \sum_{\tau=1}^{t} X_\tau X_\tau^\top \quad \mathcal{S}_t = \sum_{\tau=1}^{t} \eta_\tau X_\tau$$

Then for any $\delta > 0$, with probability at least $1 - \delta$,

$$||\mathcal{S}_t||_{V_t^{-1}} \leq R\sqrt{2 \log \frac{\det(V_t)^{1/2}}{\det(V)^{1/2}\delta}}, \quad \forall t \geq 0$$

Lemma A.18 (Corollary 8 of [69])**.** *Under the same assumptions as Lemma A.17, consider a sequence of real-valued variables $\{Z_t\}_{t=1}^{\infty}$ such that Z_t is $\mathcal{F}_t$-measurable. Then for any $\delta > 0$, with probability at least $1 - \delta$,*

$$|\sum_{\tau=1}^{t} \eta_\tau Z_\tau| \leq R\sqrt{2(V + \sum_{\tau=1}^{t} Z_\tau^2) \log\left(\frac{\sqrt{V + \sum_{\tau=1}^{t} Z_\tau^2}}{\delta\sqrt{V}}\right)}, \forall t \geq 0$$

Lemma A.19. *Under Assumption 4, $F_{i,t}(\theta)$ for $i = 1, 2, \ldots, N$ is smooth with constant $k_\mu + \frac{\lambda}{Nt}$*

Proof. By Assumption 4, $\mu(\cdot)$ is Lipschitz continuous with constant k_μ, i.e., $|\mu(\mathbf{x}^\top\theta_1) - \mu(\mathbf{x}^\top\theta_2)| \leq k_\mu|\mathbf{x}^\top(\theta_1 - \theta_2)|$. Then we can show that

$$\begin{aligned}
&||\nabla F_{i,t}(\theta_1) - \nabla F_{i,t}(\theta_2)|| \\
&= ||\frac{1}{t}\sum_{s=1}^{t} \mathbf{x}_{s,i}[\mu(\mathbf{x}_{s,i}^\top\theta_1) - \mu(\mathbf{x}_{s,i}^\top\theta_2)] + \frac{\lambda}{Nt}(\theta_1 - \theta_2)|| \\
&\leq \frac{1}{t}\sum_{s=1}^{t} ||\mathbf{x}_{s,i}[\mu(\mathbf{x}_{s,i}^\top\theta_1) - \mu(\mathbf{x}_{s,i}^\top\theta_2)]|| + \frac{\lambda}{Nt}||\theta_1 - \theta_2|| \\
&\leq \frac{1}{t}\sum_{s=1}^{t} |\mu(\mathbf{x}_{s,i}^\top\theta_1) - \mu(\mathbf{x}_{s,i}^\top\theta_2)| + \frac{\lambda}{Nt}||\theta_1 - \theta_2|| \\
&\leq \frac{k_\mu}{t}\sum_{s=1}^{t} |\mathbf{x}_{s,i}^\top(\theta_1 - \theta_2)| + \frac{\lambda}{Nt}||\theta_1 - \theta_2|| \leq (k_\mu + \frac{\lambda}{Nt})||\theta_1 - \theta_2||
\end{aligned}$$

Therefore, $\nabla F_{i,t}(\theta)$ is Lipschitz continuous with constant $k_\mu + \frac{\lambda}{Nt}$, and $\nabla F_t(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla F_{i,t}(\theta)$ is Lipschitz continuous with constant $k_\mu + \frac{\lambda}{Nt}$ as well. □