Differentially-Private Federated Linear Bandits

Abhimanyu Dubey and Alex Pentland
Media Lab and Institute for Data, Systems and Society
Massachusetts Institute of Technology
{dubeya, pentland}@mit.edu

Abstract

The rapid proliferation of decentralized learning systems mandates the need for differentially-private cooperative learning. In this paper, we study this in context of the contextual linear bandit: we consider a collection of agents cooperating to solve a common contextual bandit, while ensuring that their communication remains private. For this problem, we devise FEDUCB, a multiagent private algorithm for both centralized and decentralized (peer-to-peer) federated learning. We provide a rigorous technical analysis of its utility in terms of regret, improving several results in cooperative bandit learning, and provide rigorous privacy guarantees as well. Our algorithms provide competitive performance both in terms of pseudoregret bounds and empirical benchmark performance in various multi-agent settings.

1 Introduction

The multi-armed bandit is the classical sequential decision-making problem, involving an agent sequentially choosing actions to take in order to maximize a (stochastic) reward [45]. It embodies the central exploration-exploitation dilemma present in sequential decision-making. Practical applications of the multi-armed bandit range from recommender systems [52] and anomaly detection [11] to clinical trials [16] and finance [25]. Increasingly, however, such large-scale applications are becoming decentralized, as their data is often located with different entities, and involves cooperation between these entities to maximize performance [13, 23]. This paradigm is now known as federated learning.

The objective of the federated paradigm is to allow cooperative estimation with larger amounts of data (from multiple clients, devices, etc.) while keeping the data decentralized [28]. There has been a surge of interest in this problem from both academia and industry, owing to its overall applicability. Most research on provably private algorithms in the federated setting has been on distributed supervised learning [22] and optimization [21]. The contextual bandit problem, however, is a very interesting candidate for private methods, since the involved contexts and rewards both typically contain sensitive user information [39]. There is an increasing body of work on online learning and multi-armed bandits in cooperative settings [12, 32, 40], and private single-agent learning [42, 39], but methods for private federated bandit learning are still elusive, despite their immediate applicability.

Contributions. In this paper, we study the federated contextual bandit problem under constraints of differential privacy. We consider two popular paradigms: (a) centralized learning, where a central controller coordinates different clients [28, 50] and (b) decentralized peer-to-peer learning with delays, where agents communicate directly with each other, without a controller [13, 40, 32, 31, 53]. We provide a rigorous formulation of $(\varepsilon, \delta)$-differential privacy in the federated contextual bandit, and present two variants of FEDUCB, the first federated algorithm ensuring that each agent is private with respect to the data from all other agents, and provides a parameter to control communication.

1Originally, federated learning referred to the algorithm proposed in [29] for supervised learning, however, now the term broadly refers to the distributed cooperative learning setting [28].

Preprint. Under review.
Next, we prove rigorous bounds on the cumulative group pseudoregret obtained by \textsc{FedUCB}. In the centralized setting, we prove a high probability regret bound of $\tilde{O}(d^{3/4} \sqrt{MT/\varepsilon})$ which matches the non-private bound in terms of its dependence on $T$ and $M$, and is only a factor of $1/\sqrt{\varepsilon}$ away from the lower-bound for private bandit estimation obtained in [42]. In the decentralized case, we prove a corresponding regret bound of $\tilde{O}(d^{3/4} \sqrt{\text{diameter}(G)MT/\varepsilon})$, where $G$ is the communication network between agents. In addition to the regret bounds, we present a novel analysis of the privacy budget and regret.

2 Related Work

Multi-Agent and Distributed Bandits. Bandit learning in multi-agent distributed settings has received attention from several academic communities. Channel selection in distributed radio networks consider the (context-free) multi-armed bandit with collisions [36,38,57] and cooperative estimation over a network with delays [32,31,33]. For the contextual case, recent work has considered non-private estimation in networks with delays [14,12,49,30]. A closely-related problem is that of bandits with side information [3,4], where the single learner obtains multiple observations every round, similar to the multi-agent communicative setting. Our work builds on the remarkable work of Abbasi-Yadkori et al. [1], which in turn improves the LinUCB algorithm introduced in [34].

Differential Privacy. Our work utilizes differential privacy, a mathematically rigorous framework introduced by Dwork [17,19] that requires the behavior of an algorithm to fluctuate only slightly (in probability) with any change in its inputs. A technique to maintain differential privacy for the continual release of statistics was introduced in [10,18], known as the tree-based algorithm that privatizes the partial sums of $n$ entries by adding at most $\log n$ noisy terms. This method has been used to preserve privacy across several online learning problems, including convex optimization [27,26], online data analysis [24], collaborative filtering [6] and data aggregation [9]. In the single-agent bandit setting, differential privacy using tree-based algorithms have been explored in the multi-armed case [44,41,47] and the contextual case [42]. In particular, our work uses several elements from [42], extending their single-agent results to the federated multi-agent setting. For the multi-agent multi-armed (i.e., context-free) bandit problem, differentially private algorithms have been devised for the centralized [45] and decentralized [13] settings. Empirically, the advantages of privacy-preserving contextual bandits has been demonstrated in the work of Malekzadeh et al. [59], and Hannun et al. [22] consider a centralized multi-agent contextual bandit algorithm that use secure multi-party computations to provide privacy guarantees (both works do not have any regret guarantees).

To the best of our knowledge, this paper is the first to investigate differential privacy for contextual bandits in the federated learning setting, in both centralized and decentralized environments. Our work is most closely related to the important work of [1,42], extending it to federated learning.

3 Background and Preliminaries

We use boldface to represent vectors, e.g., $x$, and matrices, e.g., $X$. For any matrix $X$, its Gram matrix is given by $X^\top X$. Any symmetric matrix $X$ is positive semi-definite (PSD) if $\langle y^\top X y \rangle \geq 0$ for any vector $y$, and we denote PSD by $X \succeq 0$. For any PSD matrix $X$ we denote the ellipsoid $X$-norm of vector $y$ as $\|y\|_X = \sqrt{y^\top X y}$. For two matrices $X$ and $Y$, $X \succeq Y$ implies that $X - Y \succeq 0$. For any PSD matrix $X$ and vectors $u$, $v$, we denote the $X$-ellipsoid inner product as $\langle u, v \rangle_X = u^\top X v$, and drop the subscript when $X = I$. We denote the set $\{a, a + 1, \ldots, b - 1, b\}$ by the shorthand $[a,b]$, and if $a = 1$ we refer to it as $[b]$. We denote the $\gamma$-th power of graph $G$ as $G^\gamma$. ($G$, has edge $(i,j)$ if the shortest distance between $i$ and $j$ in $G$ is $\leq \gamma$). $N_{\gamma}(v)$ denotes the set of nodes in $G$ at a distance of at most $\gamma$ (including itself) from node $v \in V$.

Federated Contextual Bandit. This is an extension of the linear contextual bandit [34,1] involving a set of $M$ agents. At every trial $t \in [T]$, each agent $i \in [M]$ is presented with a decision set $\mathcal{D}_{i,t} \subset \mathbb{R}^d$ from which it selects an action $x_{i,t} \in \mathbb{R}^d$. It then obtains a reward $y_{i,t} = x_{i,t}^\top \theta^* + \eta_{i,t}$ where $\theta^* \in \mathbb{R}^d$ is an unknown (but fixed) parameter and $\eta_{i,t}$ is a noise parameter sampled i.i.d. every
trial for every agent. The objective of the agents is to minimize the cumulative group pseudoregret

\[ R_M(T) = \sum_{i=1}^{M} T \mathbb{E}[\hat{x}_{i,t} - x_{i,t}, \theta^*], \]

where \( x_{i,t} = \arg\max_{x} \mathbb{E}[D_{i,t}(x, \theta^*)] \) is optimal.

In the single-agent setting, the optimal regret is \( \tilde{O}(\sqrt{dT}) \), which has been matched by a variety of algorithms, most popular of them being the ones based on upper confidence bounds (UCB) \( [11] \)
\( [34] \). In non-private distributed contextual bandits, a group pseudoregret of \( \tilde{O}(\sqrt{dMT}) \) has been achieved \( [15] \)[49][50], which is the order-optimality we seek in our algorithms as well.

Assumptions. We assume: (a) bounded action set: \( \forall i, t, x_{i,t} \leq L \), (b) bounded mean reward: \( \forall x, \theta^*, x \leq 1 \), (c) bounded target parameter: \( \| \theta^* \| \leq S \), (d) sub-Gaussian rewards: \( \forall i, t, \eta_{i,t} \) is \( \sigma \)-sub-Gaussian, (e) bounded decision set: \( \forall i, t, d_{i,t} \) is compact, (f) bounded reward: \( \forall i, t, \eta_{i,t} \leq B \).

Differential Privacy. The contextual bandit problem involves two sets of variables that any agent must private to the other participating agents – the available decision set \( D_{i,t} \) and observed rewards \( (\eta_{i,t})_{t \in [T]} \). However, the communication between agents only depends on the chosen actions \( (x_{i,t})_{t \in [T]} \) (and not all of \( D_{i,t} \)), and hence making our technique differentially private with respect to \( (x_{i,t}, \eta_{i,t})_{t \in [T]} \) will suffice. We first denote two sequences \( S_i = ((x_{i,t}, \eta_{i,t}))_{t \in [T]} \) and \( S'_i = ((x_{i,t}', \eta_{i,t}'))_{t \in [T]} \) as \( t \)-neighbors if for each \( t' \neq t \), \( (x_{i,t'}, \eta_{i,t'}) = (x_{i,t}', \eta_{i,t}') \). We can now provide the formal definition for federated differential privacy:

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

\[ \mathbb{P}(A_j(S_i) \in S_{j,t}) \leq e^\varepsilon \cdot \mathbb{P}(A_j(S'_i) \in S_{j,t}) + \delta. \]

Our notion of federated differential privacy is formalizing the standard intuition that “the action chosen by any agent must be sufficiently impervious (in probability) to any single \((x, y)\) pair from any other agent”. The same guarantee can be obtained if each agent \( A_i \) is \((\varepsilon, \delta)\)-differentially private (in the standard contextual bandit sense, see \( [11] \)[34] with respect to any other agent \( j \)’s observations, for all \( j \). An \((M-1)\)-fold composition argument \( [17] \) over all \((M-1)\) agents would therefore provide us \((\sqrt{2(M-1)} \log(1/\delta')\varepsilon + (M-1)\varepsilon(\varepsilon - 1), (M-1)\delta + \delta')\)-differential privacy with respect to the overall sequence. To keep the notation simple, however, we adopt the \((\varepsilon, \delta, M)\) format.

4 Federated LinUCB with Differential Privacy

In this section, we introduce our algorithm for federated learning with differential privacy. For the remainder of this section, for exposition, we consider the single-agent setting and drop an additional index subscript we use in the actual algorithm (e.g., refer to the action at time \( t \) as \( x_t \) and not \( x_{i,t} \) for agent \( i \)). We build on the celebrated LinUCB algorithm, an application of the optimism heuristic to the linear bandit case \( [34] \)[11], designed for the single-agent problem. The central idea of the algorithm is, at every round \( t \), to construct a confidence set \( \mathcal{E}_t \) that contains \( \theta^* \) with high probability, followed by computing an upper confidence bound on the reward of each action within the decision set \( D_t \), and finally selecting the action with the largest UCB, i.e.,

\[ x_t = \arg\max_{x} \mathbb{E}[D_{i,t}(x, \theta)]. \]

The confidence set is an ellipsoid centered on the regularized linear regression estimate (for \( X_{<t} = [x_1^T \ x_2^T \ ... \ x_{t-1}^T]^T \) and \( y_{<t} = [y_1 \ y_2 \ ... \ y_{t-1}]^T \)):

\[ \mathcal{E}_t := \left\{ \theta \in \mathbb{R}^d : \| \theta - \hat{\theta}_t \|_{V_t} \leq \beta_t \right\}, \]

where \( \hat{\theta}_t := \min_{\theta} \mathbb{E}[\| X_{<t} \theta - y_{<t} \|_2^2 + \| \theta \|_{H_t}^2] \).

The regression solution can be given by \( \hat{\theta}_t := (G_t + H_t)^{-1} X_{<t}^T y_{<t} \), where \( G_t = X_{<t}^T X_{<t} \) is the Gram matrix of actions, \( H_t \) is a (time-varying) regularizer, and \( \beta_t \) is an appropriately chosen

---

2The pseudoregret is an expectation (over the randomness of \( \eta_{i,t} \)) of the stochastic quantity regret, and is more amenable to high-probability bounds. However, a bound over the pseudoregret can also bound the regret with high probability, e.g., by a Hoeffding concentration (see, e.g., \( [43] \).[16])

3Assuming bounded rewards is necessary for the privacy mechanism, and is standard in the literature \( [3] \)[22].

4Regarding joint differential privacy: since there is always a delay of at least 1 trial in communication, the federated setting itself guarantees that any \( A_j \) that is \((\varepsilon, \delta)\)-jointly differentially private with respect to sequences \( (S_i, S'_i) \), \( t \neq j \) will also be \((\varepsilon, \delta, M)\)-federated differentially private with respect to the sequences.
We first consider the centralized communication environment where there exists a controller that
Algorithm 1
\textbf{Algorithm 1 CENTRALIZED FedUCB}($D, M, T, \rho_{\text{min}}, \rho_{\text{max}}$)

1: \textbf{Initialization:} \forall i, set $S_{i, 1} \leftarrow 0$, $s_{i, 1} \leftarrow 0$, $\hat{Q}_{i, 0} \leftarrow 0$, $U_{i, 1} \leftarrow \rho_{\text{min}}I$, $\hat{u}_{i, 1} \leftarrow 0$.
2: \textbf{for} Each iteration $t \in [T]$ \textbf{do}
3: \hspace{0.5cm} \textbf{for} Each agent $i \in [M]$ \textbf{do}
4: \hspace{1cm} Set $V_{i, t} \leftarrow S_{i, t} + U_{i, t}$, $\hat{u}_{i, t} \leftarrow s_{i, t} + \hat{u}_{i, t}$.
5: \hspace{1cm} Receive $\mathcal{D}_{i, t}$ from environment.
6: \hspace{1cm} Compute regressor $\hat{\theta}_{i, t} \leftarrow V_{i, t}^{-1} \hat{u}_{i, t}$.
7: \hspace{1cm} Compute $\beta_{i, t}$ following Proposition 2.
8: \hspace{1cm} Select $x_{i, t} \leftarrow \arg\max_{x \in \mathcal{D}_{i, t}} (x, \hat{\theta}_{i, t}) + \beta_{i, t} \|x\|_{V_{i, t}^{-1}}$.
9: \hspace{1cm} Obtain $y_{i, t}$ from environment.
10: \hspace{1cm} Update $U_{i, t+1} \leftarrow U_{i, t} + x_{i, t} x_{i, t}^\top$, $s_{i, t+1} \leftarrow u_{i, t} + x_{i, t} y_{i, t}$.
11: \hspace{1cm} Update $\hat{Q}_{i, t} \leftarrow \hat{Q}_{i, t-1} + [x_{i, t} y_{i, t}]^\top [x_{i, t} y_{i, t}]$.
12: \hspace{1cm} if $\log \det (V_{i, t} + x_{i, t} x_{i, t}^\top + M(\rho_{\text{max}} - \rho_{\text{min}})I) - \log \det (S_{i, t}) \geq \frac{D}{\Delta t_i}$, then
13: \hspace{1cm} \hspace{1cm} \textbf{end if}
14: \hspace{1cm} \textbf{if} SYNCHRONIZE \textbf{then}
15: \hspace{1.5cm} \textbf{[\textsc{Agents]} Agent sends $Q_{i, t} \rightarrow$ PRIVATIZER and gets $\hat{U}_{i, t+1}$, $\hat{u}_{i, t+1} \leftarrow$ PRIVATIZER.}
16: \hspace{1.5cm} \textbf{[\textsc{Agents]} Agent communicates $\hat{U}_{i, t+1}$, $\hat{u}_{i, t+1}$ to controller.}
17: \hspace{1.5cm} \textbf{[\textsc{Controller]} Compute $S_{t+1} \leftarrow S_{t} + \sum_{t=1}^{\hat{U}_{i, t+1}} S_{t, i, t} \leftarrow S_{t} + \sum_{t=1}^{\hat{U}_{i, t+1}} S_{i, t}$.}
18: \hspace{1.5cm} \textbf{[\textsc{Agent]} Communication $S_{t+1}$, $S_{i, t}$ back to agent.}
19: \hspace{1.5cm} \textbf{[\textsc{Agents]} $S_{i, t+1} \leftarrow S_{t+1}$, $s_{i, t+1} \leftarrow s_{t+1}, \Delta t_i \leftarrow 0$, $U_{i, t+1} \leftarrow \rho_{\text{min}}I$, $\hat{u}_{i, t+1} \leftarrow 0$.}
20: \hspace{1.5cm} \textbf{[\textsc{Agents]} $Q_{i, t+1} \leftarrow 0$.}
21: \hspace{1cm} \textbf{end if}
22: \hspace{1cm} \textbf{else}
23: \hspace{1.5cm} $S_{i, t+1} \leftarrow S_{i, t}$, $s_{i, t+1} \leftarrow s_{i, t}, \Delta t_i \leftarrow \Delta t_i + 1$.
24: \hspace{1cm} \textbf{end if}
25: \hspace{0.5cm} \textbf{end for}
26: \hspace{0.5cm} \textbf{end for}

confidence parameter. Typically in non-private settings, the regularizer is constant, i.e., $H_t = \lambda \mathcal{I}$ \forall $t$, $\lambda > 0$ \cite{34}, however, in our case, we will carefully select $H_t$ to introduce privacy, using a strategy similar to \cite{42}. Given $V_t = G_t + H_t$, let UCB$_t(x; \theta) = (\theta, x) + \beta_t \|x\|_{V_{t-1}^{-1}}$.

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

4.1 Centralized Environment with a Controller

We first consider the centralized communication environment where there exists a controller that coordinates communication between different agents, as is typical in large-scale distributed learning \cite{1]. We consider a set of $M$ agents that each are interacting with the contextual bandit, and periodically communicate with the controller, that synchronizes them with other agents. We present the algorithm CENTRALIZED FedUCB in Algorithm 1 that details our approach.

The central idea of the algorithm is to carefully perturb the Gram matrices $V_{i, t}$ and the reward vector $u_{i, t}$ with random noise based on the sensitivity of these elements and the level of privacy required, as is typical in the literature \cite{42}. First, each agent updates its local (unsynchronized) estimates. These are used to construct the UCB in a manner identical to the standard OFUL algorithm \cite{1}. If, for any agent $i$, the log-determinant of the local Gram matrix exceeds the synchronized Gram matrix $(S_{i, t})$ by an amount $D/\Delta t_i$ (where $\Delta t_i$ is the time since the last synchronization), then it sends a signal to the controller, that synchronizes all agents with their latest action/reward pairs. The synchronization is done using a privatized version of the Gram matrix and rewards, carried out by the subroutine PRIVATIZER (Section 4.3, Alg. 2). This synchronization ensures that the heterogeneity between the agents is controlled, allowing us to control the overall regret and limit communication as well:

Proposition 1 (Communication Complexity). \textit{If Algorithm 1 is run with threshold $D$, then total rounds of communication $n$ is upper bounded by $2\sqrt{(4T/D) \cdot \log (\rho_{\text{max}}/\rho_{\text{min}} + TL^2/4\rho_{\text{min}})} + 4$.}
The PRIVATIZER subroutine provides suitable perturbations based on the privacy budget ($\varepsilon$ and $\delta$), and is inspired by the single-agent private algorithm of [42]. In this paper, we assume these to be identical for all agents (however, the algorithm and analysis hold for unique privacy budgets as well, as long as a lower bound on the budgets is known). In turn, the quantities $\varepsilon$ and $\delta$ affect the algorithm (and regret) via the quantities $\rho_{\text{min}}, \rho_{\text{max}}$, and $\kappa$ which bound the norm of perturbations at any round:

**Definition 2 (Sparsely-accurate $\rho_{\text{min}}, \rho_{\text{max}}$ and $\kappa$).** Consider a subsequence $\sigma$ of $[T] = 1, \ldots, T$ of size $n$. The bounds $0 \leq \rho_{\text{min}} \leq \rho_{\text{max}}$ and $\kappa$ are $(\alpha/2nM, \sigma)$-accurate for $(H_{i,t})_{i \in [M], t \in \sigma}$ and $(h_{i,t})_{i \in [M], t \in \sigma}$ if, for each round $t \in \sigma$ and agent $i$:

$$
\|H_{i,t}\| \leq \rho_{\text{max}}, \quad \|H_{i,t}^{-1}\| \leq 1/\rho_{\text{min}}, \quad \|h_{i,t}\|_{H_{i,t}^{-1}} \leq \kappa; \text{ with probability at least } (1 - \alpha/2nM).
$$

The motivation for obtaining accurate bounds $\rho_{\text{min}}, \rho_{\text{max}}$ and $\kappa$ stems from the fact that in the non-private case, the quantities that determine regret are not stochastic conditioned on the obtained sequence $(x_t, y_t)_{t \in [T]}$, whereas the addition of stochastic regularizers in the private case requires us to have control over their perturbations (as described above). To form the UCB, we additionally require a suitable confidence threshold $\beta_{i,t}$ for each agent, which is defined as follows:

**Definition 3 (Accurate $(\beta_{i,t})_{i \in [M], t \in [T]}$).** A sequence $(\beta_{i,t})_{i \in [M], t \in [T]}$ is $(\alpha, M, T)$-accurate for $(H_{i,t})_{i \in [M], t \in [T]}$ and $(h_{i,t})_{i \in [M], t \in [T]}$, if it satisfies $|\hat{\theta}_{i,t} - \theta^*|_{V_{i,t}} \leq \beta_{i,t}$ with probability at least $1 - \alpha$ for all rounds $t = 1, \ldots, T$ and agents $i = 1, \ldots, M$ simultaneously.

**Proposition 2.** Consider an instance of the problem where synchronization occurs exactly $n$ times on instances $\sigma$, up to and including $T$ trials, and $\rho_{\text{min}}, \rho_{\text{max}}$ and $\kappa$ are $(\alpha/2nM)$-accurate. Then, for Algorithm 1 the sequence $(\beta_{i,t})_{i \in [M], t \in [T]}$ is $(\alpha, M, T)$-accurate where:

$$
\beta_{i,t} := \sigma \sqrt{2 \log(2/\alpha) + d \log(\det(V_{i,t}))} - d \log(M \rho_{\text{min}}) + M S \sqrt{\rho_{\text{max}} + M \gamma}.
$$

The key point here is that for the desired levels of privacy ($\varepsilon$, $\delta$) and synchronization rounds ($n$), we can calculate appropriate $\rho_{\text{min}}, \rho_{\text{max}}$ and $\kappa$, which in turn provide us a UCB algorithm with guarantees. Our basic techniques are similar to [1, 42], however, the multi-agent setting introduces several new aspects in the analysis, such as the control of heterogeneity.

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

$$
\mathcal{R}_M(T) = O \left( \sigma \sqrt{M T d} \left( \log \left( \frac{\rho_{\text{max}}}{\rho_{\text{min}}} + TL^2/d \rho_{\text{min}} \right) + \frac{\sqrt{\log 2/\alpha}}{\alpha} + M S \sqrt{\rho_{\text{max}} + M \kappa} \right) \right).
$$

Furthermore, if the mean reward of the best arm and second best arm are separated by a gap $\geq \Delta$ for all $i \in [M], t \in [T]$, then Alg. 1 obtains the following gap-dependent pseudoregret w.p. $\geq (1 - \alpha)$:

$$
\mathcal{R}_M(T) = O \left( (\sigma d/\Delta) \left( \log \left( \frac{\rho_{\text{max}}}{\rho_{\text{min}}} + TL^2/d \rho_{\text{min}} \right) + \sqrt{\log 2/\alpha} \right)^{3/2} \right).
$$

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

**Remark 1 (Communication Complexity and Regret).** Theorem 1 assumes that communication is essentially $O(M \log T)$. This rate (and corresponding $D$) is chosen to provide a balance between privacy and utility, and in fact, can be altered depending on the application. In the Appendix, we demonstrate that when we allow $O(MT)$ communication (i.e., synchronize every round), the regret can be improved by a factor of $O(\sqrt{\log(T)})$ (in both the gap-dependent and independent bounds), to match the single-agent $\tilde{O}(\sqrt{dMT})$ rate. Similarly, following the reasoning in [49], we can show that with $O(M^{-1.5}d^3)$ communication (i.e., independent of $T$), we incur regret of $O(\sqrt{dMT \log^2(MT)})$.

Theorem 1 demonstrates the relationship between communication complexity (i.e., number of synchronization rounds) and the regret bound for a fixed privacy budget, via the dependence on the bounds $\rho_{\text{min}}, \rho_{\text{max}}$ and $\kappa$. We now present similar results (for a fixed privacy budget) on group regret for the decentralized setting, which is more involved, as the delays and lack of a centralized controller make it difficult to control the heterogeneity of information between agents. Subsequently, in the next section, we will present results on the privacy guarantees of our algorithms.
4.2 Decentralized Peer-to-Peer Environment

In this environment, we assume that the collection of agents communicate by directly sending messages to each other. The communication network is denoted by an undirected (connected) graph $G = (V, E)$, where edge $e_{ij} \in E$ if agents $i$ and $j$ can communicate directly. The protocol operates as follows: every trial, each agent interacts with their respective bandit, and obtains a reward. At any trial $t$, after receiving the reward, each agent $v$ sends the message $m_{v,t}$ to all its neighbors in $G$.

This message is forwarded from agent to agent $\gamma$ times (taking one trial of the bandit problem each between forwards), after which it is dropped. This communication protocol, based on the time-to-live (delay) parameter $\gamma \leq \text{diam}(G)$ is a common technique to control communication complexity, known as the LOCAL protocol [20][35][43]. Each agent $v \in V$ therefore also receives messages $m_{v',t-d(v,v')}$ from all the agents $v'$ such that $d(v, v') \leq \gamma$, i.e., from all agents in $N_\gamma(v)$.

There are several differences in this setting compared to the previous one: first, since agents receive messages from different other agents based on their position in $G$, they generally have heterogeneous information throughout (no global sync). Second, information does not flow instantaneously through $G$, and messages can take up to $\gamma$ rounds to be communicated (delay). This requires (mainly technical) changes to the centralized algorithm in order to control the regret. We present the pseudocode our algorithm DECENTRALIZED FEDUCB in the Appendix, but highlight the major differences from the centralized version as follows. The first key algorithmic change from the earlier variant is the idea of subsampling, inspired by the work of Weinberger and Ordentlich [51]. Each agent $i$ maintains $\gamma$ total estimators $\hat{\theta}_{i,g}, g = 1, \ldots, \gamma$, and uses each estimator in a round-robin fashion, i.e., at $t = 1$ each agent uses $\hat{\theta}_{1,1}$, and at $t = 2$, each agent uses $\hat{\theta}_{1,2}$, and so on. These estimators (and their associated $V_{i,g}, \mathbf{u}_{i,g}$) are updated in a manner similar to Alg. [1] if the log det of the $\gamma$th Gram matrix exceeds a threshold $D/(\Delta_{i,g}+1)$, then the agent broadcasts a request to synchronize. Each agent $j$ within the $\gamma$-clique of $i$ that receives this request broadcasts its own $V_{j,g}, \mathbf{u}_{j,g}$. Therefore, each $V_{i,g}, \mathbf{u}_{i,g}$ is updated with action/reward pairs from only the trials they were employed in (across all agents). This ensures that if a signal to synchronize the $\gamma$th set of parameters has been broadcast by an agent $i$, all agents within the $\gamma$-clique of $i$ will have synchronized their $\gamma$th parameters by the next 2 rounds they will be used again (i.e., $2\gamma$ trials later). We defer most of the intermediary results to the Appendix for brevity, and present the primary regret bound (in terms of privacy parameters $\rho_{\min}, \rho_{\max}$ and $\kappa$):

**Theorem 2** (Decentralized Private Group Regret). Assuming Proposition [2] holds, and synchronization occurs in at least $n = \Omega\left(\log(\tilde{\chi}(G))\gamma(1 + L^2)^{-1} \log\left(\frac{\rho_{\max}}{\rho_{\min}} + TL^2/d\rho_{\min}\right)\right)$ rounds, decentralized FEDUCB obtains the following group pseudoregret with probability at least $1 - \alpha$:

$$R_M(T) = O\left(\sqrt{M(\tilde{\chi}(G)\gamma)Td}\left(\log\frac{\rho_{\max}}{\rho_{\min}} + \frac{TL^2}{\gamma d\rho_{\min}}\right) + \sqrt{\log\frac{n}{\alpha}} + M\sqrt{\rho_{\max} + M\kappa}\right).$$

**Remark 2** (Decentralized Group Regret). Decentralized FEDUCB obtains an identical dependence on the privacy bounds $\rho_{\min}, \rho_{\max}$ and $\kappa$ and horizon $T$ as Algorithm [7] since the underlying bandit subroutines are identical for both. The key difference is in additional the leading factor of $\sqrt{\tilde{\chi}(G)\gamma}$, which arises from the delayed spread of information: if $G$ is dense, e.g., complete, then $\gamma = 1$ and $\tilde{\chi}(G) = 1$, since there is only one clique of $G$. In the worst case, if $G$ is a line graph, then $\tilde{\chi}(G) = M/\gamma$, giving an additional factor of $M$ (i.e., it is as good as each agent acting individually). In more practical scenarios, we expect $G$ to be hierarchical, and expect a delay overhead of $O(1)$. Note that since each agent only communicates within its clique, the factor $\sqrt{\gamma}$ is optimal, see [14].

4.3 Privacy Guarantees

We now discuss the privacy guarantees for both algorithms. Here we present results for the centralized algorithm, but our results hold (almost identically) for the decentralized case as well (see appendix).

Note that each agent interacts with data from other agents only via the cumulative parameters $\mathbf{s}_t$ and $\mathbf{S}_t$. These, in turn, depend on $\mathbf{Z}_{i,t} = U_{i,t} + H_{i,t}$ and $\mathbf{z}_{i,t} = \mathbf{u}_{i,t} + \mathbf{h}_{i,t}$ for each agent $i$, on the instances $t$ that synchronization occurs.

**Proposition 3** (see [17][42]). Consider $n \leq T$ synchronization rounds occurring on trials $\bar{\sigma} \subseteq [T]$. If the sequence $\left(\mathbf{Z}_{i,t}, \mathbf{z}_{i,t}\right)_{t \in \bar{\sigma}}$ is $(\varepsilon, \delta)$-differentially private with respect to $\left(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}\right)_{t \in \bar{\sigma}}$, for each agent $i \in [M]$, then all agents are $(\varepsilon, \delta, M)$-federated differentially private.
Algorithm 2 PRIVATIZER($\varepsilon, \delta, M, T$) for any agent $i$

1: **Initialization:**
2: If communication rounds $n$ are fixed a priori, set $m \leftarrow 1 + \left\lceil \log n \right\rceil$, else $m \leftarrow 1 + \left\lceil \log T \right\rceil$.
3: Create binary tree $T$ of depth $m$.
4: for node $n$ in $T$ do
5: Sample noise $\tilde{N} \in \mathbb{R}^{(d+1) \times (d+1)}$, where $\tilde{N}_{ij} \sim \mathcal{N}(0, 16m(L^2 + 1)^2 \log(2/\delta)/\varepsilon^2)$.
6: Store $N = (\tilde{N} + \tilde{N}^\top)/\sqrt{2}$ at node $n$.
7: **end for**
8: **Runtime:**
9: for each communication round $t \leq n$ do
10: Receive $Q_{i,t}$ from agent, and update corresponding nodes in $T$ (see [27], Alg. 5).
11: Compute $M_{i,t+1}$ using the least nodes of $T$ (see [27], Alg. 5).
12: Set $\hat{U}_{i,t+1} = U_{i,t+1} + H_{i,t}$ as top-left $d \times d$ submatrix of $M_{i,t+1}$.
13: Set $\hat{u}_{i,t+1} = u_{i,t+1} + h_{i,t}$ as first $d$ entries of last column of $M_{i,t+1}$.
14: Return $\hat{U}_{i,t+1}$, $\hat{u}_{i,t+1}$ to agent.
15: **end for**

The implementation of the private release of $(U_{i,t}, \hat{u}_{i,t})$ is done by the ubiquitous tree-based mechanism for partial sums. Following [42], we also aggregate both into a single matrix $M_{i,t} \in \mathbb{R}^{(d+1) \times (d+1)}$, by first concatenating: $A_{i,t} := [X_{i,t}, y_{i,t}] \in \mathbb{R}^{t \times (d+1)}$, and then computing $M_{i,t} = A_{i,t}^\top A_{i,t}$. Furthermore, the update is straightforward: $M_{i,t+1} = M_{i,t} + [x_{i,t} y_{i,t}]^\top [x_{i,t} y_{i,t}]$.

The tree mechanism requires at most $m = 1 + \left\lceil \log_2 T \right\rceil$ counters, therefore noise that ensures each node guarantees $(\varepsilon/\sqrt{8m \ln(2/\delta)}, \delta/2m)$-privacy is sufficient to make the sequence $(\varepsilon, \delta)$-private.

Tree-Based Mechanism. The tree-based mechanism [18] for differential privacy maintains a binary tree $T$, where the leaf nodes are the variables needed to be summed (i.e., in our case $(M_{i,t})_i$), and each parent node maintains the (matrix) sum of its children. To make the partial sums private, a noise matrix is also added to each node in $T$. We utilize additive Gaussian noise: at each node, we sample $\tilde{N} \in \mathbb{R}^{(d+1) \times (d+1)}$, where each $\tilde{N}_{ij} \sim \mathcal{N}(0, \sigma_N^2)$, and $\sigma_N^2 = 16m(L^2 + 1)^2 \log(2/\delta)/\varepsilon^2$, and symmetrize it (see step 6 of Alg. 2). The total noise $H_{i,t}$ is the sum of at most $m$ such terms, hence the variance of each element in $H_{i,t}$ is $\leq m\sigma_N^2$. We can bound the operator norm of the top-left $(d \times d)$-submatrix of each noise term. With probability at least $1 - \alpha/nM$:

$$\|H_{i,t}\|_{op} \leq \Lambda = \sqrt{32m(L^2 + 1) \log(4/\delta)(4\sqrt{d} + 2 \ln(2m\alpha)/\varepsilon)}.$$

However, as noted by [42], this $H_{i,t}$ would not always be PSD. To ensure that it is always PSD, we can shift each $H_{i,t}$ by $2\Lambda I$, giving a bound on $\rho_{max}$. Similarly, we can obtain bounds on $\rho_{min}$ and $\kappa$:

**Proposition 4.** Fix $\alpha > 0$. If each agent $i$ samples noise parameters $H_{i,t}$ and $h_{i,t}$ using the tree-based Gaussian mechanism mentioned above for all $n$ trials of $\tilde{\sigma}$ in which communication occurs, then the following $\rho_{min}$, $\rho_{max}$ and $\kappa$ are $(\alpha/2nM, \tilde{\sigma})$-accurate bounds:

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

**Remark 3 (Strategyproof Analysis).** The mechanism presented above assumes the worst-case communication, i.e., synchronization occurs every round, therefore at most $T$ partial sums will be released, and $m = 1 + \lceil \log T \rceil$. This is not true for general settings, where infrequent communication would typically require $m = O(\log \log T)$. However, if any agent is byzantine and requests a synchronization every trial, $m$ must be $1 + \lceil \log T \rceil$ to ensure privacy. In case the protocol is fixed in advance (i.e., synchronization occurs on a pre-determined set $\tilde{\sigma}$ of $n$ rounds), then we can set $m = 1 + \lceil \log n \rceil$ to achieve the maximum utility at the desired privacy budget.

**Remark 4 (Decentralized Protocol).** Decentralized FEDUCB obtains similar bounds for $\rho_{min}, \rho_{max}$ and $\kappa$, with $m = 1 + \lceil \log(T/\gamma) \rceil$. An additional term of $\log(\gamma)$ appears in $\Lambda$ and $\kappa$, since we need to maintain $\gamma$ partial sums with at most $T/\gamma$ elements in the worst-case (see Appendix). Unsurprisingly, there is no dependence on the network $G$, as privatization is done at the source itself.

**Corollary 1 ((\varepsilon, \delta)-dependent Regret).** FEDUCB with the PRIVATIZER subroutine in Alg. 2 obtains $\tilde{O}(d^{3/4} \sqrt{MT/\varepsilon})$ centralized regret and $\tilde{O}(d^{3/4} \sqrt{(\log\gamma \cdot \gamma)MT/\varepsilon})$ decentralized regret.
Figure 1: An experimental comparison of centralized FEDUCB on 3 different axes. Fig. (A) describes the variation in asymptotic per-agent regret for varying privacy budget $\rho_{\text{min}}$; (B) describes the effect of $n$ in private (solid) vs. non-private (dashed) settings; (C) describes the effect of $d$ in per-agent regret in the private setting ($n = O(M \log T), \rho_{\text{min}} = 10^4$). Experiments averaged over 100 runs.

4.4 Experiments

We provide an abridged summary of our experiments (ref. Appendix for all experiments). Here, we focus on the centralized environment, and on the variation of the regret with communication complexity and privacy budget. For all experiments, we assume $L = S = 1$. For any $d$, we randomly fix $\theta^* \in B_d(1)$. Each $D_{i,t}$ is generated as follows: we randomly sample $K \leq d^2$ actions $x$, such that for $K - 1$ actions $0.5 \leq \langle x, \theta^* \rangle \leq 0.6$ and for the optimal $x^*, 0.7 \leq \langle x^*, \theta^* \rangle \leq 0.8$ such that $\Delta \geq 0.1$ always. $y_{i,t}$ is sampled from $2 \cdot \text{Beta}(\langle x_{i,t}, \theta^* \rangle, 1 - \langle x_{i,t}, \theta^* \rangle) - 1$ such that $\mathbb{E}[y_{i,t}] = \langle x_{i,t}, \theta^* \rangle$ and $|y_{i,t}| \leq 1$. Results are in Fig. 1 and experiments are averaged on 100 trials.

**Experiment 1: Privacy Budget.** In this setting, we set $n = O(M \log T), d = 10$ (to balance communication and performance), and plot the average per-agent regret after $T = 10^7$ trials (since the regret curves are essentially flat after that) for varying $M$ and $\rho_{\text{min}}$. Figure 1A describes the results, competitive even at large privacy budget.

**Experiment 2: Communication.** To test the communication-privacy trade-off, we plot the regret curves for $M = 100$ agents on $n = O(M^{1.5}), O(M \log T), O(MT)$ communication for both private ($\rho_{\text{min}} = 10^4$) and non-private settings. We observe a tradeoff as highlighted in Remark 1.

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

5 Discussion and Conclusion

The relentless permeance of intelligent decision-making on sensitive user data necessitates the development of technically sound and robust machine learning systems; it is difficult to stress enough the importance of protecting user data in modern times. There is a significant effort in creating decentralized decision-making systems that benefit from pooling data from multiple sources while maintaining user-level privacy. This research provides an instance of such a system, along with guarantees on both its utility and privacy. From a technical perspective, we make improvements along several fronts – while there has been prior work on multi-agent private linear bandits, our work is the first to provide rigorous guarantees on private linear bandits in the multi-agent setting. Our work additionally provides the first algorithm with regret guarantees for contextual bandits in decentralized networks, extending the work of many on multi-armed bandits.

There are several unresolved questions in this line of work, as highlighted by our algorithm itself. In the decentralized case, our algorithm obtains a communication overhead of $O(\sqrt{\Delta(G_{\gamma} \cdot \gamma)})$, which we comment is an artefact of our proof technique. We conjecture that the true dependence scales as $O(\sqrt{\alpha(G_{\gamma}) + \gamma})$ (where $\alpha(\cdot)$ is the independence number of the graph), that would be obtained by an algorithm that bypasses subsampling, with a more sophisticated synchronization technique. Additionally, the lower bound obtained for private contextual bandit estimation has an $O(\varepsilon^{-1})$ dependence on the privacy budget. In the federated setting, given the intricate relationship between communication and privacy, an interesting line of inquiry would be to understand the communication-privacy tradeoff in more detail, with appropriate lower bounds for fixed communication budgets.
**Broader Impact Statement**

We envision a world that runs on data-dependent inference. Given the ease of availability of sensitive data, large-scale centralized data sources are not feasible to uphold the best interests and safety of the average user, and decentralized inference mechanisms are a feasible alternative balancing utility with important normative values as privacy, transparency and accountability. This research is a part of such a broader research goal, to create decentralized systems that maintain the security of the end-user in mind. We believe that methods like ours are a starting point for better private algorithms that can be readily deployed in industry, with appropriate guarantees as well. While there are several subsequent steps that can be improved both theoretically and practically, the design for this research from the ground-up has been to ensure robustness and security (sometimes) in lieu of efficiency.

**References**


