Differentially Private Stochastic Linear Bandits: (Almost) for Free

Osama Hanna*†, Antonious M. Girgis*‡, Christina Fragouli, and Suhas Diggavi
University of California, Los Angeles, USA.
Email:{ohanna, amgirgis, christina.fragouli, suhasdiggavi}@ucla.edu.

In this paper, we propose differentially private algorithms for the problem of stochastic linear bandits in the central, local and shuffled models. In the central model, we achieve almost the same regret as the optimal non-private algorithms, which means we get privacy for free. In particular, we achieve a regret of $\tilde{O}(\sqrt{T}+\frac{1}{\varepsilon})$ matching the known lower bound for private linear bandits, while the best previously known algorithm achieves $\tilde{O}(\frac{1}{\varepsilon}\sqrt{T})$. In the local case, we achieve a regret of $\tilde{O}(\frac{1}{\varepsilon}\sqrt{T})$ which matches the non-private regret for constant ε, but suffers a regret penalty when ε is small. In the shuffled model, we also achieve regret of $\tilde{O}(\sqrt{T}+\frac{1}{\varepsilon})$ while the best previously known algorithm suffers a regret of $\tilde{O}(\frac{1}{\varepsilon}T^{3/5})$. Our numerical evaluation validates our theoretical results. Our results generalize for contextual linear bandits with known context distributions.

I. Introduction

Stochastic linear bandits offer a sequential decision framework where a learner interacts with an environment over rounds, and decides what is the optimal (from a potentially infinite set) action to play to achieve the best possible reward (minimize her regret). In particular, at each round, the learner may take into account all past rewards and actions to decide the next action to play, and in return receive a new reward. This model has been widely adopted both in theory but also in a number of applications, including recommendation systems, health, online education, and resource allocation [1]–[4]. Motivated by the fact that many of these applications are privacy-sensitive, in this paper we explore what is the performance in terms of regret we can achieve, if we are constrained to use a privacy-preserving stochastic linear bandit algorithm.

In particular, in this paper we aim to design algorithms that preserve the privacy of the rewards, from an adversary that can observe all actions that the learner plays. For example, the central learner may make restaurant recommendations to mobile devices, may regulate the operation of on-body sensors in senior living communities, may decide what educational exercises to provide to students, or what jobs to allocate to workers. The actions the clients play - what restaurant is visited, which sensor is activated, what is the exercise solved, what is the job performed - may be naturally visible especially in public environments. What we care to protect are the rewards, that may capture private information, such as personal preferences in recommendation systems, health indices in online health, performance in online education, and income gained in resource allocation. Our goal is to design algorithms that preserve the privacy of the rewards, while still (almost) achieve the same regret as the traditional algorithms that do not take privacy into consideration.

We do so for three different setups, depicted in Figure 1, in each case measuring the privacy using Differential Privacy (DP) measures [5], [6]. In the **central DP model**, the learner is a trusted server. The adversary observes the decisions of the trusted server. The server employs a DP mechanism on aggregates of the reward realizations she collects, to ensure that the actions do not reveal information on the rewards. In the **local DP model**, the learner is an untrusted server, where the adversary (including the learner) can access the individual private rewards of the clients. The clients provide privatized rewards to the server, who then uses this noisy input to decide her next actions. In the **shuffled model**, the learner is still an untrusted server, but now a trusted node, that can act as a relay in the communication between the clients and the server, serves as a shuffler, and can randomly permute the privatized rewards before making them available to the server. A shuffler offers a privacy-amplification mechanism that has recently become popular in the literature, as it is easy to implement (simply takes a set of inputs and randomly permutes them), and may enable better privacy-regret performance [7]–[11].

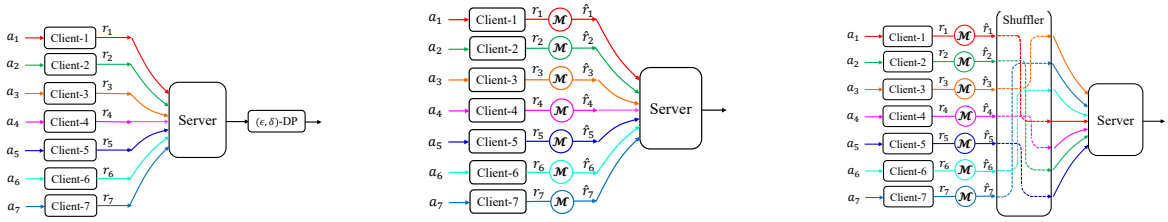
Our main contributions are as follows.

- For the **central DP model**, we design an algorithm that guarantees ε-DP (see Definition 1 in Section II) and achieves regret that matches existing lower bounds. In particular, over T rounds, it achieves regret $R_T = O\left(\sqrt{T\log T}+\frac{\log^2 T}{\varepsilon}\right)$ w.h.p., which is optimal within a $\log T$ factor: a lower bound of $O(\sqrt{T})$ is proven in [12] for non-private linear bandits, while a lower bound of $O(\frac{\log T}{\varepsilon})$ is shown in [13] for ε-DP linear bandits. Note that for $\varepsilon \approx 1$ (perhaps the most common case) the dominant term $O(\sqrt{T\log T})$ matches the regret of the best known algorithms for the non-private case (eg., LinUCB [12],

*The first and second authors made equal contribution.
† Now at Meta. ‡ Now at Google.

The work of Suhas Diggavi and Antonious M. Girgis was supported in part by NSF grants 2139304, 2007714 and 2146838. Antonious M. Girgis was also supported in part by UCLA Amazon Science Hub fellowship. The work of Christina Fragouli and Osama Hanna was supported in part by NSF under Grant 2007714 and Grant 2221871; in part by DARPA under Grant HR00112190130; and in part by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes Appendices for proofs and discussions referenced in the paper. Contact {ohanna,amgirgis}@ucla.edu for further questions about this work.



(a) Central model. (b) Local model. (c) Shuffled model.

Fig. 1: In case (a) the server is trusted, and we ensure that the publicly observable actions maintain privacy of the rewards. In (b) and (c) we maintain privacy from an untrusted server.

[14]), and hence, we get privacy for free.

• For the **local DP model**, we design an algorithm that guarantees ε_0-LDP (see Definition 2 in Section II) and achieves regret $R_T = \mathcal{O}\left(\sqrt{T\log(T)}/\varepsilon_0\right)$ w.h.p.; this regret matches the non-private regret for constant ε_0, but suffers a regret penalty when ε_0 is small. Although our algorithm does not improve the regret order as compared to the best-known algorithm for private (contextual) linear bandits in [15], it offers an alternative approach that serves as a foundation for the shuffled case.

• For the **shuffled model**, we leverage the help of a trusted shuffler to ensure both that the output of each client satisfies ε_0-LDP and that the output of the secure shuffler satisfies ε-DP requirements. Our algorithm achieves regret $R_T = \mathcal{O}\left(\sqrt{T\log(T)} + \frac{\log(T)}{\varepsilon}\right)$ w.h.p. that matches the regret of the best non-private algorithms, same as the central model. Furthermore, our algorithm outperforms the best known algorithm for private (contextual) linear bandits in [16], [17] that use shuffling.

Our results are summarized in Table I, where we also provide known results in the literature (see also discussion next). Our work is the first to study DP stochastic linear bandits. We provide order optimal algorithms for stochastic linear bandits under central, local, and shuffled DP models. Furthermore, we extend our results to DP contextual bandits with known context distribution achieving large improvements in regret bounds over state-of-the-art schemes in the central, local, and shuffled models[1]. We believe that this idea opens new techniques/directions to design private algorithms for contextual bandits. Our algorithms are simple and solve important problems: linear bandits, unstructured bandits with finitely many arms (a special case of linear bandits), and contextual linear bandits with known context distribution. In particular, the DP unstructured bandit problem has rich literature [18]–[20].

Our Work vs. Related Work. Differential Privacy (DP) algorithms have been proposed for the generic multi-armed bandits (MAB) problems [18]–[20], yet these algorithms would not work well for linear bandits, as linear bandits allow for an infinite set of actions while generic MAB have a regret that increases with the number of actions. Closer to ours is work on DP for contextual linear bandits [13], [15], [16], [21]; indeed, linear bandits can be viewed as (a special case of) contextual linear bandit setup with a single context. The work in [13] considers contextual linear bandits with DP and shows that linear regret is unavoidable. Instead, the work considers a weaker notion of privacy, JDP (joint differential privacy), in a centralized setting and proposes an algorithm that achieves a regret of $\tilde{O}(\sqrt{T}/\varepsilon)$. This does not match the best known lower bound for the centralized setting of $\Omega(\sqrt{T} + \log(T)/\varepsilon)$ [13]. Our work considers the stronger DP notion and achieves the lower bound of $\Omega(\sqrt{T} + \log(T)/\varepsilon)$ up to logarithmic factors for the special case of stochastic linear bandits. Recent work shows that contextual linear bandits can be reduced to stochastic linear bandits if the context distribution is known [22], which is the case for many application [22]. This implies direct generalizations of our algorithms to contextual linear bandits with DP and known context distribution without affecting the regret bounds. The work in [21] considers contextual linear bandits with LDP, where the contexts can be adversarial. The work proposes an algorithm that achieves a regret of $\tilde{O}(T^{3/4}/\varepsilon_0)$ and conjectures that the regret is optimal up to a logarithmic factor. The authors in [15] consider a special case, where the contexts are generated from a distribution, and propose a method that achieves a regret of $\tilde{O}(\sqrt{T}/\varepsilon_0)$ under certain assumptions on the context distribution. Our algorithm for the local model achieves the same regret order using an alternative method. The works in [16], [17] consider contextual linear bandits in the shuffled model where the best-known algorithm achieves a regret of $\tilde{O}(T^{3/5})$. Our proposed algorithms achieve a regret of $\tilde{O}(\sqrt{T} + 1/\varepsilon)$, matching the information-theoretic lower bound in [13], for stochastic linear bandits in the shuffled model. A summary of the best results for DP contextual linear bandits and our results is presented in Table I.

We mention two works in the literature studying the DP stochastic linear bandits problem, which are related to our work. After our initial posting of the paper on arxiv [23], and the completion of our work, we recently found two works [24], [25] which are closely related. The work in [24], which was published after our paper [23], proposed DP mechanisms for stochastic linear bandits using a similar approach to the batched algorithm. The main difference between their schemes and our proposed schemes is that the work in [24] focuses on designing communication-efficient schemes for DP stochastic linear bandits. This has some relationship with our results but is quite different. In particular, [24] assumes different clients with different bandit parameters that deviate from $\theta_\star$ by a

[1]This is the reason why we compare against contextual bandit work.

zero mean noise. To handle this, a large number of clients is sampled at each time slot. To get its regret result of a form similar to ours (in a different setting) [24] needs to sample $\Omega(T)$ clients, and hence, observe $\Omega(T)$ rewards, in some time slots. On the other hand, our work considers a setup where we can observe a single reward at each time slot. As the setup is different, the corresponding privacy guarantees in [24] are also not applicable to our setup. Furthermore, they consider the Gaussian mechanism to privatize the rewards that gives *approximate DP* guarantees. They need to do this as they have clients with different parameters and need to average, and they use the infinite divisibility of Gaussians for their privacy analysis, which is not true for Laplace mechanisms. On the other hand, we consider the Laplace mechanism to provide pure DP guarantees which is a stronger DP notion.

The work in [25], which was published concurrently to our work [23], primarily focuses on deriving lower bounds for differentially private contextual bandits in the central DP model, matching our upper bound in the central case and thereby showing the optimality of our scheme. Moreover, our contributions go well beyond the central DP model to include local DP and shuffled DP models as well.

Paper organization. We present the problem formulation in Section II. We design and analyze privacy-preserving linear bandit algorithms for the central model in Section III, for the local model in Section IV and for the shuffled model in Section V. We provide numerical results in Section VI.

II. Notation and Problem Formulation

Stochastic Linear Bandits. In stochastic linear bandits a learner interacts with clients over T rounds by taking a sequence of decisions and receiving rewards. In particular, at each round $t \in [T]$, the learner plays an action a_t from a set $\mathcal{A} \subset \mathbb{R}^d$ and receives a reward $r_t \in \mathbb{R}$. The reward r_t is a noisy linear function of the action, i.e., $r_t = \langle \theta_*, a_t \rangle + \eta_t$, where $\langle . \rangle$ denotes inner product, η_t is an independent zero-mean noise and $\theta_* \in \mathbb{R}^d$ is an unknown parameter vector. The goal of the learner is to minimize the total regret over the T rounds, which is calculated as:

$$R_T = T \max_{a \in \mathcal{A}} \langle \theta_*, a \rangle - \sum_{t=1}^{T} \langle \theta_*, a_t \rangle. \tag{1}$$

The regret captures the difference between the reward for the optimal action and the rewards for the actions chosen by the learner. The basic approach in all algorithms is to play actions that enable the learner to learn θ_* well enough to identify a (near) optimal action. The best known algorithms (for example, LinUCB [12], [14]) achieve a regret of order $O(\sqrt{T \log T})$, which is the best we can hope for (matches existing lower bounds [12]).

Contextual Linear Bandits. In contextual bandits, the learner observes the context of the client at time t, c_t, plays an action $a_t \in \mathcal{A}$, and receives a reward $r_t = \langle \theta_\star, \phi(a_t, c_t) \rangle + \eta_t$, where ϕ is a known feature map and η_t is noise. In this case the regret R_T is defined as $R_T = \sum_{t=1}^{T} \max_{a \in \mathcal{A}} \langle \theta_*, \phi(a, c_t) \rangle - \langle \theta_*, \phi(a_t, c_t) \rangle$. Equivalently, contextual linear bandits can be seen as linear bandits with action set that changes over time $\mathcal{A}_t = \{\phi(a, c_t) | a \in \mathcal{A}\}$.

In this paper, we make the following standard assumptions (see, e.g., [13], [14]).

Assumption 1. We consider stochastic linear bandits with:
1. Sub-Gaussian noise: $\mathbb{E}[\eta_{t+1} | \mathcal{F}_t] = 0$ and $\mathbb{E}[\exp(\lambda \eta_{t+1}) | \mathcal{F}_t] \leq \exp(\frac{\lambda^2}{2}) \forall \lambda \in \mathbb{R}$, where $\mathcal{F}_t = \sigma(a_1, r_1, ..., a_t, r_t)$ is the σ-field summarizing the information available before round t.
2. Bounded actions, unknown parameter, and rewards: $\|a\|_2 \leq 1 \ \forall a \in \mathcal{A}$, $\|\theta_*\|_2 \leq 1$ and $|r_t| \leq 1$.

Privacy Goal and Measures. Our goal in this paper is to achieve the minimum possible regret in (1) while preserving privacy of the rewards $\{r_t\}_{t \in [T]}$ (as discussed in Section I the rewards can represent sensitive information of the clients). To measure privacy, we use the popular central and local differential privacy definitions that we provide for completeness next. For simplicity, we assume that a different client plays each action (e.g., visits a recommended restaurant).

Differential Privacy (DP). We say that two sequences of rewards $\mathcal{R} = (r_1, \ldots, r_T)$ and $\mathcal{R}' = (r'_1, \ldots, r'_T)$ are neighboring if they differ in a single reward, i.e., there is a round $t \in [T]$ such that $r_t \neq r'_t$, but $r_j = r'_j$ for all $j \neq t$. To preserve privacy, we use a randomized mechanism $\mathcal{M}$ designed for stochastic linear bandits, that observes rewards and outputs publicly observable actions.

Definition 1. (Central DP [5], [6]): A randomized mechanism $\mathcal{M}$ for stochastic linear bandits is said to be (ε, δ) Differentially Private ((ε, δ)-DP) if for any two neighboring sequences of rewards $\mathcal{R} = (r_1, \ldots, r_T)$ and $\mathcal{R}' = (r'_1, \ldots, r'_T)$, and any subset of output actions $\mathcal{O} \subset \mathcal{A}^T$, $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(\mathcal{R}) \in \mathcal{O}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{R}') \in \mathcal{O}] + \delta. \tag{2}$$

When $\delta = 0$, we say that the mechanism $\mathcal{M}$ is pure differentially private (ε-DP). The DP mechanisms maintain that the distribution on the output of the mechanism does not significantly change when replacing a single client with reward r_t with another client with reward r'_t. Thus, the adversary observing the output of the DP mechanism does not infer the clients rewards.

Local Differential Privacy (LDP). If the central learner is untrusted, we need a local private mechanism $\mathcal{M}$ whose output is all the information available to the central learner. We denote the range of the output of the local mechanism by $\mathcal{Z}$.

Definition 2. (LDP [26]) A randomized mechanism $\mathcal{M} : [-1, 1] \to \mathcal{Z}$ is said to be $(\varepsilon_0, \delta_0)$ Local Differentially Private ($(\varepsilon_0, \delta_0)$-LDP) if for any rewards r_t and r'_t, and any subset of outputs $\mathcal{O} \subset \mathcal{Z}$, the algorithm $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(r_t) \in \mathcal{O}] \leq e^{\varepsilon_0} \Pr[\mathcal{M}(r'_t) \in \mathcal{O}] + \delta_0. \tag{3}$$

Similar to the DP definition, we say that $\mathcal{M}$ is pure locally differentially private (ε_0-LDP) when $\delta_0 = 0$. Observe that the input of the LDP mechanism is a single reward, and hence, each client preserves privacy of her observed reward r_t, even if the adversary knows what is the action she plays and observes a function of her reward.

In contextual linear bandits, the context c_t and the reward r_t are considered sensitive information about the client. Hence,

Algorithm	Regret Bound	Context	Privacy Model	
			Central DP	Local DP
Central DP [13]	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon}\right)$	Adversarial	(ε, δ)	N/A
LDP [21]	$\tilde{\mathcal{O}}\left(\frac{T^{3/4}}{\varepsilon_0}\right)$	Adversarial	$(\varepsilon = \varepsilon_0, \delta)$	(ε_0, δ)
LDP+shuffling [16]	$\tilde{\mathcal{O}}\left(\frac{T^{2/3}}{\varepsilon^{1/3}}\right)$	Adversarial	(ε, δ)	$\left(\varepsilon_0 = \varepsilon^{2/3}T^{1/6}, \delta\right)$
LDP [15]	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon_0}\right)$	Stochastic	$(\varepsilon = \varepsilon_0, \delta)$	(ε_0, δ)
Central DP (Theorem 1)	$\tilde{\mathcal{O}}\left(\sqrt{T} + \frac{1}{\varepsilon}\right)$	Free	$(\varepsilon, 0)$	N/A
LDP (Theorem 2)	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon_0}\right)$	Free	$(\varepsilon = \varepsilon_0, 0)$	$(\varepsilon_0, 0)$
LDP+shuffling(Theorem 3)	$\tilde{\mathcal{O}}\left(\sqrt{T} + \frac{1}{\varepsilon}\right)$	Free	(ε, δ)	$\left(\varepsilon_0 = \varepsilon T^{1/4}, 0\right)$

TABLE I: Upper part: known results. Lower part: our results. The $\tilde{\mathcal{O}}$ notation hides the dependencies on the dimension d, privacy parameter δ and $\log$ factors.

the goal of private contextual bandits is to keep both the context and the reward private. Unfortunately, a linear regret bound is unavoidable in contextual bandits under DP constraints [13]. Therefore, Shariff et al. in [13] have presented the notion of joint differential privacy (JDP) for contextual bandits. For any two sequences $\mathcal{S} = \{(\mathcal{A}_1, r_1), (\mathcal{A}_2, r_2), \ldots, (\mathcal{A}_T, r_T)\}$ and $\mathcal{S}' = \{(\mathcal{A}'_1, r'_1), (\mathcal{A}'_2, r'_2), \ldots, (\mathcal{A}'_T, r'_T)\}$, we say that $\mathcal{S}$ and $\mathcal{S}'$ are t-neighbors if it holds that $(\mathcal{A}_j, r_j) = (\mathcal{A}'_j, r'_j)$ for all $j \neq t$.

Definition 3. (JDP [13]) A randomized algorithm $\mathcal{M}$ for the contextual bandit problem is (ε, δ)-jointly differentially private (JDP) under continual observation if for any t and any t-neighboring sequences $\mathcal{S}$ and $\mathcal{S}'$, and any subset $\mathcal{S}_{>t} \subset \mathcal{A}_{t+1} \times \cdots \times \mathcal{A}_T$, it holds that:

$$\Pr[\mathcal{M}(\mathcal{S}) \in \mathcal{S}_{>t}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{S}') \in \mathcal{S}_{>t}] + \delta. \quad (4)$$

Thus, changing the pair (c_t, r_t) of a single client cannot have a significant impact on determining future actions.

System Model. We consider three different models for private stochastic linear bandits. In all three cases, our setup is that of a learner, who asks clients to play publicly observable actions, and collects the resulting rewards (see Figure 1). The models differ on whether the learner is a trusted or untrusted server, and whether a shuffler is available or not. A shuffler simply performs a random permutation on its input.

1) Central DP model: The learner is a **trusted server** who can collect the clients' rewards and take actions. Thus, the trusted server can apply a DP mechanism (see Definition 1) to preserve the privacy of the collected rewards against any adversary observing the actions of the clients.

2) LDP model: The learner is an **untrusted server**. Hence, each client needs to privatize her own reward by applying an LDP mechanism (see Definition 2) before sending it to the untrusted server. The server takes decisions on next actions using the collected privatized rewards.

3) Shuffled model: Similar to the LDP model, the learner is an **untrusted server**. However, we consider that there exists a **trusted shuffler** that collects the LDP responses of the clients and randomly permutes them before passing them to the server, see Figure 1.

Following [13], [15], [19], we consider a model where each client appears only once, hence, we have T clients in total. This assumption is practical for many applications, e.g., online shopping and ads, where there are millions of clients interacting only one time with the algorithm.

III. Stochastic Linear Bandits with central DP

In this section we consider the case where the learner is a trusted server. We present an algorithm that offers ε-DP (see Definition 1) for stochastic linear bandits, with no regret penalty: we achieve the same order regret performance as the best algorithms that operate under no privacy considerations.

Algorithm 1 ε-DP algorithm for stochastic linear bandits: central model

Input: set of actions $\mathcal{A}$, time horizon T, and privacy parameter ε.
Let $\mathcal{A}_1$ be a ζ-net for $\mathcal{A}$ as in Lemma 1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ **do**
  $\gamma_i \leftarrow \sqrt{\frac{4d}{q^i}\log\left(4|\mathcal{A}_i|T^2\right)} + \frac{2Bd^2 + 2d\log\left(4|\mathcal{A}_i|T^2\right)}{\varepsilon q^i}$.
  For $\mathcal{A}_i \subseteq \mathbf{R}^m$, $m \leq d$, let $\mathcal{C}_i$ be a core set of size at most Bm as in Lemma 2 and π_i the associated distribution.
  Pull each action $a \in \mathcal{C}_i$, $n_{ia} = \lceil \pi_i(a) q^i \rceil$ times to get rewards $r_{ia}^{(1)}, \ldots, r_{ia}^{(n_{ia})}$.
  $\bar{r}_{ia} \leftarrow \sum_{k=1}^{n_{ia}} r_{ia}^{(k)}$, $\hat{r}_{ia} \leftarrow \bar{r}_{ia} + z_{ia}$ $\forall a \in \mathcal{C}_i$, where z_{ia} is an independent noise that follows $\mathsf{Lap}(\frac{1}{\varepsilon})$.
  $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$, $\hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
  $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.

Main Idea. Our algorithm follows the structure of elimination algorithms: it runs in batches, where we maintain a "good set of actions" $\mathcal{A}_i$, in each batch i that almost surely contain the optimal one, and gradually eliminate sub-optimal actions, shrinking the sets $\mathcal{A}_i$ as i increases. As is fairly standard in elimination algorithms, in our case as well, during batch i, the learner plays actions in $\mathcal{A}_i$, calculates an updated estimate $\hat{\theta}_i$ of the unknown parameter vector θ_*, and eliminates from $\mathcal{A}_i$ actions if their estimated reward is $2\gamma_i$ from the estimated reward of the arm that appears to be best, where γ_i is the confidence of the reward estimates.

We note that our adversary observes actions generated through the estimate of θ_i. Since, the $\hat{\theta}_i$ is generated from the

private rewards, all functions of $\hat{\theta}_i$ (including estimate of next actions) is ε-DP from post-processing [6]. Our new observation on how to achieve this is as follows. **If by playing a smaller number of distinct actions we are able to identify the optimal action, we need to overall add a smaller amount of noise to guarantee privacy than if we play a larger number of distinct actions.** Indeed, if an action a is played for n_a times, the learner, to estimate θ_*, only needs to use the sum of these n_a rewards. To offer ε-DP we can perturb this sum by adding independent Laplacian noise (Lap($\frac{1}{\varepsilon}$)); clearly, the smaller the number of distinct actions we play, the smaller the overall amount of noise we need to add. Thus our algorithm, at each batch iteration i, plays actions from a carefully selected subset of $\mathcal{A}_i$, of cardinality as small as possible. The technical question we address is, starting from a continuous action space $\mathcal{A}$, how to select at each batch iteration a small cardinality subset that maintains the ability to identify the optimal action.

We next describe the steps in implementing this idea. Recall that our actions come from a set $\mathcal{A} \subseteq \mathbb{R}^d$, and we assume they are bounded, namely, $\|a\|_2 \leq 1,\ \forall a \in \mathcal{A}$ (see Assumptions 1 in Section II).

1. Our first step is to **reduce the continuous action space to a discrete action space problem**. To do so, we finely discretize $\mathcal{A}$ to create what we call a ζ-net, a discrete set of actions $\mathcal{N}_\zeta \subseteq \mathcal{A}$ such that distances are approximately preserved. Namely, for any $a \in \mathcal{A}$, there is some $a' \in \mathcal{N}_\zeta$ with $\|a' - a\|_2 \leq \zeta$. Lemma 1, proved in [27, Cor. 4.2.13], states that we can always find such a discrete set with cardinality at most $(\frac{3}{\zeta})^d + d$. As a result, all the "good sets" $\mathcal{A}_i$ will also be discrete.

Lemma 1. *(ζ-net for $\mathcal{A}$ [27]) For any set $\mathcal{A} \subseteq \{x \in \mathbb{R}^d | \|x\|_2 \leq 1\}$ that spans $\mathbb{R}^d$, there is a set $\mathcal{N}_\zeta \subseteq \mathcal{A}$ (zeta-net) with cardinality at most $(\frac{3}{\zeta})^d + d$ such that $\mathcal{N}_\zeta$ spans $\mathbb{R}^d$, and for any $a \in \mathcal{A}$, there is some $a' \in \mathcal{N}_\zeta$ with $\|a' - a\|_2 \leq \zeta$.*

2. We introduce the use of a **core set** $\mathcal{C}_i$, a subset of the actions of the set of "good actions" $\mathcal{A}_i$. During batch i, **the learner only plays actions in $\mathcal{C}_i$, each with some probability** $\pi_i(a)$. Lemma 2, proved in [28, Ch.21], states that if $\mathcal{A}_i$ spans some space $\mathbf{R}^k$, we can find a core set of size at most Bk (with B a constant) and an associated probability distribution π, so that, playing actions only from C_i enables to calculate a good estimate of $\langle a, \theta_* \rangle$ for each $a \in \mathcal{A}_i$.

Lemma 2. *(Core set for $\mathcal{A}$ [28]) For any finite set of actions $\mathcal{A} \subset \{x \in \mathbf{R}^d | \|x\|_2 \leq 1\}$ that spans $\mathbb{R}^d$, there is a constant B, a subset $\mathcal{C}$ and a distribution π on $\mathcal{C}$, that can be computed in polynomial time, such that $|\mathcal{C}| \leq Bd$, $\mathcal{C}$ spans $\mathbb{R}^d$, and for any $a \in \mathcal{A}$*

$$a^\top \left(\sum_{\alpha \in \mathcal{C}} \pi(\alpha) \alpha \alpha^\top \right)^{-1} a \leq 2d. \tag{5}$$

The computation of $\mathcal{C}, \pi$ can be formulated as a convex optimization problem with many efficient approximation algorithms available [28], [29]. For completeness, we present the Frank-Wolfe Algorithm in Appendix A, where we use it to construct the core set in our numerics in Section VI.

3. To preserve the privacy of rewards, we **perturb the sum rewards of each action by adding Laplace noise**. Adding noise affects the confidence of the reward estimates γ (step 5 in Algorithm 1 shows that γ increases as ε decreases), and thus delays the elimination of bad actions and increases the regret by an additive term of $\tilde{O}(\frac{1}{\varepsilon})$. Replacing a possibly large set $\mathcal{A}_i$ with the smaller core set $\mathcal{C}_i$ effectively decreases the cumulative noise affecting the estimate of θ_*.

Algorithm Pseudo-Code. Algorithm 1, starts by initializing the good action set $\mathcal{A}_1$ to be an $\frac{1}{T}$-net of $\mathcal{A}$ according to Lemma 1. Then, the algorithm operates in batches that grow exponentially in length, where the length of batch i is approximately q^i and $q = (2T)^{1/\log T}$[2]. In each batch i, we construct the core set C_i and the associated distribution π_i as per Lemma 2. Each action in $\mathcal{C}_i$ is pulled $n_{ia} = \lceil \pi(a) q^i \rceil$ times, where the length of batch i is $n_i = \sum_{a \in \mathcal{C}_i} n_{ia}$. To preserve privacy, the sum of the rewards of each action is perturbed with Lap$(1/\varepsilon)$ noise. The learner uses these privatized sum rewards to compute the estimate of θ_*, $\hat{\theta}_i$. At the end of batch i, the learner eliminates from $\mathcal{A}_i$ the actions with estimated mean reward, $\langle a, \hat{\theta}_i \rangle$, that fail to be within $2\gamma_i$ from the action that appears to be best, where γ_i is our confidence in the mean estimates. After the iteration $i = \log T - 1$ is completed, the learner simply plays the action that appears to be best.

Algorithm Performance. We next prove that Algorithm 1 is ε-DP and provide a bound on its regret.

Theorem 1. *Algorithm 1 is ε-differentially private. Moreover, it achieves a regret*

$$R_T \leq C \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon} \right), \tag{6}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε, T.

Proof Outline. The privacy result follows from the Laplace mechanism [6]. To bound the regret, we first argue that with probability at least $1 - \frac{1}{T}$, and for all i and all $a \in \mathcal{A}_i$, we have that $|\langle a, \hat{\theta}_i \rangle - \langle a, \hat{\theta}_\star \rangle| \leq \gamma_i$. Conditioned on this event, an action with gap Δ_a is eliminated when, or before, $\gamma_i < \Delta_a/2$. Hence, all actions in batch i have a gap that is at most $4\gamma_i$. The regret bound follows by summing $4\gamma_i n_i$ for all batches. The complete proof is provided in Section VII. □

Remark 1. We note that the high probability bound in Theorem 1 implies a bound in expectation

$$\mathbb{E}[R_T] \leq C \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon} \right). \tag{7}$$

The regret is trivially $O(T)$ and the failure probability is $\frac{1}{T}$, which overall contributes $O(1)$ to $\mathbb{E}[R_T]$.

Remark 2. The regret in Theorem 1 is optimal up to $\log T$ factor; a lower bound of $\Omega(\sqrt{T})$ is proven in [12] for the non-private case, while a lower bound of $\frac{\log T}{\varepsilon}$ is shown in [13] for private case.

Remark 3. We observe that the privacy parameter ε is typically ≈ 1. In this case, the dominating term in (11) is $O(\sqrt{T \log T})$

[2]We note that $e \leq q \leq e^2$.

which matches the regret of the best-known algorithm for the non-private case (see LinUCB in [12], [14]), and hence, we get privacy for free.

A. Stochastic Contextual Bandits with Central DP

In this section, we extend our results to the contextual linear bandits with known context distribution. In the following, we focus on the stochastic context setting where the context c_t is generated from a distribution $\mathcal{P}$ independently from other iterations. We assume that the distribution $\mathcal{P}$ is known to the learner[3]. The main idea is to use the reduction proposed in [30] to represent the contextual linear bandits with known context distribution as a stochastic linear bandits problem, and then, we apply our DP algorithm for stochastic linear bandits.

First, we briefly review the reduction for the case of known context distribution and refer the reader to [30] for a detailed description. The basic idea in [30] is to establish a linear bandit action for each possible parameter vector θ of the contextual bandit instance.

This is achieved through the use of the function $g : \mathbb{R}^d \to \mathbb{R}^d$, which computes the expected best action under the context distribution $\mathcal{P}$ with respect to the parameter θ: $g(\theta) = \mathbb{E}_{c_t\sim\mathcal{P}}[\arg\max_{a\in\mathcal{A}}\langle\phi(a, c_t), \theta\rangle]$. As stated in [30, Theorem 1], when $a_t = \arg\max_{a\in\mathcal{A}}\langle\phi(a, c_t), \theta_t\rangle$ for some $\theta_t \in \mathbb{R}^d$, then the reward generated by the contextual bandit instance can be expressed as $r_t = \langle g(\theta_t), \theta_\star\rangle + \eta'_t$, where η'_t is noise with zero mean conditioned on the history. Consequently, the reward can be viewed as generated by pulling action $g(\theta_t)$ in a linear bandit instance with an action set $\mathcal{X} = \{g(\theta)|\theta \in \Theta\}$. Moreover, the same theorem demonstrates that if a linear bandit algorithm is employed to choose $g(\theta_t) \in \mathcal{X}$ at round t and thus play action $a_t = \arg\max_{a\in\mathcal{A}}\langle\phi(a, c_t), \theta_t\rangle$, then $|R_T - R_T^L| = \tilde{O}(\sqrt{T})$ with high probability, where $R_T^L = \sum_{t=1}^T \sup_{\theta\in\Theta}\langle g(\theta) - g(\theta_t), \theta_\star\rangle$ is the regret of the algorithm on the linear bandit instance.

As a result, if the context distribution is known, then the function g is known to the learner as well as the users. Thus, we can construct a contextual bandits algorithm under joint differential privacy (JDP) constraints to privatize the contexts and rewards using our Algorithm 1 as follows. We apply our Algorithm 1 with action set $\mathcal{A} \triangleq \mathcal{X} \triangleq \{g(\theta) : \theta \in \Theta\}$. When a client receives an action $x_t \triangleq g(\theta_t)$ (from linear bandits), the client chooses an actual action a_t by solving $a_t = \arg\max_{a\in\mathcal{A}} \langle\phi(a, c_t), \theta_t\rangle$, where $\theta_t = g^{-1}(x_t)$ with ties broken arbitrarily. The client observes a reward r_t and sends it to the learner. Following Algorithm 1, at the end of the batch, the learner privatizes the aggregated rewards and updates the action set $\mathcal{X}_{i+1}$ to the next batch, see Steps $8-10$ in Algorithm 1.

Corollary 1. There exists an $(\varepsilon, 0)$-JDP algorithm for stochastic contextual bandits with known context distribution with bounded regret:

$$R_T \leq C\left(\sqrt{T\log T} + \frac{\log^2 T}{\varepsilon}\right), \tag{8}$$

with probability at least $1 - \frac{2}{T}$, where C is a constant that does not depend on ε, T.

Proof. The results are obtained by applying the algorithm explained above which is a combination of the reduction from [30] and our Algorithm 1. Observe that at any iteration $t \in [T]$, all the past history of context-reward pairs $\{(c_{t'}, r_{t'}) : t' < t\}$ are encoded in the returned reward set $\{r_{t'} : t' < t\}$. Furthermore, the past sequence rewards are $(\varepsilon, 0)$-DP from Theorem 1, where the learner uses only these private rewards to estimate the unknown parameter $\theta_\star$ and decides the new action of the next iteration. Thus, the presented algorithm is $(\varepsilon, 0)$-JDP.

The regret of our algorithm of stochastic linear bandits is bounded by $C'\left(\sqrt{T\log T} + \frac{\log^2 T}{\varepsilon}\right)$ from Theorem 1 with probability at least $1 - \frac{1}{T}$. Furthermore, from [30, Theorem 1], the difference between the regrets of the linear and contextual bandits instances $|R_T - R_T^L| = \tilde{O}(\sqrt{T})$ with probability at least $1 - 1/T$. By the triangle inequality and the union bound, it follows that the regret of the algorithm is bounded by $C\left(\sqrt{T\log T} + \frac{\log^2 T}{\varepsilon}\right)$ with probability at least $1 - 2/T$. This completes the proof of Corollary 1. ∎

Remark 4. In this section, we showed that our Algorithm 1 for DP stochastic linear bandits can be extended to give a JDP algorithm for contextual bandits with known distribution. A similar argument can be applied to the local DP model and the shuffled model in the next sections.

IV. Stochastic Linear Bandits with LDP

In this section, the learner is an untrusted server, and thus we design a linear bandit algorithm (Algorithm 2) that operates under LDP constraints.

Main Idea. As in Algorithm 1, we here also utilize a core set of actions; the difference is that, since the server is untrusted, each client privatizes her own reward before providing it to the server. Our algorithm offers an alternative approach to [15] that achieves the same regret, while using operation in batches, which may in some applications be more implementation-friendly (e.g., multi-stage clinical trials and online marketing with high response rates) [31], [32], and also forms a foundation for the Algorithm 3 we discuss in the next section.

Algorithm Pseudocode. Algorithm 2 operates like Algorithm 1, except for the addition of $\mathsf{Lap}(1/\varepsilon_0)$ noise for each reward individually as opposed to adding $\mathsf{Lap}(1/\varepsilon)$ to the sum of the rewards of each arm in the central model. The value of γ_i is adjusted to account for this change. **Algorithm Performance.** The following Theorem 2 presents the privacy-regret tradeoffs of the LDP stochastic bandits Algorithm 2. The proof is deferred to Section VIII and follows the same main steps as the proof of Theorem 1, but with the modified values of γ_i.

Theorem 2. *Algorithm 2 is ε_0-LDP. Moreover, it achieves a regret*

$$R_T \leq C(1 + \frac{1}{\varepsilon_0})\left(\sqrt{T\log T}\right), \tag{9}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε_0 and T.

[3]The knowledge of the distribution $\mathcal{P}$ can be practical in multiple cases, e.g., known age, and gender distribution. The extension to unknown context distribution is a future direction of our work.

Algorithm 2 ε_0-LDP algorithm for stochastic linear bandits: local model

Input: set of actions $\mathcal{A}$, time horizon T, and privacy parameter ε_0.
Let $\mathcal{A}_1$ be a ζ-net for $\mathcal{A}$ as in Lemma 1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ **do**
  Client side:
    Receive action a from the server. Play action a and receive a reward r.
    Send $\hat{r} = r + \mathsf{Lap}(\frac{1}{\varepsilon_0})$.
  Server side:
    Let $\mathcal{C}_i$ be a core set for $\mathcal{A}_i$ as in Lemma 2 with distribution π_i, and $n_{ia} = \lceil \pi_i(a) q^i \rceil$.
    Send each action $a \in \mathcal{C}_i$ to a set of n_{ia} clients to get rewards $\hat{r}_{ia}^{(1)}, ..., \hat{r}_{ia}^{(n_{ia})}$.
    $n_i \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia}$.
    $\gamma_i \leftarrow \sqrt{\log\left(4|\mathcal{A}_i|T^2\right)}\left(\sqrt{\frac{4d}{q^i}} + \frac{2d\sqrt{n_i}}{q^i \varepsilon_0}\right)$.
    $\hat{r}_{ia} \leftarrow \sum_{k=1}^{n_j} \hat{r}_{ia}^{(1)} \ \forall a \in \mathcal{C}_i$.
    $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$, $\hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
    $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}_\rangle} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$.
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.

Remark 5. When $\varepsilon_0 > 1$, the regret R_T would be $\mathcal{O}\left(\sqrt{T}\log(T)\right)$ that matches the non-private case. However, the constants of the regret convergence are larger than that of the non-private case.

Remark 6. (Comparison to the central (ε, δ)-DP model.) Observe that when $\varepsilon_0 < 1$, the dominating term in the regret is $R_T = \mathcal{O}\left(\frac{T\log(T)}{\varepsilon_0}\right)$. In other words, we obtain the regret of the non-private case divided by the LDP parameter ε_0. In contrast, the central DP parameter ε appears as an additive term in the regret of the central model. This difference is because, in the local model noise is added on every reward, while in the central model directly on the reward aggregates; thus the noise variance of the aggregate rewards and the confidence parameter γ_i increases in the local model.

In the high privacy regimes; for example, assume that $\varepsilon_0 = \mathcal{O}\left(\frac{1}{T^\alpha}\right)$ for some $0 < \alpha \leq \frac{1}{2}$, we get a regret R_T of order $\mathcal{O}\left(T^{\frac{1}{2}+\alpha}\right)$ that becomes linear function of T as $\varepsilon_0 \to \frac{1}{\sqrt{T}}$.

V. Stochastic Linear Bandits in the Shuffled Model

In this section, we consider the case of an untrusted server and a trusted shuffler. We propose Algorithm 3 that (almost) achieves the same regret as the best non-private algorithms.

Main idea. To use shuffling, we need to use an algorithm that operates over batches of actions, so as to be able to shuffle them. The use of a core set is critical to enable a selection of actions that lead to a good estimate for $\theta\star$. For example, if the original set $\mathcal{A}$ contains a large number of actions along one direction in the space, but only a few actions along other directions, then pulling each action in $\mathcal{A}$ once will not result in a good estimate of $\theta_\star$. Use of the core set and the associated distribution π will balance such assymetries and enable to explore multiple directions of the space a sufficient number of times to acquire a good estimate of $\theta_\star$.

Algorithm 3 DP algorithm for stochastic linear bandits: shuffled model

Input: actions $\mathcal{A}$, horizon T, privacy parameters (ε, δ).
Let $\mathcal{A}_1$ be a ζ-net for $\mathcal{A}$ as in Lemma 1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ **do**
  Client side:
    Receive action a and the value n_i from shuffler.
    Play action a and receive a reward r.
    $\varepsilon_0^{(i)} \leftarrow f_{n_i,\delta}^{-1}(\varepsilon)$.
    Send $\hat{r} = r + \mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$ to the shuffler.
  Shuffler:
    Send action $a_{\pi(j)}$ and n_i to client j, $j = [n_i]$, where π is a random permutation of $[n_i]$.
    Receive the action-reward pairs $\{(a_j, \hat{r}_{ia_j})\}_{j=1}^{n_i}$, and send them to the server.
  Server side:
    Let $\mathcal{C}_i$ be a core set for $\mathcal{A}_i$ as in Lemma 2 with distribution π_i.
    Let $n_{ia} = \lceil \pi_i(a) q^i \rceil, n_i \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia}$.
    Let $\mathcal{A}_{\mathcal{C}_i}$ be a list of n_i actions where action $a \in \mathcal{C}_i$ is repeated n_{ia} times.
    Let $a_1, ..., a_{n_i}$ be an enumeration of $\mathcal{A}_{\mathcal{C}_i}$. Send them to the shuffler
    Receive the action-reward pairs from the shuffler.
    $\gamma_i \leftarrow \sqrt{\log\left(4|\mathcal{A}_i|T^2\right)}\left(\sqrt{\frac{4d}{q^i}} + \frac{2d\sqrt{n_i}}{q^i \varepsilon_0^{(i)}}\right)$.
    $\hat{r}_{ia} \leftarrow \sum_{k=1}^{n_j} \hat{r}_{ia}^{(1)} \ \forall a \in \mathcal{C}_i$.
    $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$, $\hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
    $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$.
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.

Accordingly, we follow the same approach as in Algorithm 2 with two changes: we use a shuffler (in a manner tailored to bandits) to realize privacy amplification gains, and we adjust the amount of Laplace noise we add in each batch, depending on the batch size.

We use the trusted shuffler as follows. The actions to be played in the ith batch are shuffled by the trusted shuffler at the beginning of the batch. The shuffler asks clients to play actions in the shuffled order. Then, at the end of the batch, the shuffler reverses the shuffling operation, associates every action with its observed LDP reward, and conveys it to the untrusted learner.[4]

We adjust the amount of added Laplace noise per batch as follows. To offer privacy guarantees, we want to add noise to the rewards so that the output of the shuffler is (ε, δ)-DP for each

[4]The server cannot directly observe which action is played by which client, for instance due to geographical separation.

batch $i \in [\log(T)]$. This implies that the entire algorithm will be (ε, δ)-DP since we assume that each client contributes to only one of the batches. The privacy amplification of the shuffling depends on the size of the batch (see e.g. [10, Theorem 1]); thus the larger the batch size, the less noise needs to be added to the rewards of the clients. To ensure that the output of batch i is (ε, δ)-DP, it is sufficient to add to each reward noise $\mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$, where $\varepsilon_0^{(i)} \leftarrow f_{n_i,\delta}^{-1}(\varepsilon)$, and n_i is the size of batch i. The function $f_{n,\delta} : \mathbb{R}^+ \to \mathbb{R}^+$ captures privacy amplification via shuffling [10], [11] and is defined as follows

$$f_{n,\delta}(\varepsilon_0) = \log\left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0}\log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n}\right)\right). \tag{10}$$

Since the noise added to the rewards varies for each batch i, we modify the confidence bounds, γ_i, to reflect this. The pseudo-code is provided in Algorithm 3.

Algorithm Performance. The following theorem proves that Algorithm 3 is (ε, δ)-DP and provides an upper bound on its regret that matches the information theoretic lower bound for $\varepsilon = \tilde{O}(\frac{1}{\sqrt{T}})$.

Theorem 3. *Algorithm 3 is (ε, δ)-differentially private. Moreover, if ε is $O(\sqrt{\frac{\log(1/\delta)}{T}})$ it achieves a regret*

$$R_T \leq C\left(\sqrt{T\log T} + \frac{\sqrt{\log(1/\delta)}\log^{3/2} T}{\varepsilon}\right), \tag{11}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε and T.

Proof Outline. The proof of Theorem 3 is deferred to Section IX. The privacy guarantee is proved by reducing the scheme to one that shuffles the rewards but does not shuffle the corresponding actions and using results from [10], [11]. The regret analysis follows similar ideas as in Theorem 1 and Theorem 2.

Remark 7. Note that in our proposed shuffled model, we randomly permute the actions. We can achieve a similar regret by shuffling the rewards of each action separately by using the shuffled mechanism for scalar summation in [33]. Please see Appendix C for more details.

Remark 8. Algorithm 3 almost achieves the same order regret as the best non-private algorithms. Indeed, Theorem 3 proves that Algorithm 3 achieves a regret that matches the regret of the central DP Algorithm 1 for the high privacy regimes $\varepsilon = O(\sqrt{\log(1/\delta)/T})$. For the low privacy regime $\varepsilon > 1$, the shuffling does not offer privacy gains, $\varepsilon_0^{(i)} \approx \varepsilon$ for all $i \in [\log(T)]$ and the regret of Algorithm 3 is similar to the regret of Algorithm 2 of the local DP model. However, for the low privacy regime, the local DP model also achieves the same regret as non-private algorithms up to constant factors (see Remark 5). Hence in both cases, Algorithm 3 achieves the same order regret as Algorithm 1 which almost matches the regret of non-private algorithms.

Remark 9. Algorithm's 3 improved regret performance over Algorithm 2 is thanks to the smaller amount of noise added to rewards. In particular, the noise added in Step 9 of Algorithm 3 has variance $\frac{2}{\varepsilon_0^{(i)2}} \approx \frac{2}{n_i\varepsilon^2}$ for small ε.

Remark 10. Observe that the Gaussian noise satisfies similar concentration properties as proven in Lemma 4. Thus, we can use the Gaussian mechanism in our central and local DP model to achieve approximate DP. However, the Gaussian mechanism is not directly applied to the shuffled model algorithm, since the privacy amplification by shuffling results in [10], [11] requires a pure local DP mechanism. The privacy analysis with approximate local DP mechanisms in the shuffled framework is an open question.

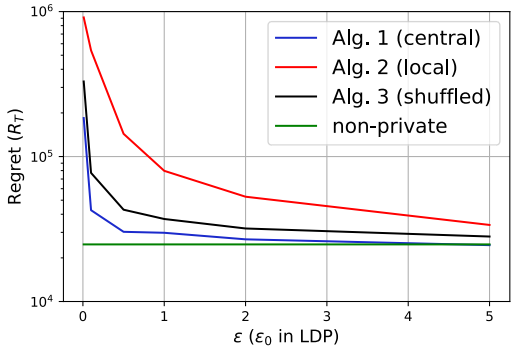
VI. Numerical Results

We here present indicative results on the performance of our proposed Algorithms 1, 2 and 3. In our numerical results, we use the Frank-Wolfe Algorithm presented in Appendix A to construct the coreset.

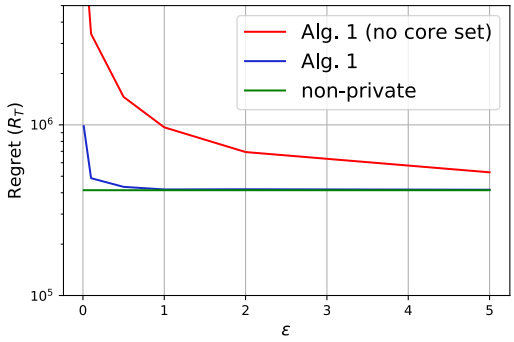
Data Generation. We generate synthetic data generated as follows. The set of actions $\mathcal{A}$ contains K actions, where each action $a \in \mathcal{A}$ is a $d = 2$-dimensional vector. The actions $a \in \mathcal{A}$ and the optimal parameter θ_* are generated uniformly at random from the unit sphere $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : ||x||_2 = 1\}$. A similar method is considered in [15]. Figure 2 plots the total regret R_T over a horizon $T = 10^6$ as a function of the privacy budget (ε or ε_0 in the case of LDP mechanisms).

Comparison of Algorithms 1, 2 and 3. In Figure 2, the set of actions $\mathcal{A}$ contains $K = 10$ actions. Figure 2 shows that the regret achieved by all three algorithms, Algorithm 1 (central model), Algorithm 2 (local model), and Algorithm 3 (shuffled model) converges to the regret of non-private stochastic linear bandit algorithms [28, Ch. 22] as $\varepsilon \to \infty$ ($\varepsilon_0 \to \infty$), albeit at different rates. As predicted from the theoretical analysis, Algorithms 1 (central) and 3 (shuffled) offer privacy (almost) for free, closely following the non-private regret. Furthermore, the central Algorithm 1 is close to the non-private case and significantly outperforms the LDP Algorithm 2. We observe that the shuffled model has a performance close to the central algorithm and outperforms the regret of the LDP Algorithm 2.

Usefulness of Core Set. In Figure 2b, we explore potential benefits on the performance of Algorithm 1 that use of the core set can offer. We consider $K = 1000$ and $T = 10^7$, and plot the regret of Algorithm 1 for two cases: (i) when we use a core set of size 2-3 actions, similar to the dimension of our space (labeled as Alg. 1), and (ii) when no core set is used, and instead the good set of actions of the batched algorithm is the whole action set (labeled as Alg. 1 no-core-set). The good action set (in line 10 of Algorithm 1) is used with the uniform distribution as our exploration policy. We find that, as expected from our theoretical analysis, using a core set enables to achieve performance very close to that of a non-private batched algorithm that adds no noise. In contrast, using (and adding noise to) the entire action space significantly degrades the performance.



(a) Central, local and shuffled models, $K = 10, T = 10^6$.



(b) Effect of core set size, $K = 1000, T = 10^7$.

Fig. 2: Regret-privacy trade-offs for stochastic linear bandits algorithms.

VII. Regret and Privacy Analysis of The Central DP Model (Proof of Theorem 1)

In this section, we prove the regret bound and the privacy guarantees of the central DP algorithm. We present the privacy analysis in Section VII-A and the regret analysis in Section VII-B.

A. Privacy Analysis

We first show that Algorithm 1 is ε-DP. Let $\bar{r}_i = [\bar{r}_{ia_1}, ..., \bar{r}_{ia_{|\mathcal{C}_i|}}]$, $\hat{r}_i = [\hat{r}_{ia_1}, ..., \hat{r}_{ia_{|\mathcal{C}_i|}}] = \bar{r}_i + z_i, z_i = [z_{ia_1}, ..., z_{ia_{|\mathcal{C}_i|}}]$, where $a_1, ..., a_{|\mathcal{C}_i|}$ is an enumeration of the elements of $\mathcal{C}_i$. We construct the concatenated reward vector denoted by $\bar{r} = [\bar{r}_1, ..., \bar{r}_{\log(T)-1}]$, and let $\hat{r} = [\hat{r}_1, ..., \hat{r}_{\log(T)-1}] = \bar{r} + z, z = [z_1, ..., z_{\log(T)-1}]$.

Now consider two neighboring sequence of rewards $\mathcal{R}, \mathcal{R}'$, that only differ in r_k, r'_k, with corresponding concatenated reward vectors $\bar{r}, \bar{r}'$. We notice that each reward in $\mathcal{R}$ appears once in $\bar{r}$, and similarly for $\mathcal{R}', \bar{r}'$. Thus, we get:

$$\|\bar{r} - \bar{r}'\|_1 \leq \max_{r_k, r'_k} |r_k - r'_k| \leq 1, \tag{12}$$

where the last inequality follows from Assumption 1 with bounded rewards $|r_k| \leq 1$. Then, from [5, Theorem 3.6], $\hat{r}$ is ε-DP. We notice that the output of Algorithm 1 depends on $r_1, ..., r_T$ only through $\hat{r}$. Hence, by post processing, Algorithm 1 is ε-DP.

B. Regret Analysis

We next prove the regret bound of Algorithm 1 for stochastic linear bandits.

Our analysis follows the known confidence bound technique in [34] by designing confidence intervals (in step 5) that take into consideration the privacy effect.

Let $K = (3T)^d$ be the size of the $\frac{1}{T}$-net set $\mathcal{N}_{1/T}$ from Lemma 1. We first bound the following regret:

$$\tilde{R}_T = T \max_{a \in \mathcal{N}_{1/T}} \langle a, \theta_* \rangle - \sum_{t=1}^{T} \langle a_t, \theta_* \rangle, \tag{13}$$

where $a_1, a_2, \ldots, a_T \in \mathcal{N}_{1/T}$. We then bound the regret R_T by showing that we only loose a constant term when we choose actions from $\mathcal{N}_{1/T}$ instead of the bigger set $\mathcal{A}$.

We start with a set of actions $\mathcal{A}_0 = \mathcal{N}_{1/T}$ with cardinality $|\mathcal{A}_0| = K$. Furthermore, we have $|\mathcal{A}_i| \leq |\mathcal{A}_{i-1}|$, and hence, we get $|\mathcal{A}_i| \leq K$ for all $i \in [\log(T)]$.

For given batch $i \in [\log(T)]$, let $\mathcal{C}_i$ be the core set of $\mathcal{A}_i$ that has at most Bd actions. At the ith batch, each action $a \in \mathcal{C}_i$ is picked n_{ia} times, where $n_{ia} = \lceil \pi_i(a) q^i \rceil$. Let $\mathcal{G}$ be the good event $\left\{ \left| \langle a, \hat{\theta}_i - \theta_* \rangle \right| < \gamma_i \ \forall i \in [\log T] \ \forall a \in \mathcal{A}_i \right\}$. Lemma 3 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof, we condition on the event $\mathcal{G}$.

We first show that the best action $a_* = \arg\max_{a \in \mathcal{N}_{1/T}} \langle a, \theta_* \rangle$ will not be eliminated at any batch $i \in [\log T]$; this is because the elimination criterion will not hold for the optimal action a_*:

$$\langle a, \hat{\theta}_i \rangle - \langle a_*, \hat{\theta}_i \rangle < (\langle a, \theta_* \rangle + \gamma_i) - (\langle a_*, \theta_* \rangle - \gamma_i) \leq 2\gamma_i \quad \forall a \in \mathcal{A}_i \ \forall i \in [\log T]. \tag{14}$$

For each sub-optimal action $a \in \mathcal{A}_0$ with $\Delta_a = \langle a_* - a, \theta_* \rangle$, let i be the smallest integer for which $\gamma_i < \frac{\Delta_a}{4}$. From the triangle inequality, we get that

$$\langle a_*, \hat{\theta}_i \rangle - \langle a, \hat{\theta}_i \rangle \geq (\langle a_*, \hat{\theta}_* \rangle - \gamma_i) - (\langle a, \hat{\theta}_i \rangle + \gamma_i) = \Delta_a - 2\gamma_i > 2\gamma_i. \tag{15}$$

This implies that a will be eliminated before the beginning of batch $i+1$. Hence, each action $a \in \mathcal{A}_{i+1}$ at batch $i+1$ has a gap at most $4\gamma_i$. Let $n_i = \sum_{a \in \mathcal{C}_i} n_{ia} \leq Bd + q^i$ denote the total number of rounds at the i-th batch. Note that the number of batches is upper bounded by $\log T$ since $\sum_{i=1}^{\log T} q^i \geq T$. When $q^i < Bd$, the regret can be bounded by $2Bd$, and when $q^i \geq Bd$, we bound $n_i \leq 2q^i$. Thus, there is universal constants C', C such that the total regret in (13) can be bounded as

$$\begin{aligned} \tilde{R}_T &\leq 2Bd \log(T) + \sum_{i=1}^{\log T} 4 n_i \gamma_{i-1} \\ &\leq 2Bd \log(T) + \sum_{i=1}^{\log T} 8 q^i \left(\sqrt{\frac{4d}{q^{i-1}} \log\left(4KT^2\right)} + \right. \end{aligned} \tag{16}$$

$$\frac{2Bd^2 + 2d\log\left(4KT^2\right)}{\varepsilon q^{i-1}}\Bigg)$$
$$\leq C'\left(d\log(T) + d\sqrt{\log T}\sum_{i=1}^{\log T} q^{(i-1)/2} + \frac{d^2\log^2 T}{\varepsilon}\right)q$$
$$\overset{(a)}{\leq} C'q\left(d\log(T) + d\sqrt{\log T}q^{\log T/2} + \frac{d^2\log^2 T}{\varepsilon}\right)$$
$$\overset{(b)}{\leq} C'q\left(d\log(T) + d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon}\right)$$
$$\overset{(c)}{\leq} C\left(d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon}\right), \tag{17}$$

where step (a) follows from the sum of a geometric series and $q > 1$, step (b) uses $q = (2T)^{1/\log T}$, and step (c) follows from the facts $q \leq e^2$, $\log T = O(\sqrt{T})$.

Hence, with probability at least $1 - \frac{1}{T}$ the regret in (13) is bounded as

$$\tilde{R}_T \leq C\left(d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon}\right). \tag{18}$$

Next, we bound the exact regret R_T. Observe that the first step in our Algorithm is to use the finite $\frac{1}{T}$-net set $\mathcal{N}_{1/T}$ of actions. Thus, for any round $t \in [T]$ and any action $a \in \mathcal{A}$, there exists an action $a' \in \mathcal{N}_{1/T}$ such that $\|a - a'\| \leq \frac{1}{T}$. As a result, we get $\langle a, \theta_*\rangle - \langle a', \theta_*\rangle \leq \|a - a'\|\|\theta_*\| \leq \frac{1}{T}$, where $\|\theta_*\| \leq 1$. Hence, there is a universal constant C such that we can bound the regret R_T as

$$\begin{aligned} R_T &= T\max_{a\in\mathcal{A}}\langle a, \theta_*\rangle - \sum_{t=1}^{T}\langle a_t, \theta_*\rangle \\ &= \left[T\max_{a\in\mathcal{A}}\langle a, \theta_*\rangle - T\max_{a'\in\mathcal{N}_{1/T}}\langle a', \theta_*\rangle\right] \\ &\quad + \left[T\max_{a'\in\mathcal{N}_{1/T}}\langle a', \theta_*\rangle - \sum_{t=1}^{T}\langle a_t, \theta_*\rangle\right] \\ &\leq T\frac{1}{T} + \tilde{R}_T \\ &= 1 + \tilde{R}_T. \end{aligned} \tag{19}$$

Hence, with probability at least $1 - \frac{1}{T}$ the regret R_T is bounded as

$$R_T \leq C\left(d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon}\right). \tag{20}$$

This concludes the proof of Theorem 1.

Lemma 3. *Let $\hat{\theta}_i$ be the least square estimate of θ_* at the end of the ith batch of Algorithm 1. Then, we have that*

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| > \gamma_i \ \forall i \in [\log T]\forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \tag{21}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{2Bd^2 + 2d\log\left(4KT^2\right)}{\varepsilon q^i}$.

Proof. Let $\hat{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\hat{r}_{ia}a$ be the private estimate of θ_* and $\bar{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\bar{r}_{ia}a$ be the non-private estimate of θ_* as $\{\bar{r}_{ia}\}$ are the non-private rewards, where $V_i = \sum_{a\in\mathcal{C}_i} n_{ia}aa^\top$. From [Chapter 21, Eqn 21.1], for each $a \in \mathcal{A}_i$, we get:

$$\Pr\left[\langle a, \bar{\theta}_i - \theta_*\rangle \geq \sqrt{2\|a\|^2_{V_i^{-1}}\log\left(\frac{1}{\beta}\right)}\right] \leq \beta, \tag{22}$$

where $\beta \in (0, 1)$ and $\|a\|^2_{V_i^{-1}} = a^\top V_i^{-1}a$. Let $V_i(\pi_i) = \sum_{a\in\mathcal{C}_i}\pi_i(a)aa^\top$ and hence we have

$$V_i = \sum_{a\in\mathcal{C}_i} n_{ia}aa^\top \geq q^i\sum_{a\in\mathcal{C}_i}\pi_i(a)aa^\top = q^iV_i(\pi_i). \tag{23}$$

Observe that for any symmetric random variable x if $\Pr[x \geq t] \leq \beta$, then $\Pr[|x| \geq t] = \Pr[x \geq t] + \Pr[-x \geq t] \leq 2\beta$. Thus, from lemma 2, we have $\|a\|^2_{V_i^{-1}} = \frac{1}{q^i}a^\top V_i(\pi_i)^{-1}a \leq \frac{2d}{q^i}$ for each $a \in \mathcal{A}_i$. By setting $\beta = \frac{1}{4KT^2}$ and $\|a\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{A}_i$ in (22), we get that:

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \theta_*\rangle\right| \geq \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)}\right] \leq \frac{1}{2KT^2}, \tag{24}$$

for each $a \in \mathcal{A}_i$. Now, we compute the effect of the privacy in estimating θ_* by bounding difference $\langle a, \bar{\theta}_i - \hat{\theta}_i\rangle$. Observe that $\hat{r}_{ia} = \bar{r}_{ia} + z_{ia}$, where $z_{ia} \sim \mathsf{Lap}(\frac{1}{\varepsilon})$, and hence, we can write $\hat{\theta}_i - \bar{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i} z_{ia}a$. Thus, for any $\alpha \in \mathcal{A}_i$, we have that:

$$\langle \alpha, \hat{\theta}_i - \bar{\theta}_i\rangle = \sum_{a\in\mathcal{C}_i}\alpha^\top V_i^{-1}az_{ia}, \tag{25}$$

where $\alpha^\top V_i^{-1}a \leq \max_{b\in\mathcal{A}_i}\|b\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{C}_i$ that holds from the fact that V_i is positive semi-definite. From Lemma 4 presented at the end of the section, by setting $b = \varepsilon$, $n = Bd$, $c = \frac{2d}{q^i}\sqrt{n}$, and $t = 2\frac{Bd^2}{\varepsilon q^i} + \frac{2d\log\left(4KT^2\right)}{\varepsilon q^i}$, we get that:

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \hat{\theta}_i\rangle\right| \geq 2\frac{Bd^2}{\varepsilon q^i} + \frac{2d\log\left(4KT^2\right)}{\varepsilon q^i}\right] \leq \frac{1}{2KT^2}, \tag{26}$$

Then, by the union bound and triangle inequality we have that

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| > \gamma_i \ \forall i \in [\log T]\forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \tag{27}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{2Bd^2 + 2d\log\left(4KT^2\right)}{\varepsilon q^i}$. This concludes the proof of Lemma 3. ∎

Lemma 4. *Let $x_i = l_iz_i$ for $i \in [n]$, where $z_i \sim \mathsf{Lap}(1/b)$ and l_i, c are constants such that $c^2 \geq \sum_{i=1}^n |l_i|^2$. Let $\bar{x} = \sum_{i=1}^n x_i$. We have that*

$$\Pr[\bar{x} \geq t] \leq \begin{cases} \exp\left(-\frac{t^2b^2}{2c^2}\right) & \text{if } t \leq \frac{c^2}{bl_{\max}} \\ \exp\left(\frac{c^2}{2l^2_{\max}} - \frac{b}{l_{\max}}t\right) & \text{if } t > \frac{c^2}{bl_{\max}} \end{cases}, \tag{28}$$

where $l_{\max} = \max_i l_i$.

The proof is provided in App. B.

VIII. Regret and Privacy Analysis of The local DP Model (Proof of Theorem 2)

In this section, we present the privacy and regret analysis of stochastic linear bandits under local DP constraints.

A. Privacy Analysis

The privacy proof is straightforward. For any client, since the reward is bounded by $|r| \leq 1$, the output $\hat{r} = r + \mathsf{Lap}(1/\varepsilon_0)$ is ε_0-LDP from [5, Theorem 3.6].

B. Regret Analysis

We next prove the regret bound of Algorithm 2 for stochastic linear bandits with LDP. Our proof is similar to the proofs of the central DP Algorithm presented in Section VII-B. Let $\tilde{R}_T$ be the regret defined in (13).

Let $\mathcal{G}$ be the good event $\left\{\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| < \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right\}$. Lemma 5 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof we condition on the event $\mathcal{G}$. When $q^i < \max\{Bd, 2\log(4KT^2)\}$, the regret can be bounded by $\max\{Bd, 2\log(4KT^2)\}$, and when $q^i \geq \max\{Bd, 2\log(4KT^2)\}$, we bound $n_i \leq 2q^i$, and hence,

$$\gamma_i \leq \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{2d}{\varepsilon_0}\sqrt{\frac{\log(4KT^2)}{q^i}} \leq (1 + \frac{1}{\varepsilon_0})2d\sqrt{\frac{\log(4KT^2)}{q^i}}.$$

By following similar steps as in the central DP, we can show that there is universal constants C', C such that the total regret in (13) can be bounded as

$$\begin{aligned}
\tilde{R}_T &\leq (Bd + 2\log(4KT^2))\log(T) + \sum_{i=1}^{\log T} 4n_i\gamma_{i-1} \\
&\leq (Bd + 2\log(4KT^2))\log(T) \\
&\quad + (1 + \frac{1}{\varepsilon_0})2d\sum_{i=1}^{\log T} 8q^i\sqrt{\frac{1}{q^{i-1}}\log\left(4KT^2\right)} \\
&\leq C'(1 + \frac{1}{\varepsilon_0})\Big(d\sqrt{d}\log^2(T) \\
&\qquad + d\sqrt{d\log T}\sum_{i=1}^{\log T} q^{(i-1)/2}\Big)q \\
&\overset{(a)}{\leq} C'(1 + \frac{1}{\varepsilon_0})q\left(d\sqrt{d}\log^2(T) + d\sqrt{d\log T}q^{\log T/2}\right) \\
&\overset{(b)}{\leq} C'(1 + \frac{1}{\varepsilon_0})q\left(d\sqrt{d}\log^2(T) + d\sqrt{dT\log T}\right) \\
&\overset{(c)}{\leq} C(1 + \frac{1}{\varepsilon_0})\left(d\sqrt{dT\log T}\right), \qquad (29)
\end{aligned}$$

where step (a) follows from the sum of a geometric series and $q > 1$, step (b) uses $q = (2T)^{1/\log T}$, and step (c) follows from the facts $q \leq e^2$, $\log^2 T = O(\sqrt{T})$.

Hence, following similar steps as in the proof of the central DP algorithm, with probability at least $1 - \frac{1}{T}$ the regret is bounded as

$$R_T \leq \tilde{R}_T + 1 \leq C(1 + \frac{1}{\varepsilon_0})\left(d\sqrt{dT\log T}\right). \qquad (30)$$

Lemma 5. *Let $\hat{\theta}_i$ be the least square estimate of θ_* at the end of the ith batch of Algorithm 2. Then, we have that*

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \qquad (31)$$

where $\gamma_i = \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{1}{q^i\varepsilon_0}\sqrt{2dn_i\log(4KT^2)}$.

Proof. Let $\hat{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\hat{r}_{ia}a$ be the private estimate of θ_* and $\bar{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\bar{r}_{ia}a$ be the non-private estimate of θ_* as $\{\bar{r}_{ia}\}$ are the non-private rewards, where $V_i = \sum_{a\in\mathcal{C}_i} n_{ia}aa^\top$ and $\hat{r}_{ia} = \sum_{j=1}^{n_{ia}}\hat{r}_{ia}^{(j)}$. Similar to the central DP in Section III, we have that

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \theta_*\rangle\right| \geq \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)}\right] \leq \frac{1}{2KT^2}, \qquad (32)$$

for each $a \in \mathcal{A}_i$. Now, we compute the effect of the LDP in estimating θ_* by bounding difference $\langle a, \bar{\theta}_i - \hat{\theta}_i\rangle$. Observe that $\hat{r}_{ia} = \sum_{j=1}^{n_{ia}}\hat{r}_{ia}^{(j)} = \bar{r}_{ia} + z_{ia}$, where $\bar{r}_{ia} = \sum_{j=1}^{n_{ia}} r_{ia}^{(j)}$ and $z_{ia} = \sum_{j=1}^{n_{ia}} z_{ia}^{(j)}$, where $z_{ia}^{(j)} \sim \mathsf{Lap}(\frac{1}{\varepsilon_0})$. Hence, we can write $\hat{\theta}_i - \bar{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i} z_{ia}a$. Thus, for any $\alpha \in \mathcal{A}_i$, we have that:

$$\langle \alpha, \hat{\theta}_i - \bar{\theta}_i\rangle = \sum_{a\in\mathcal{C}_i}\sum_{j=1}^{n_{ia}} \alpha^\top V_i^{-1}az_{ia}^{(j)}, \qquad (33)$$

where $\alpha^\top V_i^{-1}a \leq \max_{b\in\mathcal{A}_i}\|b\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{C}_i$ that holds from the fact that V_i is positive semi-definite. We also have that

$$\begin{aligned}
\sum_{a\in\mathcal{C}_i}\sum_{j=1}^{n_{ia}}(\alpha^\top V_i^{-1}a)^2 &= \sum_{a\in\mathcal{C}_i}\sum_{j=1}^{n_{ia}}\alpha^\top V_i^{-1}aa^\top V_i^{-1}\alpha \\
&= \alpha^\top V_i^{-1}\alpha \leq \frac{2d}{q^i} \qquad (34)
\end{aligned}$$

From Lemma 4 presented in Section III, by setting $b = \varepsilon_0$, $n = n_i$, $c^2 = \frac{2d}{q^i}$, and $t = \frac{1}{q^i\varepsilon_0}\sqrt{2dn_i\log(4KT^2)}$, we get that:

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \hat{\theta}_i\rangle\right| \geq \frac{1}{q^i\varepsilon_0}\sqrt{2dn_i\log(4KT^2)}\right] \leq \frac{1}{2KT^2}, \qquad (35)$$

Then, by the union bound and triangle inequality we have that

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \qquad (36)$$

where $\gamma_i = \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{1}{q^i\varepsilon_0}\sqrt{2dn_i\log(4KT^2)}$. This concludes the proof of Lemma 5. ∎

IX. Regret and Privacy Analysis of The Shuffled Model (Proof of Theorem 3)

In this section, we provide the proof of Theorem 3.

A. Privacy Analysis

We note that the data of each user j can be represented as $\cup_{a\in\mathcal{C}_i}\{(a, r_a^{(j)})\}$. We observe that our scheme is equivalent to performing the following steps

- Each user $j \in [n_i]$ sends its data $\mathcal{D}_j = \cup_{a\in\mathcal{C}_i}\{(a, r_a^{(j)})\}$ to the shuffler.
- The shuffler randomly permutes the sets $\mathcal{D}_1, ..., \mathcal{D}_{n_i}$ to get $\mathcal{D}_{\pi(1)}, ..., \mathcal{D}_{\pi(n_i)}$.
- The shuffler reveals n_i action reward pairs $(a_1, \hat{r}_{ia_1}), ..., (a_{n_i}, \hat{r}_{ia_{n_i}})$, where $(a_j, \hat{r}_{ia_j}) \in \mathcal{D}_{\pi(j)}$, and $\hat{r}_{ia_j}$ is the LDP version of r_{ia_j} ($\hat{r}_{ia_j} = r_{ia_j} + \mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$).

Hence, we shuffle the data, then feed it to an LDP mechanism with LDP parameter $\varepsilon_0^{(i)}$ (as proved in Theorem 2). As a result,

it follows from [10], [11] that the output of the shuffler is (ε_i, δ)-DP where

$$\varepsilon_i = \log\left(1 + \frac{e^{\varepsilon_0^{(i)}} - 1}{e^{\varepsilon_0^{(i)}} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0^{(i)}}\log(4/\delta)}}{\sqrt{n_i}} + \frac{8e^{\varepsilon_0^{(i)}}}{n_i}\right)\right). \tag{37}$$

By the choice of $\varepsilon_0^{(i)}$ as an inverse of the function $f_{n_i,\delta}$, we have that $\varepsilon_i = \varepsilon$ for all $i \in [\log T]$.

We observe that for any neighboring datasets D, D', there is only one user data that is different between D, D'. That user appears in exactly one batch. It follows that Algorithm 3 is (ε, δ)-DP.

B. Regret Analysis

We next prove the regret bound of Algorithm 3 for stochastic linear bandits in the shuffled model. Our proof is similar to the proofs of the LDP Algorithm presented in Section VIII-B. Let $\tilde{R}_T$ be the regret defined in (13). Let $\mathcal{G}$ be the good event $\left\{\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| < \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right\}$. Lemma 5 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof we condition on the event $\mathcal{G}$. When $q^i < Bd$, the regret can be bounded by Bd. By following similar steps as in the central DP, we can show that there is universal constants C' such that the total regret in (13) can be bounded as

$$\begin{aligned}
\tilde{R}_T &\le Bd\log(T) + \sum_{i=1}^{\log T} 4n_i\gamma_{i-1} \\
&\overset{(a)}{\le} Bd\log(T) + \sum_{i=1}^{\log T} 8q^i\sqrt{\frac{4d}{q^{i-1}}\log\left(4KT^2\right)} \\
&\quad + C'\frac{2d}{\varepsilon}\sum_{i=1}^{\log T} 8q\sqrt{\log\left(4KT^2\right)\log(1/\delta)} \\
&\le C\left(d\sqrt{T\log T} + \frac{(d\log T)^{3/2}\sqrt{\log(1/\delta)}}{\varepsilon}\right),
\end{aligned} \tag{38}$$

where step (a) follows from the fact that from the privacy analysis, when $\varepsilon_0^{(i)} \le 1$, we get that $\varepsilon = O(\varepsilon_0^{(i)}\sqrt{\frac{\log(1/\delta)}{n_i}})$.

Hence, following similar steps as in the proof of the central DP algorithm, with probability at least $1 - \frac{1}{T}$ the regret is bounded as

$$R_T \le \tilde{R}_T + 1 \le C\left(d\sqrt{T\log T} + \frac{(d\log T)^{3/2}\sqrt{\log(1/\delta)}}{\varepsilon}\right). \tag{39}$$

This completes the proof of Theorem 3.

X. Conclusion

In this paper, we proposed differentially private algorithms for stochastic linear bandits for different privacy models: central DP, local DP, and shuffled DP models. We show that our proposed Algorithms are order optimal and match the existing lower bounds up to logarithmic factors. In addition, we extended our algorithms to stochastic contextual bandits with known context distribution. We believe that this idea opens new techniques to design private algorithms for contextual bandits with unknown context distribution and adversarial contexts, which we leave as a future direction.

References

[1] J. Mary, R. Gaudel, and P. Preux, "Bandits and recommender systems," in *International Workshop on Machine Learning, Optimization and Big Data*. Springer, 2015, pp. 325–336.

[2] D. Bouneffouf, I. Rish, and G. A. Cecchi, "Bandit models of human behavior: Reward processing in mental disorders," in *International Conference on Artificial General Intelligence*. Springer, 2017, pp. 237–248.

[3] A. N. Rafferty, H. Ying, and J. J. Williams, "Bandit assignment for educational experiments: Benefits to students versus statistical power," in *International Conference on Artificial Intelligence in Education*. Springer, 2018, pp. 286–290.

[4] D. Bouneffouf and I. Rish, "A survey on practical applications of multi-armed and contextual bandits," *arXiv preprint arXiv:1904.10040*, 2019.

[5] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[6] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[7] A. Cheu, A. D. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Advances in Cryptology - EUROCRYPT 2019*, vol. 11476. Springer, 2019, pp. 375–403.

[8] 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 *SODA*. SIAM, 2019, pp. 2468–2479.

[9] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Annual International Cryptology Conference*. Springer, 2019, pp. 638–667.

[10] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022, pp. 954–964.

[11] A. M. Girgis, D. Data, S. Diggavi, A. T. Suresh, and P. Kairouz, "On the renyi differential privacy of the shuffle model," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2321–2341.

[12] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 395–411, 2010.

[13] R. Shariff and O. Sheffet, "Differentially private contextual linear bandits," vol. 31, 2018.

[14] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," *Advances in neural information processing systems*, vol. 24, 2011.

[15] Y. Han, Z. Liang, Y. Wang, and J. Zhang, "Generalized linear bandits with local differential privacy," vol. 34, 2021.

[16] E. Garcelon, K. Chaudhuri, V. Perchet, and M. Pirotta, "Privacy amplification via shuffling for linear contextual bandits," in *International Conference on Algorithmic Learning Theory*. PMLR, 2022, pp. 381–407.

[17] S. R. Chowdhury and X. Zhou, "Shuffle private linear contextual bandits," *arXiv preprint arXiv:2202.05567*, 2022.

[18] T. Sajed and O. Sheffet, "An optimal private stochastic-mab algorithm based on optimal private stopping rule," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5579–5588.

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

[20] J. Tenenbaum, H. Kaplan, Y. Mansour, and U. Stemmer, "Differentially private multi-armed bandits in the shuffle model," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] K. Zheng, T. Cai, W. Huang, Z. Li, and L. Wang, "Locally differentially private (contextual) bandits learning," vol. 33, 2020, pp. 12 300–12 310.

[22] O. Hanna, L. Yang, and C. Fragouli, "Learning from distributed users in contextual linear bandits without sharing the context," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11 049–11 062, 2022.

[23] O. A. Hanna, A. M. Girgis, C. Fragouli, and S. N. Diggavi, "Differentially private stochastic linear bandits: (almost) for free," *CoRR*, vol. abs/2207.03445, 2022, posted July 13, 2022 on arxiv. [Online]. Available: https://doi.org/10.48550/arXiv.2207.03445

[24] F. Li, X. Zhou, and B. Ji, "Differentially private linear bandits with partial distributed feedback," in *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. IEEE, September 2022, pp. 41–48.
[25] J. He, J. Zhang, and R. Zhang, "A reduction from linear contextual bandit lower bounds to estimation lower bounds," in *International Conference on Machine Learning*. PMLR, July 2022, pp. 8660–8677.
[26] 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.
[27] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.
[28] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.
[29] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
[30] O. A. Hanna, L. Yang, and C. Fragouli, "Contexts can be cheap: Solving stochastic contextual bandits with linear bandit algorithms," in *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 2023, pp. 1791–1821.
[31] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg, "Batched bandit problems," *The Annals of Statistics*, pp. 660–681, 2016.
[32] H. Esfandiari, A. Karbasi, A. Mehrabian, and V. Mirrokni, "Regret bounds for batched bandits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 7340–7348.
[33] A. Cheu, M. Joseph, J. Mao, and B. Peng, "Shuffle private stochastic convex optimization," *arXiv preprint arXiv:2106.09805*, 2021.
[34] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Proceedings of IEEE 36th annual foundations of computer science*. IEEE, 1995, pp. 322–331.
[35] X. Chen, D. Simchi-Levi, and Y. Wang, "Privacy-preserving dynamic personalized pricing with demand learning," *Management Science*, vol. 68, no. 7, pp. 4878–4898, 2022.

Osama Hanna received the B.S. and M.S. degrees in electrical engineering from the Faculty of Engineering, Cairo University in 2014, and Nile University, Egypt, in 2018 respectively. He is currently a research scientist at Meta, Menlo Park, CA, USA, and a Ph.D. candidate at the Electrical and Computer Engineering Department, University of California at Los Angeles, Los Angeles (UCLA). His research interests are machine learning, information theory, and algorithms. He received the Award of Excellence from Cairo University in 2014, the Masters Fellowship and a Graduate Research Assistantship from Nile University in 20142018, and the Electrical and Computer Engineering Department Fellowship from UCLA in 20182019.

Antonious M. Girgis is currently a research scientist at Google, Mountain View, CA, USA. He received the B.Sc. degree in electrical engineering from Cairo University, Egypt, in 2014, the M.Sc. degree in electrical engineering from Nile University, Egypt, in 2018, and the Ph.D. degree in the electrical and computer engineering from the University of California, Los Angeles (UCLA), in 2023. He was the receipt of the 2021 ACM Conference on Computer and Communications Security (CCS) best paper award, and the receipt of the distinguished Ph.D. dissertation award in signals and systems from the ECE department, UCLA. He was an Exchange Research Assistant at Sabanci University, Turkey, from 2016 to 2017. He received the Masters Fellowship and a graduate Research Assistantship from Nile University for the years 2014-2018. He received the Electrical and Computer Engineering Department Fellowship from UCLA for the year 2018/2019, and the 2022 Amazon Ph.D fellowship. His research interests include privacy, machine learning, information theory, and optimization.

Christina Fragouli (Fellow, IEEE) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in electrical engineering from UCLA. She is a Professor with the Electrical and Computer Engineering Department, University of California at Los Angeles (UCLA). She has worked with the Information Sciences Center, AT&T Labs, Florham Park, NJ, USA, and the National University of Athens. She also vis- ited Bell Laboratories, Murray Hill, NJ, USA, and DIMACS, Rutgers University. From 2006 to 2015, she was an Assistant and an Associate Professor with the School of Computer and Communication Sciences, EPFL, Switzerland. Her current research interests include compression for machine learning applications, coding techniques, wireless networks, and network security. She has served as the 2022 IEEE Information Theory Society President, an Information Theory Society Distinguished Lecturer, and an Associate Editor for IEEE COMMUNICATIONS LETTERS, Journal on Computer Communication (Elsevier), IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION THEORY, and IEEE TRANSACTIONS ON MOBILE COMMUNICATIONS. She has served in multiple IEEE committees, and received awards for her work.

Suhas Diggavi is currently a Professor of Electrical and Computer Engineering at UCLA. His undergraduate education is from IIT, Delhi and his PhD is from Stanford University. He has worked as a principal member research staff at AT&T Shannon Laboratories and directed the Laboratory for Information and Communication Systems (LICOS) at EPFL. At UCLA, he directs the Information Theory and Systems Laboratory.

His research interests include information theory and its applications to several areas including machine learning, security & privacy, wireless networks, data compression, cyber-physical systems, bio-informatics and neuroscience; more information can be found at http://licos.ee.ucla.edu.

He has received several recognitions for his research from IEEE and ACM, including the 2013 IEEE Information Theory Society & Communications Society Joint Paper Award, the 2021 ACM Conference on Computer and Communications Security (CCS) best paper award, the 2013 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) best paper award, the 2006 IEEE Donald Fink prize paper award among others. He was selected as a Guggenheim fellow in 2021. He also received the 2019 Google Faculty Research Award, 2020 Amazon faculty research award and 2021 Facebook/Meta faculty research award. He served as a IEEE Distinguished Lecturer and also served on board of governors for the IEEE Information theory society (2016-2021). He is a Fellow of the IEEE.

He is the Editor-in-Chief of the IEEE BITS Information Theory Magazine and has been an associate editor for IEEE Transactions on Information Theory, ACM/IEEE Transactions on Networking and other journals and special issues, as well as in the program committees of several IEEE conferences. He has also helped organize IEEE and ACM conferences including serving as the Technical Program Co-Chair for 2012 IEEE Information Theory Workshop (ITW), the Technical Program Co-Chair for the 2015 IEEE International Symposium on Information Theory (ISIT) and General co-chair for ACM Mobihoc 2018. He has 8 issued patents.

Supplementary Material

Appendix A
Frank-Wolfe Algorithm for Core set

To find the core set in our algorithms we use the FrankWolfe algorithm which starts with an initial distribution π_0 that is uniform over $\mathcal{A}$ and updates it according to

$$\pi_{k+1}(a) = (1-\gamma_k)\pi_k(a) + \gamma_k \mathbb{I}\{a_k = a\},\ a_k = \arg\max_{a\in\mathcal{A}} \|a\|^2_{V(\pi_k)^{-1}},$$

$$\gamma_k = \frac{(1/d)\|a_k\|^2_{V(\pi_k)^{-1}} - 1}{\|a_k\|^2_{V(\pi_k)^{-1}} - 1},\ V(\pi_k) = \sum_{a\in\mathcal{A}} \pi(a) a a^\top.$$

The algorithm will terminate when $\|a\|^2_{V(\pi_k)^{-1}} \le 2d\ \forall a \in \mathcal{A}$. The complexity of the Frank-Wolfe algorithm is $\tilde{O}(|\mathcal{A}|d^3 + d^4)$, which is polynomial in the number of actions. Please note that from the termination condition we have that $\|a_k\|^2_{V(\pi_k)^{-1}} > 2d$ at each iteration, which implies that γ_k is always positive.

Appendix B
Proof of Lemma 4

Lemma. *Let $x_i = l_i z_i$ for $i \in [n]$, where $z_i \sim \mathsf{Lap}(1/b)$ and l_i, c are constants such that $c^2 \ge \sum_{i=1}^n |l_i|^2$. Let $\bar{x} = \sum_{i=1}^n x_i$. We have that*

$$\Pr[\bar{x} \ge t] \le \begin{cases} \exp\left(-\frac{t^2 b^2}{2c^2}\right) & \text{if } \ t \le \frac{c^2}{b l_{\max}} \\ \exp\left(\frac{c^2}{2l^2_{\max}} - \frac{b}{l_{\max}} t\right) & \text{if } \ t > \frac{c^2}{b l_{\max}} \end{cases}, \tag{40}$$

where $l_{\max} = \max_i l_i$.

Proof. The proof follows from the concentration results of the Laplace distribution (e.g., see). We have that

$$\begin{aligned} \Pr[\bar{x} \ge t] &= \Pr\left[\exp(\lambda \bar{x}) \ge e^{\lambda t}\right] && \forall\ \lambda \ge 0 \\ &\overset{(a)}{\le} \frac{\mathbb{E}[\exp(\lambda \bar{x})]}{e^{\lambda t}} \\ &\overset{(b)}{=} \frac{\prod_{i=1}^n \mathbb{E}\left[e^{\lambda x_i}\right]}{e^{\lambda t}} \\ &\overset{(c)}{\le} \frac{\prod_{i=1}^n e^{\lambda^2 \frac{l_i^2}{2b^2}}}{e^{\lambda t}} && \forall\ 0 \le \lambda \le \frac{b}{l_{\max}} \\ &= \frac{e^{\lambda^2 \frac{c^2}{2b^2}}}{e^{\lambda t}} && \forall\ 0 \le \lambda \le \frac{b}{l_{\max}} \end{aligned} \tag{41}$$

where $l_{\max} = \max_i l_i$, step (a) follows from Markov's inequality and step (b) follows from the fact that $z_1, \ldots, z_n$ are independent Laplace random variables. Step (c) follows from the fact that z_i is sub-exponential random variable with proxy $\frac{l_i^2}{2b^2}$. By choosing $\lambda = \frac{tb^2}{c^2}$ when $t < \frac{c^2}{b l_{\max}}$ and $\lambda = \frac{b}{l_{\max}}$ when $t > \frac{c^2}{b l_{\max}}$, we get that

$$\Pr[\bar{x} \ge t] \le \begin{cases} \exp\left(-\frac{t^2 b^2}{2c^2}\right) & \text{if } \ t \le \frac{c^2}{b l_{\max}} \\ \exp\left(\frac{c^2}{2l^2_{\max}} - \frac{b}{l_{\max}} t\right) & \text{if } \ t > \frac{c^2}{b l_{\max}} \end{cases} \tag{42}$$

This completes the proof of Lemma 4. ■

Appendix C
Alternative Algorithm for the Shuffled Model

In this subsection we present an alternative scheme for the shuffled model that achieves the same regret as Algorithm 3. The algorithm uses the shuffled protocol for summing scalars presented in [33]. The pseudo-code is presented in Algorithm 4.

Theorem 4. *Algorithm 4 is (ε, δ)-differentially private. Moreover, for $\varepsilon = O(\sqrt{\frac{\log(1/\delta)}{T}})$ it achieves a regret*

$$R_T \le C\left(\sqrt{T \log T} + \frac{\sqrt{\log(1/\delta)}\log^2 T}{\varepsilon}\right), \tag{43}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε and T.

Algorithm 4 DP algorithm for stochastic linear bandits: shuffled model

Input: set of actions $\mathcal{A}$, time horizon T, and privacy parameters (ε, δ).
Let $\mathcal{A}_1$ be a ζ-net for $\mathcal{A}$ as in Lemma 1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ **do**
  Let $\mathcal{C}_i$ be a core set for $\mathcal{A}_i$ as in Lemma 2 with distribution π_i.
  Let $n_{ia} = \lceil \pi_i(a) q^i \rceil, n_i \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia}$.
  for $a \in \mathcal{C}_i$ **do**
    Let n_{ia} users pull arm a and observe a reward $r_{ia}^j, j = 1, ..., n_{ia}$.
    Use shuffled protocol for summing scalars in [33] to compute the private sum $\hat{r}_{ia}$.
  $\gamma_i \leftarrow \sqrt{\frac{4d}{q^i} \log\left(4|\mathcal{A}_i|T^2\right)} + \frac{2Bd^2 + 2d\log\left(4|\mathcal{A}_i|T^2\right)}{\varepsilon q^i} \sqrt{180 \log(2/\delta)}$.
  $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top, \ \hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
  $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$.
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.

Proof. The privacy proof follows from Lemma 3.1 in [33]. The regret analysis follows similar steps as in the proofs of the central DP Algorithm presented in Section VII-B. Let R_T be the regret defined in (13).
Let $\mathcal{G}$ be the good event $\left\{\left|\langle a, \hat{\theta}_i - \theta_* \rangle\right| < \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right\}$. Lemma 6 below shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof we condition on the event $\mathcal{G}$.

Lemma 6. *Let $\hat{\theta}_i$ be the least square estimate of θ_* at the end of the ith batch of Algorithm 2. Then, we have that*

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_* \rangle\right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \tag{44}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i} \log\left(4|\mathcal{A}_i|T^2\right)} + \frac{2Bd^2 + 2d\log\left(4|\mathcal{A}_i|T^2\right)}{\varepsilon q^i} \sqrt{180 \log(2/\delta)}$.

Proof. Let $\hat{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$ be the private estimate of θ_* and $\bar{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} \bar{r}_{ia} a$ be the non-private estimate of θ_* as $\{\bar{r}_{ia}\}$ are the non-private rewards, where $V_i = \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$ and $\hat{r}_{ia} = \sum_{j=1}^{n_{ia}} \hat{r}_{ia}^{(j)}$. Similar to the central DP in Section III, we have that

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \theta_* \rangle\right| \geq \sqrt{\frac{4d}{q^i} \log\left(4KT^2\right)}\right] \leq \frac{1}{2KT^2}, \tag{45}$$

for each $a \in \mathcal{A}_i$. To bound the effect of privacy in the shuffled model in estimating θ_*, we bound the difference $\langle a, \bar{\theta}_i - \hat{\theta}_i \rangle$. By letting $z_{ia} = \hat{r}_{ia} - \bar{r}_{ia}$, we can write $\hat{\theta}_i - \bar{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} z_{ia} a$. Thus, for any $\alpha \in \mathcal{A}_i$, we have that:

$$\langle \alpha, \hat{\theta}_i - \bar{\theta}_i \rangle = \sum_{a \in \mathcal{C}_i} \alpha^\top V_i^{-1} a z_{ia}, \tag{46}$$

where $\alpha^\top V_i^{-1} a \leq \max_{b \in \mathcal{A}_i} \|b\|_{V_i^{-1}}^2 \leq \frac{2d}{q^i}$ for each $a \in \mathcal{C}_i$ that holds from the fact that V_i is positive semi-definite. From Algorithm 1 in [33], the noise z_{ia} is zero mean $(\frac{180 \log(2/\delta)}{\varepsilon^2})$-subgaussian random variable. The result follows similar to Lemma 3 using the concentration of subgaussian random variables, union bound and triangle inequality. ■

Following the remaining steps similar to the analysis of the central case, we get that

$$R_T \leq C \left(\sqrt{T \log T} + \frac{\sqrt{\log(1/\delta)} \log^2 T}{\varepsilon} \right), \tag{47}$$

with probability at least $1 - \frac{1}{T}$. ■