Differentially Private Linear Bandits with Partial Distributed Feedback

Fengjiao Li, Xingyu Zhou, and Bo Ji

***Abstract*—In this paper, we study the problem of global reward maximization with only partial distributed feedback. This problem is motivated by several real-world applications (e.g., cellular network configuration, dynamic pricing, and policy selection) where an action taken by a central entity influences a large population that contributes to the global reward. However, collecting such reward feedback from the entire population not only incurs a prohibitively high cost, but often leads to privacy concerns. To tackle this problem, we consider differentially private distributed linear bandits, where only a subset of users from the population are selected (called clients) to participate in the learning process and the central server learns the global model from such partial feedback by iteratively aggregating these clients' local feedback in a differentially private fashion. We then propose a unified algorithmic learning framework, called differentially private distributed phased elimination (DP-DPE), which can be naturally integrated with popular differential privacy (DP) models (including central DP, local DP, and shuffle DP). Furthermore, we prove that DP-DPE achieves both sublinear regret and sublinear communication cost. Interestingly, DP-DPE also achieves privacy protection "for free" in the sense that the additional cost due to privacy guarantees is a lower-order additive term. Finally, we conduct simulations to corroborate our theoretical results and demonstrate the effectiveness of DP-DPE.**

I. Introduction

The bandit learning models have been widely adopted for many sequential decision-making problems, such as clinical trials, recommender systems, and configuration selection. Each action (called arm), if selected in a round, generates a (noisy) reward. By observing such reward feedback, the learning agent gradually learns the unknown parameters of the model (e.g., mean rewards) and decides the action in the next round. The objective here is to maximize the cumulative reward over a finite time horizon, balancing the tradeoff between *exploitation* and *exploration*. While the stochastic multi-armed bandits (MAB) model is useful for these applications [1], one key limitation is that actions are assumed to be independent, which, however, is usually not the case in practice. Therefore, the linear bandit model that captures the correlation among actions has been extensively studied [2]–[4].

In this paper, we introduce a new linear bandit setting where the reward of an action could be from a large population. Take the cellular network configuration as an example (see Fig. 1). The configuration (antenna tilt, maximum output power, inactivity timer, etc.) of a base station (BS), denoted by $x \in \mathbb{R}^d$, influences all the users under the coverage of this BS [5]. After a configuration is applied, the BS receives a reward in terms of the network-level performance, which accounts for the performance of all users within the coverage (e.g., average user-perceived Quality of Experience (QoE)). Specifically, let the mean global reward of configuration x be $f(x) = \langle \theta^*, x \rangle$, where $\theta^* \in \mathbb{R}^d$ represents the unknown global parameter. While some configuration may work best for a specific user, only one configuration can be applied at the BS at a time, which, however, simultaneously influences all the users within the coverage. Therefore, the goal here is to find the best configuration that maximizes the global reward (i.e., the network-level performance).

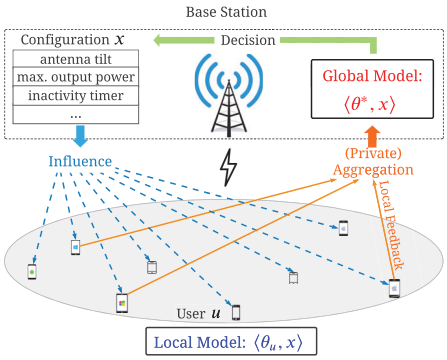


Fig. 1. Cellular network configuration: a motivating application of global reward maximization with partial feedback in a linear bandit setting.

At first glance, it seems that one can address the above problem by applying existing linear bandit algorithms (e.g., LinUCB [4]) to learn the global parameter θ^*. However, this would require collecting reward feedback from the entire population, which could incur a prohibitively high cost or could even be impossible to implement in practice when the population is large. To learn the global parameter, one natural way is to sample a subset of users from the population and aggregate this distributed partial feedback. This leads to a new problem we consider in this paper: *global reward maximization with partial feedback in a distributed linear bandit setting.* As in many distributed supervised learning problems [7]–[9], privacy protection is also of significant importance in our setting as clients' local feedback may contain their sensitive information. In summary, we are interested in the following fundamental question: *How to privately achieve global reward*

This work is supported in part by the NSF grants under CNS-2112694 and CNS-2153220.

Fengjiao Li (fengjiaoli@vt.edu) and Bo Ji (boji@vt.edu) are with the Department of Computer Science, Virginia Tech, Blacksburg, VA, USA. Xingyu Zhou (xingyu.zhou@wayne.edu) is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA.

ISBN 978-3-903176-49-2 © 2022 IFIP

TABLE I
SUMMARY OF MAIN RESULTS

Algorithm[1]	Regret[2]	Communication cost[3]	Privacy
DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}\right)$	$O(dT^{\alpha})$	None
CDP-DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha}\sqrt{\ln(1/\delta)\log(kT)}/\epsilon\right)$	$O(dT^{\alpha})$	(ϵ,δ)-DP
LDP-DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha/2}\sqrt{\ln(1/\delta)\log(kT)}/\epsilon\right)$	$O(dT^{\alpha})$	(ϵ,δ)-LDP
SDP-DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha}\ln(d/\delta)\sqrt{\log(kT)}/\epsilon\right)$	$O(dT^{3\alpha/2})$ (bits)	(ϵ,δ)-SDP

[1]DPE is the non-private DP-DPE algorithm; CDP-DPE, LDP-DPE, and SDP-DPE represent the DP-DPE algorithm in the central, local, and shuffle models, respectively, which guarantee (ϵ,δ)-DP, (ϵ,δ)-LDP, and (ϵ,δ)-SDP, respectively.
[2]In the regret upper bounds, T is the time horizon, k is the number of actions, d is the dimension of the action space, and α is a design parameter that can be used to tune the tradeoff between the regret and the communication cost. We ignore lower-order terms for simplicity.
[3]While the communication cost of CDP-DPE and LDP-DPE is measured in the number of real numbers transmitted between the clients and the server, SDP-DPE directly uses bits for reporting feedback. A detailed discussion is provided in our online technical report [6].

maximization with only partial distributed feedback?

To that end, we introduce a new model called *differentially private distributed linear bandit (DP-DLB)*. In DP-DLB, there is a global linear bandit model $f(x) = \langle \theta^*, x\rangle$ with an unknown parameter $\theta^* \in \mathbb{R}^d$ at the central server (e.g., the BS); each user u of a large population has a local linear bandit model $f_u(x) = \langle \theta_u, x\rangle$, which represents the mean local reward for user u. Here, we assume that each user u has a local parameter $\theta_u \in \mathbb{R}^d$, motivated by the fact that the mean local reward (e.g., the expected QoE of a user under a certain network configuration) varies across the users. In addition, each local parameter θ_u is unknown and is assumed to be a realization of a random vector with the mean being the global model parameter θ^*. The server makes decisions based on the estimated global model, which can be learned through sampling a subset of users (referred to as clients) and iteratively aggregating these distributed partial feedback. While sampling more clients could improve the learning accuracy and thus lead to a better performance, it also incurs a higher communication cost. Therefore, it is important to address this tradeoff in the design of communication protocols. Furthermore, to protect users' privacy, we resort to *differential privacy (DP)* to guarantee that clients' sensitive information will not be inferred by an adversary. Therefore, the goal is to maximize the cumulative global reward (or equivalently minimize the regret due to not choosing the optimal action in hindsight) in a communication-efficient manner while providing privacy guarantees for the participating clients. Our main contributions are summarized as follows.

- We present the first work that considers global reward maximization with partial feedback in the distributed linear bandit setting. In addition to the traditional tradeoff between exploitation and exploration, learning with distributed feedback introduces two practical challenges: communication efficiency and privacy concerns. This adds an extra layer of difficulty in the design of learning algorithms.
- To address these challenges, we introduce a DP-DLB model and develop a carefully-crafted algorithmic learning framework, called differentially private distributed phased elimination (DP-DPE), which allows the server and the clients to work in concert and can be naturally integrated with several state-of-the-art DP trust models (including central model, local model, and shuffle model). This unified framework enables us to systemically study the key regret-communication-privacy tradeoff.
- We then establish the regret-communication-privacy tradeoff of DP-DPE in various settings including the non-private case as well as the central, local, and shuffle DP models. Our main results are summarized in Table I. These results reveal that DP-DPE achieves privacy "for-free" in the central and shuffle models, in the sense that the additional regret due to privacy protection is only a lower-order additive term. Moreover, this is the first work that considers the shuffle model in distributed linear bandits to attain a better regret-privacy tradeoff, i.e., guaranteeing a similar privacy protection as the strong local model while achieving the same regret as the central model. We further perform simulations on synthetic data to corroborate our theoretical results.

Due to space limitations, we provide all the detailed proofs of our results in our online technical report [6].

II. RELATED WORK

We discuss the most relevant work here and provide a more detailed discussion in our technical report [6].

Distributed bandits. Our model is related to multi-agent collaborative learning in the distributed bandits setting [10]–[15]. The most relevant work to ours is the distributed linear bandit problem studied in [15]. Similarly, they design a distributed phased elimination algorithm where a central server aggregates data provided by the local clients and iteratively eliminates suboptimal actions. However, there are two key differences: (i) they consider the standard group regret minimization problem with homogeneous clients that have the same unknown parameter; (ii) the clients send the rewards to the central server without any privacy protection.

Federated bandits. Another line of related work is bandits in the federated setting [16]–[20], among which [19] and [20] are most relevant. In addition to different model and problem formulation we consider, we also highlight our main technical contributions compared to these works. While a phased elimination algorithm is also employed in [19], there are two key differences: (i) They do not consider the correlation among

the actions. Specifically, they consider a linear reward for contextual bandits while still studying MAB with independent actions, each of which is associated with a distinct parameter vector. Differently, the linear bandits formulation in our work is used to capture the correlation among the actions; (ii) When aggregating users' feedback for learning the global parameter, we protect users' data privacy through rigorous differential privacy guarantees, which is not considered in their design. While DP is also employed to protect users' data privacy in [20], they require that both the Gram matrix of actions (of size $O(d^2)$) and reward vectors (of size $O(d)$) be periodically communicated. Differently, our algorithm only requires that private average local reward for the chosen actions (of size $O(d \log \log d)$) be communicated in each phase. Moreover, while they only consider the central DP model, we provide a unified algorithmic learning framework that can be integrated with different DP models. In particular, our proposed DP-DPE algorithm integrated with the shuffle DP model can achieve a better regret-communication-privacy tradeoff (see Table I).

Differentially private bandits. Since proposed in [21], DP has become the *de facto* privacy-preserving model in many applications, including online learning [22] and bandit problems [23]. Specifically, in [24]–[26], MAB has been studied in the central, local, and shuffle DP models, respectively. In [27], the authors explore DP in contextual linear bandits and introduce a joint DP model. As a stronger privacy notion, local DP is also studied for contextual linear bandits [28] and Bayesian optimization [29]. However, none of them considers shuffle DP in the linear bandits setting. Moreover, our DP-DPE algorithm can be naturally integrated with several different DP models.

III. System Model and Problem Formulation

We begin with some notations: $[N] \triangleq \{1, \ldots, N\}$ for any positive integer N; $|S|$ denotes the cardinality of set S; $\|x\|_2$ denotes the ℓ_2-norm of vector x; the inner product is denoted by $\langle \cdot, \cdot \rangle$. For a positive definite matrix $A \in \mathbb{R}^{d \times d}$, the weighted ℓ_2-norm of vector $x \in \mathbb{R}^d$ is defined as $\|x\|_A \triangleq \sqrt{x^\top A x}$.

A. Global Reward Maximization with Partial Feedback

We consider the global reward maximization problem over a large population, which is a sequential decision making problem. In each round t, the learning agent (e.g., the BS or the policy maker) selects an action x_t from a finite decision set $\mathcal{D} \subseteq \{x \in \mathbb{R}^d : \|x\|_2^2 \leq 1\}$ with $|\mathcal{D}| = k$. This action leads to a global reward with mean $\langle \theta^*, x_t \rangle$, where $\theta^* \in \mathbb{R}^d$ with $\|\theta^*\|_2 \leq 1$ is unknown to the agent. This global reward captures the overall effectiveness of action x_t over a large population $\mathcal{U}$. The local reward of action x_t at user u has a mean $\langle \theta_u, x_t \rangle$, where $\theta_u \in \mathbb{R}^d$ is the local parameter, which is assumed to be a realization of a random vector with mean θ^* and is also unknown. Let $x^* \triangleq \operatorname{argmax}_{x \in \mathcal{D}} \langle \theta^*, x \rangle$ be the unique global optimal action. Then, the objective of the agent is to maximize the cumulative global reward, or equivalently, to minimize the regret defined as follows:

$$R(T) \triangleq T \langle \theta^*, x^* \rangle - \sum_{t=1}^{T} \langle \theta^*, x_t \rangle. \tag{1}$$

At first glance, standard linear bandit algorithms (e.g., LinUCB in [4]) can be applied to addressing the above problem. However, the exact reward here is a global quantity, which is the average over the entire population. The learning agent may not be able to observe this exact reward, since collecting such global information from the entire population incurs a prohibitively high cost, is often impossible to implement in practice, and could lead to privacy concerns.

B. Differentially Private Distributed Linear Bandits

To address the above problem, we consider a *differentially private distributed linear bandit (DP-DLB)* formulation, where there are two important entities: a central server (which wants to learn the global model) and participating clients (i.e., a subset of users from the population who are willing to share their feedback). In the following, we discuss important aspects of the DP-DLB formulation.

Server. The server aims to learn the global linear bandit model, i.e., unknown parameter θ^*. In each round t, it selects an action x_t with the objective of maximizing the cumulative global reward $\sum_{t=1}^{T} \langle \theta^*, x_t \rangle$. Without observing the exact reward of action x_t, the server collects only partial feedback from a subset of users sampled from the population, called *clients*, and then aggregates this partial feedback to update the estimate of the global parameter θ^*. Based on the updated model, the server chooses an action in the next round.

Clients. We assume that each participating client is randomly sampled from the population and is independent from each other and also from other randomness. Specifically, we assume that local parameter θ_u at client u satisfies $\theta_u = \theta^* + \xi_u$, where $\xi_u \in \mathbb{R}^d$ is a zero-mean σ-sub-Gaussian random vector[4] and is independently and identically distributed (*i.i.d.*) across all clients. Let U_t be the set of clients in round t. After action x_t is chosen by the server in round t, each client $u \in U_t$ observes a noisy local reward: $y_{u,t} = \langle \theta_u, x_t \rangle + \eta_{u,t}$, where $\eta_{u,t}$ is a conditionally 1-sub-Gaussian[5] noise and *i.i.d.* across the clients and over time. We also assume that the local rewards are bounded, i.e., $\|y_{u,t}\|_2 \leq B$, for all $u \in \mathcal{U}$ and $t \in [T]$.

Communication. The communication happens when the clients report their feedback to the server. At the beginning of each communication step, each participating client reports feedback to the server based on the local reward observations during a certain number of rounds. In particular, the time duration between reporting feedback is called a phase. By aggregating such feedback from the clients, the server estimates the global parameter θ^* and adjusts its decisions in the following rounds accordingly. We assume that the clients do not quit before a phase ends. By slightly abusing the notation, we use U_l to denote the set of clients in the l-th phase.

The communication cost is a critical factor in DP-DLB. As in [15], we define the communication cost as the total number

[4]A random vector $\xi \in \mathbb{R}^d$ is said to be σ-sub-Gaussian if $\mathbb{E}[\xi] = 0$ and $v^\top \xi$ is σ-sub-Gaussian for any unit vector $v \in \mathbb{R}^d$ and $\|v\|_2 = 1$ [30].

[5]Consider noise sequence $\{\eta_t\}_{t=1}^{\infty}$. As in the general linear bandit model [2], η_t is assumed to be conditionally 1-sub-Gaussian, meaning $\mathbb{E}[e^{\lambda \eta_t} | x_{1:t}, \eta_{1:t}] \leq \exp(\lambda^2/2)$ for all $\lambda \in \mathbb{R}$, where $a_{i:j}$ denotes the subsequence $a_i, \ldots, a_j$.

of real numbers (or bits, depending on the adopted DP model) communicated between the server and the clients. Let L be the number of phases in T rounds, and let N_l be the number of real numbers (or bits) communicated in the l-th phase. Then, the total communication cost, denoted by $C(T)$, is

$$C(T) \triangleq \sum_{l=1}^{L} |U_l| N_l. \tag{2}$$

Data privacy. In practice, even if users are willing to share their feedback, they typically require privacy protection as a premise. To that end, we resort to *differential privacy (DP)* [21] to formally address the privacy concerns in the learning process. More importantly, instead of only considering the standard central model where the central server is responsible for protecting the privacy, we will also incorporate other popular DP models, including the stronger local model (where each client directly protects her data) [31] and the recently proposed shuffle model (where a trusted shuffler between clients and server is adopted to amplify privacy) [32], in a unified algorithmic learning framework.

IV. Algorithm Design

In this section, we first present the key challenges associated with the introduced DP-DLB model and then explain how the developed DP-DPE framework addresses these challenges, followed by a brief description of DP-DPE instantiations with three different DP models (central, local, and shuffle).

A. Key Challenges

To solve the problem of global reward maximization with partial distributed feedback using the DP-DLB formulation, we face four key challenges, discussed in detail below.

As in the standard stochastic bandits problem, there is an uncertainty due to noisy rewards of each chosen action, which is called the *action-related uncertainty*. In addition to this, we face another type of uncertainty related to the sampled clients in DP-DLB, called the *client-related uncertainty*. The client-related uncertainty lies in estimating the global model at the server based on randomly sampled clients with *biased* local models. Note that the global model may not be accurately estimated even if exact rewards of the sampled clients are known when the number of clients is insufficient. Therefore, the first challenge lies in *simultaneously addressing both types of uncertainty in a sample-efficient way* (Challenge ⓐ).

To handle the newly introduced client-related uncertainty, we must sample a sufficiently large number of clients so that the global parameter can be accurately estimated using the partial distributed feedback. However, too many clients result in a large communication cost (see Eq. (2)). Therefore, the second challenge is to *decide the number of sampled clients to balance the regret (due to the client-related uncertainty) and the communication cost* (Challenge ⓑ).

Finally, to ensure privacy guarantees for the clients, one needs to add additional perturbations (or noises) to the local feedback. *Such randomness introduces another type of uncertainty to the learning process* (Challenge ⓒ), and *it is unclear how to integrate different trust DP models into a unified algorithmic learning framework* (Challenge ⓓ). These add an extra layer of difficulty to the design of learning algorithms.

Main ideas. In the following, we present our main ideas for addressing the above challenges. We propose a phased elimination algorithm that gradually eliminates suboptimal actions by periodically aggregating the local feedback from the sampled clients in a privacy-preserving manner. To address the multiple types of uncertainty when estimating the global reward (ⓐ and ⓒ), we carefully construct a confidence width to incorporate all three types of uncertainty. To achieve a sublinear regret while saving communication cost (ⓑ), we increase both the phase length and the number of clients exponentially. To ensure privacy guarantees (ⓓ), we introduce a PRIVATIZER that can be easily tailored under different DP models. The PRIVATIZER is a process consisting of tasks to be collaboratively completed by the clients, the server, and/or even a trusted third party. To keep it general, we use $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$ to denote a PRIVATIZER, where $\mathcal{R}$ is the procedure at each client (usually a local randomizer), $\mathcal{S}$ is a trusted third party that helps privatize data (e.g., a shuffler that permutes received messages), and $\mathcal{A}$ is an analyzer operated at the central server. Next, we will show how to integrate these main ideas into a unified algorithmic learning framework.

B. Differentially Private Distributed Phased Elimination

With the main ideas presented above, we now propose a unified algorithmic learning framework, called *differentially private distributed phased elimination (DP-DPE)*, which is presented in Algorithm 1. The DP-DPE runs in phases and operates with the coordination of the central server and the participating clients in a synchronized manner. At a high level, each phase consists of the following three steps:

- **Action selection (Lines 4-6):** computing a near-G-optimal design (i.e., a distribution) over a set of possibly optimal actions and playing these actions;
- **Clients sampling and private feedback aggregation (Lines 7-16):** sampling participating clients and aggregating their local feedback in a privacy-preserving fashion;
- **Parameter estimation and action elimination (Lines 17-19):** using (privately) aggregated data to estimate θ^* and eliminating actions that are likely to be suboptimal.

In the following, we describe the detailed operations of DP-DPE. We begin by giving some necessary notations. Consider the l-th phase. Let t_l and T_l be the index of the starting round and the length of the l-th phase, respectively. Then, let $\mathcal{T}_l \triangleq \{t \in [T] : t_l \leq t < t_l + T_l\}$ be the round indices in the l-th phase, let $\mathcal{T}_l(x) \triangleq \{t \in \mathcal{T}_l : x_t = x\}$ be the time indices in the l-th phase when action x is selected, and let $\mathcal{D}_l \subseteq \mathcal{D}$ be the set of active actions in the l-th phase.

Action selection (Lines 4-6): In the l-th phase, the action set $\mathcal{D}_l$ consists of active actions that are possibly optimal. We compute a distribution $\pi_l(\cdot)$ over $\mathcal{D}_l$ and choose actions according to $\pi_l(\cdot)$. We briefly explain the intuition below. Let $V(\pi) \triangleq \sum_{x \in \mathcal{D}} \pi(x) x x^\top$ and $g(\pi) \triangleq \max_{x \in \mathcal{D}} \|x\|^2_{V(\pi)^{-1}}$.

Algorithm 1 Differentially Private Distributed Phased Elimination (DP-DPE)

1: **Input:** $\mathcal{D} \subseteq \mathbb{R}^d$, $\alpha \in (0,1)$, $\beta \in (0,1)$, and σ_n
2: **Initialization:** $l = 1$, $t_1 = 1$, $\mathcal{D}_1 = \mathcal{D}$, and $h_1 = 2$
3: **while** $t_l \leq T$ **do**
4: Find a distribution $\pi_l(\cdot)$ over $\mathcal{D}_l$ such that $g(\pi_l) \triangleq \max_{x \in \mathcal{D}_l} \|x\|^2_{V(\pi_l)^{-1}} \leq 2d$ and $|\text{supp}(\pi_l)| \leq 4d \log\log d + 16$, where $V(\pi_l) \triangleq \sum_{x \in \mathcal{D}_l} \pi_l(x) x x^\top$
5: Let $T_l(x) = \lceil h_l \pi_l(x) \rceil$ for each $x \in \text{supp}(\pi_l)$ and $T_l = \sum_{x \in \text{supp}(\pi_l)} T_l(x)$
6: Play each action $x \in \text{supp}(\pi_l)$ exactly $T_l(x)$ times if not reaching T
7: Randomly select $\lceil 2^{\alpha l} \rceil$ participating clients U_l
Operations at each client
8: **for** each client $u \in U_l$ **do**
9: **for** each action $x \in \text{supp}(\pi_l)$ **do**
10: Compute average local reward over $T_l(x)$ rounds: $y_l^u(x) = \frac{1}{T_l(x)} \sum_{t \in \mathcal{T}_l(x)} (\langle \theta_u, x \rangle + \eta_{u,t})$
11: **end for**
12: Let $\vec{y}_l^u = (y_l^u(x))_{x \in \text{supp}(\pi_l)}$
Apply the PRIVATIZER $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$
The local randomizer $\mathcal{R}$ at each client:
13: Run the local randomizer $\mathcal{R}$ and send the output $\mathcal{R}(\vec{y}_l^u)$ to $\mathcal{S}$
14: **end for**
Computation $\mathcal{S}$ at a trusted third party:
15: Run the computation function $\mathcal{S}$ and send the output $\mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u \in U_l})$ to the analyzer $\mathcal{A}$
The analyzer $\mathcal{A}$ at the server:
16: Generate the privately aggregated statistics: $\tilde{y}_l = \mathcal{A}(\mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u \in U_l}))$
17: Compute the following quantities:

$$\begin{cases} V_l = \sum_{x \in \text{supp}(\pi_l)} T_l(x) x x^\top \\ G_l = \sum_{x \in \text{supp}(\pi_l)} T_l(x) x \tilde{y}_l(x) \\ \tilde{\theta}_l = V_l^{-1} G_l \end{cases}$$

18: Find low-rewarding actions with confidence width W_l:

$$E_l = \left\{ x \in \mathcal{D}_l : \max_{b \in \mathcal{D}_l} \langle \tilde{\theta}_l, b - x \rangle > 2W_l \right\}$$

19: Update: $\mathcal{D}_{l+1} = \mathcal{D}_l \backslash E_l$, $h_{l+1} = 2h_l$, $t_{l+1} = t_l + T_l$, and $l = l + 1$
20: **end while**

According to the analysis in [2, Chapter 21], if action $x \in \mathcal{D}$ is played $\lceil h\pi(x) \rceil$ times (where h is a positive constant), the estimation error associated with the action-related uncertainty for action x is at most $\sqrt{2g(\pi)\log(1/\beta)/h}$ with probability $1 - \beta$ for any $\beta \in (0,1)$. That is, for a fixed number of rounds, a distribution $\pi(\cdot)$ with a smaller value of $g(\pi)$ helps achieve a better estimation. Note that minimizing $g(\cdot)$ is a well-known *G-optimal design* problem [33]. By the Kiefer-Wolfowitz Theorem [34], one can find a distribution π^* minimizing $g(\cdot)$ with $g(\pi^*) = d$, and the support set[6] of π^*, denoted by $\text{supp}(\pi^*)$, has a size no greater than $d(d+1)/2$. In our problem, however, it suffices to solve it near-optimally, i.e., finding a distribution π_l such that $g(\pi_l) \leq 2d$ with $|\text{supp}(\pi_l)| \leq 4d \log\log d + 16$ (Line 4), which follows from [35, Proposition 3.7]. The near-G-optimal design reduces the complexity to $O(kd^2)$ while keeping the same order of regret.

Clients sampling and private feedback aggregation (Lines 7-16): The central server randomly samples a subset U_l of $\lceil 2^{\alpha l} \rceil$ users (called clients) from the population $\mathcal{U}$ to participate in the global bandit learning (Line 7). Each sampled client $u \in U_l$ collects their local reward observations of each chosen action $x \in \text{supp}(\pi_l)$ by the server and computes the average $y_l^u(x)$ as feedback (Line 10). Before being used to estimate the global parameter by the central server, these feedback $\vec{y}_l^u \triangleq (y_l^u(x))_{x \in \text{supp}(\pi_l)} \in \mathbb{R}^{|\text{supp}(\pi_l)|}$ are processed by a PRIVATIZER $\mathcal{P}$ to ensure differential privacy. Recall that a PRIVATIZER $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$ is a process completed by the clients, the server, and/or a trusted third party. In particular, according to the privacy requirement under different DP models, the PRIVATIZER $\mathcal{P}$ enjoys flexible instantiations (see Section IV-C). Generally, a PRIVATIZER works in the following manner: each client u runs the randomizer $\mathcal{R}$ on its local average reward $\vec{y}_l^u$ (over T_l pulls) and then sends the resulting (potentially private) messages $\mathcal{R}(\vec{y}_l^u)$ to $\mathcal{S}$ (Line 13). The computation function in $\mathcal{S}$ operates on these messages and then sends results $\mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u \in U_l})$ to the analyzer $\mathcal{A}$ at the central server (Line 15). Finally, the analyzer $\mathcal{A}$ aggregates received messages (potentially in a privacy-preserving manner) and outputs a private averaged local reward $\tilde{y}_l(x)$ (over participating clients U_l) for each action $x \in \text{supp}(\pi_l)$ (Line 16). We provide the rigorous formulation of different DP models for PRIVATIZER $\mathcal{P}$ in our technical report [6], with corresponding detailed instantiations of $\mathcal{R}, \mathcal{S}$, and $\mathcal{A}$.

Parameter estimation and action elimination (Lines 17-19): Using privately aggregated feedback (i.e., the private averaged local reward $\tilde{y}_l$ of the chosen actions $x \in \text{supp}(\pi_l)$), the central server computes the least-square estimator $\tilde{\theta}_l$ (Line 17). We perform action elimination based on the following confidence width:

$$W_l \triangleq \left(\underbrace{\sqrt{\frac{2d}{|U_l| h_l}}}_{\text{action-related}} + \underbrace{\frac{\sigma}{\sqrt{|U_l|}}}_{\text{client-related}} + \underbrace{\sigma_n}_{\text{privacy noise}} \right) \sqrt{2 \log\left(\frac{1}{\beta}\right)}, \tag{3}$$

where σ is the standard variance associated with client sampling, σ_n is related to the privacy noise determined by the DP model, and β is the confidence level. We choose this confidence width based on the concentration inequality for sub-Gaussian variables. Specifically, the three terms in Eq. (3) capture the action-related uncertainty, client-related uncertainty, and the added noise for privacy guarantees, respectively.

[6]The support set of a distribution π over set $\mathcal{D}$, denoted by $\text{supp}_{\mathcal{D}}(\pi)$, is the subset of elements with a nonzero $\pi(\cdot)$, i.e., $\text{supp}_{\mathcal{D}}(\pi) \triangleq \{x \in \mathcal{D} : \pi(x) \neq 0\}$. We drop the subscript $\mathcal{D}$ in $\text{supp}_{\mathcal{D}}(\pi)$ for notational simplicity.

This privacy noise σ_n depends on the adopted DP model. Using this confidence width W_l and the estimated global model parameter $\tilde{\theta}_l$, we can identify a subset of suboptimal actions E_l with high probability (Line 18). At the end of the l-th phase, we update the set of active actions $\mathcal{D}_{l+1}$ by eliminating E_l from $\mathcal{D}_l$ and double h_l (Line 19).

C. DP-DPE under Different DP Models

We now briefly explain how to instantiate the PRIVATIZER $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$ in DP-DPE using three representative DP trust models: the central, local, and shuffle models. In addition, we also present the formal definition of the privacy guarantees regarding $\mathcal{P}$ under each trust model, which further implies respective privacy guarantee of DP-DPE according to the post-processing property of DP [36, Proposition 2.1]. We provide the detailed descriptions in our technical report [6].

DP-DPE under the central DP model (CDP-DPE). Under the central DP model, each client trusts the server, and the outputs of the server on two neighboring datasets (differing by only one client) must be indistinguishable [21]. To achieve this, the PRIVATIZER functions as follows: while both $\mathcal{R}$ and $\mathcal{S}$ are simply identity mappings, $\mathcal{A}$ injects well-tuned Gaussian noise to the aggregated statistics for privacy. That is,

$$\tilde{y}_l = \mathcal{A}\left(\{\vec{y}_l^u\}_{u\in U_l}\right) = \frac{1}{|U_l|}\sum_{u\in U_l}\vec{y}_l^u + (\gamma_1, \ldots, \gamma_{s_l}), \qquad (4)$$

where $s_l \triangleq |\text{supp}(\pi_l)|$, $\gamma_j \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_{nc}^2)$, and the variance σ_{nc}^2 is chosen according to the ℓ_2-sensitivity of the average $\frac{1}{|U_l|}\sum_{u\in U_l}\vec{y}_l^u$. Consider a particular phase l. The PRIVATIZER $\mathcal{P}$ is (ϵ, δ)-differentially-private (or (ϵ, δ)-DP) if the following is satisfied for any pair of $U_l, U_l' \subseteq \mathcal{U}$ that differ by at most one client and for any output $\tilde{y}$ of $\mathcal{A}$:

$$\mathbb{P}[\mathcal{A}(\{\vec{y}_l^u\}_{u\in U_l}) = \tilde{y}] \leq e^\epsilon \cdot \mathbb{P}[\mathcal{A}(\{\vec{y}_l^u\}_{u\in U_l'}) = \tilde{y}] + \delta.$$

DP-DPE under the local DP model (LDP-DPE). Under the local DP model, since clients do not trust the server, each client with a local randomizer $\mathcal{R}$ is responsible for privacy protection by injecting Gaussian noise; $\mathcal{S}$ is an identity mapping; $\mathcal{A}$ is a simple aggregation function. That is,

$$\tilde{y}_l = \frac{1}{|U_l|}\sum_{u\in U_l}\mathcal{R}(\vec{y}_l^u) = \frac{1}{|U_l|}\sum_{u\in U_l}\left(\vec{y}_l^u + (\gamma_{u,1}, \ldots, \gamma_{u,s_l})\right), \qquad (5)$$

where $\gamma_{u,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_{nl}^2)$ and the variance σ_{nl}^2 is chosen according to the sensitivity of $\vec{y}_l^u$. Consider any phase l. Let Y_u be the set of all possible values of the average local reward $\vec{y}_l^u$ for client u. The PRIVATIZER $\mathcal{P}$ is (ϵ, δ)-local-differentially-private (or (ϵ, δ)-LDP) if the following is satisfied for any client u, for any pair of $\vec{y}, \vec{y}' \in Y_u$, and for any output $o \in \{\mathcal{R}(\vec{y}) | \vec{y} \in Y_u\}$:

$$\mathbb{P}[\mathcal{R}(\vec{y}) = o] \leq e^\epsilon \cdot \mathbb{P}[\mathcal{R}(\vec{y}') = o] + \delta.$$

DP-DPE under the shuffle DP model (SDP-DPE). Under the shuffle model, without a trusted server, we instantiate DP-DPE by building on the vector summation protocol recently proposed in [37]. Specifically, each local randomizer $\mathcal{R}$ encodes its inputs by adding random bits; the analyzer $\mathcal{A}$ outputs the random vector whose expectation is the average of the input vectors; beyond that, we leverage a third-party shuffler $\mathcal{S}$, which uniformly at random permutes users' messages (in bits) to hide their sources. That is,

$$\tilde{y}_l = \mathcal{P}\left(\{\vec{y}_l^u\}_{u\in U_l}\right) = \mathcal{A}(\mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u\in U_l})), \qquad (6)$$

where the additional randomness introduced by $\mathcal{S}$ allows each local $\mathcal{R}$ to inject only a small amount of noise σ_{ns} while still guaranteeing a private view at the analyzer $\mathcal{A}$. Consider any phase l. We use $(\mathcal{S}\circ\mathcal{R})(U_l) \triangleq \mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u\in U_l})$ to denote the composite mechanism. Formally, the PRIVATIZER $\mathcal{P}$ is (ϵ, δ)-shuffle-differentially-private (or (ϵ, δ)-SDP) if the following is satisfied for any pair of U_l, $U_l' \subseteq \mathcal{U}$ that differ by one client and for any possible output z of $\mathcal{S}\circ\mathcal{R}$:

$$\mathbb{P}[(\mathcal{S}\circ\mathcal{R})(U_l) = z] \leq e^\epsilon \cdot \mathbb{P}[(\mathcal{S}\circ\mathcal{R})(U_l') = z] + \delta.$$

V. MAIN RESULTS

In this section, we study the performance of DP-DPE under different DP models in terms of regret and communication cost. We start with the non-private DP-DPE algorithm (called DPE, with $\tilde{y}_l = \frac{1}{|U_l|}\sum_{u\in U_l}\vec{y}_l^u$ and $\sigma_n = 0$ for all l) and present the main result in Theorem 1.

Theorem 1 (DPE): Let $\beta = 1/(kT)$ and $\sigma_n = 0$ in Algorithm 1. Then, the non-private DP-DPE algorithm achieves the following expected regret:

$$\mathbb{E}[R(T)] = O(\sqrt{dT\log(kT)}) + O(\sigma T^{1-\alpha/2}\sqrt{\log(kT)}),$$

with a communication cost of $O(dT^\alpha)$.

Remark 1: Theorem 1 gives a problem-independent regret upper bound for DPE. We can observe an obvious tradeoff between regret and communication cost, captured by the value of α. While a larger α leads to a smaller regret, it also incurs a larger communication cost. Setting $\alpha = 2/3$ gives $O(T^{2/3})$ for both regret and communication cost.

In Theorem 2, we present the performance of DP-DPE under different DP models in terms of regret, communication cost, and privacy guarantee. Let $S \triangleq 4d\log\log d + 16$.

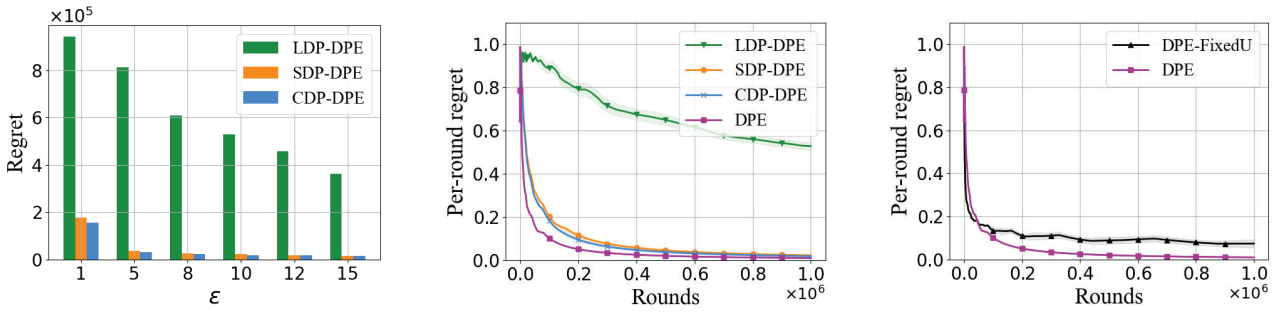
Theorem 2: Let $\beta = 1/(kT)$. DP-DPE under different DP models with the following parameters achieves the corresponding results in Table I:

(i) **CDP-DPE**. Set $\sigma_{nc} = O\left(\frac{B\sqrt{d\ln(1/\delta)}}{\epsilon|U_l|}\right)$ in (4) for each phase l and $\sigma_n = 2\sigma_{nc}\sqrt{Sd}$ in (3);

(ii) **LDP-DPE**. Set $\sigma_{nl} = O\left(\frac{B\sqrt{d\ln(1/\delta)}}{\epsilon}\right)$ in (5) for each phase l and $\sigma_n = 2\sigma_{nl}\sqrt{Sd/|U_l|}$ in (3);

(iii) **SDP-DPE**. Set $\sigma_{ns} = O\left(\frac{B\sqrt{d}\ln(d/\delta)}{\epsilon|U_l|}\right)$ in (6) for each phase l and $\sigma_n = 2\sigma_{ns}\sqrt{Sd}$ in (3).

Remark 2 (Privacy "for-free"): Comparing the above results with Theorem 1 for the non-private case, we observe that the DP-DPE algorithm enables us to achieve privacy guarantees "for free" in the central and shuffle DP models, in the sense that the additional regret due to privacy protection is



(a) (b) (c)

Fig. 2. Performance comparisons of different algorithms. The shaded area indicates the standard deviation. (a) Final cumulative regret vs. the privacy budget ϵ. (b) Per-round regret vs. time with privacy parameters $\epsilon = 10$ and $\delta = 0.25$. (c) Per-round regret vs. time for two non-private algorithms. Here, we choose the number of clients in DPE-FixedU to be $U = 97$ based on the calculation.

only a lower-order additive term. Essentially, this is because the uncertainty introduced by privacy noise is dominated by the client-related uncertainty, which can be captured by our carefully designed confidence width W_l in Eq. (3) and our choice of σ_n for different PRIVATIZERS.

Remark 3 (Regret-privacy tradeoff under the shuffle model): Consider the regret due to privacy protection. From Theorem 2, we can see that while the local DP model ensures a stronger privacy guarantee compared to the central DP model, it introduces an additional regret of $O(T^{1-\alpha/2})$ compared to $O(T^{1-\alpha})$ in the central DP model. The shuffle DP model, however, leads to a much better tradeoff between regret and privacy, achieving nearly the same regret guarantee as the central DP model, yet assuming a similar trust model to the local DP model (i.e., without a trustworthy central server).

Remark 4 (Communication cost): Both CDP-DPE and LDP-DPE consume the same amount of communication resources as the non-private DP-DPE algorithm, measured by the number of real numbers [15]. In contrast, SDP-DPE relies only on binary feedback from the clients, and thus, the communication cost is measured by the number of bits. It is worth noting that sending messages consisting of real numbers could be difficult in practice on finite computers [38], [39], and hence in this case, it is desirable to use SDP-DPE, which incurs a communication cost of $O(dT^{3\alpha/2})$ bits.

VI. NUMERICAL RESULTS

In this section, we conduct simulations to evaluate the performance of DP-DPE. The detailed setting of our simulations is as follows: $d = 20, k = 10^3, \sigma = 0.1, |\mathcal{U}| = 10^5, \alpha = 0.8$, and $T = 10^6$. We perform 20 independent runs for each set of simulations.

First, we study the regret performance of DP-DPE under different DP models. Recall that we use CDP-DPE, LDP-DPE, and SDP-DPE to denote DP-DPE in the central, local, and shuffle DP models, respectively. In Fig. 2(a), we present the cumulative regret at the end of T rounds for the three algorithms under different values of privacy budget ϵ. We can observe an obvious tradeoff between the privacy budget and the regret performance for all the DP models: the cumulative regret decreases as the privacy requirement becomes less stringent (i.e., a larger ϵ). In addition, it also reflects the regret-privacy tradeoff across different DP models. That is, with the same privacy budget ϵ, while LDP-DPE has the largest regret yet without requiring the clients to trust anyone else (neither the server nor a third party), CDP-DPE achieves the smallest regret but relies on the assumption that the clients trust the server. Interestingly, SDP-DPE achieves a regret fairly close to that of CDP-DPE, yet without the need to trust the server. This is well aligned with our theoretical results that SDP-DPE achieves a better regret-privacy tradeoff.

In addition, we are also interested in the regret loss due to privacy protection and how efficiently DP-DPE performs the global bandit learning. Fix the privacy parameters $\epsilon = 10$ and $\delta = 0.25$. In Fig. 2(b), we plot how the per-round regret of the three algorithms (i.e., CDP-DPE, LDP-DPE, and SDP-DPE) varies over time compared to the non-private DP-DPE algorithm (i.e., DPE). We observe that LDP-DPE incurs the largest regret while ensuring the strongest privacy guarantee (i.e., (ϵ, δ)-LDP). On the other hand, the regret performance of CDP-DPE and SDP-DPE is very close to that of DPE (that does not ensure any privacy guarantee), under the assumption of a trusted central server and a trusted third party shuffler, respectively. This observation, along with our theoretical results, shows that DP-DPE can indeed achieve privacy "for-free" under the central and shuffle models.

Finally, we show that the exponentially-increasing client-sampling plays a key role in balancing the regret and the communication cost. To this end, we compare DPE (i.e., non-private DP-DPE) with another non-private algorithm, called DPE-FixedU in Fig. 2(c). DPE-FixedU is similar to DPE but samples only a fixed number U of participating clients in each phase (i.e., the participating clients are different, but the number of clients in each phase is fixed, in contrast to our increasing sampling schedule). For a fair comparison, we choose the value of U such that the communication cost is the same under DPE and DPE-FixedU, i.e., $U = \lceil \frac{\sum_{l=1}^{L} |U_l| \cdot N_l}{\sum_{l=1}^{L} N_l} \rceil$. The results show that DPE learns much faster than DPE-FixedU while incurring the same communication cost.

VII. Conclusion

In this paper, we studied a new problem of global reward maximization with partial distributed feedback. This problem is motivated by several practical applications where the expected reward of an action represents the overall performance over a large population. In such scenarios, it is often difficult, if not impossible, to collect exact reward feedback. To that end, we proposed a differentially private distributed linear bandits formulation, where the learning agent samples clients and interacts with them by iteratively aggregating such partial distributed feedback in a privacy-preserving fashion. We then developed a unified algorithmic learning framework, called DP-DPE, which can be naturally integrated with different DP models, and systematically established the regret-communication-privacy tradeoff.

In this work, we assumed that actions are correlated through a common linear function with parameter θ^*. One interesting direction for future work is to extend linear functions to general (possibly non-convex) functions via kernelized bandits. In addition, our work also raises several interesting questions that are worth investigating. For example, can we further improve the communication efficiency by using advanced shuffle protocols? Can we generalize our formulation to studying reinforcement learning problems?

References

[1] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[2] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[3] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits." in *NIPS*, vol. 11, 2011, pp. 2312–2320.

[4] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[5] A. Mahimkar, A. Sivakumar, Z. Ge, S. Pathak, and K. Biswas, "Auric: using data-driven recommendation to automatically generate cellular configuration," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 807–820.

[6] F. Li, X. Zhou, and B. Ji, "Differentially private linear bandits with partial distributed feedback," 2022. [Online]. Available: https://arxiv.org/abs/2207.05827

[7] R. Bassily, V. Feldman, K. Talwar, and A. Thakurta, "Private stochastic convex optimization with optimal rates," *arXiv preprint arXiv:1908.09970*, 2019.

[8] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[9] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2521–2529.

[10] M. Agarwal, V. Aggarwal, and K. Azizzadenesheli, "Multi-agent multi-armed bandits with limited communication," *arXiv preprint arXiv:2102.08462*, 2021.

[11] N. Cesa-Bianchi, C. Gentile, Y. Mansour, and A. Minora, "Delay and cooperation in nonstochastic bandits," in *Conference on Learning Theory*. PMLR, 2016, pp. 605–622.

[12] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, "Decentralized cooperative stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[13] A. Dubey *et al.*, "Kernel methods for cooperative multi-agent contextual bandits," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2740–2750.

[14] ——, "Cooperative multi-agent bandits with heavy tails," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2730–2739.

[15] Y. Wang, J. Hu, X. Chen, and L. Wang, "Distributed bandit learning: Near-optimal regret with efficient communication," *arXiv preprint arXiv:1904.06309*, 2019.

[16] C. Shi and C. Shen, "Federated multi-armed bandits," in *35th AAAI Conference on Artificial Intelligence*, 2021.

[17] C. Shi, C. Shen, and J. Yang, "Federated multi-armed bandits with personalization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2917–2925.

[18] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, "Federated bandit: A gossiping approach," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 1, pp. 1–29, 2021.

[19] R. Huang, W. Wu, J. Yang, and C. Shen, "Federated linear contextual bandits," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[20] A. Dubey and A. Pentland, "Differentially-private federated linear bandits," *arXiv preprint arXiv:2010.11425*, 2020.

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[22] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2012, pp. 24–1.

[23] N. Mishra and A. Thakurta, "(nearly) optimal differentially private stochastic multi-arm bandits," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 2015, pp. 592–601.

[24] A. C. Tossou and C. Dimitrakakis, "Algorithms for differentially private multi-armed bandits," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[25] W. Ren, X. Zhou, J. Liu, and N. B. Shroff, "Multi-armed bandits with local differential privacy," *arXiv preprint arXiv:2007.03121*, 2020.

[26] J. Tenenbaum, H. Kaplan, Y. Mansour, and U. Stemmer, "Differentially private multi-armed bandits in the shuffle model," *arXiv preprint arXiv:2106.02900*, 2021.

[27] R. Shariff and O. Sheffet, "Differentially private contextual linear bandits," *arXiv preprint arXiv:1810.00068*, 2018.

[28] K. Zheng, T. Cai, W. Huang, Z. Li, and L. Wang, "Locally differentially private (contextual) bandits learning," *arXiv preprint arXiv:2006.00701*, 2020.

[29] X. Zhou and J. Tan, "Local differential privacy for bayesian optimization," *arXiv preprint arXiv:2010.06709*, 2020.

[30] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[31] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[32] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 375–403.

[33] F. Pukelsheim, *Optimal design of experiments*. SIAM, 2006.

[34] J. Kiefer and J. Wolfowitz, "The equivalence of two extremum problems," *Canadian Journal of Mathematics*, vol. 12, pp. 363–366, 1960.

[35] T. Lattimore, C. Szepesvari, and G. Weisz, "Learning with good feature representations in bandits and in RL with a generative model," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5662–5670.

[36] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[37] A. Cheu, M. Joseph, J. Mao, and B. Peng, "Shuffle private stochastic convex optimization," *arXiv preprint arXiv:2106.09805*, 2021.

[38] C. L. Canonne, G. Kamath, and T. Steinke, "The discrete gaussian for differential privacy," *arXiv preprint arXiv:2004.00010*, 2020.

[39] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," *arXiv preprint arXiv:2102.06387*, 2021.