Distributed Linear Bandits With Differential Privacy

Fengjiao Li, *Member, IEEE*, Xingyu Zhou, *Member, IEEE*, and Bo Ji, *Senior Member, IEEE*

***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 distributed linear bandits with differential privacy, where 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 show 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. In addition, as a by-product of our techniques, the same results of "free" privacy can also be achieved for the standard differentially private linear bandits. Finally, we conduct simulations to corroborate our theoretical results and demonstrate the effectiveness of DP-DPE.**

***Index Terms*—Linear bandits, global reward maximization, partial distributed feedback, differential privacy, regret, communication cost.**

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 this application [2], 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 [3], [4], [5].

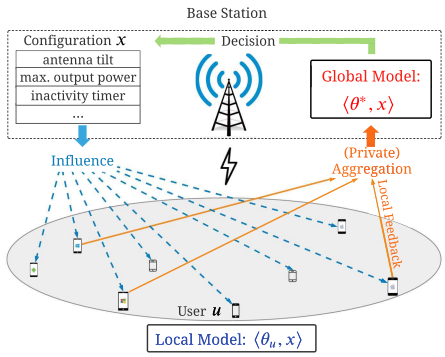

Fig. 1. Cellular network configuration: A motivating application of global reward maximization with partial feedback in a linear bandit setting.

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), with feature representation[1] $x \in \mathbb{R}^d$, influences all the users under the coverage of this BS [6]. 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 throughput). 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).

Manuscript received 12 June 2023; revised 26 December 2023; accepted 30 January 2024. Date of publication 6 February 2024; date of current version 30 April 2024. This work was supported in part by NSF under Grant CNS-2312833, Grant CNS-2112694, Grant CNS-2153220, and Grant CNS-2312835, in part by the Fundamental Research of Shanxi Province under Grant 202303021222030, in part by Commonwealth Cyber Initiative (CCI), and in part by Nokia Corporation. An earlier version of this paper was presented at IEEE/IFIP WiOpt 2022 [DOI: 10.23919/WiOpt56218.2022.9930524]. Recommended for acceptance by Dr. Chunxiao Jiang. *(Corresponding author: Bo Ji.)*

Fengjiao Li is with the Shanxi University, Taiyuan 030006, China, and also with the Virginia Tech, Blacksburg, VA 24061 USA (e-mail: fengjiaoli@sxu.edu.cn).

Xingyu Zhou is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA (e-mail: xingyu.zhou@wayne.edu).

Bo Ji is with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: boji@vt.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSE.2024.3362978, provided by the authors.

Digital Object Identifier 10.1109/TNSE.2024.3362978

[1] Similar to many linearly parameterized bandits (e.g., [5]), we may represent each configuration by a d-dimensional feature vector through some feature mapping.

At first glance, it seems that one can address the above problem by applying existing linear bandit algorithms (e.g., LinUCB [5]) 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,* which can be also applied to several other practical applications, including dynamic pricing and public policy selection [7], [8]. As in many distributed supervised learning problems [9], [10], [11], 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 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 throughput 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, 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 a new distributed linear bandit setting where only partial feedback is available, leading to a novel problem of global reward maximization with distributed partial feedback. In addition to the traditional exploitation and exploration tradeoff, 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 study the key regret-communication-privacy tradeoff systemically.
- We then establish the regret-communication-privacy tradeoff of DP-DPE in various settings, including the non-private case and the central, local, and shuffle DP models. Our main results are summarized in Table I. From Table I, we observe that the additional regret incurred by privacy is only a lower-order additive term, which is dominated by the regret from learning (i.e., $\tilde{O}(T^{1-\alpha}/\varepsilon)$ vs.[2] $\tilde{O}(T^{1-\alpha/2})$). In this sense, we say that DP-DPE might achieve privacy "for free" following [12]. Moreover, this is the first work considering the shuffle model in distributed linear bandits to attain a better regret-privacy tradeoff, i.e., guaranteeing 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.
- Finally, we provide an interesting discussion about achieving privacy "for free". We first highlight an interesting connection between our introduced DP-DLB formulation and the differentially private stochastic convex optimization (DP-SCO) problem in terms of achieving privacy "for free". This bridge between our online bandit learning and standard supervised learning might be of independent interest. Furthermore, differential privacy may also be ensured "for free" for standard linear bandits as well with minor modifications of our developed techniques.

II. 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 containing an infinite number of users, 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 $x^* \triangleq \text{argmax}_{x\in\mathcal{D}} \langle \theta^*, x\rangle$ be the unique global optimal action. Then, the objective of the agent is to maximize the cumulative global reward, or equivalently, to

[2]Here the $\tilde{O}(\cdot)$ notation hides the dependence on $polylog(T)$, the dimension d, and privacy parameter δ.

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	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha}\sqrt{\ln(1/\delta)\log(kT)}/\varepsilon\right)$	$O(dT^{\alpha})$	(ε,δ)-DP
LDP-DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha/2}\sqrt{\ln(1/\delta)\log(kT)}/\varepsilon\right)$	$O(dT^{\alpha})$	(ε,δ)-LDP
SDP-DPE	$O\left(T^{1-\alpha/2}\sqrt{\log(kT)}+d^{3/2}T^{1-\alpha}\ln(d/\delta)\sqrt{\log(kT)}/\varepsilon\right)$	$O(dT^{3\alpha/2})$ (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, we ignore lower-order terms for simplicity. 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.
[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 Section IV.

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 [5]) can be applied to address 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 and aggregates partial feedback from a subset of users sampled from the population, called *clients*, and then 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[3] and is independently and identically distributed (*i.i.d.*) across all clients. Let U_t be the set of clients in round t. After the server takes action x_t at 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[4] noise and *i.i.d.* across the clients and over time. Assume that the local rewards are bounded, i.e., $|y_{u,t}| \le B$, for all $u \in \mathcal{U}$ and $t \in [T]$.

Communication: 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 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 [14], we define the communication cost as the total number 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 N_l be the number of real numbers (or bits) communicated in the l-th phase. Then, the 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. *Differential privacy (DP)* [15] is a mathematical framework for ensuring the privacy of individuals in datasets. Specifically, by observing the calculation/statistics/model update from a set of individual data, an adversary cannot infer too much information about any specific individual. In this sense, DP can protect any existing or future attacks in that any adversary tries to infer any individual's information would fail no matter how much computation power they have or how much side information they have (i.e., even though the adversary has access to all the others' information except the targeted one). To that end, we

[3]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 $\|v\|_2 = 1$ [13].

[4]Consider noise sequence $\{\eta_t\}_{t=1}^{\infty}$. As in the general linear bandit model [3], η_t is assumed to be conditionally 1-sub-Gaussian, meaning $\mathbb{E}[e^{\lambda\eta_t}|x_{1:t}, \eta_{1:t}] \le \exp(\lambda^2/2)$ for all $\lambda \in \mathbb{R}$, where $a_{i:j}$ denotes the subsequence $a_i, \ldots, a_j$.

resort to DP 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 privacy, we will also incorporate other popular DP models, including the stronger local model (where each client directly protects her data) [16] and the recently proposed shuffle model (where a trusted shuffler between clients and server is adopted to amplify privacy) [17], in a unified algorithmic learning framework.

III. 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.

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 bandit problem [18], there is uncertainty due to the noisy rewards of each chosen action, which is called *action-related uncertainty*. In addition, we face another type of uncertainty related to the sampled clients in DP-DLB, called *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 the 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 (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: We design a phased elimination algorithm as in [19] 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 (e.g., 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 run at the central server. Next, we 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 $g(\pi) \triangleq \max_{x\in\mathcal{D}} \|x\|^2_{V(\pi)^{-1}}$. According to the analysis in [3, 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 [20]. By the Kiefer-Wolfowitz Theorem [21], one can find a distribution π^* minimizing $g(\cdot)$ with $g(\pi^*) = d$, and the support set[5] 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

[5]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.

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

1: **Input:** $\mathcal{D} \subseteq \mathbb{R}^d$, $\alpha, \beta \in (0,1)$, 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 $g(\pi_l) \triangleq \max_{x\in\mathcal{D}_l} \|x\|^2_{V(\pi_l)^{-1}} \leq 2\,d$ and $|\text{supp}(\pi_l)| \leq 4d\log\log d + 16$, where $V(\pi_l) \triangleq \sum_{x\in\mathcal{D}_l} \pi_l(x) xx^\top$
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 thePRIVATIZER $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$
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:

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

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**

$g(\pi_l) \leq 2\,d$ with $|\text{supp}(\pi_l)| \leq 4d\log\log d + 16$ (Line 4), which follows from [19, 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). Then, 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 detailed discussions in Section IV). 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 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 Section IV, with corresponding detailed instantiations of $\mathcal{R}, \mathcal{S}$, and $\mathcal{A}$.

Parameter estimation and action elimination (Lines 17–19): Using privately aggregated feedback $\tilde{y}_l$, the central server computes the least-square estimator $\tilde{\theta}_l$ (Line 17). Action elimination is based on the following confidence width:

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

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 (3) capture the action-related uncertainty, client-related uncertainty, and the added noise for privacy guarantees, respectively. 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).

Finally, we make two remarks about DP-DPE.

Remark 3.1: While a finite number of actions is assumed in this paper, one could extend it to the case with an infinite number of actions by using the covering argument [3, Lemma 20.1]. Specifically, when the action set $\mathcal{D} \subseteq \mathbb{R}^d$ is infinite, we can replace $\mathcal{D}$ with a finite set $\mathcal{D}_{\varepsilon_0} \subseteq \mathbb{R}^d$ with $|\mathcal{D}_{\varepsilon_0}| \leq (3/\varepsilon_0)^d$ such that for all $x \in \mathcal{D}$, there exists an $x' \in \mathcal{D}_{\varepsilon_0}$ with $\|x - x'\|_2 \leq \varepsilon_0$.

Remark 3.2: In Algorithm 1, we assume that $\mathcal{D}_l$ spans $\mathbb{R}^d$ such that matrices $V(\pi_l)$ and V_l are invertible. Then, one could find the near optimal design $\pi_l(\cdot)$ (Line 4) and compute the least-square estimator $\tilde{\theta}_l$ (Line 17). When $\mathcal{D}_l$ does not span $\mathbb{R}^d$, one can simply work in the smaller space $\text{span}(\mathcal{D}_l)$ [19].

IV. DP-DPE Under Different DP Models

In this section, we formalize DP models integrated with our DP-DLB formulation and provide concrete instantiations for the PRIVATIZER in DP-DPE according to three representative DP trust models: the central, local, and shuffle models.

A. DP-DPE Under the Central DP Model

In the central DP model, we assume that each client trusts the server, and hence, the server can collect clients' raw data (i.e., the local reward $\vec{y}_l^u$ in our case). The privacy guarantee is that any adversary with arbitrary auxiliary information cannot infer a particular client's data by observing the output of the server. To achieve this, the central DP model requires that the outputs of the server on two neighboring datasets differing in only one client are indistinguishable [15]. Before presenting the formal definition in our case, recall that DP-DPE (Algorithm 1) runs in phases, and that in each phase l, a set of *new* clients U_l participate in the global bandit learning by providing their feedback. Let[6] $\mathcal{U}_T \triangleq (U_l)_{l=1}^L \in \mathcal{U}^*$ be the sequence of all the participating clients in the total L phases (T rounds). We use $\mathcal{M}(\mathcal{U}_T) = (x_1, \ldots, x_T) \in \mathcal{D}^T$ to denote the sequence of actions chosen in T rounds by the central server. Intuitively, we are interested in a randomized algorithm such that the output $\mathcal{M}(\mathcal{U}_T)$ does not reveal "much" information about any particular client $u \in \mathcal{U}_T$. Formally, we have the following definition.

Definition 4.1. (Differential Privacy (DP)): For any $\varepsilon \geq 0$ and $\delta \in [0, 1]$, a DP-DPE instantiation is (ε, δ)-differentially private (or (ε, δ)-DP) if for every $\mathcal{U}_T, \mathcal{U}_T' \subseteq \mathcal{U}$ differing on a single client and for any subset of actions $Z \subseteq \mathcal{D}^T$,

$$\mathbb{P}[\mathcal{M}(\mathcal{U}_T) \in Z] \leq e^{\varepsilon} \mathbb{P}[\mathcal{M}(\mathcal{U}_T') \in Z] + \delta. \tag{4}$$

According to the post-processing property of DP (cf. Proposition 2.1 in [22]) and parallel-composition (thanks to the uniqueness of client sampling), it suffices to guarantee that the final analyzer $\mathcal{A}$ in $\mathcal{P}$ is (ε, δ)-DP. That is, for any phase l, the PRIVATIZER $\mathcal{P}$ is (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 $\mathcal{A}$:

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

To achieve this, we utilize the standard Gaussian mechanism at the server side to guarantee (ε, δ)-DP. Specifically, in each phase l, the participating clients send their average local rewards $\{\vec{y}_l^u\}_{u \in U_l}$ directly to the central server, and the central server adds Gaussian noise to the average local feedback (over clients) before estimating the global parameter. That is, in the central DP model, both $\mathcal{R}$ and $\mathcal{S}$ of the PRIVATIZER $\mathcal{P}$ are identity mapping while $\mathcal{A}$ adds Gaussian noise when computing the average. In this case, $\mathcal{P} = \mathcal{A}$, and the private aggregated feedback for the chosen actions in the l-th phase can be represented as

$$\begin{aligned} \tilde{y}_l &= \mathcal{P}\left(\{\vec{y}_l^u\}_{u \in U_l}\right) = \mathcal{A}\left(\{\vec{y}_l^u\}_{u \in U_l}\right) \\ &= \frac{1}{|U_l|} \sum_{u \in U_l} \vec{y}_l^u + (\gamma_1, \ldots, \gamma_{s_l}), \end{aligned} \tag{5}$$

[6]We use the superscript * to indicate that the length could be varying.

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 based on the ℓ_2 sensitivity of the average $\frac{1}{|U_l|} \sum_{u \in U_l} \vec{y}_l^u$. In the rest of the paper, we will continue to use s_l instead of $|\text{supp}(\pi_l)|$ to denote the number of actions chosen in the l-th phase for notational simplicity. With the above definition, we present the privacy guarantee of DP-DPE in the central DP model in Theorem 4.1.

Theorem 4.2: The DP-DPE instantiation using the PRIVATIZER in (5) with $\sigma_{nc} = \frac{2B\sqrt{2s_l \ln(1.25/\delta)}}{\varepsilon |U_l|}$ guarantees (ε, δ)-DP.

The relatively high trust model in the central DP is not always feasible in practice since some clients do not trust the server and are not willing to share any of their sensitive data. This motivates the introduction of a strictly stronger notion of privacy protection called the local DP [16], which is the main focus of the next subsection.

B. DP-DPE Under the Local DP Model

In the local DP model, any data sent by any client must already be private. In other words, even though an adversary can observe the data sent from a client to the server, the adversary cannot infer any sensitive information about the client. Mathematically, this requires a local randomizer $\mathcal{R}$ at each user's side to generate approximately indistinguishable outputs on any two different data inputs. In particular, let Y_u be the set of all possible values of the average local reward $\vec{y}_l^u$ for client u. Then, we have the following formal definition.

Definition 4.3. (Local Differential Privacy (LDP)): For any $\varepsilon \geq 0$ and $\delta \in [0, 1]$, a DP-DPE instantiation is (ε, δ)-local differentially private (or (ε, δ)-LDP) if for any client u, every two datasets $\vec{y}, \vec{y}' \in Y_u$ satisfies

$$\mathbb{P}[\mathcal{R}(\vec{y}) = o] \leq e^{\varepsilon} \mathbb{P}[\mathcal{R}(\vec{y}') = o] + \delta, \tag{6}$$

for every possible output $o \in \{\mathcal{R}(\vec{y}) | \vec{y} \in Y_u\}$.

That is, an instantiation of DP-DPE is (ε, δ)-LDP if the local randomizer $\mathcal{R}$ in $\mathcal{P}$ is (ε, δ)-DP. To this end, the randomizer $\mathcal{R}$ at each client employs a Gaussian mechanism, the shuffler $\mathcal{S}$ is a simple identity mapping, and the analyzer $\mathcal{A}$ at the server side conducts a simple averaging. Then, the overall output of the PRIVATIZER is the following:

$$\tilde{y}_l = \frac{1}{|U_l|} \sum_{u \in U_l} \mathcal{R}(\vec{y}_l^u) = \frac{1}{|U_l|} \sum_{u \in U_l} \left(\vec{y}_l^u + (\gamma_{u,1}, \ldots, \gamma_{u,s_l})\right), \tag{7}$$

where $\gamma_{u,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_{nl}^2)$, and the variance σ_{nl}^2 is based on the sensitivity of $\vec{y}_l^u$. With the above definition, we present the privacy guarantee of DP-DPE in the local DP model in Theorem 4.2.

Theorem 4.4: The DP-DPE instantiation using the PRIVATIZER in (7) with $\sigma_{nl} = \frac{2B\sqrt{2s_l \ln(1.25/\delta)}}{\varepsilon}$ guarantees (ε, δ)-LDP.

Although the local DP model offers a stronger privacy guarantee compared to the central DP model, it often comes at a price of the regret performance. As we will see, the regret performance of DP-DPE under the local DP model is much worse than that under the central DP model. Therefore, a fundamental question

is whether there is a PRIVATIZER for DP-DPE that can achieve the same regret as in the central DP PRIVATIZER while assuming similar trust model as in the local DP PRIVATIZER. This motivates us to consider a recently proposed *shuffle DP model* [17], [23].

C. DP-DPE Under the Shuffle DP Model

In the shuffle DP model, between the clients and the server, there exists a shuffler that permutes a batch of clients' randomized data before they are observed by the server so that the server cannot distinguish between two clients' data. Thus, an additional layer of randomness is introduced via shuffling, which can often be easily implemented using cryptographic primitives (e.g., mixnets) due to its simple operation [24]. Due to this, the clients now tend to trust the shuffler but still do not trust the central server as in the local DP model. This new trust model offers a possibility to achieve a better regret-privacy tradeoff. This is because the additional randomness of the shuffler creates a *privacy blanket* so that by adding much less random noise, each client can now hide her information in the crowd, i.e., privacy amplification by shuffling [25].

Formally, a standard one-round shuffle protocol consists of all the three parts: a (local) randomizer $\mathcal{R}$, a shuffler $\mathcal{S}$, and an analyzer $\mathcal{A}$. In this protocol, the clients trust the shuffler but not the analyzer. Hence, the privacy objective is to ensure that the outputs of the shuffler on two neighboring datasets are indistinguishable from the analyzer's point of view. Note that each client still does not send her raw data to the shuffler even though she trusts it. Due to this, a shuffle protocol often also offers a certain level of LDP guarantee.

In our case, the online learning procedure will proceed in multiple phases rather than a simple one-round computation. Thus, we need to guarantee that all the shuffled outputs are indistinguishable. To this end, we define the (composite) mechanism $\mathcal{M}_s(\mathcal{U}_T) \triangleq ((\mathcal{S}\circ\mathcal{R})(U_1), (\mathcal{S}\circ\mathcal{R})(U_2), \ldots, (\mathcal{S}\circ\mathcal{R})(U_L))$, where $(\mathcal{S}\circ\mathcal{R})(U_l) \triangleq \mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u\in U_l})$. We say a DP-DPE instantiation satisfies the shuffle differential privacy (SDP) if the composite mechanism $\mathcal{M}_s$ is DP, which leads to the following formal definition.

Definition 4.5. (Shuffle Differential Privacy (SDP)): For any $\varepsilon \geq 0$ and $\delta \in [0,1]$, a DP-DPE instantiation is (ε,δ)-shuffle differential privacy (or (ε,δ)-SDP) if for any pair $\mathcal{U}_T$ and $\mathcal{U}_T'$ that differ by one client, the following is satisfied for all $Z \subseteq Range(\mathcal{M}_s)$:

$$\mathbb{P}[\mathcal{M}_s(\mathcal{U}_T) \in Z] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}_s(\mathcal{U}_T') \in Z] + \delta. \tag{8}$$

Then, for any phase l, the PRIVATIZER $\mathcal{P}$ is (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 $\mathcal{S}\circ\mathcal{R}$:

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

We present the concrete pseudocode of $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{A}$ for the shuffle DP model PRIVATIZER $\mathcal{P}$ in Algorithm 2 (see Appendix A), which builds on the vector summation protocol recently proposed in [26]. Here, we provide a brief description of the process. Essentially, the noise added in the shuffle model PRIVATIZER relies on the upper bound of ℓ_2 norm of the input vectors. However, each component operates on each coordinate of the input vectors independently. Recall that the input of the shuffle model PRIVATIZER is $\{\vec{y}_l^u\}_{u\in U_l}$ and that each chosen action x corresponds to a coordinate in the s_l-dimentional vector. Consider the coordinate j_x corresponding to action x, and the entry $y_l^u(x)$ at client u. First, the local randomizer $\mathcal{R}$ encodes the input $y_l^u(x)$ via a fixed-point encoding scheme [17] and ensures privacy by injecting binomial noise. Specifically, given any scalar $w \in [0,1]$, it is first encoded as $\widehat{w} = \bar{w} + \gamma_1$ using an accuracy parameter $g \in \mathbb{N}$, where $\bar{w} = \lfloor wg \rfloor$ and $\gamma_1 \sim \texttt{Ber}(wg - \bar{w})$ is a Bernoulli random variable. Then, a binomial noise $\gamma_2 \sim \texttt{Bin}(b,p)$ is generated, where $b \in \mathbb{N}$ and $p \in (0,1)$ controls the level of the privacy noise. The output of the local randomizer for each coordinate is simply a collection of $g+b$ bits, where $\widehat{w}+\gamma_2$ bits are 1's and the rest are 0's. Combining these $g+b$ bits for each coordinate j_x for $x \in \text{supp}(\pi_l)$ yields the final outputs of the local randomizer $\mathcal{R}$ for the vector $\vec{y}_l^u$. Note that the output bits for each coordinate are marked with the coordinate index so that they will not be mixed up in the following procedures. After receiving the bits from all participating clients, the shuffler $\mathcal{S}$ simply permutes these bits uniformly at random and sends the output to the analyzer $\mathcal{A}$ at the central server. The analyzer $\mathcal{A}$ adds the received bits, removes the bias introduced by encoding and binomial noise (through simple shifting operations), and divides the result by $|U_l|$ for each coordinate. Finally, the analyzer $\mathcal{A}$ outputs a random s_l-dimensional vector $\tilde{y}_l$, whose expectation is the average of the input vectors. That is, $\mathbb{E}[\tilde{y}_l] = \frac{1}{|U_l|}\sum_{u\in U_l}\vec{y}_l^u$ (which is proven in our Appendix A.3). In the shuffle model PRIVATIZER, the three parameters g, b, and p need to be properly chosen according to the privacy requirement. Then, the final privately aggregated data is the following:

$$\tilde{y}_l = \mathcal{P}\left(\{\vec{y}_l^u\}_{u\in U_l}\right) = \mathcal{A}(\mathcal{S}(\{\mathcal{R}(\vec{y}_l^u)\}_{u\in U_l})). \tag{9}$$

With the above definition, we present the privacy guarantee of DP-DPE in the shuffle DP model in Theorem 4.3.

Theorem 4.6: For any $\varepsilon \in (0,15)$ and $\delta \in (0,1/2)$, the DP-DPE instantiation using the PRIVATIZER specified in Algorithm 2 guarantees (ε,δ)-SDP.

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 5.1.

Theorem 5.1 (DPE): Let $\beta = 1/(kT)$ and $\sigma_n = 0$ in Algorithm 1. Then, the non-private DP-DPE algorithm achieves the following expected regret:

$$\mathbb{E}[R(T)] = O(\sqrt{dT\log(kT)}) + O\left(\sigma T^{1-\alpha/2}\sqrt{\log(kT)}\right), \tag{10}$$

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

We present a proof sketch below and provide detailed proof in Appendix B.1.

Proof: We begin by showing a concentration inequality $P\{\langle\tilde{\theta}_l - \theta^*, x\rangle \geq W_l\} \leq 2\beta$, which indicates that in the l-th

phase, the estimation error for the global reward of each action is bounded by W_l *w.h.p.* Then, the optimal action stays in the active set the whole time *w.h.p.*, and the regret incurred by one pull is bounded by $4W_{l-1}$ in the l-th phase. Finally, summing up the regret over rounds in all phases yields the regret upper bound. The analysis of the communication cost is quite straightforward. In the l-th phase, only the local average reward of each chosen action in this phase is communicated, whose amount is bounded by $(4d \log\log d + 16)$ according to the near-G-optimal design [19, Proposition 3.7]. Hence, the communication cost is proportional to the total number of clients involved in the entire learning process. □

Remark 5.2: Theorem 5.1 gives a problem-independent regret upper bound for DPE. We can observe an obvious tradeoff between regret and communication cost, captured by α. While a larger α leads to a smaller regret, it incurs a larger communication cost. Setting $\alpha = 2/3$ gives $O(T^{2/3})$ for both regret and communication cost.

Remark 5.3 ((Sub-)optimality): Note that one natural lower bound for our setting is $\Omega(\sqrt{dT})$, the one for the standard linear bandits with finite arms [3], where there is no client-related uncertainty (i.e., $\sigma = 0$). In this setting, the upper bound derived in (10) matches the existing lower bound up to a logarithmic term. As to the general case with $\sigma > 0$, we can still see the (near)-optimality of our upper bound for the case with user-sampling parameter $\alpha > 1$. When sampling fewer users with $\alpha \in (0, 1)$, the second term of the regret upper bound in (10) that relies on α becomes dominant and cannot be ignored. However, the aforementioned lower bound $\Omega(\sqrt{dT})$ is derived under the standard linear bandit setting, which is irrelevant to the user sampling parameter α. Therefore, we leave it as our future work to close this gap between this natural lower bound and the derived (α-dependent) upper bound in (10).

In Theorem 5.2, 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 5.2: 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 $\sigma_{nc} = O(\frac{B\sqrt{d\ln(1/\delta)}}{\varepsilon|U_l|})$ in (5) for each phase l and $\sigma_n = 2\sigma_{nc}\sqrt{Sd}$ in (3);

(ii) *LDP-DPE:* Set $\sigma_{nl} = O(\frac{B\sqrt{d\ln(1/\delta)}}{\varepsilon})$ in (7) for each phase l and $\sigma_n = 2\sigma_{nl}\sqrt{Sd/|U_l|}$ in (3);

(iii) *SDP-DPE:* Set $\sigma_{ns} = O(\frac{B\sqrt{d}\ln(d/\delta)}{\varepsilon|U_l|})$ in (9) for each phase l and $\sigma_n = 2\sigma_{ns}\sqrt{Sd}$ in (3).

We provide the detailed proofs in Appendix B.2 and make the following remarks.

Remark 5.5 (Privacy "for-free"): Comparing the above results with Theorem 5.1 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 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 (3) and our choice of σ_n for different PRIVATIZERs. See more discussions on achieving privacy "for-free" in Section VII-A.

Remark 5.6 (Regret-privacy tradeoff): Consider the regret due to privacy protection by comparing the regret performance column in Table I of all the DP-DPE algorithms. We can see an additional term in regret performance associated with each DP-DPE algorithm. Specifically, while the local DP model ensures a stronger privacy guarantee compared to the central DP model, it introduces an additional regret of $\tilde{O}(T^{1-\alpha/2})$ compared to $\tilde{O}(T^{1-\alpha})$ 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 5.7 (Communication cost): Both CDP-DPE and LDP-DPE consume the same amount of communication resources as DPE, measured by the number of real numbers [14]. 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 [27], [28], and hence in this case, it is desirable to use SDP-DPE, which incurs a communication cost of $O(dT^{3\alpha/2})$ bits.

Remark 5.8 (Pure DP extension): While we use the Gaussian mechanism to ensure approximate DP (i.e., (ε, δ)-DP), we claim that our proposed scheme in this paper can be effectively integrated with the Laplace mechanism, which ensures a pure DP and achieves nearly the same regret performance. We provide how to modify the algorithm and derive the theoretical results for the Laplace mechanism in Appendix C.

VI. NUMERICAL RESULTS

In this section, we conduct simulations to evaluate 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

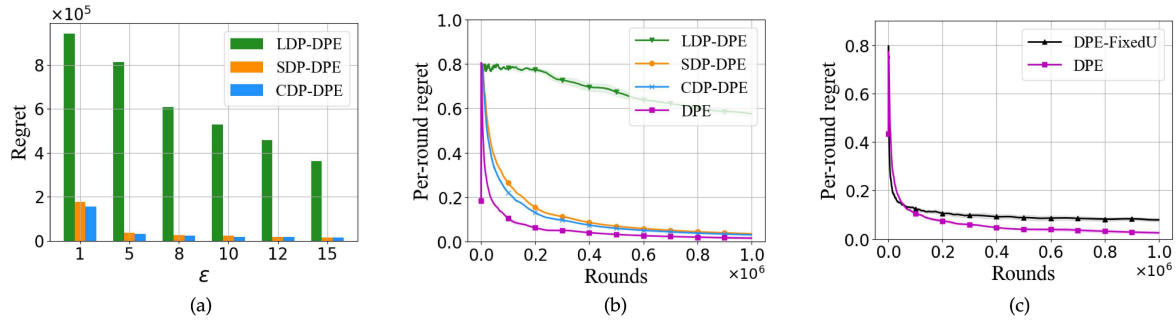

Fig. 2. Performance evaluation of DP-DPE. The shaded area indicates the standard deviation. (a) Final cumulative regret vs. the privacy budget ε. (b) Per-round regret vs. time with privacy parameters $\varepsilon = 10$ and $\delta = 0.1$. (c) Comparison between two non-private algorithms. Here, we choose the number of clients in DPE-FixedU to be $U = 97$ based on the calculation.

TABLE II
COMPARISON OF COMMUNICATION COST UNDER LINUCB AND PE WITH DIFFERENT VALUES OF α

Algorithms	DPE $\alpha = 0.5$	DPE $\alpha = 0.6$	DPE $\alpha = 0.7$	DPE $\alpha = 0.8$	DPE $\alpha = 0.9$	LinUCB	PE
Communication cost ($\times 10^4$)	0.70	0.81	1.05	1.69	3.27	5.00	5.00
# of participating users ($\times 10^4$)	0.04	0.10	0.23	0.55	1.34	5.00	5.00

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 $\varepsilon = 10$ and $\delta = 0.1$. 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.

Regarding the communication efficiency of our proposed algorithm, we also 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 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.

Finally, as discussed in Section VII-B, we also compare DPE with the the-state-of-the-art for standard linear bandit problem, i.e. LinUCB and PE, and present the regret comparison in Fig. 3 and communication and sample efficiency in Table II. The results show that DPE can achieve a regret close to that of (adapted) LinUCB and PE by adjusting sampling parameter α while always consuming less communication cost and involving fewer users.

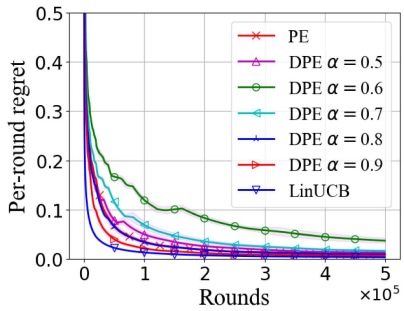

Fig. 3. LinUCB vs PE vs DPE with different values of α.

VII. DISCUSSIONS

A. On Achieving Privacy "for Free"

Following the remark on privacy "for-free" (Remark 5.3), in this section, we first study differentially private linear bandits and then draw an interesting connection between bandit online learning and supervised learning.

1) Differentially Private Linear Bandits: Motivated by the cellular configuration problem, we consider the distributed linear bandits with partial feedback in the main content and propose the DP-DPE algorithmic framework to address the newly introduced challenges. However, we highlight that our developed

techniques with minor modifications can also achieve similar results in terms of regret and privacy for the standard linear bandits, where there is no client-related uncertainty ($\sigma = 0$), i.e., $\theta_u = \theta^*$ in our notations. That is, we can design differentially private linear bandits where one can also achieve privacy "for free" in the central and shuffle DP models (similar to Remarks 5.3). This might be of independent interest to the bandit learning community. We provide the detailed description of differentially private linear bandits in Appendix D.

Remark 7.1: We can achieve the above "for-free" results because the sensitive information in linear bandits are only rewards, which is in sharp contrast to linear *contextual* bandits where both contexts and rewards need to be protected. In this case, the best known private regrets in the central, local and shuffle model are $\tilde{O}(\frac{\sqrt{T}}{\sqrt{\varepsilon}})$ [29], $\tilde{O}(\frac{T^{3/4}}{\sqrt{\varepsilon}})$ [30], and $\tilde{O}(\frac{T^{3/5}}{\varepsilon^{2/5}})$ [31], respectively.

2) Connection With Supervised Learning: In addition, we draw an interesting connection of our novel bandit online learning problem to private (distributed) supervised learning problems, through which we provide more intuition on why DP-DPE can achieve privacy "for free". In particular, we compare our problem with differentially private stochastic convex optimization (DP-SCO) [9], where the goal is to approximately minimize the population loss[7] over convex and Lipschitz loss functions given n *i.i.d.* d-dimensional samples from a population distribution while protecting privacy under different trust models. More specifically, via noisy stochastic gradient descent (SGD), the excess losses[8] in DP-SCO under various trust models are roughly as follows:

$$\text{Central \& Shuffle Model [8], [25]} : \widetilde{O}\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\varepsilon}\right), \quad (11)$$

$$\text{Local Model [32]} : \widetilde{O}\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{\sqrt{n}\varepsilon}\right). \quad (12)$$

Recall our main results in Table I and ignore all the logarithmic terms for clarity. Now, one can easily see that in both problems, privacy protection is achieved "for free" in the central and shuffle models, in the sense that the second term (i.e., the additional privacy-dependent term) is a lower-order term (with respect to n or T) compared to the first term (see (11) and Table I). On the other hand, under the much stronger local model, in both problems, the additional privacy-dependent term is of the same order as the first term. We tend to believe that the above interesting connection is not a coincidence. Rather, it provides us with a sharp insight into our introduced DP-DLB formulation. In particular, we know that the first term $1/\sqrt{n}$ in DP-SCO comes from standard concentration results, i.e., how independent samples approximate the true population parameter. Similarly, in our problem, the first term $\sqrt{d}T^{1-\alpha/2}$ comes from the concentration due to client sampling, which is used to approximate the true unknown population parameter θ^*. On the other hand, the second term in DP-SCO is privacy-dependent and comes from the average of noisy gradients. Similarly, in our problem, the second term is due to the average of the local reward vectors with added noise for preserving privacy.

In addition to these useful insights, we believe that this interesting connection also opens the door to a series of important future research directions, in which one can leverage recent advances in DP-SCO to improve our main results (dependence on d, communication efficiency, etc.).

B. Comparison With The-State-of-The-Art

Some perceptive readers might think reducing the model to a problem where each user u can observe *i.i.d.* rewards with mean $\langle \theta^*, x \rangle$ by treating $\langle \theta_u - \theta^*, x \rangle$ as an additional noise to η_t. In this case, we may solve our problem with the existing solutions to the traditional linear bandits. However, they exhibit the following significant limitations.

Note that the uncertainty introduced by the additional noise has to be addressed by sampling enough clients, e.g., one client per round. Considering DP, this problem essentially reduces to the differential private linear bandit (also discussed in our Section VII-A1) with a larger noise variance, where the same results in terms of regret (order-wise) and privacy can be achieved. However, one new user is sampled in each round to collect reward observation, which requires exactly T users in total to obtain the desired regret while ensuring the privacy guarantee. Instead, the DP-DPE framework in this work provides an approach where it collects feedback from multiple clients for the selected action in each round while each client serves for multiple rounds to maintain (or improve) sample efficiency. Specifically, it samples $\lceil 2^{\alpha l} \rceil$ clients for 2^l plays (rounds in the l-th phase), which is $O(T^\alpha)$ users in total. In addition, by only collecting feedback after preprocessing reward observations at the end of each phase, this carefully designed DP-DPE algorithmic framework reduces the communication cost from exactly T to $O(dT^\alpha)$. We have to mention that choosing $\alpha < 1$, however, will incur a larger privacy cost (see Table I). Therefore, there is a tradeoff between the regret penalty due to privacy and the communication and sampling efficiency, which can be balanced by tuning α properly. Meanwhile, we run simulations of the non-private algorithms: DPE, LinUCB in [5], and PE in [3], and present the results in Fig. 3 and Table II. The results show that DPE can achieve similar regret performance (by adjusting parameter α) to LinUCB and PE while improving user-sampling efficiency and communication efficiency significantly for each $\alpha \in (0, 1)$.

C. Extensions to Non-Linear Bandits

In this work, we study the problem of global reward maximization with distribution feedback in the stochastic linear bandit model where *direct reward observations* are not available. Note that the same challenge (i.e., no direct /partial reward feedback) could also exist in other general bandit models, e.g., generalized linear bandits and kernelized bandits. We believe our algorithmic framework incorporating different DP models can

[7]The population loss for a solution w is given by $\mathcal{L}(w) \triangleq \mathbb{E}_{z\in\mathcal{D}}[l(w, z)]$, where w is the chosen solution (e.g., weights of a classifier), z is a testing sample from the population distribution $\mathcal{D}$, and l is a convex loss function of w.

[8]The excess loss measures the gap between the chosen solution and the optimal solution in terms of the population loss. See [32].

be extended through careful accommodation for the parametric generalized linear bandits. Specifically, one may refer to [34] to update the estimator of $\tilde{\theta}_l$ and the confidence width W_l for the upper/lower confidence bound (UCB/LCB) of each active arm used in the elimination rule in any particular phase l. However, our algorithmic framework may not be extended directly to the non-parametric kernelized bandits. We study the new challenges and present the solutions in our recent paper [8].

VIII. Related Work

Bandit models and their variants have proven to be useful for many real-world applications and have been extensively studied (see, e.g., [3], [18], [35] and references therein). This paper, different from most existing studies with exact reward feedback available, considers a new linear bandit setting where the agent has to learn with partial distributed feedback. While this setting shares some similarities with distributed bandits, federated bandits, and multi-agent cooperative bandits, our motivation and model are very different from theirs, which leads to different regret definitions (global regret vs. group regret; see Section II) and algorithmic solutions. In the following, we discuss the most relevant work in the literature and highlight the key differences.

Linear bandits: Different from the standard stochastic multi-armed bandits (MAB) model with independent arms, the linear bandit model captures the correlation among actions via an unknown parameter [4], [36], [37]. The best-known regret upper bound for stochastic linear bandits is $O(d\sqrt{T\log(T)})$ in [4], which holds for an almost arbitrary, even infinite, bounded subset of a finite-dimensional vector space. For a special setting where the set of actions is finite and does not change over time, it is shown in [3] that a *phased elimination with G-optimal exploration* algorithm guarantees a regret upper bounded by $O(\sqrt{dT\log(kT)})$. This new bound is better by a factor of $\sqrt{d}$, which deserves the effort when $d \geq \log(k)$. However, none of these studies consider the scenario where an action influences a large population and the exact reward feedback is unavailable, which is a key challenge in our problem. Note that the linear bandits model we consider is different from the contextual linear bandits in [5], [38] where the parameter is not shared by actions (although assuming linear reward function), and thus, the actions are not correlated through the parameter.

Differentially private online learning and bandits: Since proposed in [15], differential privacy (DP) has become the *de facto* privacy preserving model in many applications, including online learning [39] and bandits problems [40]. Specifically, in [41], [42], [43], MAB has been studied in the central, local, and shuffle DP models, respectively. We refer interested readers to [44] for state-of-the-art results on private MABs under all three models. In [29], the authors explore DP in contextual linear bandits and introduce joint DP as ensuring the standard DP incurs a linear regret. As stronger privacy protection, local DP is also studied for contextual linear bandits [30] and Bayesian optimization [45]. Very recently, shuffle model for linear *contextual* bandits have been studied in [31]. As already highlighted in Remark 7.1, the additional protection of context information leads to a higher cost of privacy compared to linear bandits considered in our paper, where only rewards are private information. One concurrent work [46] with our conference paper study the standard linear bandits in all the three DP models as ours while ensuring pure DP. However, different from the unified algorithmic framework in this paper, their algorithms in different DP models are independently designed, and their shuffle model requires the shuffler to do more than shuffling.

Distributed bandits: Another line of related work is on multi-agent collaborative learning in the distributed bandits setting [14], [47], [48], [49], [50], [51]. The most relevant work to ours is the distributed linear bandit problem studied in [14]. 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 data privacy protection.

Federated bandits: Federated learning (FL) has received substantial attention since its introduction in [52]. The main idea of FL is to enable collaborative learning among heterogeneous devices while preserving data privacy. Very recently, bandit problems have also been studied in the federated setting, where the underlying problem is a bandit one, including *federated multi-armed bandits* [53], [54], [55], *federated linear bandits* [56], [57], and *federated Bayesian optimization* [58], [59]. Among all the above work, the two most relevant studies are [57] and [56]. While they both consider the case where all heterogeneous users share the same unknown parameter with heterogeneous decision sets, in our problem setting, the users have heterogeneous unknown local parameters.

In addition to the differences in model and problem formulation, we also highlight our main technical contributions compared to these works in the following. First, when aggregating users' data for learning the global parameter, we protect users' data privacy using rigorous differential privacy guarantees, which, however, is not considered in [14] or [57]. Besides, the work [57] did not consider the correlation among the actions, which is captured by a common linear parameter in our setting. However, they consider a linear reward for contextual bandits while still studying multi-armed bandits with independent actions, each of which is associated with a distinct parameter vector. While DP is also employed to protect users' data privacy in [56],[9] they require that both the Gram matrix of actions (of size $O(d^2)$) and reward vectors (of size $O(d)$) be periodically communicated using some DP mechanisms (e.g., the Gaussian mechanism). Instead, in our algorithm, only private average local reward for the chosen actions (of size $O(d\log\log d)$) would be communicated in each phase. Moreover, while they only consider a variant of the central DP model, our DP-DPE solution provides a unified algorithmic learning framework, which can be instantiated with different DP models. Specifically, DP-DPE with the shuffle model enables us to achieve a finer

[9] As shown in a recent work [60], both the privacy guarantee and regret bounds in [56] have gaps.

regret-communication-privacy tradeoff (see Table I). That is, not only can it achieve nearly the same regret performance as the central model (yet without trusting the central server), but it requires the users to report feedback in bits only throughout the learning process.

Recently, we also extended our setup to the non-linear case by considering kernelized bandits [8].

Despite the above work regarding federated bandits, one may wonder whether we can follow the idea of federated learning to share clients' locally learned model parameters only. This way, one can avoid sharing raw data, which is another way of protecting clients' data privacy. However, we argue that the additional benefit is marginal. On the one hand, by employing different DP mechanisms, our proposed DP-DPE algorithms already ensure provable privacy guarantees. On the other hand, the communication cost of transmitting the (private) average rewards is nearly the same as that of transmitting the local model parameters. Specifically, in each phase, a client in our DP-DPE algorithm needs to send a $|\text{supp}(\pi_l)|$-dimensional vector in DP-DPE, compared to a d-dimensional vector when sending the local model parameters. Therefore, the difference is marginal since we have $|\text{supp}(\pi_l)| \leq 4d \log\log d + 16$.

Reinforcement learning: Note that reinforcement learning (RL) [61] is a generalization of bandits with a distinct new feature – the agent's actions not only yield immediate rewards but also influence the environment's future state(s). In other words, bandits is a special and simple case of RL where the horizon length of each episode is one, and hence, the action will not impact the state for the next step as the episode just restarts. In this sense, our study in bandits (dealing with a stateless environment) could shed light on distributed RL, including efficient communication design and differentially private algorithmic framework design, which might be of independent interest to the RL community.

IX. Conclusion

In this article, we studied a new bandit learning problem where it is often difficult, if not impossible, to collect exact reward feedback. To address it, we proposed a differentially private distributed linear bandits formulation, where the learning agent samples clients and interacts with them by iteratively aggregating 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. Moreover, our current privacy guarantee under the shuffle model is only approximated DP. One promising future direction is to explore pure DP in the shuffle model by building upon the recent advance in MAB [44]. Finally, our work also raises several interesting questions that are worth investigating. For example, can we further improve communication efficiency by using advanced shuffle protocols? Can we generalize our formulation to studying reinforcement learning problems?

References

[1] F. Li, X. Zhou, and B. Ji, "Differentially private linear bandits with partial distributed feedback," in *Proc. IEEE 20th Int. Symp. Model. Optim. Mobile Ad hoc Wireless Netw.*, 2022, pp. 41–48.

[2] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[3] T. Lattimore and C. Szepesvári, *Bandit Algorithms*.Cambridge, U.K.: Cambridge Univ. Press, 2020.

[4] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2312–2320.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[6] A. Mahimkar, A. Sivakumar, Z. Ge, S. Pathak, and K. Biswas, "Auric: Using data-driven recommendation to automatically generate cellular configuration," in *Proc. Proc. ACM SIGCOMM Conf.*, 2021, pp. 807–820.

[7] D. Bouneffouf and I. Rish, "A survey on practical applications of multi-armed and contextual bandits," *IEEE Congr. Evol. Comput.*, 2020, pp. 1–8.

[8] F. Li, X. Zhou, and B. Ji, "(Private) kernelized bandits with distributed biased feedback," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 1, pp. 1–47, Mar. 2023, doi: 10.1145/3579318.

[9] R. Bassily, V. Feldman, K. Talwar, and A. Thakurta, "Private stochastic convex optimization with optimal rates," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 11282–11291.

[10] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," in *Proc. NIPS 2017 Workshop: Mach. Learn Phone Consum. Devices*, 2017.

[11] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2521–2529.

[12] N. Agarwal and K. Singh, "The price of differential privacy for online learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 32–40.

[13] P. Bühlmann and S. Van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*.Berlin, Germany: Springer, 2011.

[14] Y. Wang, J. Hu, X. Chen, and L. Wang, "Distributed bandit learning: Near-optimal regret with efficient communication," in *Proc. 18th Int. Conf. Learn. Representations*, 2020, pp. 1–31.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory cryptogr. Conf.*, 2006, pp. 265–284.

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

[17] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2019, pp. 375–403.

[18] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, pp. 1–122, 2012.

[19] T. Lattimore, C. Szepesvari, and G. Weisz, "Learning with good feature representations in bandits and in RL with a generative model," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5662–5670.

[20] F. Pukelsheim, *Optimal Design of Experiments*.Philadelphia, PA, USA: SIAM, 2006.

[21] J. Kiefer and J. Wolfowitz, "The equivalence of two extremum problems," *Can. J. Math.*, vol. 12, pp. 363–366, 1960.

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

[23] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proc. 30th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2019, pp. 2468–2479.

[24] A. Bittau et al., "Prochlo: Strong privacy for analytics in the crowd," in *Proc. 26th Symp. Operating Syst. Princ.*, 2017, pp. 441–459.

[25] E. Garcelon, K. Chaudhuri, V. Perchet, and M. Pirotta, "Privacy amplification via shuffling for linear contextual bandits," in *Proc. Int. Conf. Algorithmic Learn. Theory*2022, pp. 381–407.

[26] A. Cheu, M. Joseph, J. Mao, and B. Peng, "Shuffle private stochastic convex optimization," in *Proc. 10th Int. Conf. Learn. Representations*, 2022, pp. 1–28.
[27] C. L. Canonne, G. Kamath, and T. Steinke, "The discrete Gaussian for differential privacy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15676–15688.
[28] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete Gaussian mechanism for federated learning with secure aggregation," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5201–5212.
[29] R. Shariff and O. Sheffet, "Differentially private contextual linear bandits," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 4296–4306.
[30] K. Zheng, T. Cai, W. Huang, Z. Li, and L. Wang, "Locally differentially private (contextual) bandits learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12300–12310.
[31] S. R. Chowdhury and X. Zhou, "Shuffle private linear contextual bandits," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 3984–4009.
[32] F. Li, X. Zhou, and B. Ji, "Differentially private linear bandits with partial distributed feedback," in *Proc. IEEE 20th Int. Symp. Model. Optim. Mobile, Ad hoc, Wireless Netw.*, 2022, pp. 41–48.
[33] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *J. Amer. Stat. Assoc.*, vol. 113, no. 521, pp. 182–201, 2018.
[34] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, "Parametric bandits: The generalized linear case," in *Proc. Adv. Neural Inf. Process. Syst.*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23, 2010, pp. 586–594.
[35] A. Slivkins, "Introduction to multi-armed bandits," *Found. Trends Mach. Learn.*, vol. 12, pp. 1–286, 2019.
[36] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proc. 21st Annu. Conf. Learn. Theory*, 2008, pp. 355–366.
[37] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. Operations Res.*, vol. 35, no. 2, pp. 395–411, 2010.
[38] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 208–214.
[39] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Proc. Conf. Learn. Theory*, vol. 23, 2012, pp. 24.1–24.34.
[40] N. Mishra and A. Thakurta, "(Nearly) optimal differentially private stochastic multi-arm bandits," in *Proc. 31st Conf. Uncertainty Artif. Intell.*, 2015, pp. 592–601.
[41] A. C. Tossou and C. Dimitrakakis, "Algorithms for differentially private multi-armed bandits," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2087–2093.
[42] W. Ren, X. Zhou, J. Liu, and N. B. Shroff, "Multi-armed bandits with local differential privacy," 2020, *arXiv:2007.03121*.
[43] J. Tenenbaum, H. Kaplan, Y. Mansour, and U. Stemmer, "Differentially private multi-armed bandits in the shuffle model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 24956–24967.
[44] S. R. Chowdhury and X. Zhou, "Distributed differential privacy in multi-armed bandits," in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–34.
[45] X. Zhou and J. Tan, "Local differential privacy for Bayesian optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11152–11159.
[46] O. A. Hanna, A. M. Girgis, C. Fragouli, and S. Diggavi, "Differentially private stochastic linear bandits: (almost) for free," 2022, *arXiv:2207.03445*.
[47] M. Agarwal, V. Aggarwal, and K. Azizzadenesheli, "Multi-agent multi-armed bandits with limited communication," *J. Mach. Learn. Res.*, vol. 23, pp. 9529–9552, 2021.
[48] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, "Decentralized cooperative stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 4529–4540.
[49] A. Dubey et al., "Cooperative multi-agent bandits with heavy tails," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2730–2739.
[50] N. Cesa-Bianchi, C. Gentile, Y. Mansour, and A. Minora, "Delay and cooperation in nonstochastic bandits," in *Proc. Conf. Learn. Theory*, 2016, pp. 605–622.
[51] A. Dubey et al., "Kernel methods for cooperative multi-agent contextual bandits," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2740–2750.
[52] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[53] C. Shi and C. Shen, "Federated multi-armed bandits," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 9603–9610.
[54] C. Shi, C. Shen, and J. Yang, "Federated multi-armed bandits with personalization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2917–2925.
[55] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, "Federated bandit: A gossiping approach," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 1, pp. 1–29, 2021.
[56] A. Dubey and A. Pentland, "Differentially-private federated linear bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6003–6014.
[57] R. Huang, W. Wu, J. Yang, and C. Shen, "Federated linear contextual bandits," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 27057–27068.
[58] Z. Dai, K. H. Low, and P. Jaillet, "Federated Bayesian optimization via thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9687–9699.
[59] Z. Dai, B. K. H. Low, and P. Jaillet, "Differentially private federated Bayesian optimization with distributed exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 9125–9139.
[60] X. Zhou and S. R. Chowdhury, "On differentially private federated linear contextual bandits," in *Proc. 12th Int. Conf. Learn. Representations*, 2024.
[61] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, "Reinforcement learning: Theory and algorithms," CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep. 12, 2019.

Fengjiao Li (Member, IEEE) received the B.E. degree in electronics and communications from the Taiyuan University of Technology, Taiyuan, China, in 2012, the M.E. degrees in electronics and communications from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015, and the Ph.D. degree in computer science and applications from Virginia Tech, Blacksburg, VA, USA, in 2022. She is currently an Assistant Professor with Shanxi University, Taiyuan. Her research interests include online learning, networking, modeling, optimization, and analysis.

Xingyu Zhou (Senior Member, IEEE) received the B.E. degree (with Highest Hons.) in electrical engineering from the Beijing University of Posts and Communications, Beijing, China, in 2012, the M.E. degree (with Highest Hons.) in electrical engineering from Tsinghua University, Beijing, in 2015, and the Ph.D. degree in Electronics and Communications Engineering from The Ohio State University, Columbus, OH, USA, in 2020, with Presidential Fellowship. He is currently an Assistant Professor of Electronics and Communications Engineering with Wayne State University, Detroit, MI, USA. His research interest include trustworthy data-driven decision-making, with a focus on private, fair, and robust bandits and reinforcement learning.

Bo Ji (Senior Member, IEEE) received the B.E. and M.E. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from the Ohio State University, Columbus, OH, USA, in 2012. He is currently an Associate Professor of computer science and a College of Engineering Faculty Fellow with Virginia Tech, Blacksburg, VA, USA. Before joining Virginia Tech, he was an Associate/Assistant Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA, from 2014 to 2020. From 2013 to 2014, he was also a Senior Member of the Technical Staff with AT&T Labs, San Ramon, CA, USA. His research interests include the intersections of networking, machine learning, security and privacy, and spatial computing.