# Federated Linear Bandits with Finite Adversarial Actions

**Li Fan**
University of Virginia
lf2by@virginia.edu

**Ruida Zhou**
Texas A&M University
ruida@tamu.edu

**Chao Tian**
Texas A&M University
chao.tian@tamu.edu

**Cong Shen**
University of Virginia
cong@virginia.edu

## Abstract

We study a federated linear bandits model, where $M$ clients communicate with a central server to solve a linear contextual bandits problem with finite adversarial action sets that may be different across clients. To address the unique challenges of *adversarial finite* action sets, we propose the `FedSupLinUCB` algorithm, which extends the principles of SupLinUCB and OFUL algorithms in linear contextual bandits. We prove that `FedSupLinUCB` achieves a total regret of $\tilde{O}(\sqrt{dT})$, where $T$ is the total number of arm pulls from all clients, and $d$ is the ambient dimension of the linear model. This matches the minimax lower bound and thus is order-optimal (up to polylog terms). We study both asynchronous and synchronous cases and show that the communication cost can be controlled as $O(dM^2 \log(d) \log(T))$ and $O(\sqrt{d^3M^3} \log(d))$, respectively. The `FedSupLinUCB` design is further extended to two scenarios: (1) variance-adaptive, where a total regret of $\tilde{O}(\sqrt{d\sum_{t=1}^{T}\sigma_t^2})$ can be achieved with $\sigma_t^2$ being the noise variance of round $t$; and (2) adversarial corruption, where a total regret of $\tilde{O}(\sqrt{dT} + dC_p)$ can be achieved with $C_p$ being the total corruption budget. Experiment results corroborate the theoretical analysis and demonstrate the effectiveness of `FedSupLinUCB` on both synthetic and real-world datasets.

## 1 Introduction

In the canonical formulation of contextual bandits, a single player would repeatedly make arm-pulling decisions based on contextual information with the goal of maximizing the long-term reward. With the emerging *federated learning* paradigm (McMahan et al., 2017) where multiple clients and a server jointly learn a global model with each client locally updating the model with its own data and server only aggregating the local models periodically, researchers have started exploring contextual bandits algorithms in such federated learning setting (Dubey and Pentland, 2020; Huang et al., 2021; Li and Wang, 2022a,b). This federated contextual bandits framework broadens the applicability of contextual bandits to practical scenarios such as recommender systems, clinical trials, and cognitive radio. In these applications, although the goal is still to maximize the cumulative reward for the overall system, decision-making and observations are naturally distributed at the participating clients.

Several intrinsic challenges arise with the federated contextual bandit formulation. One important issue is that besides regret, we should also take into account the communication cost, which is usually the system bottleneck. To reduce the communication cost while maintaining the same regret guarantee, the clients should transmit the necessary information to the server only when the local

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

information has accumulated to the extent that it would affect the decision-making. Compared with the centralized contextual bandits, which have a linearly growing communication cost, algorithms for federated contextual bandits attempt to achieve a comparable regret with sub-linear communication cost.

Second, most existing studies on federated contextual bandits focus on the synchronous communication scenario (Huang et al., 2021; Li and Wang, 2022b), in which all participating clients first upload local information and then download updated global information from the server in each communication round. This stringent communication requirement is often not met in practice. A recent work of Li and Wang (2022a) studies the asynchronous federated linear bandit problem. However, communications for different clients are not independent in their approach because the upload from one client may trigger the server to perform downloads for all clients. To address this issue, He et al. (2022a) proposes FedLinUCB, which enables independent synchronizations between clients and the server.

Third, the majority of prior studies on federated linear bandits focused on the infinite-arm setting (Li and Wang, 2022b,a; He et al., 2022a) (see Section 2 for a detailed literature review). From a methodology point of view, these papers largely build on the OFUL principle (Abbasi-Yadkori et al., 2011). One notable exception is Huang et al. (2021), which studies synchronous communication with fixed contexts and proposes the Fed-PE algorithm based on the phased elimination G-optimal design (Lattimore and Szepesvári, 2020). To the best of our knowledge, no prior result exists for federated linear bandits with finite arms and time-evolving adversarial contexts, which is the focus of our work.

Table 1: Comparison of this paper with related works

| System | Action | Algorithm | Regret | Communication |
|---|---|---|---|---|
| Single-player | infinite arm | OFUL (Abbasi-Yadkori et al., 2011) | $d\sqrt{T}\log T$ | N/A |
| Single-player | finite fixed arm | PE + G-optimal (Lattimore and Szepesvári, 2020) | $O(\sqrt{dT\log T})$ | N/A |
| Single-player | finite adversarial arm | SupLinUCB (Chu et al., 2011) | $O(\sqrt{dT\log^3 T})$ | N/A |
| Federated (Async) | infinite arm | FedLinUCB (He et al., 2022a) | $O(d\sqrt{T}\log T)$ | $O(dM^2\log T)$ |
| Federated (Async) | infinite arm | Async-LinUCB (Li and Wang, 2022a) | $O(d\sqrt{T}\log T)$ | $O(dM^2\log T)$ |
| Federated (Sync) | infinite arm | DisLinUCB (Wang et al., 2019) | $O(d\sqrt{T}\log^2 T)$ | $O(dM^{3/2})$ |
| Federated (Sync) | finite fixed arm | Fed-PE (Huang et al., 2021) | $O(\sqrt{dT\log T})$ | $O(d^2MK\log T)$ |
| Federated (Async) | finite adversarial arm | FedSupLinUCB (This work) | $O(\sqrt{dT\log^3 T})$ | $O(dM^2\log d\log T)$ |
| Federated (Sync) | finite adversarial arm | FedSupLinUCB (This work) | $O(\sqrt{dT\log^3 T})$ | $O(d^{3/2}M^{3/2}\log(d))$ |

$d$: the dimension of the unknown parameter, $M$: the number of clients, $K$: the number of finite actions, $T$: the total arm pulls from all clients.

**Main contributions.** Our main contributions are summarized as follows.

- We develop a general federated bandits framework, termed `FedSupLinUCB`, for solving the problem of federated linear contextual bandits with finite adversarial actions. `FedSupLinUCB` extends SupLinUCB (Chu et al., 2011; Ruan et al., 2021) and OFUL (Abbasi-Yadkori et al., 2011), two important principles in (single-player, centralized) linear bandits, to the federated bandits setting with a carefully designed layered successive screening.
- We instantiate `FedSupLinUCB` with both asynchronous and synchronous client activities. For the former setting, we propose Async-`FedSupLinUCB` where communication is triggered only when the cumulative local information impacts the exploration uncertainty to a certain extent. We prove that Async-`FedSupLinUCB` achieves $\tilde{O}(\sqrt{dT})$ regret with $O(dM^2\log d\log T)$ communication cost, which not only reduces the regret by $\sqrt{d}$ compared with previous results on asynchronous federated linear bandits with infinite arms, but also matches the minimax lower bound up to polylog terms, indicating that Async-`FedSupLinUCB` achieves order-optimal regret.
- For synchronous communications, we propose Sync-`FedSupLinUCB`, which has a refined communication design where only certain layers are communicated, as opposed to the complete information. Sync-`FedSupLinUCB` achieves order-optimal regret of $\tilde{O}(\sqrt{dT})$ with horizon-independent communication cost $O(\sqrt{d^3M^3}\log d)$. Compared with the best previous result (Huang et al., 2021) which achieves the same order-optimal regret but only for *fixed* actions, we show that it is the *finite* actions that fundamentally determines the regret behavior in the federated linear bandits setting.
- We further develop two extensions of `FedSupLinUCB`: (1) Variance-adaptive `FedSupLinUCB`, for which a total regret of $\tilde{O}(\sqrt{d\sum_{t=1}^{T}\sigma_t^2})$ is achieved, where $\sigma_t^2$ is the noise variance at round $t$. (2)

Adversarial corruption `FedSupLinUCB`, for which a total regret of $\tilde{O}(\sqrt{dT} + dC_p)$ is achieved, where $C_p$ is the total corruption budget.

## 2 Related Works

The linear bandit model, as a generalization of finite armed bandits with linear contextual information, has been extensively studied. The setting of infinite arm sets solved by LinUCB was analyzed in (Dani et al., 2008; Abbasi-Yadkori et al., 2011), which achieves regret $\tilde{O}(d\sqrt{T})$ with appropriate confidence width (Abbasi-Yadkori et al., 2011) and matches the lower bound (Dani et al., 2008) up to logarithmic factors. In contrast, algorithms like SupLinRel (Auer, 2002) and SupLinUCB (Chu et al., 2011) achieve $\tilde{O}(\sqrt{dT})$ in the setting of finite time-varying adversarial arm sets under $K \ll 2^d$, with a lower bound $\Omega(\sqrt{dT})$ (Chu et al., 2011). The SupLinUCB algorithm was later optimized and matches the lower bound up to iterated logarithmic factors in Li et al. (2019). As a special case of the finite arm setting, if the arm set is time-invariant, an elimination-based algorithm (Lattimore and Szepesvári, 2020) via G-optimal design can be applied to achieve similar optimal performance.

The federated linear bandits problems were studied under the settings of infinite arm set (Dubey and Pentland, 2020; Li et al., 2020; Li and Wang, 2022a) and time-invariant finite arm set (Huang et al., 2021), while the time-varying finite arm set setting has not been well explored. A finite time-varying arm set has many meaningful practical applications such as recommendation system (Li et al., 2010; Chu et al., 2011), and the distributed (federated) nature of the applications naturally falls in the federated linear bandits problem with finite time-varying arms. The paper fills this gap by generalizing the SupLinUCB algorithm to the federated setting.

We study both the asynchronous setting (Li and Wang, 2022a) (He et al., 2022a), where clients are active on their own and full participation is not required, and the synchronous setting (Shi et al., 2021; Dubey and Pentland, 2020), where all the clients make decisions at each round and the communication round requires all the clients to upload new information to the server and download the updated information. We design algorithms so as to reduce the communication cost while maintaining optimal regret. Technically, the communication cost is associated with the algorithmic adaptivity, since less adaptivity requires fewer updates and thus fewer communication rounds. The algorithmic adaptivity of linear bandits algorithms was studied in the single-player setting (Han et al., 2020) (Ruan et al., 2021). It was also considered in the federated setting (Wang et al., 2019; Huang et al., 2021; Salgia and Zhao, 2023).

## 3 System Model and Preliminaries

### 3.1 Problem Formulation

We consider a federated linear contextual bandits model with $K$ finite but possibly time-varying arms. The model consists of $M$ clients and one server in a star-shaped communication network. Clients jointly solve a linear bandit problem by collecting local information and communicating with the central server through the star-shaped network in a federated manner, with no direct communications among clients. The only function of the server is to aggregate received client information and to send back updated information to clients. It cannot directly play the bandits game.

Specifically, some clients $I_t \subseteq [M]$ are active at round $t$. Client $i \in I_t$ receives $K$ arms (actions to take) associated with contexts $\{x_{t,a}^i\}_{a\in[K]} \subset \mathbb{R}^d$ with $\|x_{t,a}^i\|_2 \leq 1$. Here we adopt the oblivious adversarial setting, where all contexts are chosen beforehand, and not dependent on previous game observation. Client $i$ then pulls an arm $a_t^i \in [K]$ based on the information collected locally as well as previously communicated from the server. A reward $r_{t,a_t^i}^i = \theta^\top x_{t,a_t^i}^i + \epsilon_t$ is revealed privately to client $i$, where $\theta \in \mathbb{R}^d$ is an unknown weight vector with $\|\theta\|_2 \leq 1$ and $\epsilon_t$ is an independent 1-sub-Gaussian noise. At the end of round $t$, depending on the communication protocol, client $i$ may exchange the collected local information with the server so that it can update the global information.

We aim to design algorithms to guide the clients' decision-making and overall communication behaviors. We analyze two patterns of client activity. 1) **Synchronous**: all $M$ clients are active at each round. 2) **Asynchronous**: one client is active at each round. For the latter case, we further assume that client activity is independent of data and history. Denote by $T_i$ the number of times client

$i$ is active. In the former case, $T_i = T_j, \forall i, j \in [M]$, while in the latter case, $T_i$ may be different among clients. We define $T = \sum_{i=1}^{M} T_i$ as the total number of arm pulls from all clients.

The performance is measured under two metrics – *total regret* and *communication cost*, which concern the decision-making effectiveness and the communication efficiency respectively. Denote by $P_T^i = \{t \in [T] \mid i \in I_t\}$ the set of time indices at which client $i$ is active, with $|P_T^i| = T_i$. The total regret is defined as

$$R_T = \sum_{i=1}^{M} R_T^i = \sum_{i=1}^{M} \mathbb{E}\left[\sum\nolimits_{t \in P_T^i} r_{t,a_t^{i,*}}^i - r_{t,a_t^i}^i\right], \tag{1}$$

where $a_t^{i,*} = \arg\max_{a \in [K]} \theta^\top x_{t,a}^i$. We define the communication cost as the total number of communication rounds between clients and the server.

### 3.2 Preliminaries

**Information encoding.** In the linear bandits setting (federated or not), the information a client acquires is usually encoded by the gram matrix and the action-reward vector. Specifically, when the client has observed $n$ action-reward pairs $\{(x_t, r_t)\}_{t=1}^n$, the information is encoded by matrix $A_n = \sum_{t=1}^n x_t x_t^\top$ and vector $b_n = \sum_{t=1}^n r_t x_t$. Denote by $\mathrm{Encoder}(\cdot)$ this encoding function, i.e., $A_n, b_n \leftarrow \mathrm{Encoder}(\{x_t, r_t\}_{t=1}^n)$.

**Communication criterion.** Communication in our proposed framework is data-dependent, in the same spirit as the "doubling trick" introduced in Abbasi-Yadkori et al. (2011) to reduce the computation complexity in single-player linear bandits. The key idea is that communication is triggered only when the cumulative local information, represented by the determinant of the gram matrix $A_n$, affects the exploration uncertainty to a great extent and hence the client needs to communicate with the server. Detailed communication protocols will be presented in each algorithm design.

**Synchronization procedure.** Denote by $\mathrm{Sync}()$ a routine that $n$ clients (client 1, ..., client $n$) first communicate their local gram matrices and action-reward vectors to the server, and the server then aggregates the matrices (and vectors) into one gram matrix (and action-reward vector) and transmits them back to the $n$ clients. Specifically, each client $i$ holds newly observed local information $(\Delta A^i, \Delta b^i)$, which is the difference between the client's current information $(A^i, b^i)$ and the information after the last synchronization. In other words, $(\Delta A^i, \Delta b^i)$ is the information that has not been communicated to the server. The server, after receiving the local information $\{(\Delta A^i, \Delta b^i)\}_{i=1}^n$, updates the server-side information $(A^{ser}, b^{ser})$ by $A^{ser} \leftarrow A^{ser} + \sum_{i=1}^n \Delta A^i, b^{ser} \leftarrow b^{ser} + \sum_{i=1}^n \Delta b^i$ and sends them back to each of the $n$ clients. Each client $i$ will then update the local information by $A^i \leftarrow A^{ser}, b^i \leftarrow b^{ser}$. The procedure is formally presented in Algorithm 1.

**Algorithm 1** $\mathrm{Sync}(s$, server, client $1, \ldots$ client $n)$

1: **for** $i = 1, 2, \ldots, n$ **do** ▷ Client-side local information upload
2: Client $i$ sends the local new layer $s$ information $(\Delta A_s^i, \Delta b_s^i)$ to the server
3: **end for**
4: Update server's layer $s$ information: ▷ Server-side information aggregation and distribution

$$A_s^{ser} \leftarrow A_s^{ser} + \sum\nolimits_{i=1}^n \Delta A_s^i, \quad b_s^{ser} \leftarrow b_s^{ser} + \sum\nolimits_{i=1}^n \Delta b_s^i$$

5: Send server information $A_s^{ser}, b_s^{ser}$ back to all clients
6: **for** $i = 1, 2, \ldots, n$ **do**
7: $A_s^i \leftarrow A_s^{ser}, b_s^i \leftarrow b_s^{ser}, \Delta A_s^i \leftarrow 0, \Delta b_s^i \leftarrow 0$ ▷ Client $i$ updates the local information
8: **end for**

## 4 The `FedSupLinUCB` Framework

In this section, we present a general framework of federated bandits for linear bandits with finite oblivious adversarial actions. Two instances (asynchronous and synchronous) of this general framework will be discussed in subsequent sections.

**Building block: SupLinUCB.** As the name suggests, the proposed `FedSupLinUCB` framework is built upon the principle of SupLinUCB (Chu et al., 2011; Ruan et al., 2021). The information $(A, b)$ is useful in the sense that the reward corresponding to an action $x$ can be estimated within confidence interval $x^\top \hat{\theta} \pm \alpha \|x\|_{A^{-1}}$, where $\hat{\theta} = A^{-1}b$. It is shown in Abbasi-Yadkori et al. (2011) that in linear bandits (even with an infinite number of actions) with $\alpha = \tilde{O}(\sqrt{d})$, the true reward is within the confidence interval with high probability. Moreover, if the rewards in the action-reward vector $b$ are mutually independent, $\alpha$ can be reduced to $O(1)$. The former choice of $\alpha$ naturally guarantees $\tilde{O}(d\sqrt{T})$ regret. However, to achieve regret $\tilde{O}(\sqrt{dT})$, it is critical to keep $\alpha = O(1)$. This is fulfilled by the SupLinUCB algorithm (Chu et al., 2011) and then recently improved by Ruan et al. (2021). The key intuition is to successively refine an action set that contains the optimal action, where the estimation precision of sets is geometrically strengthened. Specifically, the algorithm maintains $(S+1)$ layers of information pairs $\{(A_s, b_s)\}_{s=0}^{S}$, and the rewards in the action-reward vectors are mutually independent, except for layer 0. The confidence radius for each layer $s$ is $w_s = 2^{-s} d^{1.5}/\sqrt{T}$.

**Algorithm 2** `S-LUCB`

**Initialization**: $S = \lceil \log d \rceil$, $\overline{w}_0 = d^{1.5}/\sqrt{T}$, $\overline{w}_s \leftarrow 2^{-s}\overline{w}_0, \forall s \in [1:S]$.
$\alpha_0 = 1 + \sqrt{d \ln(2M^2T/\delta)}$, $\alpha_s \leftarrow 1 + \sqrt{2\ln(2KMT \ln d/\delta)}, \forall s \in [1:S]$
**Input**: Client $i$ (with local information $A^i, b^i, \Delta A^i, \Delta b^i$), contexts set $\{x^i_{t,1}, \ldots, x^i_{t,K}\}$
$A^i_{t,s} \leftarrow A^i_s + \Delta A^i_s, b^i_{t,s} \leftarrow b^i_s + \Delta b^i_s$ or $A^i_{t,s} \leftarrow A^i_s, b^i_{t,s} \leftarrow b^i_s$ for `lazy update`
$\hat{\theta}_s \leftarrow (A^i_{t,s})^{-1} b^i_{t,s}$, $\hat{r}^i_{t,s,a} = \hat{\theta}_s^\top x^i_{t,a}$, $w^i_{t,s,a} \leftarrow \alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}, \forall s \in [0:S], \forall a \in [K]$
$s \leftarrow 0$; $\mathcal{A}_0 \leftarrow \{a \in [K] \mid \hat{r}^i_{t,0,a} + w^i_{t,0,a} \geq \max_{a \in [K]} (\hat{r}^i_{t,0,a} - w^i_{t,0,a})\}$ ▷ Initial screening
**repeat** ▷ Layered successive screening
  **if** $s = S$ **then**
    Choose action $a^i_t$ arbitrarily from $\mathcal{A}_S$
  **else if** $w^i_{t,s,a} \leq \overline{w}_s$ for all $a \in \mathcal{A}_s$ **then**
    $\mathcal{A}_{s+1} \leftarrow \{a \in \mathcal{A}_s \mid \hat{r}^i_{t,s,a} \geq \max_{a' \in \mathcal{A}_s} (\hat{r}^i_{t,s,a'}) - 2\overline{w}_s\}$; $s \leftarrow s+1$
  **else**
    $a^i_t \leftarrow \arg\max_{\{a \in \mathcal{A}_s, w^i_{t,s,a} > \overline{w}_s\}} w^i_{t,s,a}$
  **end if**
**until** action $a^i_t$ is found
Take action $a^i_t$ and and receive reward $r^i_{t,a^i_t}$
$\Delta A^i_s \leftarrow \Delta A^i_s + x^i_{t,a^i_t} x^{i\top}_{t,a^i_t}$, $\Delta b^i_s \leftarrow \Delta b^i_s + r^i_{t,a^i_t} x^i_{t,a^i_t}$ ▷ Update local information
**Return** layer index $s$

**FedSupLinUCB.** `S-LUCB`, presented in Algorithm 2, combines the principles of SupLinUCB and OFUL (Abbasi-Yadkori et al., 2011) and is the core subroutine for `FedSupLinUCB`. We maintain $S = \lceil \log d \rceil$ information layers, and the estimation accuracy starts from $d^{1.5}/\sqrt{T}$ of layer 0 and halves as the layer index increases. Finally, it takes $\Theta(\log d)$ layers to reach the sufficient accuracy of $\sqrt{d/T}$ and achieves the minimax-optimal regret.

When client $i$ is active, the input parameters $(A^i, b^i)$ contain information received from the server at the last communication round, and $(\Delta A^i, \Delta b^i)$ is the new local information collected between two consecutive communication rounds. $\{x^i_{t,1}, \ldots, x^i_{t,K}\}$ is the set of contexts observed in this round. Client $i$ can estimate the unknown parameter $\theta$ either with all available information or just making a lazy update. This choice depends on the communication protocol and will be elaborated later. During the decision-making process, client $i$ first makes arm elimination at layer 0 to help bootstrap the accuracy parameters. Then, it goes into the layered successive screening in the same manner as the SupLinUCB algorithm, where we sequentially eliminate suboptimal arms depending on their empirical means and confidence widths. After taking action $a^i_t$ and receiving the corresponding reward $r^i_{t,a^i_t}$, client $i$ updates its local information set $(\Delta A^i_s, \Delta b^i_s)$ by aggregating the context into layer $s$ in which we take the action, before returning layer $s$.

## 5 Asynchronous FedSupLinUCB

In the asynchronous setting, only one client is active in each round. Note that global synchronization and coordination are not required, and all inactive clients are idle.

### 5.1 Algorithm

We first initialize the information for all clients and the server (gram matrix and action-reward vector) in each layer $s \in [0:S]$. We assume only one client $i_t$ is active at round $t$. It is without loss of generality since if multiple clients are active, we can queue them up and activate them in turn. More discussion of this equivalence can be found in He et al. (2022a); Li and Wang (2022a). The active client chooses the action, receives a reward, updates local information matrices of layer $s$ with a lazy update according to S-LUCB, and decides whether communication with the server is needed by the criterion in Line 7 of Algorithm 3. If communication is triggered, we synchronize client $i_t$ with the server by Algorithm 1.

**Algorithm 3** Async-FedSupLinUCB

1: **Initialization**: $T$, $C$, $S = \lceil \log d \rceil$
2: $\{A_s^{ser} \leftarrow I_d, b_s^{ser} \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization
3: $\{A_s^i \leftarrow I_d, \Delta A_s^i, b_s^i, \Delta b_s^i \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization
4: **for** $t = 1, 2, \cdots, T$ **do**
5: Client $i_t = i$ is active, and observes $K$ contexts $\{x_{t,1}^i, x_{t,2}^i, \cdots, x_{t,K}^i\}$
6: $s \leftarrow$ S-LUCB $\left(\text{client } i, \{x_{t,1}^i, x_{t,2}^i, \cdots, x_{t,K}^i\}\right)$ with lazy update
7: **if** $\frac{\det\left(A_s^i + \Delta A_s^i\right)}{\det(A_s^i)} > (1+C)$ **then**
8: Sync($s$, server, clients $i$) for each $s \in [0:S]$
9: **end if**
10: **end for**

### 5.2 Performance Analysis

**Theorem 5.1.** *For any $0 < \delta < 1$, if we run Algorithm 3 with $C = 1/M^2$, then with probability at least $1-\delta$, the regret of the algorithm is bounded as $R_T \leq \tilde{O}\left(\sqrt{d\sum_{i=1}^M T_i}\right) = \tilde{O}\left(\sqrt{dT}\right)$. Moreover, the corresponding communication cost is bounded by $O(dM^2 \log d \log T)$.*

*Remark 1.* The minimax lower bound of the expected regret for linear contextual bandits with $K$ adversarial actions is $\Omega(\sqrt{dT})$, given in Chu et al. (2011). Theorem 5.1 indicates that Async-FedSupLinUCB achieves order-optimal regret (up to polylog term) with $O(dM^2 \log d \log T)$ communication cost. To the best of our knowledge, this is the first algorithm that achieves the (near) optimal regret in federated linear bandits with finite adversarial actions.

*Remark 2.* Without any communication, each client would execute SupLinUCB (Chu et al., 2011) for $T_i$ rounds locally, and each client can achieve regret of order $\tilde{O}(\sqrt{dT_i})$. Therefore, the total regret of $M$ clients is upper bound by $R_T \leq \sum_{i=1}^M \sqrt{dT_i}\,\text{polylog}(T) \leq \sqrt{dM\sum_{i=1}^M T_i}\,\text{polylog}(T)$, where the last inequality becomes equality when $T_i = T_j, \forall i,j \in [M]$. Compared with conducting $M$ independent SupLinUCB algorithms locally, Async-FedSupLinUCB yields an average *per-client gain of* $1/\sqrt{M}$, demonstrating that communications in the federated system can speed up local linear bandits decision-making at clients.

*Remark 3.* Most previous federated linear bandits consider the infinite action setting, based on the LinUCB principle (Abbasi-Yadkori et al., 2011). Async-FedSupLinUCB considers a finite adversarial action setting and has a $\sqrt{d}$ reduction on the regret bound. Fed-PE proposed in Huang et al. (2021) also considers the finite action setting. However, their action sets are fixed. We generalize their formulation and take into account a more challenging scenario, where the finite action set can be chosen adversarially. The regret order is the same as Fed-PE (ignoring the ploylog term), indicating

that it is the *finite* actions as opposed to *fixed* actions that fundamentally leads to the $\sqrt{d}$ regret improvement in the federated linear bandits setting.

**Communication cost analysis of** `FedSupLinUCB`**.** We sketch the proof for the communication cost bound in Theorem 5.1 in the following, while deferring the detailed proofs for the regret and the communication cost to Appendix C.

We first study the communication cost triggered by some layer $s$. Denote by $A_{t,s}^{ser}$ the gram matrix in the server aggregated by the gram matrices uploaded by all clients up to round $t$. Define $T_{n,s} = \min\{t \in [T] | \det(A_{t,s}^{ser}) \geq 2^n\}$, for each $n \geq 0$. We then divide rounds into epochs $\{T_{n,s}, T_{n,s} + 1, \cdots, \min(T_{n+1,s} - 1, T)\}$ for each $n \geq 0$. The number of communications triggered by layer $s$ within any epoch can be upper bounded by $2(M + 1/C)$ (see Lemma C.1), and the number of non-empty epochs is at most $d \log(1 + T/d)$ by Lemma A.1. Since there are $S = \lceil \log d \rceil$ layers and synchronization among all layers is performed once communication is triggered by any layer (Line 8 in Algorithm 3), the total communication cost is thus upper-bounded by $O(d(M + 1/C) \log d \log T)$. Plugging $C = 1/M^2$ proves the result.

We note that although choosing a larger $C$ would trigger fewer communications, the final choice of $C = 1/M^2$ takes into consideration both the regret and the communication cost, i.e., to achieve a small communication cost while maintaining an order-optimal regret.

## 6 Synchronous `FedSupLinUCB`

In the synchronous setting, all clients are active and make decisions at each round. Though it can be viewed as a special case of the asynchronous scenario (clients are active and pulling arms in a round-robin manner), the information update is broadcast to all clients. In other words, the key difference from the asynchronous scenario besides that all clients are active at each round is that when a client meets the communication criterion, *all* clients will upload local information to the server and download the updated matrices. This leads to a higher communication cost per communication round, but in this synchronous scenario, knowing all clients are participating allows the communicated information to be well utilized by other clients. This is in sharp contrast to the asynchronous setting, where if many other clients are active in the current round, uploading local information to the clients seems unworthy. To mitigate the total communication cost, we use a more refined communication criterion to enable time-independent communication cost.

### 6.1 The Algorithm

The Sync-`FedSupLinUCB` algorithm allows each client to make decisions by the `S-LUCB` subroutine. Note that the decision-making is based on all available local information instead of the lazy update in the Async-`FedSupLinUCB` algorithm. The communication criterion involves the count of rounds since the last communication, which forces the communication to prevent the local data from being obsolete. Some layers may trigger the communication criterion either because the local client has gathered enough new data or due to having no communication with the server for too long. We categorize these layers in the CommLayers and synchronize all the clients with the server.

### 6.2 Performance Analysis

**Theorem 6.1.** *For any $0 < \delta < 1$, if we run Algorithm 4 with $D = \frac{T_c \log T_c}{d^2 M}$, with probability at least $1 - \delta$, the regret of the algorithm is bounded as $R_T \leq \tilde{O}(\sqrt{dMT_c})$ where $T_c$ is the total per-client arm pulls. Moreover, the corresponding communication cost is bounded by $O(\sqrt{d^3 M^3} \log d)$.*

*Remark 4.* Theorem 6.1 demonstrates Sync-`FedSupLinUCB` also achieves the minimax regret lower bound while the communication cost is independent of $T_c$. It is particularly beneficial for large $T_c$. Especially, the number of total rounds in the synchronous scenario is $T = MT_c$, while in the asynchronous setting, we have $T = \sum_{i=1}^{M} T_i$ rounds.

**Communication cost analysis of** `Sync-FedSupLinUCB`**.** We sketch the proof for the communication cost bound in Theorem 6.1 below, while deferring the detailed proofs for the regret and the communication cost to Appendix D.

**Algorithm 4** Sync-FedSupLinUCB

**Initialization**: $T_c$, $D$, $S = \lceil \log d \rceil$, $t^s_{last} \leftarrow 0, \forall s \in [0:S]$, CommLayers $\leftarrow \emptyset$.
$\{A^{ser}_s \leftarrow I_d, b^{ser}_s \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization
$\{A^i_s \leftarrow I_d, \Delta A^i_s, b^i_s, \Delta b^i_s \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization
**for** $t = 1, 2, \cdots, T_c$ **do**
  **for** $i = 1, 2, \cdots, M$ **do**
    Client $i_t = i$ is active, and observes $K$ contexts $\{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}$
    $s \leftarrow$ S-LUCB $\left(\text{client } i, \{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}\right)$
    **if** $(t - t^s_{last}) \log \frac{\det\left(A^i_s + \Delta A^i_s\right)}{\det(A^i_s)} > D$ **then**
      Add $s$ to CommLayers
    **end if**
  **end for**
**end for**
**for** $s \in$ CommLayers **do**
  Sync($s$, server, clients $[M]$); $t^s_{last} \leftarrow t$, CommLayers $\leftarrow \emptyset$
**end for**

We call the chunk of consecutive rounds without communicating information in layer $s$ (except the last round) an *epoch*. Information in layer $s$ is collected locally by each client and synchronized at the end of the epoch, following which the next epoch starts. Denoted by $A^{all}_{p,s}$ the synchronized gram matrix at the end of the $p$-th epoch. For any value $\beta > 0$, there are at most $\lceil \frac{T_c}{\beta} \rceil$ epochs that contain more than $\beta$ rounds by pigeonhole principle. If the $p$-th epoch contains less than $\beta$ rounds, then $\log\left(\frac{\det\left(A^{all}_{p,s}\right)}{\det\left(A^{all}_{p-1,s}\right)}\right) > \frac{D}{\beta}$ based on the communication criterion and the fact that $\sum_{p=1}^{P} \log \frac{\det\left(A^{all}_{p,s}\right)}{\det\left(A^{all}_{p-1,s}\right)} \leq R_s = O(d \log(T_c))$ (see Equation (6)). The number of epochs containing rounds fewer than $\beta$ is at most $O(\lceil \frac{R_s}{D/\beta} \rceil)$. Noting that $D = \frac{T_c \log(T_c)}{d^2 M}$, the total number of epochs for layer $s$ is at most $\lceil \frac{T_c}{\beta} \rceil + \lceil \frac{R_s \beta}{D} \rceil = O(\sqrt{\frac{T_c R_s}{D}}) = O(\sqrt{d^3 M})$ by taking $\beta = \sqrt{\frac{DT_c}{R_s}}$. The total communication cost is thus upper bounded by $O(SM\sqrt{d^3 M}) = O(\log(d)\sqrt{d^3 M^3})$.

## 7 Extensions of FedSupLinUCB

In this section, we extend the FedSupLinUCB algorithm to address two distinct settings in federated systems: scenarios characterized by heterogeneous variances, and those affected by adversarial corruptions.

### 7.1 Federated Heteroscedastic Linear Bandits

We have so far focused on the federated linear bandits with 1-sub-Gaussian reward noises. In this section, we adapt Async-FedSupLinUCB to the case where the reward noises have *heterogeneous* variances, which extends the *heteroscedastic linear bandits* as studied in Zhou et al. (2021); Zhou and Gu (2022) to the asynchronous federated setting, where one client is active at a time. Specifically, the reward noises $\{\epsilon_t\}_{t \in [T]}$ are independent with $|\epsilon_t| \leq R, \mathbb{E}[\epsilon_t] = 0$ and $\mathbb{E}[\epsilon_t^2] \leq \sigma_t^2$, where $\sigma_t$ is known to the active client.

We propose a variance-adaptive Asyc-FedSupLinUCB and analyze its regret and the communication cost in the theorem below, with the algorithm and the proof details in Appendix E due to space constraint. The regret is significantly less than that of the Async-FedSupLinUCB when the variances $\{\sigma_t^2\}$ are small.

**Theorem 7.1.** *For any $0 < \delta < 1$, if we run the variance-adaptive Async-FedSupLinUCB algorithm in Appendix E with $C = 1/M^2$, with probability at least $1 - \delta$, the regret is bounded as $R_T \leq \tilde{O}(\sqrt{d \sum_{t=1}^{T} \sigma_t^2})$, and the communication cost is bounded by $O(dM^2 \log^2 T)$.*

### 7.2 Federated Linear Bandits with Corruption

We further explore asynchronous federated linear bandits with adversarial corruptions, where an adversary inserts a corruption $c_t$ to the reward $r_t$ of the active client at round $t$. The total corruption is bounded by $\sum_{t=1}^{T} |c_t| \leq C_p$. We incorporate the idea of linear bandits with adversarial corruption studied in He et al. (2022b) to the proposed `FedSupLinUCB` framework and propose the Robust Async-`FedSupLinUCB` algorithm, with details in Appendix F. Robust Async-`FedSupLinUCB` can achieve the optimal minimax regret (matching the lower bound in He et al. (2022b)) while incurring a low communication cost.

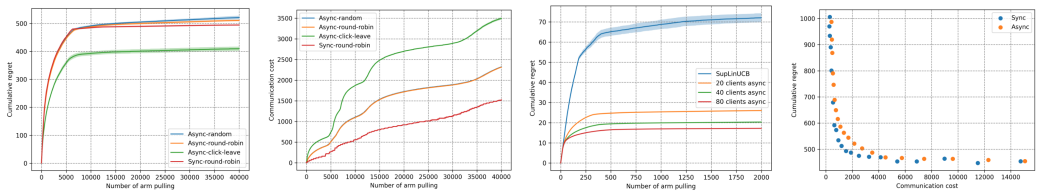
**Theorem 7.2.** *For any $0 < \delta < 1$, if we run the Robust Async-`FedSupLinUCB` algorithm in Appendix F with $C = 1/M^2$, with probability at least $1 - \delta$, the regret is bounded as $R_T \leq \tilde{O}(\sqrt{dT} + dC_p)$, and the communication cost is bounded by $O(dM^2 \log d \log T)$.*

## 8 Experiments

We have experimentally evaluated `FedSupLinUCB` in the asynchronous and synchronous settings on both synthetic and real-world datasets. We report the results in this section.

### 8.1 Experiment Results Using Synthetic Dataset

We simulate the federated linear bandits environment specified in Section 3. With $T = 40000$, $M = 20$, $d = 25$, $\mathcal{A} = 20$, contexts are uniformly randomly sampled from an $l_2$ unit sphere, and reward $r_{t,a} = \theta^\top x_{t,a} + \epsilon_t$, where $\epsilon_t$ is Gaussian distributed with zero mean and variance $\sigma = 0.01$. It should be noted that while $M$ clients participate in each round in the synchronous scenario, only one client is active in the asynchronous case. In the plots, the $x$-axis coordinate denotes the number of arm pulls, which flattens the actions in the synchronous setting.




(a) Regret: arrival patterns. (b) Communication: arrival patterns. (c) Regret: client numbers. (d) Regret vs communications.

Figure 1: Experimental results with the synthetic dataset.

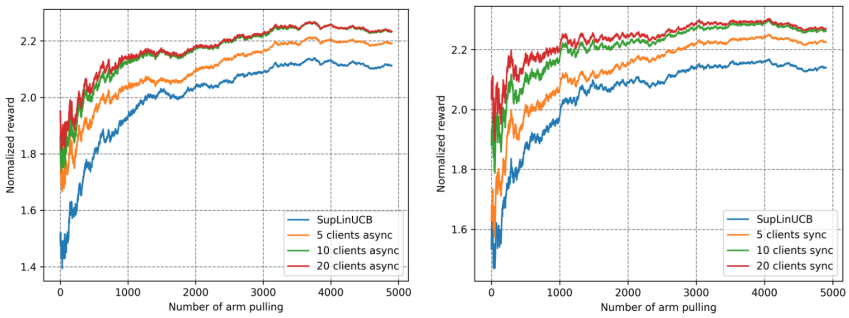
**Arrival pattern.** We first investigate the impact of different arrival patterns (the sequence of activating clients): (1) **Random**, which randomly allocates $T/M$ arm pulls in $[T]$ for each client. (2) **Round-robin**, i.e. $[1, 2, 3, \cdots, M, 1, 2, 3, \cdots M, \cdots]$. (3) **Click-leave**, i.e. $[1, 1, \cdots, 2, 2, \cdots, \cdots, M, M, \cdots]$. The regret and the communication cost of these three arrival patterns in the synthetic experiment are reported in Figure 1(a) and Figure 1(b), respectively. We note that although the upper bound analysis in our proof is for the worst-case instance, the numerical results suggest that different arrival patterns result in diverse regret performances. Round-robin and random patterns are more challenging since both local bandit learning and each client's policy updates happen relatively slowly. The click-leave pattern, which is the closest to the centralized setting, achieves the best regret. In addition, compared with Async-`FedSupLinUCB` , Sync-`FedSupLinUCB` achieves better cumulative regrets with a higher communication cost.

**Amount of clients.** The per-client cumulative regret as a function of $T_c = T/M$ with different amounts of clients is plotted in Figure 1(c). In comparison to the baseline SupLinUCB, `FedSupLinUCB` algorithms achieve better regret via communication between clients and the server. We can see from the experiment that `FedSupLinUCB` significantly reduces the per-client regret compared with SupLinUCB, and achieves a better regret as $M$ increases in both asynchronous and synchronous settings.

**Trade-off between regrets and communications.** We evaluate the tradeoff between communication and regret by running FedSupLinUCB with different communication threshold values $C$ and $D$ in asynchronous and synchronous settings respectively. The results are reported in Figure 1(d), where each scattered dot represents the communication cost and the cumulative regret that FedSupLinUCB has achieved with a given threshold value at round $T = 40000$. We see a clear tradeoff between the regret and the communication. More importantly, Sync-FedSupLinUCB achieves a better tradeoff than Async-FedSupLinUCB.

### 8.2 Experiment Results Using Real-world Dataset

We further investigate how efficiently the federated linear bandits algorithm performs in a more realistic and difficult environment. We have carried out experiments utilizing the real-world recommendation dataset MovieLens 20M (Harper and Konstan, 2015). Following the steps in Li and Wang (2022b), we first filter the data by maintaining users with above 2500 movie ratings and treating rating points greater than 3 as positive, ending up with $N = 37$ users and 121934 total movie rating interactions. Then, we follow the process described in Cesa-Bianchi et al. (2013) to generate the contexts set, using the TF-IDF feature $d = 25$ and the arm set $K = 20$. We plot the per-client normalized rewards of the FedSupLinUCB algorithm with different client numbers $M$ in synchronous and asynchronous cases respectively. Note that the per-client cumulative rewards here are normalized by a random strategy. From Figure 2(a) and Figure 2(b), we can see that in both synchronous and asynchronous experiments, FedSupLinUCB has better rewards than SupLinUCB, and the advantage becomes more significant as the number of users increases.




(a) Async-FedSupLinUCB. (b) Sync-FedSupLinUCB.

Figure 2: Experimental results with the real-world MovieLens-20M dataset.

## 9 Conclusion

We studied federated linear bandits with finite adversarial actions, a model that has not been investigated before. We proposed FedSupLinUCB that extends the SupLinUCB and OFUL principles to the federated setting in both asynchronous and synchronous scenarios, and analyzed their regret and communication cost, respectively. The theoretical results proved that FedSupLinUCB is capable of approaching the minimal regret lower bound (up to polylog terms) while only incurring sublinear communication costs, suggesting that it is the *finite* actions that fundamentally determines the regret behavior in the federated linear bandits setting. Furthermore, we examined the extensions of the algorithm design to the variance-adaptive and adversarial corruption scenarios.

## Acknowledgments and Disclosure of Funding

The work of LF and CS was supported in part by the U.S. National Science Foundation (NSF) under grants 2143559, 2029978, and 2132700.

## References


Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Cesa-Bianchi, N., Gentile, C., and Zappella, G. (2013). A gang of bandits. *Advances in neural information processing systems*, 26.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. *21st Annual Conference on Learning Theory*, pages 355–366.

Dubey, A. and Pentland, A. (2020). Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33:6003–6014.

Han, Y., Zhou, Z., Zhou, Z., Blanchet, J., Glynn, P. W., and Ye, Y. (2020). Sequential batch learning in finite-action linear contextual bandits. *arXiv preprint arXiv:2004.06321*.

Harper, F. M. and Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):1–19.

He, J., Wang, T., Min, Y., and Gu, Q. (2022a). A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. *arXiv preprint arXiv:2207.03106*.

He, J., Zhou, D., Zhang, T., and Gu, Q. (2022b). Nearly optimal algorithms for linear contextual bandits with adversarial corruptions. *Advances in neural information processing systems*.

Huang, R., Wu, W., Yang, J., and Shen, C. (2021). Federated linear contextual bandits. *Advances in Neural Information Processing Systems*, 34:27057–27068.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Li, C. and Wang, H. (2022a). Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6529–6553. PMLR.

Li, C. and Wang, H. (2022b). Communication efficient federated learning for generalized linear bandits. *arXiv preprint arXiv:2202.01087*.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670.

Li, T., Song, L., and Fragouli, C. (2020). Federated recommendation system via differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2592–2597. IEEE.

Li, Y., Wang, Y., and Zhou, Y. (2019). Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*, pages 2173–2174. PMLR.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, pages 1273–1282, Fort Lauderdale, FL, USA.

Ruan, Y., Yang, J., and Zhou, Y. (2021). Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–87.

Salgia, S. and Zhao, Q. (2023). Distributed linear bandits under communication constraints. In *International Conference on Machine Learning*, pages 29845–29875. PMLR.

Shi, C., Shen, C., and Yang, J. (2021). Federated multi-armed bandits with personalization. In *International Conference on Artificial Intelligence and Statistics*, pages 2917–2925. PMLR.

Wang, Y., Hu, J., Chen, X., and Wang, L. (2019). Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*.

Zhou, D. and Gu, Q. (2022). Computationally efficient horizon-free reinforcement learning for linear mixture mdps. *arXiv preprint arXiv:2205.11507*.

Zhou, D., Gu, Q., and Szepesvari, C. (2021). Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR.

## A Supporting Lemmas

**Lemma A.1.** *(Lemma 11 in Abbasi-Yadkori et al. (2011)) Let $\{X_t\}_{t=1}^{\infty}$ be a sequence in $\mathbb{R}^d$, $V$ be a $d \times d$ positive definite matrix, and define $\bar{V}_t = V + \sum_{s=1}^{t} X_s X_s^\top$. Then, we have*

$$\log\left(\frac{\det\left(\bar{V}_n\right)}{\det(V)}\right) \leq \sum_{t=1}^{n} \|X_t\|^2_{\bar{V}_{t-1}^{-1}}.$$

*Further, if $\|X_t\|_2 \leq L$ for all $t$, then*

$$\sum_{t=1}^{n} \min\left\{1, \|X_t\|^2_{\bar{V}_{t-1}^{-1}}\right\} \leq 2\left(\log\det\left(\bar{V}_n\right) - \log\det V\right) \leq 2\left(d\log\left(\left(\mathrm{trace}(V) + nL^2\right)/d\right) - \log\det V\right).$$

*Finally, if $\lambda_{min}(V) \geq \max(1, L^2)$, then*

$$\sum_{t=1}^{n} \|X_t\|^2_{\bar{V}_{t-1}^{-1}} \leq 2\log\frac{\det\left(\bar{V}_n\right)}{\det(V)}.$$

**Lemma A.2.** *(Lemma 12 in Abbasi-Yadkori et al. (2011)). Let $A, B$, and $C$ be positive semi-definite matrices such that $A = B + C$. Then, we have*

$$\sup_{x \neq \mathbf{0}} \frac{x^\top A x}{x^\top B x} \leq \frac{\det(A)}{\det(B)}.$$

**Theorem A.1.** *(Theorem 2 in Abbasi-Yadkori et al. (2011)). Let $\{\mathcal{F}_i\}_{i=0}^{\infty}$ be a filtration. Let $\{x_i\}_{i=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process such that $x_i$ is $\mathcal{F}_{i-1}$-measurable and $\|x_i\| \leq 1$ almost surely. Let $\{\epsilon_i\}_{i=1}^{\infty}$ be a real-valued stochastic process such that $\varepsilon_i$ is $\mathcal{F}_i$-measurable and is sub-Gaussian with variance proxy 1 when conditioned on $\mathcal{F}_{i-1}$. Fix $\theta \in \mathbb{R}^d$ such that $\|\theta\| \leq 1$. Let $A_n = I + \sum_{i=1}^{n} x_i x_i^\top$, $r_i = x_i^\top \theta + \varepsilon_i$, and $\hat{\theta}_n = A_n^{-1} \sum_{i=1}^{n} r_i x_i$. For every $\delta > 0$, we have*

$$\mathbb{P}\left[\forall n \geq 0 : \left\|\hat{\theta}_n - \theta\right\|_{A_n} \leq 1 + \sqrt{d\ln\left(\frac{1+n}{\delta}\right)}\right] \geq 1 - \delta,$$

*where we define $\|x\|_A = \sqrt{x^\top A x}$. Furthermore, when the above event holds, we have for every $n \geq 0$ and any vector $x \in \mathbb{R}^d$ that*

$$\left|x^\top(\hat{\theta}_n - \theta)\right| \leq \left(1 + \sqrt{d\ln\left(\frac{1+n}{\delta}\right)}\right)\sqrt{x^\top A_n^{-1} x}.$$

**Lemma A.3.** *(Adapted from Lemma B.1 in He et al. (2022a)) Under the setting of Theorem 5.1, establish $C = 1/M^2, \alpha_0 = 1 + \sqrt{d\ln(2M^2T/\delta)}$. In layer 0, with probability at least $1 - \delta$, the good event $\mathcal{E}_0$ happens:*

$$\mathcal{E}_0 \triangleq \left\{\left|x_{t,a}^{i\top}\hat{\theta}_{t,s}^i - x_{t,a}^{i\top}\theta\right| \leq w_{t,s,a}^i, \forall i \in [M], a \in [K], t \in [T], s = 0\right\}.$$

**Lemma A.4.** *(Lemma 31 in Ruan et al. (2021)). Given $\theta, x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ such that $\|\theta\| \leq 1$, for all $i \in [n]$, let $r_i = x_i^\top \theta + \epsilon_i$ where $\epsilon_i$ is an independent sub-Gaussian random variable with variance proxy 1. Let $A = I + \sum_{i=1}^{n} x_i x_i^\top$, and $\hat{\theta} = A^{-1}\sum_{i=1}^{n} r_i x_i$. For any $x \in \mathbb{R}^d$ and any $\alpha > 0$, we have*

$$\mathbb{P}\left[|x^\top(\theta - \hat{\theta})| > (\alpha+1)\|x\|_{A^{-1}}\right] \leq 2\exp\left(-\alpha^2/2\right).$$

## B Lemmas for the SupLinUCB Subroutine

We present several useful lemmas that are based on Algorithm 2. Recall that $\Psi_{t,s}$ represents the index set of rounds up to and including round $t$ during which an action is taken in layer $s$. That is,

$$\Psi_{t,s} = \{t' \in [t] : \exists i \in [M],\ a_{t'}^i \text{ is chosen in layer } s\}, \forall s \in [0:S].$$

Similar to Lemma 4 in Chu et al. (2011), we claim that the rewards associated with rounds within each $\Psi_{t,s}, s \in [S]$ (excluding layer 0) are mutually independent.

**Lemma B.1.** *For each $t \in [T]$ each $s \in [S]$, given any fixed sequence of contexts $\{x^i_{t,a}, t \in \Psi_{t,s}\}$, the rewards $\{r^i_{t,s,a}, t \in \Psi_{t,s}\}$ are independent random variables with means $\mathbb{E}[r^i_{t,s,a}] = \theta^\top x^i_{t,s,a}$.*

*Proof of Lemma B.1.* For each $s \in [S]$ and each time $t$, the procedure of generating $\Psi_{t,s}$ only depends on the information in previous layers $\cup_{\sigma<s}\Psi_{t,\sigma}$ and confidence width $\{w^i_{t,s,a}, a \in [K]\}$. From its definition, $w^i_{t,s,a}$ only depends on $\{x_{\tau,a_\tau}, \tau \in \Psi_{t-1,s}\}$ and on the current context $x^i_{t,a}$. Thus the procedure of generating $\Psi_{t,s}$ does not depend on rewards $\{r_{\tau,a_\tau}, \tau \in \Psi_{t-1,s}\}$, and therefore the rewards are independent random variables when conditioned on $\Psi_{t,s}$. □

Given the above-mentioned statistical independence property, and by referring to Lemma A.4, we can establish the following lemma for each layer $s \in [S]$.

**Lemma B.2.** *Suppose the time index set $\Psi_{t,s}$ is constructed so that for fixed $x_{\tau,a_\tau}$ with $\tau \in \Psi_{t,s}$, the rewards $\{r_{\tau,a_\tau}\}$ are independent random variables with mean $\mathbb{E}[r_{\tau,a_\tau}] = \theta^\top x_{\tau,a_\tau}$. For any round $t \in [T]$, if client $i_t = i$ is active and chooses arm $a_t$ in layer $s \in [S]$, then with probability at least $1 - \frac{\delta}{MT\ln d}$, we have for any $a_t \in [K]$:*

$$\left|\hat{r}_{t,s,a_t} - \theta^\top x^i_{t,a_t}\right| \le w^i_{t,s,a_t} = \alpha_s \|x^i_{t,a_t}\|_{(A^i_{t,s})^{-1}}.$$

For layer 0, we employ the self-normalized martingale concentration inequality as outlined in He et al. (2022a). By resorting to Lemma A.3, we obtain the following:

**Lemma B.3.** *For any round $t \in [T]$, given that client $i_t = i$ is active in round $t$ and arm $a_t$ is chosen in layer 0, with probability at least $1 - \delta$, we have for any $a_t \in [K]$:*

$$\left|\hat{r}_{t,0,a_t} - \theta^\top x^i_{t,a_t}\right| \le w^i_{t,0,a_t} = \alpha_0 \|x^i_{t,a_t}\|_{(A^i_{t,0})^{-1}}.$$

Summarizing the discussions presented in Lemma B.2 and Lemma B.3, we now proceed to define the following good event:

**Lemma B.4.** *Define the good event $\mathcal{E}$ as:*

$$\mathcal{E} \triangleq \left\{\left|\hat{r}_{t,s,a} - x^{i\top}_{t,a}\theta\right| \le w^i_{t,s,a}, \forall i \in [M], a \in [K], t \in [T], s \in [0:S]\right\}. \tag{2}$$

*We have $\mathbb{P}[\mathcal{E}] \ge 1 - \delta$.*

Conditioned on the good event $\mathcal{E}$, the ensuing lemma illustrates that the optimal arm persists in the candidate set, and that the regret experienced in each layer aligns with the order of the confidence width.

**Lemma B.5.** *Conditioned on the good event $\mathcal{E}$, for $t \in [T]$, assume that client $i$ is active and chooses an action $a_t \in \mathcal{A}_s$, and recall $(a^i_t)^*$ represents the optimal arm in the current round. For any $s' \le s$, we have:*

$$(a^i_t)^* = \arg\max_{a\in[K]} \theta^\top x^i_{t,a} = \arg\max_{a\in\mathcal{A}_{s'}} \theta^\top x^i_{t,a}.$$

*Proof of Lemma B.5.* For any time step $t \in [T]$, when the good event $\mathcal{E}$ holds, by the arm elimination rule in layer 0, we have

$$\hat{r}_{t,0,a^*} + w_{t,0,a^*} \ge \max_{a\in[K]} \theta^\top x_{t,a} \ge \max_{a\neq a^*} \theta^\top x_{t,a} \ge \max_{a\neq a^*}(\hat{r}_{t,0,a} - w_{t,0,a}).$$

Thus, $a^* \in \mathcal{A}_0$. For each layer $s' < s$, we have:

$$\hat{r}_{t,s',a^*} + w_{t,s',a^*} \ge \max_{a\in\mathcal{A}_{s'}} \theta^\top x_{t,a} \ge \max_{a\neq a^*, a\in\mathcal{A}_{s'}} \theta^\top x_{t,a} \ge \max_{a\neq a^*, a\in\mathcal{A}_{s'}} (\hat{r}_{t,s',a} - w_{t,s',a}).$$

Thus, we derive $\hat{r}_{t,s',a^*} \ge \max_{a\in\mathcal{A}_{s'}}(\hat{r}_{t,s',a}) - 2\overline{w}_{s'}$, which follows from $w_{t,s',a} \le \overline{w}_{s'}$ for all $a \in \mathcal{A}_{s'}$ by the arm elimination rule in Line 10 Algorithm 2. Therefore, arm eliminations will preserve the best arm. □

The forthcoming lemma demonstrates that, under the good event, the regret experienced in each layer aligns with the order of the corresponding confidence width.

**Lemma B.6.** *Conditioned on the good event $\mathcal{E}$, for $t \in [T]$ client $i \in [M]$ and $s \in [S]$, it holds that:*

$$\mathbb{I}[a_t \text{ is chosen in layer } 0](\max_{a \in \mathcal{A}_0} \theta^\top x_{t,a} - \theta^\top x_{t,a_t}) \leq 4w_{t,0,a_t}, \tag{3}$$

$$\mathbb{I}[a_t \text{ is chosen in layer } s](\max_{a \in \mathcal{A}_s} \theta^\top x_{t,a} - \theta^\top x_{t,a_t}) \leq 8\overline{w}_s. \tag{4}$$

*Proof of Lemma B.6.* If an action is taken in layer 0, we have that

$$a_t = \arg \max_{a \in \mathcal{A}_0, w_{t,0,a} > \overline{w_0}} w_{t,0,a},$$

and

$$\begin{aligned}
\max_{a \in \mathcal{A}_0} \theta^\top (x_{t,a} - \theta^\top x_{t,a_t}) &\leq \max_{a \in \mathcal{A}_0} \theta^\top x_{t,a} - \min_{a \in \mathcal{A}_0} \theta^\top x_{t,a} \\
&\leq \max_{a \in \mathcal{A}_0} (\hat{\theta}_0^\top x_{t,a} + w_{t,0,a}) - \min_{a \in \mathcal{A}_0} (\hat{\theta}_0^\top x_{t,a} - w_{t,0,a}) \\
&\leq 4 \max_{a \in \mathcal{A}_0} w_{t,0,a} \\
&= 4w_{t,0,a_t}.
\end{aligned}$$

The second inequality is conditioned on the good event $\mathcal{E}$, and the third inequality arises from the arm elimination rule. If an action is taken in layer $s$, we establish the following:

$$a_t = \arg \max_{a \in \mathcal{A}_s, w_{t,s,a} > \overline{w}_s} w_{t,s,a},$$

and

$$\begin{aligned}
\max_{a \in \mathcal{A}_s} (\theta^\top x_{t,a} - \theta^\top x_{t,a_t}) &\leq \max_{a \in \mathcal{A}_{s-1}} (\hat{\theta}_{s-1}^\top x_{t,a} + w_{t,s-1,a}) - \min_{a \in \mathcal{A}_{s-1}} (\hat{\theta}_{s-1}^\top x_{t,a} - w_{t,s-1,a}) \\
&\leq 2 \max_{a \in \mathcal{A}_{s-1}} w_{t,s-1,a} + \max_{a \in \mathcal{A}_{s-1}} \hat{\theta}_{s-1}^\top - \min_{a \in \mathcal{A}_{s-1}} \hat{\theta}_{s-1}^\top x_{t,a} \\
&\leq 2 \max_{a \in \mathcal{A}_{s-1}} w_{t,s-1,a} + 2\overline{w}_{s-1} \\
&\leq 4\overline{w}_{s-1} \leq 8\overline{w}_s.
\end{aligned}$$

The first inequality is based on the good event $\mathcal{E}$, the third inequality follows the arm elimination rule, and the fourth inequality is due to $w_{t,s-1,a} \leq \overline{w}_{s-1}$ for all $a \in \mathcal{A}_{s-1}$. □

# C Supporting Lemmas and Proofs for Async-FedSupLinUCB

**Lemma C.1.** *(Lemma 6.2 in He et al. (2022a)) In any epoch from round $T_{n,s}$ to round $T_{n+1,s} - 1$, the number of communications is at most $2(M + 1/C)$.*

**Proof outline of Async-FedSupLinUCB.** First, we reorganize the arrival pattern, demonstrating that the rearranged system parallels the original system, and present the requisite definitions for our analysis. Second, we deploy a virtual global model encapsulating information about all clients up to round $t$, subsequently interconnecting the local models with this global model. Lastly, we derive upper bounds on the regret and communication cost in each layer $s \in [0 : S]$ prior to aggregating them to yield the total regret and communication costs, respectively.

Suppose that client $i$ communicates with the server at rounds $t_1, t_2$ with $t_1 < t_2$ and does not communicate during the rounds in between. The actions and information gained by client $i$ at the rounds $t_1 < t < t_2$ do not impact other clients' decision-making, since the information is kept local without communication. Therefore, we can reorder the arrival of clients appropriately while keeping the reordered system equivalent to the original system.

More specifically, suppose client $i$ communicates with the server at two rounds $t_m$ and $t_n$ and does not communicate in the rounds in between (even if she is active). We reorder all the active rounds of client $i$ in $t_m < t < t_n$ and place them sequentially after the round $t_m$. Hence, the arrival of clients can be reordered such that each client communicates with the server and keeps active until the next client's communication begins. We assume that the sequence of communication rounds in the reordered arrival pattern is $0 = t_0 < t_1 < t_2 < \cdots < t_N = T$, where in rounds $t_i \leq t < t_{i+1}$, the

active client is the same. Details of the reordering process are given in Definition C.2. Due to the equivalence between the original system and the reordered system, we carry out the proofs in the reordered system. Note that only one client $i_t$ is active at round $t$, we will write $a_t = a_t^{i_t}$, $x_t = x_{t,a_t}^{i_t}$ and $r_t = r_{t,a_t}^i$ for simplicity.

**Definition C.1.** ***Client information.*** *Recall for each client $i \in [M]$, we denote by $L_i(t)$ the last round when client $i$ communicated with the server before and including round $t$. E.g., $L_i(t) = t$ if client $i$ communicates at round $t$. For each round $t$ each client $i$ and each layer $s$, the information that has been uploaded by client $i$ to the server is: $A_{t,s}^{i,up} = \sum_{t'=1}^{L_i(t)} x_{t'} x_{t'}^\top \mathbb{I}\{i_{t'} = i, a_t \text{ in layer } s\}, b_{t,s}^{i,up} = \sum_{t'=1}^{L_i(t)} r_{t'} x_{t'} \mathbb{I}\{i_{t'} = i, a_t \text{ in layer } s\}$, and the local information in the buffer that has not been uploaded to the server is: $\Delta A_{t,s}^{\mathrm{i}} = \sum_{t'=L_i(t)+1}^{t} x_{t'} x_{t'}^\top \mathbb{I}\{i_{t'} = i, a_t \text{ in layer } s\}, \Delta b_{t,s}^i = \sum_{t'=L_i(t)+1}^{t} r_{t'} x_j \mathbb{I}\{i_{t'} = i, a_t \text{ in layer } s\}$.*

***Server information.*** *The information in the server is the data uploaded by all clients up to round $t$: $A_{t,s}^{ser} = I + \sum_{i=1}^M A_{t,s}^{i,up}, b_{t,s}^{ser} = \sum_{i=1}^M b_{t,s}^{i,up}$.*

***Time index set.*** *Denote by $\Psi_{t,s}$ the time index set when the action $a_t^i$ is chosen in layer $s$. It can be expressed as $\Psi_{t,s} = \{t' \in [t], a_{t'}^i \text{ in layer } s,\ i \in [M]\}, s \in [0:S]\}$.*

***Virtual global information.*** *We define a virtual global model that contains all the information up to round $t$ as: $A_{t,s}^{all} = I + \sum_{t' \in \Psi_{t,s}} x_{t'} x_{t'}^\top, b_{t,s}^{all} = \sum_{t' \in \Psi_{t,s}} r_{t'} x_{t'}$.*

The information that is stored on the server and all the information that has not yet been uploaded by clients are combined to generate the global information: $A_{t,s}^{all} = A_{t,s}^{ser} + \sum_{i=1}^M \Delta A_{t,s}^i, b_{t,s}^{all} = b_{t,s}^{ser} + \sum_{i=1}^M \Delta b_{t,s}^i$.

Before presenting the proof, we define *good event* $\mathcal{E}$ as

$$\mathcal{E} \triangleq \left\{ \left| x_{t,a}^{i\top} \hat{\theta}_{t,s}^i - x_{t,a}^{i\top} \theta \right| \le w_{t,s,a}^i, \forall i \in [M], a \in [K], t \in [T], s \in [0:S] \right\}.$$

Recall $\hat{\theta}_{t,s}^i$ is the estimate of $\theta$ by client $i$, and $x_{t,a}^i$ and $w_{t,s,a}^i$ is the corresponding context and confidence width of the action taken at round $t$. The following lemma shows the good event happens with high probability, similar to the result in Lemma B.4.

**Lemma C.2.** *It holds that $\mathbb{P}[\mathcal{E}] \ge 1 - \delta$.*

Conditioned on the good event, to upper bound the regret, we bound the confidence width in each layer via the size of each time index set in the lemma below.

**Lemma C.3.** *Conditioned on the good event $\mathcal{E}$, for each $s \in [0 : S-1]$ we have:*

$$\sum_{t \in \Psi_{T,s}} w_{t,s,a}^i \le \alpha_s \sqrt{2(1+MC)} \sqrt{2d|\Psi_{T,s}| \log |\Psi_{T,s}|} + \alpha_s dM \log(1 + T/d).$$

Noting that $|\Psi_{T,s}| \le T$ naturally holds, we give a tighter (dimension-dependent) bound on the size of $\Psi_{T,0}$ so as to mitigate the larger coefficient $\alpha_0$ as follows.

**Lemma C.4.** *The size of $\Psi_{T,0}$ can be bounded by $|\Psi_{T,0}| \le T \log T \log(2MT/\delta)/d$.*

We postpone the proofs of Lemma C.3 and Lemma C.4 until the end of this section, and instead focus on presenting the regret analysis next. Equipped with the previous lemmas, we are ready to analyze the total regret.

*Proof of Theorem 5.1.* (**Regret analysis**) The total regret can be decomposed w.r.t. layers as follows:

$$R_T = \mathbb{E} \sum_{t \in \Psi_{T,0}} (r_{t,a_t^{i,*}}^i - r_{t,a_t}^i) + \sum_{s=1}^{S} \mathbb{E} \sum_{t \in \Psi_{T,s}} (r_{t,a_t^{i,*}}^i - r_{t,a_t}^i).$$

Conditioned on the good event $\mathcal{E}$, we first bound the regret in layer 0 by

$$\mathbb{E} \sum_{t \in \Psi_{T,0}} (r_{t,a_t^{i,*}}^i - r_{t,a_t}^i) \le \sum_{t \in \Psi_{T,0}} 4 w_{t,0,a_t}$$

$$\leq 4\alpha_0\sqrt{2(1+MC)}\sqrt{2d|\Psi_{T,0}|\log|\Psi_{T,0}|} + 4\alpha_0 dM\log(1+T/d)s \leq \tilde{O}(\sqrt{(1+MC)dT}).$$

The first inequality follows Lemma B.6, the second inequality is from Lemma C.3, and the last inequality is due to Lemma C.4. We next bound the regret in each layer $s \in [1:S-1]$ similarly by

$$\sum_{t\in\Psi_{T,s}} \mathbb{E}\left[r^i_{t,a_t^{i,*}} - r^i_{t,a_t}\right] \leq \sum_{t\in\Psi_{T,s}} 8\overline{w}_s \leq \sum_{t\in\Psi_{T,s}} 8w_{t,s,a_t}$$
$$\leq 8\alpha_s\sqrt{2(1+MC)}\sqrt{2d|\Psi_{T,s}|\log|\Psi_{T,s}|} + 8\alpha_s dM\log(1+T/d) \leq \tilde{O}(\sqrt{(1+MC)dT})$$

where the first inequality follows Lemma B.6, the second inequality is from the arm selection rule in line 13 Algorithm 2, and the third inequality is from Lemma C.3. For the last layer $S$, we have:

$$\sum_{t\in\Psi_{T,S}} \mathbb{E}\left[r^i_{t,a_t^{i,*}} - r^i_{t,a_t}\right] \leq \sum_{t\in\Psi_{T,S}} 8\overline{w}_S \leq 8\overline{w}_S|\Psi_{T,S}| \leq 8\overline{w}_S T \leq 8\sqrt{dT}.$$

Finally, with Lemma C.2, we have $R_T \leq \tilde{O}(\sqrt{(1+MC)dT})$.

(**Communication cost analysis**) Next, we study the communication cost in an asynchronous setting. For each layer $s$, $i \geq 0$, we define $T_{n,s} = \min\{t \in [T] | \det(A^{ser}_{t,s}) \geq 2^i\}$. We divide rounds in each layer into epoch $\{T_{n,s}, T_{n,s}+1, .., \min(T, T_{n+1,s}-1)\}$, and the communication rounds in the epoch $T_{n,s} \leq t \leq T_{n+1,s}-1$ can be bound by Lemma C.1. Let $N'$ be the largest integer such that $T_{N',s}$ is not empty. According to Lemma A.1 that $\log(\det(A^{all}_{t,s})) \leq d\log(1+|\Psi_{T,s}|/d)$, $N' \leq d\log(1+T/d)$. The total number of epochs of layer $s$ is bounded by $d\log(1+T/d)$. By lemma C.1 the communication rounds in layer $s$ is bounded by $O((M+1/C))d\log T$. There are $S = \lceil \log d \rceil$ in the FedSupLinUCB algorithm, the total communication cost is thus upper bound by $O(d(M+1/C)\log d\log T)$. Plugging in $C = 1/M^2$ proves the result.

□

**Definition C.2.** ***(Reorder function)*** *Without loss of generality, we assume all clients communicate with the server at round $t_0 = 0$, and the sequence of rounds that clients communicate with the server in the original system is $0 \leq t_0 < t_1 < t_2 < ... < t_N \leq T$. Define $I_{t,i} = \mathbb{I}$(client $i$ communicates with the server at round $t$). Denote by $L_i(t)$ the last communication round of client $i$ before and including round $t$:*

$$L_i(t) := \inf\{u : \sum_{t'=0}^{u} I_{t',i} = \sum_{t'=0}^{t} I_{t',i}\}.$$

*Denote by $N_i(t)$ the next communication round of client $i$ including and after round $t$:*

$$N_i(t) := \inf\{u : \sum_{t'=t}^{u} I_{t',i} = 1\}.$$

*The round $t \in [T]$ in the original system is placed in round $\phi(t)$ by the reordering function $\phi : [T] \to [T]$. We first reorder the communication round, suppose two consecutive communication rounds $t_n$ and $t_{n+1}$ with $t_n < t_{n+1}$, and client $i$ is active at round $t_n$ and client $j$ is active at round $t_{n+1}$.*

$$\phi(t_{n+1}) = \phi(t_n) + \sum\nolimits_{t'=t_n}^{N_i(t_n)} I(i_{t'} = i) - 1.$$

*Then we reorder the no-communication rounds, assuming client $i$ is active at round $t$ and does not communicate at this round. We first find the last communication round of client $i$ as $L_i(t)$, and place round $t$ by $\phi(t)$:*

$$\phi(t) = \phi(L_i(t)) + \sum\nolimits_{t'=L_i(t)}^{t} I(i_{t'} = i) - 1.$$

**Lemma C.5.** *(Adapted from Lemma 6.5 in He et al. (2022a)) For each round $t \in [T]$ each layer $s \in [0:S]$ and each client $i \in [M]$ , we have:*

$$A^{ser}_{t,s} = I + \sum_{i=1}^{M} A^{i,up}_{t,s} \succeq \frac{1}{C}\Delta A^i_{t,s}.$$

*Further averaging the inequality above over $M$ clients, we have:*

$$A_{t,s}^{ser} = I + \sum_{i=1}^{M} A_{t,s}^{i,up} \succeq \frac{1}{MC} \sum_{i=1}^{M} \Delta A_{t,s}^{i}.$$

*Proof of Lemma C.5.* Without loss of generality, we consider client $i$ and fix any round $t \in [T]$. Let $t_1 \leq t$ be the last round such that client $i$ was active at round $t_1$. If client $i$ communicated with the server at round $t_1$, and chose action $a_{t_1}$ at layer $s$, then we have

$$A_{t,s}^{ser} = I + \sum_{i=1}^{M} A_{t,s}^{i,up} \succeq \frac{1}{C} \Delta A_{t_1,s}^{i} = 0$$

for other layers $s' \neq s$, according to the determinant-based communication criterion, we have:

$$\det\big(A_{t_1,s'}^{i} + \Delta A_{t_1,s'}^{i}\big) < (1+C) \det\big(A_{t_1,s'}^{i}\big).$$

By Lemma A.2 we have

$$A_{t,s'}^{i} = A_{t_1,s'}^{i} \succeq \frac{1}{C} \Delta A_{t_1,s'}^{i}.$$

Otherwise, if no communication happened at round $t_1$, by the communication criterion, at the end of round $t_1$, for each layer $s \in [0:S]$, we have $A_{t_1,s}^{i} \succeq \frac{1}{C} \Delta A_{t_1,s}^{i}$. Note that $\{A_{t_1,s}^{i},\ s \in [0:S]\}$ are the downloaded gram matrices from last communication before round $t_1$, so it must satisfy $A_{t_1,s}^{i} \preceq A_{t_1,s}^{ser}$ for all $s \in [0:S]$. For round $t$, since client $i$ is inactive from round $t_1$ to $t$, we have for all $s \in [0:S]$:

$$A_{t,s}^{ser} \succeq A_{t_1,s}^{ser} \succeq A_{t_1,s}^{i} \succeq \frac{1}{C} \Delta A_{t_1,s}^{i} = \frac{1}{C} \Delta A_{t,s}^{i}$$

where the last equality holds for inactivation, which completes the proof of the first claim. Further average the above inequality over all clients $i \in [M]$, and we get:

$$A_{t,s}^{ser} = I + \sum_{i=1}^{M} A_{t,s}^{i,up} \succeq \frac{1}{MC} \sum_{i=1}^{M} \Delta A_{t,s}^{i}.$$

□

Recall that client $i$ utilizes $A_{t,s}^{i}$ and $b_{t,s}^{i}$ to make the decision at round $t$, which were received from the server during the last communication. The following lemma establishes a connection between the gram matrix of the virtual global model and the gram matrix in the active client at round $t$.

**Lemma C.6.** *In the reordered arrival pattern, for any $1 \leq t_1 < t_2 \leq T$, suppose client $i$ communicates with the server at round $t_1$, and keep active during rounds $t_1 \leq t \leq t_2 - 1$. Then for rounds $t_1 + 1 \leq t \leq t_2 - 1$, it holds that for each $s \in [0:S]$:*

$$A_{t,s}^{i} \succeq \frac{1}{1+MC} A_{t,s}^{all}.$$

*Proof of Lemma C.6.* Client $i$ is the only active client from round $t_1$ to $t_2 - 1$ and only communicated with the server at round $t_1$, which implies that for $t_1 + 1 \leq t \leq t_2 - 1 \forall s \in [0:S]$, we have

$$A_{t,s}^{i} = I + \sum_{i=1}^{M} A_{t_1,s}^{i,up} = I + \sum_{i=1}^{M} A_{t,s}^{i,up} \succeq \frac{1}{1+MC} (I + \sum_{i=1}^{M} A_{t,s}^{i,up} + \sum_{i=1}^{M} \Delta A_{t,s}^{i}) \succeq \frac{1}{1+MC} A_{t,s}^{all}$$

where the second equality holds due to the fact that no clients communicate with the server from round $t_1 + 1$ to $t_2 - 1$, and the first inequality follows Lemma C.5. □

*Proof of Lemma C.3.* For $t \in \Psi_{T,s}$, if no communication happened at round $t$, under Lemma C.6 and Lemma A.2, we can connect confidence width at the local client with the global gram matrix as:

$$\|x_{t,a}^{i_t}\|_{(A_{t,s}^{i_t})^{-1}} \leq \sqrt{1+MC} \|x_{t,a}^{i}\|_{(A_{t,s}^{all})^{-1}}.$$

It remains to control the communication rounds in $\Psi_{T,s}$. We define

$$T_n = \min\left\{t \in \Psi_{T,s} \mid \det\left(A_{t,s}^{all}\right) \geq 2^n\right\},$$

and let $N'$ be the largest integer such that $T_{N'}$ is not empty. According to Lemma A.1, we have:

$$\log\big(\det\big(A_{t,s}^{all}\big)\big) \leq d\log(1 + |\Psi_{T,s}|/d).$$

Thus, $N' \leq d\log(1 + T/d)$. For each time interval from $T_n$ to $T_{n+1}$ and each client $i \in [M]$, suppose client $i$ communicates with the server more than once, and communication rounds sequentially are $T_{n,1}, T_{n,2}, \ldots, T_{n,k} \in [T_n, T_{n+1})$. Then for each $j = 2, \ldots, k$, since client $i$ is active at rounds $T_{n,j-1}$ and $T_{n,j}$, we have

$$\|x_{T_{n,j}}\|_{(A^i_{T_{n,j},s})^{-1}} \leq \|x_{T_{n,j}}\|_{(A^i_{T_{n,j-1}+1,s})^{-1}} \leq \sqrt{1+MC}\|x_{T_{n,j}}\|_{((A^{all}_{T_{n,j-1}+1,s})^{-1}}.$$

Since $\det\Big(A^{all}_{T_{n+1}-1,s}\Big)/\det\Big(A^{all}_{T_{n,j-1}+1,s}\Big) \leq 2^{n+1}/2^n = 2$, by the definition of $T_n$, we have:

$$\|x_{T_{n,j}}\|_{(A^i_{T_{n,j},s})^{-1}} \leq \sqrt{2(1+MC)}\|x_{T_{n,j}}\|_{(A^{all}_{T_{n+1}-1,s})^{-1}} \leq \sqrt{2(1+MC)}\|x_{T_{n,j}}\|_{(A^{all}_{T_{n,j},s})^{-1}},$$

where the second inequality comes from $A^{all}_{T_{n+1}-1,s} \succeq A^{all}_{T_{n,j},s}$. Specifically, for round $T_{i,1}$ the first communication round, we can bound the confidence width by 1. Thus, for the communication rounds in $\Psi_{T,s}$, we have:

$$\sum_{t\in\Psi_{T,s},round\ t\ comm} \|x^{i_t}_{t,a}\|_{(A^{i_t}_{t,s})^{-1}} \leq MN' + \sum_{t\in\Psi_{T,s},round\ t\ comm} \sqrt{2(1+MC)}\|x^{i_t}_{t,a}\|_{(A^{all}_{t,s})^{-1}}.$$

Finally, we put all rounds in $\Psi_{T,s}$ together:

$$\begin{aligned}
\sum_{t\in\Psi_{T,s}} w_{t,s,a} &= \alpha_s \sum_{t\in\Psi_{T,s}} \|x^{i_t}_{t,a}\|_{(A^{i_t}_{t,s})^{-1}} \\
&\leq \alpha_s \sum_{t\in\Psi_{T,s}} \sqrt{2(1+MC)}\|x^{i_t}_{t,a}\|_{(A^{all}_{t,s})^{-1}} + \alpha_s MN' \\
&\leq \alpha_s\sqrt{2(1+MC)}\sqrt{2d|\Psi_{T,s}|\log|\Psi_{T,s}|} + \alpha_s dM\log(1+T/d)
\end{aligned}$$

where the second inequality follows Lemma A.1. □

*Proof of Lemma C.4.* Based on the algorithm, if we choose an action in layer 0, the selected arm is

$$a_t = \arg\max_{a\in\mathcal{A}_0, w_{t,0,a}>\overline{w}_0} w_{t,0,a},$$

and the corresponding confidence width satisfies $w_{t,0,a_t} > \overline{w}_0$. Furthermore,

$$\begin{aligned}
\overline{w}_0|\Psi_{T,0}| \leq \sum_{t\in\Psi_{T,0}} w_{t,0,a_t} &= \alpha_0 \sum_{t\in\Psi_{T,0}} \|x_{t,a_t}\|_{(A^{i_t}_{t,0})^{-1}} \\
&\leq \alpha_0\sqrt{2(1+MC)}\sqrt{2d|\Psi_{T,s}|\log|\Psi_{T,s}|} + \alpha_0 dM\log(1+T/d),
\end{aligned}$$

where the last inequality is by Lemma C.3. We can thus conclude that $\Psi_{T,0} \leq T\log T\log(2MT/\delta)/d$.

□

# D Supporting Lemmas and Proofs for Sync-FedSupLinUCB

**Proof outline of Sync-FedSupLinUCB.** To prove a high-probability regret bound, we first define the good event $\mathcal{E}$ in the following lemma, under which the regret bound is derived.

**Lemma D.1.** *Define* $\mathcal{E} \triangleq \{\left|x^{i\top}_{t,a}\hat{\theta}^i_{t,s} - x^{i\top}_{t,a}\theta\right| \leq w^i_{t,s,a}, \forall i \in [M], a \in [K], t \in [T_c], 0 \leq s \leq S\}$. *Then,* $\mathbb{P}[\mathcal{E}] \geq 1 - \delta$.

Define client $i$'s one-step regret at round $t$ as $\mathsf{reg}_t^i = \theta^\top(x^i_{t,a_t^{i*}} - x^i_{t,a_t})$. Let $\mathsf{reg}^i_{t,s} = \mathsf{reg}^i_t$ if action $a_t$ is chosen in layer $s$; otherwise $\mathsf{reg}^i_{t,s} = 0$. The total regret can be written as

$$R_T = \sum_{i=1}^{M}\sum_{t=1}^{T_c} \mathsf{reg}_t^i = \sum_{s=0}^{S}\sum_{i=1}^{M}\sum_{t=1}^{T_c} \mathsf{reg}^i_{t,s}.$$

Fix an arbitrary $s \in \{0, 1, \ldots, S\}$, we analyze the total regret induced by the actions taken in layer $s$, i.e., $R_{s,T_c} = \sum_{i=1}^{M}\sum_{t=1}^{T_c} \mathsf{reg}^i_{t,s}$. The analysis can be carried over to different $s$ in the same manner.

We call the chunk of consecutive rounds without communicating information in layer $s$ (except the last round) an *epoch*. In other words, information in layer $s$ is collected locally by each client and synchronized at the end of the epoch, following which the next epoch starts. The set of rounds that at least one client is pulling an arm in layer $s$ can then be divided into multiple consecutive epochs, and we further dichotomize these epochs into good and bad epochs in the following definition.

**Definition D.1.** ***(Good epoch)*** *Suppose the set of rounds that at least one client is pulling an arm in layer $s$ are divided into $P$ epochs and denoted by $A^{all}_{p,s}, b^{all}_{p,s}$ the synchronized gram matrix and reward-action vector at the end of the $p$-th epoch. $P$ epochs can then be dichotomized into $\mathcal{P}_s^{good} \triangleq \left\{p \in [P] : \frac{\det(A^{all}_{p,s})}{\det(A^{all}_{p-1,s})} \le 2\right\}, \mathcal{P}_s^{bad} \triangleq [P] \setminus \mathcal{P}_s^{good}$, where $A^{all}_{0,s} \triangleq I$. We say round $t$ is* good *if the epoch containing round $t$ belongs to $\mathcal{P}_s^{good}$; otherwise $t$ is* bad.

We bound regrets in layer $s$ induced by the good and bad epochs separately in the following lemmas. Recall $\Psi_{t,s}$ is the time index set when the action $a_t^i$ is chosen in the $s$ layer.

**Lemma D.2.** *Conditioned on the good event $\mathcal{E}$, for each layer $s \in [0 : S]$, the regret induced by good epochs of layer $s$ is bounded as $\sum_{t\in\Psi_{T_c,s},t \text{ is good}} \mathsf{reg}^i_{t,s} \le \tilde{O}\left(\alpha_s\sqrt{d|\Psi_{T_c,s}|\log(MT_c)}\right)$.*

**Lemma D.3.** *Define $D = \frac{T_c \log T_c}{d^2 M}$ and $R_s = d\log\left(1 + \frac{|\Psi_{T_c,s}|}{d}\right)$. Conditioned on the good event $\mathcal{E}$, for each layer $s \in [0 : S]$, the regret induced by bad epochs of layer $s$ is bounded as $\sum_{t\in\Psi_{T_c,s},t \text{ is bad}} \mathsf{reg}^i_{t,s} \le O\left(\alpha_s M\sqrt{D}R_s\right)$.*

**Lemma D.4.** *We have $|\Psi_{T_c,s}| \le \tilde{O}(\frac{MT_c}{d})$.*

*Proof of Theorem 6.1.* (**Regret analysis**) For each $s \in [0 : S]$, the regret induced in layer $s$ is bounded by:

$$\begin{aligned} R_{s,T_c} &\le \sum_{t\in\Psi_{T_c,s},t \text{ is good}} \mathsf{reg}^i_{t,s} + \sum_{t\in\Psi_{T_c,s},t \text{ is bad}} \mathsf{reg}^i_{t,s} \\ &\le O(\alpha_s\sqrt{d|\Psi_{T_c,s}|\log(MT)} + \alpha_s M\sqrt{D}R_s) \le \tilde{O}(\sqrt{dMT_c}) \end{aligned}$$

where the second inequality is from Lemmas D.2 and D.3, and the last inequality is due to Lemma D.4. The total regret can thus be bounded as $R_T = \sum_{s=0}^{S} R_{s,T_c} = \tilde{O}(\sqrt{dMT_c})$. □

*Proof of Lemma D.2.* If $t$ is good and belongs to the $p$-th epoch, we have by Lemma A.2 that

$$w^i_{t,s,a} = \alpha_s\|x^i_{t,a}\|_{(A^i_{t,s})^{-1}} \le \sqrt{2}\alpha_s\|x^i_{t,a}\|_{(A^{all}_{p,s})^{-1}} \le 2\alpha_s\|x^i_{t,a}\|_{(A^{all}_{p-1,s})^{-1}}. \tag{5}$$

Within $p$-th good epoch, we have

$$A^{all}_{p-1,s} + \sum_{i=1}^{M}\sum_{t\in p\text{-th good epoch}} x^i_{t,a^i_t}(x^i_{t,a^i_t})^\top = A^{all}_{p,s},$$

which together with inequality (5) and the last inequality in the elliptical potential lemma (Lemma A.1) imply that

$$\sum_{i=1}^{M}\sum_{t\in p\text{-th good epoch}} \|x^i_{t,a^i_t}\|^2_{(A^i_{t,s})^{-1}} \le 4\log\frac{\det(A^{all}_{p,s})}{\det(A^{all}_{p-1,s})}.$$

Thus under event $\mathcal{E}$, the regret induced by good epochs of layer $s$ is

$$\begin{aligned}\sum_{(i,t)\in\Psi_{T,s},t \text{ is good}} reg^i_{t,s} &\le \sum_{(i,t)\in\Psi_{T,s},t \text{ is good}} 8w^i_{t,s,a^i_t} \\ &\le 8\sqrt{|\Psi_{T,s}| \sum_{(i,t)\in\Psi_{T,s},t \text{ is good}} (w^i_{t,s,a^i_t})^2} \\ &= \tilde{O}\left(\alpha_s\sqrt{d|\Psi_{T,s}|\log(MT)}\right),\end{aligned}$$

where the first inequality is from Lemma B.6, the second inequality is by Cauchy-Schwartz inequality, and the last relation is from

$$\sum_{p=1}^{P}\log\frac{\det\left(A^{all}_{p,s}\right)}{\det\left(A^{all}_{p-1,s}\right)} = \log\det\left(A^{all}_{P,s}\right) \le d\log\left(1+\frac{|\Psi_{T,s}|}{d}\right) = R_s. \tag{6}$$

□

*Proof of Lemma D.3.* Denote by $R_s = d\log\left(1+\frac{|\Psi_{T,s}|}{d}\right)$. It follows that the number of bad epochs is at most $O(R_s)$. Moreover, the regret within a bad epoch of length $n$ can be upper bounded as $O(M+\alpha_s M\sqrt{D})$ by applying the elliptical potential lemma for each client $i$ and the communication condition, where the extra 1 in the upper bound is due to that at most $M$ clients trigger the communication condition at the end of the $p$-th epoch. We thus have

$$\sum_{t \text{ is bad}} reg^i_{t,s} \le \sum_{t\in\Psi_{T,s} \text{ is bad}} 8w^i_{t,s} = O\left(MR_s + \alpha_s M\sqrt{D}R_s\right) = O\left(\alpha_s M\sqrt{D}R_s\right).$$

□

*Proof of Lemma D.4.* Recall $D = \frac{T\log(T)}{d^2M}$. Note that if $\alpha_s\sqrt{d|\Psi_{T,s}|\log(T)} = O(\alpha_s M\sqrt{D}R_s)$, we have $|\Psi_{T,s}| = \tilde{O}(M^2Dd) = \tilde{O}(\frac{MT}{d})$. Otherwise $|\Psi_{T,s}|\overline{w}^s_t = O(\alpha_s\sqrt{d|\Psi_{T,s}|\log(T)})$, which implies $|\Psi_{T,s}| = \tilde{O}(\frac{\alpha_s^2 d}{(\overline{w}^s_t)^2}) = \tilde{O}(\frac{MT4^s}{d})$. □

# E Variance-adaptive Async-FedSupLinUCB

The variance-adaptive SupLinUCB subroutine is presented in Alg. 5, while the complete variance-adaptive Async-FedSupLinUCB is given in Alg. 6.

## E.1 Algorithm

## E.2 Supporting Lemmas and Proofs

**Theorem E.1.** *(Theorem 4.3 in Zhou and Gu (2022)) Let $\{\mathcal{F}_t\}_{t=1}^{\infty}$ be a filtration, and $\{x_t,\eta_t\}_{t\ge1}$ be a stochastic process such that $x_t\in\mathbb{R}^d$ is $\mathcal{F}_t$-measurable and $\eta_t\in\mathbb{R}$ is $\mathcal{F}_{t+1}$-measurable. Let $\sigma,\epsilon>0,\theta^*\in\mathbb{R}^d$. For $t\ge1$, let $y_t=\langle\theta^*,x_t\rangle+\eta_t$ and suppose that $\eta_t,x_t$ also satisfy*

$$\mathbb{E}[\eta_t \mid \mathcal{F}_t] = 0, \mathbb{E}[\eta_t^2 \mid \mathcal{F}_t] \le \sigma^2, |\eta_t| \le R, \|x_t\|_2 \le 1.$$

*For $t\ge1$, let $Z_t = I+\sum_{i=1}^t x_i x_i^\top, b_t=\sum_{i=1}^t y_i x_i, \theta_t = Z_t^{-1}b_t$, and*

$$\begin{aligned}\beta_t =& 12\sqrt{\sigma^2 d\log\left(1+tL^2/(d)\right)\log\left(32(\log(R/\epsilon)+1)t^2/\delta\right)} \\ &+ 24\log\left(32(\log(R/\epsilon)+1)t^2/\delta\right)\max_{1\le i\le t}\left\{|\eta_i|\min\left\{1,\|\mathbf{x}_i\|^{-1}_{\mathbf{Z}^{-1}_{i-1}}\right\}\right\} + 6\log\left(32(\log(R/\epsilon)+1)t^2/\delta\right)\epsilon.\end{aligned}$$

*Then, for any $0<\delta<1$, we have with probability at least $1-\delta$ that,*

$$\forall t\ge1, \left\|\sum_{i=1}^t x_i\eta_i\right\|_{Z_t^{-1}} \le \beta_t, \quad \|\theta_t-\theta^*\|_{Z_t} \le \beta_t + \|\theta^*\|_2.$$

**Algorithm 5** Variance-adaptive SupLinUCB subroutine: VS-LUCB

**Initialization**: $S \leftarrow \lceil \log R + \log T \rceil$, $\overline{w}_0 = dR^2$, $\overline{w}_s \leftarrow 2^{-s}\overline{w}_0, \forall s \in [1:S]$,
$\alpha_0 = \tilde{O}(\sqrt{d}), \alpha_s = 1 + \sqrt{2\ln(2KMT\ln d/\delta)}, \rho = 1/\sqrt{T}, \gamma = R^{1/2}/d^{1/4}$.
**Input**: Client $i$ (with local information $A^i, b^i, \Delta A^i, \Delta b^i$), contexts set $\{x^i_{t,1}, \ldots, x^i_{t,K}\}$
$A^i_{t,s} \leftarrow A^i_s, b^i_{t,s} \leftarrow b^i_s$ for `lazy update`
$\hat{\theta}_s \leftarrow (A^i_{t,s})^{-1} b^i_{t,s}$, $\hat{r}^i_{t,s,a} = \hat{\theta}_s^\top x^i_{t,a}$, $w^i_{t,s,a} \leftarrow \alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}, \forall s \in [0:S], \forall a \in [K]$.
$s \leftarrow 0$; $\mathcal{A}_0 \leftarrow \{a \in [K] \mid \hat{r}^i_{t,0,a} + w^i_{t,0,a} \geq \max_{a \in [K]}(\hat{r}^i_{t,0,a} - w^i_{t,0,a})\}$ ▷ Initial screening
**repeat** ▷ Layered successive screening
  **if** $s = S$ **then**
    Choose action $a^i_t$ arbitrarily from $\mathcal{A}_S$
  **else if** $w^i_{t,s,a} \leq \overline{w}_s$ for all $a \in \mathcal{A}_s$ **then**
    $\mathcal{A}_{s+1} \leftarrow \{a \in \mathcal{A}_s \mid \hat{r}^i_{t,s,a} \geq \max_{a' \in \mathcal{A}_s}(\hat{r}^i_{t,s,a'}) - 2\overline{w}_s\}$; $s \leftarrow s+1$
  **else**
    Choose $a_t = \arg\max_{\{a \in \mathcal{A}_s, w^i_{t,s,a} > \overline{w}_s\}} w^i_{t,s,a}$
  **end if**
**until** action $a_t$ is found
Take action $a_t$ and and receive reward $r^i_{t,a_t}$ and variance $\sigma_t$
$\overline{\sigma}_t = \max\{\sigma_t, \rho, \gamma \|x^i_{t,a_t}\|^{1/2}_{(A^i_{t,s})^{-1}}\}$
$\Delta A^i_s \leftarrow \Delta A^i_s + x^i_{t,a_t} x^{i\top}_{t,a_t}/\overline{\sigma}^2_t$, $\Delta b^i_s \leftarrow \Delta b^i_s + r^i_{t,a_t} x^i_{t,a_t}/\overline{\sigma}^2_t$ ▷ Update local information
Return layer index $s$

**Algorithm 6** Variance-adaptive Async-`FedSupLinUCB`

**Initialization**: $T$, $C$, $S = \lceil \log R + \log T \rceil$
$\{A^{ser}_s \leftarrow I_d, b^{ser}_s \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization
$\{A^i_s \leftarrow I_d, \Delta A^i_s, b^i_s, \Delta b^i_s \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization
**for** $t = 1, 2, ..., T$ **do**
  Client $i_t = i$ is active, and observes $K$ contexts $\{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}$
  $s =$ `VS-LUCB` $\left(clienti, \{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}\right)$ with the lazy update
  **if** $\frac{\det\left(A^i_s + \Delta A^i_s\right)}{\det(A^i_s)} > (1+C)$ **then**
    Sync($s$, server, clients $i$) for each $s \in [0:S]$
  **end if**
**end for**

**Lemma E.1.** *(Adapted from Lemma B.1 in Zhou and Gu (2022)). Let $\{\sigma_t, \beta_t\}_{t \geq 1}$ be a sequence of non-negative numbers, $\rho, \gamma > 0, \{x_t\}_{t \geq 1} \subset \mathbb{R}^d$ and $\|x_t\|_2 \leq 1$. Let $\{Z_t\}_{t \geq 1}$ and $\{\bar{\sigma}_t\}_{t \geq 1}$ be recursively defined as follows:*

$$Z_1 = I; \quad Z_{t+1} = Z_t + x_t x_t^\top / \bar{\sigma}_t^2, \quad \forall t \geq 1, \bar{\sigma}_t = \max\{\sigma_t, \rho, \gamma \|x_t\|^{1/2}_{Z_t^{-1}}\}.$$

*Let $\iota = \log\big(1 + T/(d\rho^2)\big)$. Then we have*

$$\sum_{t=1}^{T} \min\{1, \beta_t \|x_t\|_{Z_t^{-1}}\} \leq 2d\iota + 2\beta_T \gamma^2 d\iota + 2\sqrt{d\iota}\sqrt{\sum_{t=1}^{T} \beta_t^2(\sigma_t^2 + \rho^2)}.$$

Following a similar proof structure to Async-`FedSupLinUCB`, we employ a novel Bernstein-type self-normalized martingale inequality, proposed by Zhou and Gu (2022), for layer 0 to manage the variance information. We define $\alpha_0 = \beta_T$ as specified in Theorem E.1, and establish the following lemma, analogous to Lemma B.3.

**Lemma E.2.** *For any round $t \in [T]$, if client $i_t = i$ is active in round $t$ and arm $a_t$ is chosen in layer 0, with probability at least $1 - \delta$, with $\alpha_0 = \tilde{O}(\sqrt{d})$ we have for any $a_t \in [K]$:*

$$\left|\hat{r}_{t,0,a_t} - \theta^\top x^i_{t,a_t}\right| \leq w^i_{t,0,a_t} = \alpha_0 \|x^i_{t,a_t}\|_{(A^i_{t,0})^{-1}}.$$

We define *good event* $\mathcal{E}$ as $\mathcal{E} \triangleq \left\{ \left| x_{t,a}^{i\top} \hat{\theta}_{t,s}^{i} - x_{t,a}^{i\top} \theta \right| \leq w_{t,s,a}^{i}, \forall i \in [M], a \in [K], t \in [T], s \in [0:S] \right\}$. In a manner similar to the proof of Lemma B.4, we have that $\mathbb{P}[\mathcal{E}] \geq 1 - \delta$.

**Lemma E.3.** *Conditioned on the event $\mathcal{E}$, the regret in layer $0$ can be bounded by* $\mathsf{reg}_{\textit{layer } 0} \leq \tilde{O}(d)$.

*Proof of Lemma E.3.* We set $\overline{w}_0 = dR^2$ to provide a tighter bound for the size of $\Psi_{T,0}$. Mirroring the proof methodology in Lemma C.4, we establish the following:

$$\begin{aligned}\overline{w}_0 |\Psi_{T,0}| &\leq \alpha_0 \sum_{t \in \Psi_{T,0}} \|x_{t,i}\|_{(A_{t,s}^{i_t})^{-1}} \leq 2d\iota + 2\alpha_0 \gamma^2 d\iota + 2\alpha_0 \sqrt{d\iota} \sqrt{\sum_{t \in \Psi_{T,0}} (\sigma_t^2 + \rho^2)} \\ &\leq 2d\iota + 2\alpha_0 \gamma^2 d\iota + 2\alpha_0 \sqrt{d\iota} \sqrt{|\Psi_{T,0}|(R^2 + \rho^2)}.\end{aligned}$$

The first inequality results from the arm selection rule of layer 0, the second is derived from Lemma E.1, and the third arises due to the constraint $\sigma_t^2 \leq R^2$. Consequently, we infer that $\Psi_{T,0} \leq O(d^2 R^2 / \overline{w}_0^2)$. We can then bound the regret in layer $0$ as follows:

$$\mathsf{reg}_{\text{layer } 0} \leq 4\alpha_0 \sum_{t \in \Psi_{T,0}} \|x_{t,i}\|_{(A_{t,s}^{i_t})^{-1}} \leq 8d\iota + 8\alpha_0 \gamma^2 d\iota + 8\alpha_0 \sqrt{d\iota} \sqrt{|\Psi_{T,0}|(R^2 + \rho^2)} \leq \tilde{O}(d).$$

□

**Lemma E.4.** *Conditioned on the event $\mathcal{E}$, the regret of each layer $s \in [1 : S-1]$ can be bounded by* $\mathsf{reg}_{\textit{layer } s} \leq \tilde{O} \sqrt{d \sum_t \sigma_t^2}$.

*Proof of Lemma E.4.* For $s \in \{1, 2, ..., S-1\}$, the rewards in each layer $s$ are mutually independent, as proven in Lemma B.1. We deduce:

$$\begin{aligned}\mathsf{reg}_{\text{layer } s} &\leq 8\overline{w}_s |\Psi_{T,s}| \leq 8\alpha_s \sum_{t \in \Psi_{T,s}} \|x_{t,i}\|_{(A_{t,s}^{i_t})^{-1}} \\ &\leq \alpha_s \sum_{t \in \Psi_{T,s}} \|x_{t,a}^{i_t}\|_{(A_{t,s}^{all})^{-1}} + \alpha_s dM \log(1 + T/d) \\ &\leq \tilde{O}(\sqrt{d \sum_{t \in \Psi_{T,s}} \sigma_t^2}).\end{aligned}$$

The first inequality arises from Lemma B.6, the second is a result of the arm selection rule in Line 13, the third derives from Lemma C.3, and the final inequality is attributable to Lemma E.1. □

For the final layer $S$, applying Lemma B.6 and setting $\overline{w}_S = d/T$, we have $\mathsf{reg}_{\text{layer } S} \leq 8\overline{w}_s |\Psi_S| \leq \tilde{O}(d)$.

**Proof of the communication bound in Theorem 7.1.** Having established the bound for regret in each layer, we have demonstrated that $R_T \leq \tilde{O}\left(\sqrt{d \sum_{t=1}^{T} \sigma_t^2}\right)$. Given that we set $\overline{w}_0 = dR^2$ and $\overline{w}_S = d/T$, it requires $S = \log(\overline{w}_0 / \overline{w}_S) = \Theta(\log R + \log T)$ layers to achieve the desired accuracy. The number of communications triggered by layer $s$ can be upper bounded by $O(dM^2 \log(T)$ (Lemma C.1). Consequently, we are able to constrain the overall communication cost to $\tilde{O}(dM^2 \log^2 T)$.

# F Corruption Robust Async-FedSupLinUCB

The corruption robust SupLinUCB subroutine is presented in Alg. 7, while the complete corruption robust Async-FedSupLinUCB is given in Alg. 8.

**Algorithm 7** Corruption Robust SupLinUCB subroutine: CS-LUCB

**Initialization**: $S = \lceil \log d \rceil$, $\overline{w}_0 = d^{1.5}/\sqrt{T}$, $\overline{w}_s \leftarrow 2^{-s}\overline{w}_0$, $\gamma = \sqrt{d}/C_p$.
$\alpha_0 = 1 + \sqrt{d\ln(2M^2T/\delta)} + \gamma C_p$, $\alpha_s \leftarrow 1 + \sqrt{2\ln(2KMT\ln d/\delta)} + \gamma C_p, \forall s \in [1:S]$
**Input**: Client $i$ (with local information $A^i, b^i, \Delta A^i, \Delta b^i$), contexts set $\{x^i_{t,1}, \ldots, x^i_{t,K}\}$
$A^i_{t,s} \leftarrow A^i_s, b^i_{t,s} \leftarrow b^i_s$ for `lazy update`
$\hat{\theta}_s \leftarrow (A^i_{t,s})^{-1} b^i_{t,s}$, $\hat{r}^i_{t,s,a} = \hat{\theta}_s^\top x^i_{t,a}$, $w^i_{t,s,a} \leftarrow \alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}, \forall s \in [0:S], \forall a \in [K]$.
$s \leftarrow 0$; $\mathcal{A}_0 \leftarrow \{a \in [K] \mid \hat{r}^i_{t,0,a} + w^i_{t,0,a} \geq \max_{a \in [K]}(\hat{r}^i_{t,0,a} - w^i_{t,0,a})\}$. ▷ Initial screening
**repeat** ▷ Layered successive screening
  **if** $s = S$ **then**
    Choose action $a^i_t$ arbitrarily from $\mathcal{A}_S$
  **else if** $w^i_{t,s,a} \leq \overline{w}_s$ for all $a \in \mathcal{A}_s$ **then**
    $\mathcal{A}_{s+1} \leftarrow \{a \in \mathcal{A}_s \mid \hat{r}^i_{t,s,a} \geq \max_{a' \in \mathcal{A}_s}(\hat{r}^i_{t,s,a'}) - 2\overline{w}_s\}$; $s \leftarrow s+1$
  **else**
    $a^i_t \leftarrow \arg\max_{\{a \in \mathcal{A}_s, w^i_{t,s,a} > \overline{w}_s\}} w^i_{t,s,a}$
  **end if**
**until** action $a^i_t$ is found
Take action $a^i_t$ and and receive reward $r^i_{t,a^i_t}$
$\eta_t = \min\{1, \gamma/\|x^i_{t,a_t}\|_{(A^i_{t,s})^{-1}}\}$
$\Delta A^i_s \leftarrow \Delta A^i_s + \eta_t x^i_{t,a^i_t} x^{i\top}_{t,a^i_t}$, $\Delta b^i_s \leftarrow \Delta b^i_s + \eta_t r^i_{t,a^i_t} x^i_{t,a^i_t}$ ▷ Update local information
**Return** layer index $s$

**Algorithm 8** Corruption Robust Async-`FedSupLinUCB`

**Initialization**: $T$, $C$, $S = \lceil \log d \rceil$
$\{A^{ser}_s \leftarrow I_d, b^{ser}_s \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization
$\{A^i_s \leftarrow I_d, \Delta A^i_s, b^i_s, \Delta b^i_s \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization
**for** $t = 1, 2, \cdots, T$ **do**
  Client $i_t = i$ is active, and observes $K$ contexts $\{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}$
  $s \leftarrow$ `CS-LUCB` $\big($client $i, \{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}\big)$ with lazy update
  **if** $\frac{\det(A^i_s + \Delta A^i_s)}{\det(A^i_s)} > (1+C)$ **then**
    Sync($s$, server, clients $i$) for each $s \in [0:S]$
  **end if**
**end for**

### F.1 Algorithm

### F.2 Supporting Lemmas and Proof

When confronted with adversarial corruption, we utilize a weighted ridge regression in which the weight assigned to each selected action depends on its confidence. Further, we expand the confidence width to accommodate this corruption, with $\alpha_0 = 1 + \sqrt{d\ln(2M^2T/\delta)} + \gamma C_p$ and $\alpha_s = 1 + \sqrt{2\ln(2KMT\ln d/\delta)} + \gamma C_p$ as proposed in He et al. (2022b). In our analysis of layer 0, we adapt Lemma B.1 from He et al. (2022b) to fit a federated scenario, yielding the following lemma:

**Lemma F.1.** *(Adapted from Lemma B.1 in He et al. (2022b)) Under the setting of Theorem 5.1, in the layer 0, with probability at least $1-\delta$, the following event $\mathcal{E}_0$ happens:*

$$\mathcal{E}_0 \triangleq \{\Big|x^{i\top}_{t,a}\hat{\theta}^i_{t,s} - x^{i\top}_{t,a}\theta\Big| \leq w^i_{t,s,a}, \forall i \in [M], a \in [K], t \in [T], s = 0\}.$$

For each layer $s \in [S]$, the rewards are mutually independent, analogous to the proof of Lemma B.1. We can restate the lemma as follows:

**Lemma F.2.** *Suppose the time index set $\Psi_{t,s}$ is constructed so that for fixed $x_{\tau,a_\tau}$ with $\tau \in \Psi_{t,s}$, the rewards $\{r_{\tau,a_\tau}\}$ are independent random variables with means $\mathbb{E}[r_{\tau,a_\tau}] = \theta^\top x_{\tau,a_\tau} + c_\tau$. For any round $t \in [T]$, if client $i_t = i$ is active and chooses arm $a_t$ in layer $s \in [S]$, with probability at least*

$1 - \frac{\delta}{MT \ln d}$, *we have for any* $a_t \in [K]$*:*

$$\left|\hat{r}_{t,s,a_t} - \theta^\top x^i_{t,a_t}\right| \leq w^i_{t,s,a_t} = \alpha_s \|x^i_{t,a_t}\|_{(A^i_{t,s})^{-1}}.$$

After combining the aforementioned events, we redefine the good event in the presence of corruption as follows:

$$\mathcal{E} \triangleq \left\{\left|x^{i\top}_{t,a}\hat{\theta}^i_{t,s} - x^{i\top}_{t,a}\theta\right| \leq w^i_{t,s,a}, \forall i \in [M], a \in [K], t \in [T], s \in [0:S]\right\}.$$

Similar to proof of Lemma B.4, we have that $\mathbb{P}[\mathcal{E}] \geq 1 - \delta$.

**Lemma F.3.** *Conditioned on the good event* $\mathcal{E}$*, the regret of layer* $s \in [0 : S-1]$ *can be bounded as follows:* reg*layers* $\leq \tilde{O}(\sqrt{dT} + dC_p)$.

*Proof of Lemma F.3.* Under the condition of the good event $\mathcal{E}$, we adopt a similar approach to the regret decomposition analysis presented in He et al. (2022b) to bound the regret in each layer $s \in [0 : S-1]$.

$$\mathbb{E}\sum_{t\in\Psi_{T,s}}(r^i_{t,a^{i,*}_t} - r^i_{t,a_t}) \leq \sum_{t\in\Psi_{T,s}} 8w_{t,s,a_t} = \sum_{t\in\Psi_{T,s}} 8\alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}} \tag{7}$$

$$= \underbrace{8\alpha_s \sum_{t\in\Psi_{T,s},\eta_t=1} \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}}_{I_1} + \underbrace{8\alpha_s \sum_{t\in\Psi_{T,s},\eta_t<1} \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}}_{I_2}. \tag{8}$$

The first inequality is derived from Lemma B.6, while Equation (8) follows from the definition of $\eta_t$. For the term $I_1$, we consider the rounds with $\eta_t = 1$, assuming these rounds can be listed as $\{k_1, k_2, ..., k_n\}$. To analyze this, we construct the auxiliary matrix $B_{t,s} = I + \sum_{j=1}^n x_{k_j} x_{k_j}^\top I\{k_j \leq t\}$. Using the definition of $A^i_{t,s}$, we can establish the inequality $A^i_{t,s} \succeq \frac{1}{1+MC} A^{all}_{t,s} \succeq \frac{1}{1+MC} B_{t,s}$.

Then we have

$$I_1 = \sum_{t\in\Psi_{T,s},\eta_t=1} 8\alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}$$

$$\leq 8\alpha_s \sqrt{2(1+MC)}\sqrt{2d|\Psi_{T,s}|\log|\Psi_{T,s}|} + 8\alpha_s dM\log(1+T/d) \leq \tilde{O}(\sqrt{dT}),$$

where the first inequality follows from Lemma C.3, and the second inequality is obtained by noting that the size of $\Psi_{T,0}$ is bounded by $\tilde{O}(T/d)$, as stated in Lemma C.4 particularly for layer 0.

For the term $I_2$, using the property $\eta_t < 1$, we can express $\eta_t$ as $\eta_t = \gamma/\|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}$, which implies:

$$I_2 = \sum_{t\in\Psi_{T,s},\eta_t<1} 8\alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}$$

$$\leq \sum_{t\in\Psi_{T,s},\eta_t<1} 8\frac{\alpha_s}{\gamma}\eta_t x^{i\top}_{t,a}(A^i_{t,s})^{-1}x^i_{t,a} \leq \frac{\alpha_s}{\gamma} d\log(T) \leq \tilde{O}(dC_p),$$

where the first inequality is derived from the definition of $\eta_t$, the second inequality is obtained from the elliptical potential lemma, as referenced in Lemma A.1, and the third inequality stems from the definition of $\alpha_s$.

By combining $I_1$ and $I_2$, we can ultimately bound the regret in each layer $s \in [0 : S-1]$ as $\mathsf{reg}_{\mathsf{layer}s} \leq \tilde{O}(\sqrt{dT} + dC_p)$. □

For the regret that occurs in the last layer $S$, we can derive the following bound:

$$\sum_{t\in\Psi_{T,S}} \mathbb{E}\left[r^i_{t,a^{i,*}_t} - r^i_{t,a_t}\right] \leq \sum_{t\in\Psi_{T,S}} 8\overline{w}_S \leq 8\overline{w}_S|\Psi_{T,S}| \leq 8\overline{w}_S T \leq 8\sqrt{dT}.$$

The first inequality is from Lemma B.6, and the last inequality follows from $\overline{w}_S = \sqrt{d/T}$.

**Proof of the communication bound in Theorem 7.2.** By combining the regret in each layer, we can conclude that $R_T \leq \tilde{O}(\sqrt{dT} + dC_p)$. Note that, based on the definition of $\eta_t \leq 1$ and Lemma A.1, it follows that $\log\big(\det\big(A_{t,s}^{all}\big)\big) \leq d\log(1 + |\Psi_{T,s}|/d)$. Additionally, by following a similar proof as in Lemma C.1, we can bound the number of communication rounds in layer $s$ by $O(dM^2 \log T)$. Considering that the FedSupLinUCB algorithm has $S = \lceil \log d \rceil$ layers, the total communication cost is therefore upper bounded by $O(dM^2 \log d \log T)$.