Improved rate for Locally Differentially Private Linear Bandits

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we propose a stochastic linear contextual bandit algorithm that ensures local differential privacy (LDP). Our algorithm is $(\epsilon, \delta)-$Locally Differentially Private and guarantees $\tilde{O}\left(\sqrt{d}T^{3/4}\right)$ regret with high probability . This is a factor of $d^{1/4}$ improvement over the previous state-of-the-art (SOTA)(Zheng et al., 2020). Furthermore, our regret guarantee improves to $\tilde{O}\left(\sqrt{dT}\right)$ when the action space is well-conditioned. This rate matches the optimal non-private asymptotic rate, thus demonstrating that we can achieve privacy for free even in the stringent LDP model. Our algorithm is the first algorithm that achieves $\tilde{O}(\sqrt{T})$ regret in a privacy setting that is stronger than the central settings.

1 Introduction

The stochastic linear contextual bandit problem consists of a sequence of T "rounds" of interaction between a learner and an environment. In the tth round, a learner receives context c_t, which determines a *decision set* $D_t \coloneqq \{\phi(c_t, a)|a \in \mathcal{A}\} \subset \mathbb{R}^d$ where $\mathcal{A}$ is a set of possible actions and $\phi(\cdot, \cdot)$ is a function that maps context-action pairs to $\mathbb{R}^d$. Then the learner chooses an "action" $a_t \in \mathcal{A}$ which corresponds to a "decision" $x_t \coloneqq \phi(c_t, a_t) \in D_t$, and receives a reward y_t such that $\mathbb{E}[y_t|c_t, a_t] = \langle \theta^\star, x_t \rangle$ for some unknown $\theta^\star \in \mathbb{R}^d$. Similar to other bandit settings, we measure the performance of our algorithm by evaluating the regret, defined as the gap between the cumulative rewards of our algorithm and the best possible cumulative reward:

$$\begin{aligned} Regret_T &= \sum_{t=1}^{T} \left[\max_{a \in \mathcal{A}} \langle \theta^\star, \phi(c_t, a) \rangle - \langle \theta^\star, \phi(c_t, a_t) \rangle \right] \\ &= \sum_{t=1}^{T} \left[\max_{x \in D_t} \langle \theta^\star, x \rangle - \langle \theta^\star, x_t \rangle \right] \end{aligned}$$

Our goal is to come up with an algorithm that achieves sublinear regret ($R_T \leq o(T)$), which means that on average we are doing as well as the best possible actions in hindsight. This problem has been well-studied in the literature, and the optimal regret is $O(d\sqrt{T})$ (Lattimore & Szepesvári, 2020; Li et al., 2019). Furthermore, we want to ensure that our algorithm enforces a *privacy* guarantee - which we will quantify via the framework of local differential privacy (LDP).

To motivate this contextual bandit setting and the need to ensure privacy, consider a personalized medical app where each user has their own treatment plan based on their medical history and the weekly data they provide to the central server (app provider). This application can be modeled as a contextual bandit problem by letting the medical history/weekly data be the context, the treatment plan be the action, and the user's health outcome be the reward. It is clear that the context/action and the reward are sensitive information that the user wishes to protect. Thus, in order to maximize health outcomes we need a contextual bandit algorithm, while in order to protect sensitive information we need the algorithm to ensure privacy.

Our work will ensure a "local" model of privacy, in contrast to a weaker "central" model. In a central privacy model, the users send their raw data to the central server that will be responsible for making decisions. Then the server would inject sufficient noise into its decisions so that potential attackers cannot tell if a particular user's data is used by the server or not. Even though this central model provides protection from outside

attackers, it requires the user to have complete trust in the server. Another way to protect users' privacy is the local model, which is the model that we consider in this paper. In this model, before sending their data to the server, each user would inject noise into their own data. Thus, the local model ensures that every user's data is safe without relying on any external sources. Consider a framework with T local users, each holding their private data, and a central server that collects information from these users. Each user applies a randomized mechanism to their data before sending it to the server. The system satisfies $(\epsilon, \delta)-$Locally Differential Privacy (LDP) if the following condition holds:

Definition 1.1. (Local Differential Privacy Dwork & Roth (2014)) A randomized algorithm $M\colon \mathcal{X} \mapsto \mathcal{S}$ satisfies $(\epsilon, \delta)-$ local differential privacy ((ϵ, δ)-LDP) if for any pair of users $x, x' \in \mathcal{X}$ and any event $E \subseteq S$, it holds that:

$$P\left[M(x) \in E\right] \leq \exp(\epsilon) P\left[M(x') \in E\right] + \delta$$

Roughly speaking, LDP ensures that the output of a randomized algorithm using any pair of users would be *almost* indistinguishable with high probability. This local model is a stronger notion and more user-friendly notion of privacy than regular DP, making it appealing for real-life application Cormode et al. (2018). However, it is also a lot more difficult to recover the asymptotically optimal $\tilde{O}(\sqrt{T})$ regret guarantee since the private mechanism in the central model Shariff & Sheffet (2018) fails in the local model. Indeed, the current best regret guarantee for LDP is only $\tilde{O}\left((dT)^{3/4}/\sqrt{\epsilon} + d\sqrt{T}\right)$ Zheng et al. (2020). Another line of work on private stochastic contextual linear is the shuffle model Erlingsson et al. (2019); Cheu et al. (2019). In this model, there exists a trusted shuffler between the users and the server. The shuffler receives noisy data from the users, and permutes them before sending the data to the server. This shuffling step adds another layer of protection which allows for finer privacy-utility trade-offs compared to the local model. In this model, recent works by Chowdhury & Zhou (2022) and Garcelon et al. (2022) achieve the regret of $\tilde{O}(dT^{3/5})$ and $\tilde{O}(dT^{2/3})$ respectively. However, both of these works fail to achieve the $\tilde{O}\left(\sqrt{T}\right)$ regret in any setting. Furthermore, since both works rely on privacy amplification by shuffling, their guarantees only work for $\epsilon \leq O(1/T^{3/10})$ and $\epsilon \leq O(1/T^{1/4})$ respectively, which are a lot smaller than what is used in practice (which is typically $O(1)$). Therefore, a question naturally arises:

Is it possible to achieve $\tilde{O}\left(\sqrt{T}\right)$ regret in a stronger privacy setting than the central model?

In this paper, we will provide sufficient conditions under which the answer to the above question is *yes*.

Contributions. We introduce a new private variant of the LinUCB algorithm (Chu et al., 2011; Abbasi-yadkori et al., 2011) where the confidence set is constructed using the predictions of an online learner (Abbasi-Yadkori et al., 2012). By carefully choosing the online learner and the loss function, this new approach allows us to construct tighter confidence sets for the unknown parameter $\theta^\star$ which results in a $O\left(\sqrt{d}T^{3/4}/\epsilon\right)$ regret with high probability, improving the best known bound of $\tilde{O}\left((dT)^{3/4}/\sqrt{\epsilon}\right)$ (Zheng et al., 2020) whenever $\epsilon \geq \frac{1}{\sqrt{d}}$. Further, when the minimum eigenvalue of the expected gram matrix $\mathbb{E}_t[x_t x_t^T]$ is bounded from below, the regret guarantee of the new algorithm improves to $\tilde{O}\left(\sqrt{dT}/\epsilon\right)$, recovering the asymptotic regret guarantee of the central model while guaranteeing LDP. This regret, to the best of our knowledge, is the first $\tilde{O}(\sqrt{T})$ regret guarantee for LDP stochastic contextual linear bandit in any setting. Finally, we test our algorithm in the same experiments as in (Chowdhury & Zhou, 2022) and show that our algorithm has empirical improvements over previous works.

2 Problem Setup

Let $\mathcal{A}$ be an action space and $\mathcal{C}$ a context space. In the stochastic contextual linear bandit setting that we are considering, in every round $t \in [T]$, the learner receives an i.i.d random context $c_t \in \mathcal{C}$ and a function $\phi(\cdot, \cdot) : \mathcal{C} \times \mathcal{A} \mapsto \mathbb{R}^d$, and picks an action a_t corresponding to $x_t \coloneqq \phi(c_t, a_t) \in \mathcal{D}_t$. The learner then receives a random reward $y_t = \langle \theta^\star, x_t \rangle + \eta_t$ where η_t is zero-mean and independent R^2- subgaussian random variable.

Algorithm 1 Private (Contextual) Online LinUCB

Input: Privacy parameters ϵ, δ, failure parameter β, covariance matrix $\Sigma = \mathbb{E}[\eta_{x,t}\eta_{x,t}^T]$, minimum eigenvalue λ_{min} of $\mathbb{E}[x_t x_t^T]$, domain diameter D, time horizon T, universal constant C, threshold $\bar{\lambda}$.
Initialize $\theta_1 = 0$, $\tilde{V}_0 = I_{d\times d}$, $\tilde{u}_0 = 0$, $\sigma = 2\sqrt{2\log(1.25/\delta)}/\epsilon$, $\Delta^2 = 0$.
if $\lambda_{min} \leq \bar{\lambda}$ **then**
  $\Delta^2 = \bar{\lambda}$
end if
Set clipping constant $G = 2\sqrt{2\log(2T/\beta)}(2D\sigma + 2\sigma + D\Delta + \Delta) + 2\log(2T/\beta)(2D\Delta^2 + 4D\Delta\sigma + 2\Delta\sigma + 2D\sigma^2 + 2\sigma^2 + DC\sigma^2) + 2D + 2$.
for $t = 1 \dots T$ **do**
  Local step performed by user t:
  Receive θ_t, $\tilde{V}_{t-1}$, $\tilde{u}_{t-1}$ from the server. Construct the confidence set C_{t-1} using Lemma 3.3.
  $(x_t, \tilde{\theta}_t) = \arg\max_{(x,\theta)\in D_t \times C_{t-1}} \langle x, \theta\rangle$
  Play x_t and observe reward y_t
  Perturb x_t with a small amount of noise: $\bar{x}_t \leftarrow x_t + \zeta_t$ where $\zeta_t \sim N\left(0, \Delta^2 I_d\right)$.
  Update $\tilde{x}_t = \bar{x}_t + \eta_{x,t}$, $\tilde{y}_t = y_t + \eta_{y,t}$ where $\eta_{x,t} \sim N(0, \sigma^2 I_d)$, $\eta_{y,t} \sim N(0, \sigma^2)$.
  Get the loss $l_t(\theta) = (\langle \tilde{x}_t, \theta_t\rangle - \tilde{y}_t)^2 - \|\theta\|_\Sigma^2$
  Compute $g_t = 2\tilde{x}_t\left(\langle \tilde{x}_t, \theta_t\rangle - \tilde{y}_t\right) - 2\Sigma\theta_t$
  Send $\tilde{x}_t$, $\tilde{y}_t$, and $g_t^{clip} = \frac{g_t}{\|g_t\|}\min\{G, \|g_t\|\}$ to the server.
  Server step:
  Sent θ_t, $\tilde{x}_t$, $\tilde{y}_t$, and g_t^{clip} to Maler (Algorithm 3) and get back θ_{t+1}.
  Update the history $\{\theta_1, \dots, \theta_t\} \cup \{\theta_{t+1}\}$
  Update $\tilde{V}_t = \tilde{V}_{t-1} + \tilde{x}_t\tilde{x}_t^T$, $\tilde{u}_t = \tilde{u}_{t-1} + \langle \theta_t, \tilde{x}_t\rangle \tilde{x}_t$
end for

Then we define the regret as:

$$Regret_T = \sum_{t=1}^{T}\left[\max_{x\in D_t}\langle \theta^\star, x\rangle - \langle \theta^\star, x_t\rangle\right]$$

For $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ as $[n]$. We use the standard big-O notation to hide constants and $\tilde{O}$ to hide additional logarithmic terms. Throughout the paper, $\|\cdot\|$ is used to indicate the Euclidean norm unless specified otherwise. A symmetric matrix $M \in \mathbb{R}^{d\times d}$ is a positive-semidefinite matrix if $x^T M x \geq 0$ for any $x \in \mathbb{R}^d$ and we define its associated norm as $\|x\|_M = \sqrt{x^T M x}$. We also define $\mathbb{E}_x[\cdot]$ as the expectation over the randomness of some random variable x and $\log(x)$ as the natural logarithm of x.

Assumption: We assume the reward, and the norm of the parameter $\theta^\star$ and feature map $x_t = \phi(c_t, a_t)$ are all bounded: $\|x_t\| \leq 1$, $|y_t| \leq 1$, $|\langle x_t, \theta^\star\rangle| \leq 1$ for all $t \in [T]$, and $\|\theta^\star\| \leq D$. Note that these are all standard assumptions from the literature Shariff & Sheffet (2018); Chowdhury & Zhou (2022).

3 Online LDP LinUCB

Our private method described in Algorithm 1 is a private variant of the LinUCB algorithm (Chu et al., 2011; Abbasi-yadkori et al., 2011). The main task of the algorithm is to derive an ellipsoid confidence set defined as

$$C_{t-1} \coloneqq \left\{\theta \in \mathbb{R}^d : \|\theta - V_{t-1}^{-1}u_{t-1}\|_{V_{t-1}} \leq \beta_t\right\} \tag{1}$$

where $V_t = \sum_{i=1}^{t} x_t x_t^T$, β_t is the width of the confidence set, and $u_{t-1} = \sum_{i=1}^{t-1} x_i y_i$. For appropriate β_t, the optimal parameter $\theta^\star$ is inside the ellipsoid with high probability for all $t \in [T]$. LinUCB identifies $x_t \in D_t$ and $\theta_t \in C_{t-1}$ that maximizes $\langle x, \theta\rangle$ and plays x_t. Overall, LinUCB guarantees the following regret:

$$Regret_T \leq \tilde{O}\left(\max_t \beta_t\sqrt{dT}\right)$$

Thus, as long as we can design a tight confidence ellipsoid (small β_t), our linear bandit algorithm will have a small regret.

To ensure privacy, we unfortunately cannot update our algorithm with the true value of V_t and u_t. Instead, we have to use private approximations $\tilde{V}_t$ and $\tilde{u}_t$ to define an analogous $\tilde{C}_{t-1}$. Let $\tilde{V}_t = V_t + H_t$ and $\tilde{u}_t = u_t + h_t$. Assuming $\|H_t\| \leq \rho_{max}$, $\|h_t\|_{H_t^{-1}} \leq \nu$ for some $\rho_{max}, \nu \geq 0$, then from (Shariff & Sheffet, 2018), we know that the confidence width is bounded by:

$$\beta_t \leq \tilde{O}\left(\sqrt{d} + \sqrt{\rho_{max}} + \nu\right) \tag{2}$$

Since higher β_t leads to higher regret, we would like to minimize the error measures ρ_{max} and ν introduced by the private approximations.

We can now shed light on why the regret guarantees for local models are worse than for the central model. In the central model, since the server is allowed to see the raw data x_t and y_t, the server can compute $\tilde{V}_t$ and $\tilde{u}_t$ privately using the tree-aggregation mechanism (Chan et al., 2011; Dwork et al., 2010). Then, we have $\rho_{max} \leq \tilde{O}(\sqrt{d}/\epsilon)$ and $Regret_T \leq \tilde{O}(d\sqrt{T} + d^{3/4}\sqrt{T}/\sqrt{\epsilon})$. However, in the local model, since each user perturbs their data before sending them to the server, applying tree-aggregation is off the table. If we naively apply the Gaussian Mechanism with T rounds of compositions, ρ_{max} now is $\tilde{O}(\sqrt{dT}/\epsilon)$ and the regret becomes $\tilde{O}(d\sqrt{T} + d^{3/4}T^{3/4}/\sqrt{\epsilon})$.

In this section, we propose a new approach for designing the confidence sets LDP LinUCB. Our approach is based on the *online-to-confidence-set conversion* in (Abbasi-Yadkori et al., 2012) where the main idea is that the predictions of any online algorithm that predicts the responses of the chosen inputs in a sequential manner can be "converted" to a confidence set. In each round t, the online algorithm will receive x_t, y_t, predict θ_t, and suffer the loss $l_t(\theta_t) = (\langle \theta_t, x_t \rangle - y_t)^2$. The goal of the online learner is to discover the "true" value $\theta_\star$. We measure its performance via its own notion of regret (Regret_{OL}), and we define M_T to be a known upper-bound on Regret_{OL}.

$$\text{Regret}_{OL} \triangleq \sum_{t=1}^{T} l_t(\theta_t) - l_t(\theta^\star) \tag{3}$$

$$M_T \geq \text{Regret}_{OL} \tag{4}$$

Intuitively, a low regret means that the online learner is able to predict a good approximate of the optimal $\theta^\star$. Now, Abbasi-Yadkori et al. (2012) show how to use the bound M_T to construct a confidence set with the width bounded by:

$$\beta_T \leq \tilde{O}\left(\sqrt{M_T}\right)$$

Since the width of the confidence ellipsoid depends on the regret of the online learner, one could hope that with carefully designed online learner and loss function, the confidence width would be small and we can see improvements in the final regret bound for LinUCB. We will now show that this is indeed the case and Algorithm 1 using this approach can achieve $\tilde{O}(\sqrt{dT})$ regret when the second-moment matrix $\mathbb{E}[x_t x_t^T] \succeq \lambda_{min} I$ for some $\lambda_{min} \approx O(1)$.

Our algorithm elaborates on this strategy with two key ideas. First, instead of using the intuitive square loss $l_t(\theta_t) = (\langle \theta_t, x_t \rangle - y_t)^2$, we use the more peculiar choice $l_t(\theta) = (\langle \tilde{x}_t, \theta_t \rangle - \tilde{y}_t)^2 - \|\theta\|_\Sigma^2$ for some to-be-specified Σ. Second, we employ the advanced online learning algorithm Maler (Wang et al., 2020b) as the online learner.

The reason for the choice of the loss is a bit technical. Intuitively, if we want the online learner to accurately approximate $\theta^\star$, we want $\theta^\star$ to be the minimizer of the loss $l_t(\theta)$. However, due to the noise in x_t and y_t, $\theta^\star$ is not the minimizer of the square loss $(\langle \tilde{x}_t, \theta_t \rangle - \tilde{y}_t)^2$. To counteract this issue, we incorporate a negative regularizer term, $-\|\theta\|_\Sigma^2$, which serves to neutralize the variance introduced by the privacy noise in x_t. Now, with the added regularizer, $\theta^\star$ becomes the minimizer of $\mathbb{E}[l_t(\theta)]$. At first glance, this new loss appears to be intractable because it is non-convex. However, it is convex *in expectation*, which is sufficient to guarantee an $O(\sqrt{T})$ regret. This in turn translates to $\tilde{O}(T^{3/4})$ for the final regret bound.

The online learner can potentially do even better than $O(\sqrt{T})$ regret in certain favorable settings. Specifically, when $\mathbb{E}[x_t x_t^T] \succeq \lambda_{min} I$ for $\lambda_{min} > 0$, $l_t(\theta)$ is *strongly convex in expectation*, despite not being convex. This necessitates the use of an online learner capable of adapting to such advantageous scenarios and refining the regret to $O(\log T)$. This is precisely the scenario where Maler proves invaluable. Maler (described in Algorithm 3) is an online learner that adjusts to achieve the optimal regret across various types of loss functions, including convex, strongly convex, and exp-concave. We demonstrate that implementing Maler with our loss function yields a dimension-independent regret of $O(\log T)$ with high probability. This allows us to attain a regret of $\tilde{O}(\sqrt{dT})$,which is not only asymptotically optimal but also improves upon the non-private worst-case regret guarantee for general settings by an order of $O(\sqrt{d})$ (see Section 3.2 for more discussion).

Remark 3.1. Let us discuss one specific example when the condition $\mathbb{E}[x_t x_t^T] \succeq \lambda_{min} I$ is satisfied, giving our algorithm the optimal regret guarantee of $\tilde{O}(\sqrt{dT})$. Assuming we have k available actions, and let each action be $x_i \sim N(0, \sigma^2 I_d)$. Then, for the condition $\mathbb{E}\left[x_t x_t^T\right] \succeq \lambda_{min} I$ for some $\lambda_{min} \approx O(1)$ to be true, we need to show that $\mathbb{E}\left[v^T x_t x_t^T v\right] \geq C$ where C is a positive constant and v is a unit vector. Notice that $\mathbb{E}\left[v^T x_t x_t^T v\right] \geq \mathbb{E}\left[\min_{1\leq i\leq k} v^T x_i x_i^T v\right]$, thus if we can show a constant lower bound for $\mathbb{E}\left[\min_{1\leq i\leq k} v^T x_i x_i^T v\right]$ then we are done. We have $v^T x_i \sim N(0, \sigma^2)$ (since v is a unit vector) for every $i \in [k]$, thus by Lemma G.15, $P\left[|x_i| \leq t\right] \leq \frac{\sqrt{2}}{\sigma\sqrt{\pi}} t$ for $t > 0$. Then, we can apply Theorem G.14 to get $\mathbb{E}\left[\min_{1\leq i\leq k} v^T x_i x_i^T v\right] = E\left[\min_{1\leq i\leq k} |\langle v, x_i\rangle|^2\right] \geq \frac{\sigma^2\pi}{6k^2}$. Thus, as long as the number of actions is not too large, this example would fall under our favorable setting.

Before we prove the utility guarantee of Algorithm 1, let us first show that Algorithm 1 is $(\epsilon, \delta)-$LDP.

Theorem 3.2. *(Privacy Guarantee) Algorithm 1 guarantees $(\epsilon, \delta)-$LDP.*

Proof. Let us define the local step of Algorithm 1 as the local mechanism M_t. Let x_t' and y_t' be the action and reward of a new user at time t. By the boundedness assumption, we have $\max_{x_t, x_t' \in \mathcal{X}} \|x_t - x_t'\| \leq 2$ and $\max_{y_t, y_t' \in \mathcal{Y}} |y_t - y_t'| \leq 2$ for all $t \in [T]$. Thus, by the classic Gaussian Mechanism in (Dwork et al., 2010), the outputs $\tilde{x}_t$ and $\tilde{y}_t$ of the local mechanism M_t satisfy $(\epsilon, \delta)-$LDP for all t. Further, since θ_t and g_t^{clip} are computed using a sequence of private parameters $\tilde{x}_1, \tilde{y}_1, \ldots, \tilde{x}_{t-1}, \tilde{y}_{t-1}$, θ_t and g_t^{clip} also satisfy $(\epsilon, \delta)-$LDP by post-processing. Consequently, Algorithm 1 guarantees $(\epsilon, \delta)-$LDP for every user $t \in [T]$, as each local mechanism M_t is $(\epsilon, \delta)-$LDP. □

To make the analysis more succinct and easier to follow, let us define the "good event" $\mathcal{E}$ as in Section A in the Appendix. Roughly speaking, $\mathcal{E}$ is the event in which a small number of standard martingale concentration bounds hold simultaneously. Then, from Lemma A.1, we know that the good event $\mathcal{E}$ happens with high probability. Now, we can show the following result on the confidence set.

Lemma 3.3. *We define $\tilde{V}_{N-1} = \sum_{t=1}^{N-1} \tilde{x}_t \tilde{x}_t^T$, $\tilde{u}_{N-1} = \sum_{t=1}^{N-1} \langle \theta_t, \tilde{x}_t \rangle \tilde{x}_t$ (θ_t is the prediction of the online learner), and $\hat{\theta}_N = \tilde{V}_{N-1}^{-1} \tilde{u}_{N-1}$. Assuming $\|\theta_t\| \leq D$, then under event $\mathcal{E}$, the true parameter $\theta^\star$ lies in the set:*

$$C_{N-1} = \left\{\theta \in R^d : \|\theta - \hat{\theta}_N\|^2_{\tilde{V}_{N-1}} \leq M_N + K_N\right\}$$

for any $N \geq 1$ and

$$K_N = \gamma D \Delta^2 \log(T/\beta) \sqrt{N \sum_{t=1}^{N} \|\theta_t - \theta^\star\|^2} + \gamma \left(R + \frac{D\sqrt{\log(1/\delta)}}{\epsilon} + D\Delta\right)^2 \log(T/\beta)$$
$$+ \gamma \left(\frac{\log(1/\delta)}{\epsilon^2} + \Delta^2\right) \log(T/\beta) \sum_{t=1}^{N} \|\theta_t - \theta^\star\|^2$$

for a sufficient large constant $\gamma > 0$.

We now provide a sketch of the proof of Lemma 3.3 below. For the full proof, refer to section C in the appendix.

Proof. Our proof follows the proof of Theorem 1 in (Abbasi-Yadkori et al., 2012). From our definition of M_N and the loss l_t, we have:

$$M_N \geq \sum_{t=1}^{N} (\langle \tilde{x}_t, \theta_t \rangle - \tilde{y}_t)^2 - \|\theta_t\|_\Sigma^2 - (\langle \tilde{x}_t, \theta^\star \rangle - \tilde{y}_t)^2 + \|\theta^\star\|_\Sigma^2$$

Plugging in $\tilde{y}_t = \langle x_t, \theta^\star \rangle + r_t + \eta_{y,t}$ and $\tilde{x}_t = x_t + \zeta_t + \eta_{x,t}$:

$$= \sum_{t=1}^{N} (\langle x_t, \theta_t - \theta^\star \rangle + \langle \eta_{x,t}, \theta_t \rangle + \langle \zeta_t, \theta_t \rangle - r_t - \eta_{y,t})^2 - \|\theta_t\|_\Sigma^2 - (\langle \eta_{x,t}, \theta^\star \rangle + \langle \zeta_t, \theta^\star \rangle - r_t - \eta_{y,t})^2 + \|\theta^\star\|_\Sigma^2$$

Let us denote $z_t = r_t + \eta_{y,t}$. Now expanding the squares and rearranging the terms we get:

$$\sum_{t=1}^{N} (\langle x_t, \theta_t - \theta^\star \rangle)^2 \leq M_N + \sum_{t=1}^{N} \underbrace{-2(\langle \eta_{x,t} + \zeta_t, \theta_t \rangle - z_t)\langle x_t, \theta_t - \theta^\star \rangle}_{A_t} \underbrace{+2 z_t \langle \eta_{x,t} + \zeta_t, \theta_t - \theta^\star \rangle}_{B_t} \underbrace{-\langle \zeta_t, \theta_t \rangle^2 + \langle \zeta_t, \theta^\star \rangle^2}_{C_t}$$
$$\underbrace{-2(\langle \zeta_t, \theta_t \rangle \langle \eta_{x,t}, \theta_t \rangle - \langle \zeta_t, \theta^\star \rangle \langle \eta_{x,t}, \theta^\star \rangle)}_{D_t} \underbrace{-(\theta_t - \theta^\star)^T (\eta_{x,t} \eta_{x,t}^T - \Sigma)(\theta_t + \theta^\star)}_{E_t} \tag{5}$$

Under event $\mathcal{E}$, we have

$$\begin{aligned} A_t + B_t &= -\sum_{t=1}^{N} 2(\langle \eta_{x,t} + \zeta_t, \theta_t \rangle - z_t)\langle x_t, \theta_t - \theta^\star \rangle + 2 z_t \langle \eta_{x,t} + \zeta_t, \theta_t - \theta^\star \rangle \\ &\leq \tilde{O}\left(\sqrt{\sum_{t=1}^{N} (\langle x_t, \theta_t - \theta^\star \rangle)^2} + \sqrt{\sum_{t=1}^{N} \langle \eta_{x,t} + \zeta_t, \theta_t - \theta^\star \rangle^2} \right) \end{aligned} \tag{6}$$

Notice that the first sum in the right-hand side of Eq.6 is exactly the sum in the left-hand side of Eq.5. Thus, we can use Proposition G.8 and G.9 to bound this term. For the second term in the right-hand side of Eq.6, we can again use the fact that we are under the good event $\mathcal{E}$ to control the sum.

The sum of C_t and D_t can be written as follows:

$$\begin{aligned} \sum_{t=1}^{N} \langle \zeta_t, \theta^\star \rangle^2 - \langle \zeta_t, \theta_t \rangle^2 + 2(\langle \zeta_t, \theta^\star \rangle \langle \eta_{x,t}, \theta^\star \rangle - \langle \zeta_t, \theta_t \rangle \langle \eta_{x,t}, \theta_t \rangle) &= \sum_{t=1}^{N} \langle \zeta_t, \theta^\star - \theta_t \rangle \langle \zeta_t, \theta^\star + \theta_t \rangle \\ &\quad + 2(\theta^\star - \theta_t)^T \zeta_t \eta_{x,t}^T (\theta^\star - \theta_t) + 2\theta_t^T \zeta_t r_t^T (\theta^\star - \theta_t) + 2(\theta^\star - \theta_t)^T \zeta_t r_t^T \theta_t \end{aligned}$$

Using norm bound of Gaussian random vector and corollary G.12:

$$C_t + D_t \leq O\left(D\Delta^2 \log(T/\beta) \sqrt{N \sum_{t=1}^{N} \|\theta_t - \theta^\star\|^2} \right) + O\left(\frac{D\Delta\sqrt{\log(1/\delta)}}{\epsilon} \sqrt{\log(T/\beta) \sum_{t=1}^{T} \|\theta_t - \theta^\star\|^2} \right)$$

For the term E_t in Eq.5, since $\mathbb{E}\left[\eta_{x,t}\eta_{x,t}^T\right] = \Sigma$, it is a Martingale difference sequence and by Theorem G.2 we have:

$$|\sum_{t=1}^{N} (\theta_t - \theta^\star)^T (\eta_{x,t}\eta_{x,t}^T - \Sigma)(\theta_t + \theta^\star)| \leq \tilde{O}\left(\frac{D \log(1/\delta) \log(T/\beta)}{\epsilon^2} \sqrt{\sum_{t=1}^{N} \|\theta_t - \theta^\star\|^2} \right)$$

Now we can combine the bounds of all the terms and use Proposition G.8 to get:

$$\sum_{t=1}^{N} (\langle \tilde{x}_t, \theta_t - \theta^\star \rangle)^2 \leq M_N + K_N$$

Let us denote the set C_{N-1} as the ellipsoid underlying the covariance matrix $\tilde{V}_{N-1} = I + \sum_{t=1}^{N-1} \tilde{x}_t \tilde{x}_t^T$ and centering at

$$\begin{aligned}\hat{\theta}_N &= \arg\min_{\theta \in R^d} \left(\|\theta\|_2^2 + \sum_{t=1}^{N-1} (\langle \tilde{x}_t, \theta_t - \theta \rangle)^2 \right) \\ &= \tilde{V}_{N-1}^{-1} \left(\sum_{t=1}^{N-1} \langle \theta_t, \tilde{x}_t \rangle \tilde{x}_t \right) \\ &= \tilde{V}_{N-1}^{-1} \tilde{u}_{N-1}\end{aligned}$$

We can thus express the ellipsoid as:

$$\hat{C}_{N-1} = \left\{ \theta \in R^d : (\theta - \hat{\theta}_N)^T \tilde{V}_{N-1} (\theta - \hat{\theta}_N) + \|\hat{\theta}_N\|_2^2 + \sum_{t=1}^{N-1} (\langle \tilde{x}_t, \theta_t - \hat{\theta}_N \rangle)^2 \leq M_N + K_N \right\}$$

The ellipsoid is contained in a larger ellipsoid

$$\hat{C}_{N-1} \subseteq C_{N-1} = \left\{ \theta \in R^d : \|\theta - \hat{\theta}_N\|_{\tilde{V}_{n-1}}^2 \leq M_N + K_N \right\}$$

Thus, $\theta^\star$ lies in C_{N-1} with high probability. □

With Lemma 3.3 in hand, we can show a general regret bound:

Theorem 3.4. *(Utility guarantee) Recall that M_T is the regret of our online learner (see equation (4)), and K_T is as defined in Lemma 3.3. Under event $\mathcal{E}$, the regret of Algorithm 1 is:*

$$Regret_T \leq \tilde{O}\left(\sqrt{M_T + K_T} \sqrt{2Td \log\left(1 + \frac{T}{d}\right)} \right)$$

As we can see, this regret bound is quite similar to the regret bound of its non-private counterpart assuming M_T and K_T can be controlled. Now we show that at worst, this bound is $\tilde{O}\left(\sqrt{d}T^{3/4}/\epsilon\right)$ which is $O\left(d^{1/4}\right)$ improvement over the current best known bound for LDP Stochastic Linear Bandit whenever $\epsilon \geq \frac{1}{\sqrt{d}}$. Then, we show that in certain settings, this bound becomes $\tilde{O}(\sqrt{dT})$, which to the best of our knowledge, is the new state-of-the-art bounds for any stronger privacy model than the central model.

3.1 $\tilde{O}(\sqrt{dT})$ regret bound

From Theorem 3.4, it is clear that if we want to have $\tilde{O}(\sqrt{T})$ regret, we need our online learner to have logarithmic regret i.e $M_T = O(\log T)$. However, for this to be true, one might think that we need our loss to be either strongly convex or exp-concave with a sufficiently large strong-convexity/exp-concavity constant. Unfortunately, the loss $l_t(\theta) = (\langle \tilde{x}_t, \theta_t \rangle - \tilde{y}_t)^2 - \|\theta\|_\Sigma^2$ does not fall into either of these family of functions. Surprisingly, it may still be possible to guarantee low regret with high probability using $l_t(\theta_t)$. Denote $L(\theta_t) = \mathbb{E}[l_t(\theta_t)]$. Then:

$$\nabla L(\theta_t) = \mathbb{E}\left[2(\bar{x}_t + \eta_{x,t})(\langle \bar{x}_t, \theta_t \rangle + \langle \eta_{x,t}, \theta_t \rangle - y_t - \eta_{y,t}) - 2\Sigma\theta_t\right]$$

Since $\eta_{x,t}, \eta_{y,t}$ are zero-mean and $\bar{x}_t, \eta_{x,t}, \eta_{y,t}$ are independent:

$$\begin{aligned}\nabla L(\theta_t) &= \mathbb{E}[2\bar{x}_t(\langle \bar{x}_t, \theta_t \rangle - y_t) + 2\eta_{x,t}\langle \eta_{x,t}, \theta_t \rangle - 2\Sigma\theta_t] \\ \Rightarrow \nabla^2 L(\theta_t) &= \mathbb{E}[2\bar{x}_t \bar{x}_t^T] = \mathbb{E}[2x_t x_t^T + 2\zeta_t \zeta_t^T]\end{aligned}$$

Since $x_t x_t^T$ is a positive semi-definite matrix for all $t \in [T]$, we have $\mathbb{E}[x_t x_t^T] \succeq \lambda_{min} I_d$ for some $\lambda_{min} \geq 0$. Thus, $l_t(\theta_t)$ is $\mu-$strongly convex in expectation where $\mu = 2\left(\lambda_{min} + \Delta^2\right)$. This is great news since even though $l_t(\theta_t)$ is not strongly convex, we can still show that the online learner Maler guarantees logarithmic regret with high probability using the following lemma:

Lemma 3.5. *(Strongly convex regret) Assuming $L(\theta) = E[l_t(\theta_t)]$ is $\mu-$strongly convex and $\max_{t,t'} \|\theta_t - \theta'_t\| \leq 2D$. Then w.p at least $1-\beta$, Maler (Algorithm 3) under the event $\mathcal{E}$ guarantees:*

$$\sum_{t=1}^{T} l_t(\theta_t) - l_t(\theta^\star) \leq O\left(\left(\frac{G^2}{\mu} + GD\right)\log T + \frac{G^2}{\mu}\log(1/\beta)\right)$$

Furthermore, we have:

$$\sum_{t=1}^{T} \|\theta_t - \theta^\star\|^2 \leq O\left(\left(\frac{G^2}{\mu^2} + \frac{GD}{\mu}\right)\log T + \frac{G^2}{\mu^2}\log(1/\beta)\right)$$

The proof for Lemma 3.5 is provided in Section B in the Appendix. Notice that from this lemma, we immediately have a high probability bound for the confidence ellipsoid in Lemma 3.3. Specifically, with $\beta \geq \frac{1}{T}$, we have:

$$M_T \leq O\left(\left(\frac{G^2}{\lambda_{min} + \Delta^2} + GD\right)\log T\right)$$

And,

$$\begin{aligned} K_T = O&\left(D\Delta^2 \log(T/\beta)\sqrt{T\log T\left(\frac{G^2}{(\lambda_{min}+\Delta^2)^2} + \frac{GD}{\lambda_{min}+\Delta^2}\right)}\right) \\ &+ O\left(\left(R + \frac{D\sqrt{\log(1/\delta)}}{\epsilon} + D\Delta\right)^2 \log(T/\beta)\right) \\ &+ O\left(\log^2 T\left(\frac{\log(1/\delta)}{\epsilon^2} + \Delta^2\right)\left(\frac{G^2}{(\lambda_{min}+\Delta^2)^2} + \frac{GD}{\lambda_{min}+\Delta^2}\right)\right) \end{aligned}$$

Let $H = \frac{G^2}{(\lambda_{min}+\Delta^2)^2} + \frac{GD}{\lambda_{min}+\Delta^2}$. Then:

$$K_T \leq O\left(D\Delta^2 \log(T/\beta)\sqrt{TH\log T} + \left(R + \frac{D\sqrt{\log(1/\delta)}}{\epsilon}\right)^2 \log(T/\beta) + \frac{H\log(1/\delta)}{\epsilon^2}\log^2 T\right)$$

Now plugging K_T and M_T into Theorem 3.4 we get the following corollary:

Corollary 3.6. *Assuming $\|\theta_t\| \leq D$. Under event $\mathcal{E}$, for any $\lambda_{min} \geq 0$ such that $\mathbb{E}[x_t x_t^T] \succeq \lambda_{min} I_{d\times d}$ and $\beta \geq 1/T$, the regret of Algorithm 1 with Maler as the online learner and with threshold $\bar{\lambda} = \frac{1}{T^{1/4}}$ is upper bounded by*

$$\begin{aligned} &O\left(\left(\left(R + \frac{D\sqrt{\log(1/\delta)}}{\epsilon}\right)\sqrt{\log(T/\beta)} + \sqrt{D\Delta^2\log(T/\beta)\sqrt{TH\log T}} + \frac{\log T\sqrt{H\log(1/\delta)}}{\epsilon}\right)\right. \\ &\qquad \left.\times\sqrt{dT\log(T/d)}\right) \\ &\leq \tilde{O}\left(\min\left\{\frac{\sqrt{d}T^{3/4}}{\epsilon}, \frac{\sqrt{dT}}{\epsilon\lambda_{min}}\right\}\right) \end{aligned}$$

where $H = \frac{G^2}{(\lambda_{min}+\Delta^2)^2} + \frac{GD}{\lambda_{min}+\Delta^2}$.

Let us discuss the implication of the regret bound in Corollary 3.6. Consider the best-case scenario, which is when $\lambda_{min} = O(1)$ (e.g. x_t follows some random distribution with constant variance). Then, $\Delta^2 = 0$ and Algorithm 1 is $(\epsilon, \delta)-$LDP and guarantees $\tilde{O}(\sqrt{dT}/\epsilon)$ regret with high probability. Thus, Algorithm 1 exactly recovers the asymptotic rate of non-private LinUCB. Further, by running our online learner on a bounded

domain, our online regret depends on the radius D of the domain (which is set by the user) rather than the dimension d of the feature vector x_t. As a result, we are able to improve the dimension dependence from $O(d^{3/4})$ in previous works to $O(\sqrt{d})$. Our regret bound would get worse as λ_{min} decreases. However, when $\lambda_{min} \leq \frac{1}{T^{1/4}}$, notice that now $\Delta^2 = \frac{1}{T^{1/4}}$ and Algorithm 1 guarantees $\tilde{O}(\sqrt{d}T^{3/4}/\epsilon)$ regret, which is the same asymptotic rate as the current best regret for LDP (contextual) linear bandit (Shariff & Sheffet, 2018) but with improved dimension dependence. Overall, the regret of Algorithm 1 is always between $\tilde{O}(\sqrt{dT}/\epsilon)$ and $\tilde{O}(\sqrt{d}T^{3/4}/\epsilon)$.

Remark 3.7. Since the regret in (Chowdhury & Zhou, 2022) is $\tilde{O}(dT^{3/5})$, Algorithm 1 has a better regret as long as $\lambda_{min} \geq \Omega\left(\frac{1}{T^{1/10}}\right)$ while also providing stronger privacy guarantee and having no restriction on ϵ.

3.2 Comparisons with previous results

In the worst-case scenario ($\lambda_{min} \leq \frac{1}{T^{1/4}}$), Algorithm 1 provides a regret bound of $\tilde{O}(\sqrt{d}T^{3/4}/\epsilon + \sqrt{dT})$ with high probability. This surpasses the SOTA result for LDP stochastic linear bandit, which is $\tilde{O}((dT)^{3/4}/\sqrt{\epsilon} + d\sqrt{T})$ whenever $\epsilon \geq \frac{1}{\sqrt{d}}$ (which covers many practical scenarios where ϵ is typically larger than 1). Although both exhibit the same asymptotic rate of $\tilde{O}(T^{3/4})$, Algorithm 1 demonstrates superior dimension dependence in both private and non-private terms. The key to this improvement lies in the unique approach of Algorithm 1 concerning noise injection and the employment of an online learner with a dimension-free regret. In (Zheng et al., 2020), the privacy guarantee is achieved by adding a Gaussian matrix H_t to V_t and a Gaussian vector h_t to u_t. From the concentration inequality of the Gaussian matrices, $\|H_t\|$ is bounded by $\tilde{O}(\sqrt{dT}/\epsilon)$ with high probability. Thus, plugging this back in Eq. 1 and Eq. 2 yields the $\tilde{O}((dT)^{3/4}/\sqrt{\epsilon} + d\sqrt{T})$ regret. On the other hand, Algorithm 1 injects noise directly to x_t and y_t, instead of V_t and u_t. Thus, we are not restricted by the $\sqrt{d}$ factor that comes from the concentration bound of the Gaussian matrix. The regret now hinges on the performance of the online learner Maler, which is $O(D\sqrt{T})$ where D is the user-set bound for θ_t. As a result, we are able to improve the dimension dependence to $O(\sqrt{d})$.

In the favorable settings (λ_{min} is $O(1)$), the regret of Algorithm 1 is improved to $O(\sqrt{dT})$. Comparatively, this shows a notable advancement over other private algorithms. Specifically, Algorithm 1 outperforms the shuffle model (Chowdhury & Zhou, 2022), which has a regret of $\tilde{O}(dT^{3/5})$, and matches the rate of the central model. However, our algorithm demonstrates a more favorable dimension dependence, attributable to the same reasons discussed in relation to (Zheng et al., 2020). Interestingly, Algorithm 1 also exhibits better dimension dependence than even non-private algorithms (Abbasi-Yadkori et al., 2012; Abbasi-yadkori et al., 2011) in this specific setting. This is because the non-private models consider worst-case bounds universally, suggesting that these bounds might be further optimized under certain favorable conditions.

4 Online model selection for LDP LinUCB

From the previous section, we show that Algorithm 1 achieves $\tilde{O}(\sqrt{dT}/\epsilon)$ regret when $\lambda_{min} \approx O(1)$ and degrades to $\tilde{O}(\sqrt{d}T^{3/4}/\epsilon)$ when $\lambda_{min} \leq \frac{1}{T^{1/4}}$. However, to run Algorithm 1, we need to know the minimum eigenvalue λ_{min} of $\mathbb{E}[x_t x_t^T]$ (the online learner does not need to know λ_{min} but we still need λ_{min} to set the confidence width). One might think that we can run a grid search through a range of values for λ_{min} to find the one that works the best. However, running the algorithm multiple times using the same dataset can degrade our privacy guarantee (though in practice people still tune their algorithms). Moreover, in a truly online setting it may be that the minimum eigenvalue observed during this "training" period does not capture the later values. In this section, we present a new algorithm that bypasses this problem.

Algorithm 2 is built around a meta-learner that has access to M distinct base learners (which are $M-1$ copies of Algorithm 1 and 1 copy of Algorithm 7 initialized with different guesses of $\lambda_{min} = \lambda_i$ for $i \in [M]$). The reason we have to use two different types of base learners is that model selection algorithms usually require anytime regret guarantees, which Algorithm 1 satisfies only under the condition where perturbation of x_t is unnecessary (i.e., $\Delta^2 = 0$). Thus, in scenarios where the actual λ_{min} is small, necessitating the perturbation of x_t, we instead utilize the anytime variant of Algorithm 1, which is obtained by applying the classic "doubling trick" to Algorithm 1, as described in Algorithm 7. This variant serves as the first base learner of Algorithm 2. Then, by setting $\lambda_i = \frac{2^{i-1}}{T^{1/8}}$ for $i \in [M]$, we can guarantee that at least one of our

Algorithm 2 Private Bandits Combiner

Input: Receive base learner 1 (Algorithm 7) and base learners i for $i \in (1, M]$(Algorithm 1). Constants $C_1, \ldots, C_M$, $\alpha_1, \ldots, \alpha_M$, $R_1, \ldots, R_M$, T users, failure probability β, universal constant p, privacy noise variance σ^2, privacy parameters ϵ, δ, covariance matrix Σ, domain diamater D, universal constant C, power constant k.
Initialize base learner 1 with ϵ, δ, β, Σ, D, C, $\lambda_{min} = \frac{1}{T^{1/8}}$ and $k = 1/8$.
Initialize each base learner i with ϵ, δ, β, Σ, D, C, $\lambda_{min} = \lambda_i = \frac{2^{i-1}}{T^{1/8}}$ for $i \in (1, M]$, and $\bar{\lambda} = 1/T^{1/8}$.
Set $T(i, 0) = 0$ and $\hat{\mu}_0^i = 0$ for all i, and set $I_1 = \{1, \ldots, M\}$.
Initialize $\theta_1^i = 0$, $\tilde{V}_0^i = I_{d\times d}$, $\tilde{u}_0^i = 0$ for all $i \in [M]$.

for $t = 1 \ldots T$ **do**
For the server:
Set $U(i, t-1) = \hat{\mu}^i_{T(i,t-1)} + \min\left(1, \frac{C_i T(i,t-1)^{\alpha_i} + p\sqrt{T(i,t)(1+\sigma^2)\log\left(\frac{T^3 M \log T(i,t)(1+\sigma^2)}{\beta}\right)}}{T(i,t-1)}\right) - \frac{R_i}{T}$ for all i.
Set $i_t = \arg\max_{i \in I_t} U(i, t-1)$.

For the local user t:
Receive the base learner index i_t and $\theta_t^{i_t}$, $\tilde{V}_{t-1}^{i_t}$, $\tilde{u}_{t-1}^{i_t}$ from the server.
User t follows the policy of the base learner i_t and play $x_t^{i_t}$, receive reward $y_t^{i_t}$, and θ_{t+1}.
Send the noisy feature vector $\tilde{x}_t^{i_t}$, noisy reward $\tilde{y}_t^{i_t}$, and parameter θ_{t+1} to the server.

For the server:
Update $T(i_t, t) = T(i_t, t-1) + 1$ and $T(j, t) = T(j, t-1)$ for $j \neq i_t$.
Update $\theta_{t+1}^{i_t} = \theta_{t+1}$, $\tilde{V}_t^i = \tilde{V}_{t-1}^i + \tilde{x}_t^{i_t}(\tilde{x}_t^{i_t})^T$, $\tilde{u}_t^{i_t} = \tilde{u}_{t-1}^{i_t} + \langle \theta_t^{i_t}, \tilde{x}_t^{i_t} \rangle \tilde{x}_t^{i_t}$.
Update $\theta_{t+1}^i = \theta_t^i$, $\tilde{V}_t^i = \tilde{V}_{t-1}^i$, $\tilde{u}_t^i = \tilde{u}_{t-1}^i$ for all $i \neq i_t$.
Update $\hat{\mu}^{i_t}_{T(i_t,t)} = \frac{1}{T(i_t,t)} \sum_{\tau=1}^{T(i_t,t)} \tilde{y}_t^{i_t}$.
if $\sum_{\tau=1}^{T(i_t,t)} \hat{\mu}_{\tau-1}^{i_t} - \hat{t}_\tau^{i_t} \geq C_{i_t} T(i_t, t)^{\alpha_{i_t}} + p\sqrt{T(i,t)(1+\sigma^2)\log\left(\frac{T^3 M \log T(i,t)(1+\sigma^2)}{\beta}\right)}$ **then**
$I_t = I_{t-1} - \{i_t\}$
else
$I_t = I_{t-1}$
end if
end for

guesses would be at most a constant factor away from the actual λ_{min} if $\lambda_{min} \geq \frac{1}{T^{1/8}}$. In the scenario where $\lambda_{min} < \frac{1}{T^{1/8}}$, the actual value of λ_{min} is not as important. This is due to our setting of the threshold $\bar{\lambda}$ at $\frac{1}{T^{1/8}}$. Under this condition, the base learner with the smallest λ estimate automatically ensures a regret of $\tilde{O}(\sqrt{d}T^{3/4})$. The goal of the meta-learner is to "combine" the outputs of M base learners into one output in such a way that the final regret is not much worse than if we had selected the best base learner in hindsight.

We employ the Bandit Combiner Algorithm in (Cutkosky et al., 2020) as our meta-learner. We note that there are more recent meta-learners with more refined guarantees and techniques (Pacchiano et al., 2023; Cutkosky et al., 2021), as well as techniques that work even in the fully adversarial setting (Agarwal et al., 2017; Pacchiano et al., 2020). However, (Cutkosky et al., 2020) is somewhat easier to use in our setting because it applies out-of-the-box to combine base learners whose individual regret bounds have different asymptotic rates.

The Bandit Combiner Algorithm (Algorithm 2) employs the use of the Upper Confidence Bound (UCB) strategy by treating each base learner as an arm in a multi-armed bandit setup. This approach involves using the average reward received by each base learner and their respective regret to establish the upper-confidence bound. Through this method, the algorithm sequentially identifies the most effective learner. Overall,

Algorithm 2 guarantees $O(R_T^J)$ regret where R_T^J is the regret of the best base learner. For a more detailed discussion of the algorithm, refer to (Cutkosky et al., 2020). We have the following guarantee for Algorithm 2:

Theorem 4.1. *(see Corollary 2 (Cutkosky et al., 2020)) Let* $\eta_1 = \frac{\epsilon T^{1/8}}{\sqrt{dT}(P\log^{3/2}(T)\epsilon T^{1/8}+1)}$ *and* $\eta_i = \frac{\epsilon}{P'\log^{3/2}(T)\sqrt{dT}}$, $C_1 = P\log^{3/2}(T)\sqrt{d}\left(\frac{\epsilon T^{1/8}+1}{\epsilon T^{1/8}}\right)$ *and* $C_i = P'\log^{3/2}(T)\frac{\sqrt{d}}{\epsilon\lambda_i}$ *for positive constants* P *and* P', $\alpha_1 = \frac{3}{4}$ *and* $\alpha_i = \frac{1}{2}$ *for* $i \in [2, M]$, *and set* R_i *via:*

$$R_i = C_i T^{\alpha_i} + \frac{(1-\alpha_i)^{\frac{1-\alpha_i}{\alpha_i}}(1+\alpha_i)^{\frac{1}{\alpha_i}}}{\alpha_i^{\frac{1-\alpha_i}{\alpha_i}}} C_i^{\frac{1}{\alpha_i}} T\eta_i^{\frac{1-\alpha_i}{\alpha_i}} + 288\log(T^3N/\beta)T\eta_i + \sum_{k\neq i}\frac{1}{\eta_k}$$

for all i. *Let* j *be the index of the base learner with the smallest regret. If* $\lambda_{min} \geq \frac{1}{T^{1/8}}$, *then w.p at least* $1-3\beta$, *the regret of Algorithm 2 under event* $\mathcal{E}$ *satisfies:*

$$Regret_T \leq \tilde{O}\left(\frac{\sqrt{dT}}{\epsilon\lambda_{min}^2}\right)$$

If $0 \leq \lambda_{min} < \frac{1}{T^{1/8}}$, *then w.p at least* $1-3\beta$, *the regret of Algorithm 2 under event* $\mathcal{E}$ *satisfies:*

$$Regret_T \leq \tilde{O}\left(\sqrt{d}T^{5/6} + \frac{\sqrt{d}T^{17/24}}{\epsilon}\right)$$

Overall, Algorithm 2 guarantees $\tilde{O}(\sqrt{dT}/\epsilon)$ regret when λ_{min} is $O(1)$ without requiring the knowledge of λ_{min}. The guarantee then degrades with a rate of $O(1/\lambda_{min}^2)$, yet it remains below $\tilde{O}\left(\sqrt{d}T^{5/6} + \frac{\sqrt{d}T^{17/24}}{\epsilon}\right)$. This result reveals the trade-offs involved when combining base learners that have different asymptotic regret guarantees. Since the general regret guarantee of Algorithm 2 is $\tilde{O}\left(C_j^{\frac{1}{\alpha_j}} T\eta_j^{\frac{1-\alpha_j}{\alpha_j}} + \sum_{k\neq i}\frac{1}{\eta_k}\right)$, different asymptotic rate with different a_j requires different settings of η_j for the final rate to be optimal. Thus, to adapt to the $\tilde{O}(\sqrt{dT})$ rate, we suffer the worst case rate of $\tilde{O}(\sqrt{d}T^{5/6})$ instead of $\tilde{O}(\sqrt{d}T^{3/4})$ as in Algorithm 1. However, if we have a reason to believe that we are not in the favorable setting or if we want to preserve the worst case rate of Algorithm 1, we can instead run Algorithm 1 with $\lambda_{min} = 0$ and $\bar{\lambda} = \frac{1}{T^{1/4}}$ to always guarantee $\tilde{O}(\sqrt{d}T^{3/4}/\epsilon)$ regret.

5 Experiments

In this section, we will compare the empirical performance of Algorithm 1 to that of previous works. Specifically, we compare our algorithm to Shuffle Private LinUCB (SDP) Chowdhury & Zhou (2022), Joint Differentially Private LinUCB (JDP) Shariff & Sheffet (2018), Locally Private LinUCB (LDP) Zheng et al. (2020), and the Non-private LinUCB (LinUCB) Abbasi-yadkori et al. (2011). For our algorithm (OnlineUCB), we implement Algorithm 1 with Maler as the online learner. In our experiment, we consider 100 arms with dimension $d = 5$. We run our algorithm over $T = 20000$ rounds and average the regrets over 50 trials. We generate the optimal parameter $\theta^\star$ and the feature vector x_t by sampling a $(d-1)-$dimensional vector of norm $1/\sqrt{2}$ uniformly at random and append it with a $1/\sqrt{2}$ entry. We also use Bernoulli rewards to ensure boundedness. This is the same exact setting as in Chowdhury & Zhou (2022). As we can see from Figure 1, in the low-privacy domain ($\epsilon = 1$ and $\epsilon = 10$), our OnlineUCB algorithm significantly outperforms not only the previous best-known LDP algorithm but also SDP and JDP (which use the weaker shuffle and central models of differential privacy respectively) while having stronger privacy guarantee. This result is consistent with the theory since when there exists a $\lambda_{min} = O(1)$ such that $\mathbb{E}_t\left[x_t x_t^T\right] \succeq \lambda_{min} I$ (which is our experiment settings), OnlineUCB has a better regret guarantee than all of the previous private stochastic linear bandits algorithms. In the high privacy domain ($\epsilon = 0.2$), OnlineUCB does not do as well but still outperforms LDP LinUCB and has comparable performance to Shuffle LinUCB. We also note that even though we use Maler to be consistent with the theory, one could also use Online Gradient Descent (which runs a lot faster than Maler) as the online learner to get the same empirical performance.

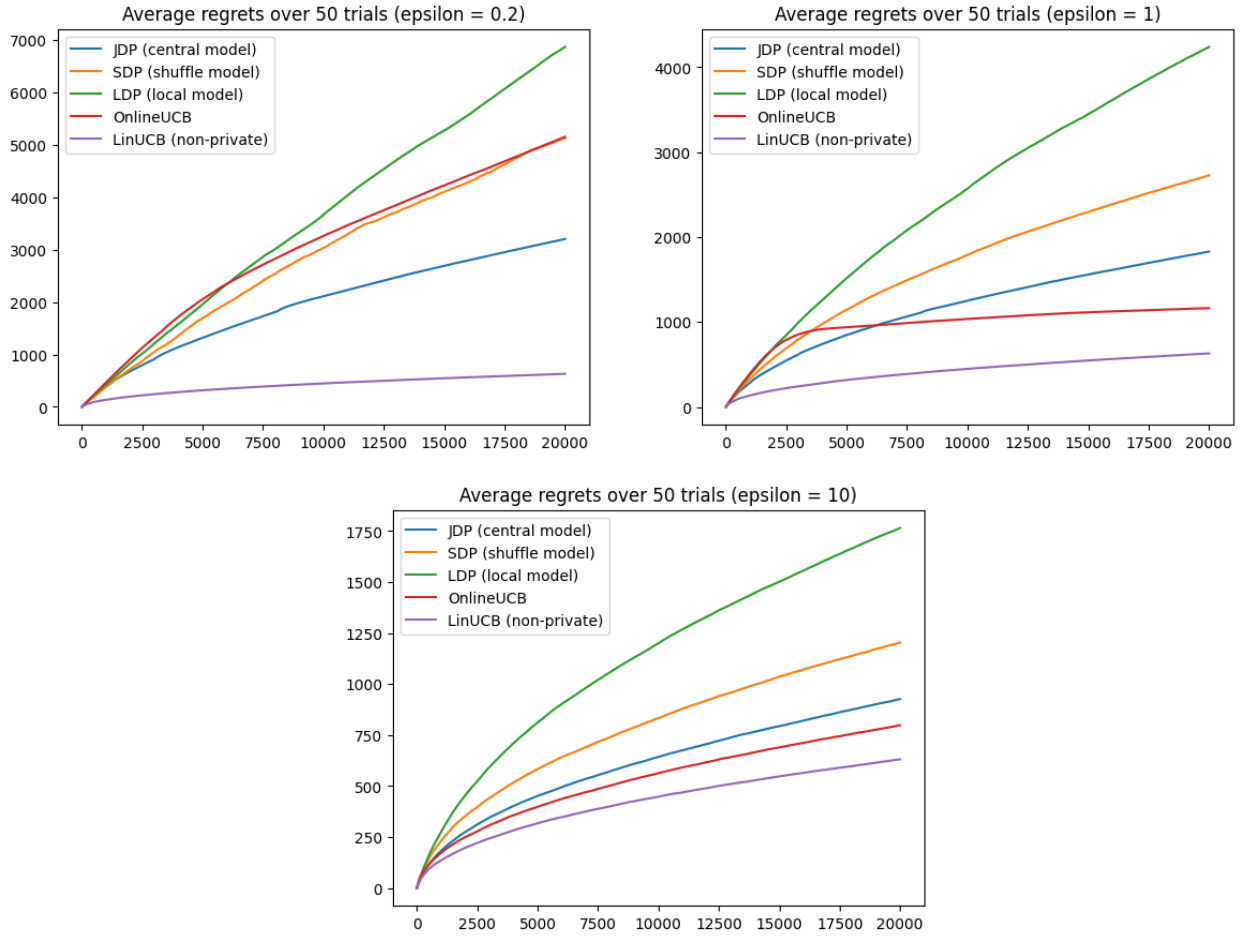

Figure 1: (a) $\epsilon = 0.2$ (b) $\epsilon = 1$ (c) $\epsilon = 10$

6 Conclusions

In this paper, we present a new algorithm for differentially private linear (contextual) stochastic bandit in the local settings that uses an online learner to construct the confidence set. By carefully choosing the online learner as well as the loss function sent to the online learner, our algorithm guarantees the regret $\tilde{O}\left(\min\left\{\frac{\sqrt{d}T^{3/4}}{\epsilon}, \frac{\sqrt{dT}}{\epsilon\lambda_{min}}\right\}\right)$ with high probability where λ_{min} is the lower bound on $\mathbb{E}_t[x_t x_t^T]$. Thus, when $\lambda_{min} \approx O(1)$, our algorithm is the first algorithm that guarantees $\tilde{O}(\sqrt{T})$ for the LDP setting. Further, by running the online learner on a bounded domain, we are able to improve the regret dependence on the dimension d of the feature vector x_t from $O(d^{3/4})$ to $O(\sqrt{d})$. There are several limitations that one could further explore to improve the results of this paper. The most natural question is if it is possible to guarantee $\tilde{O}\left(\sqrt{T}\right)$ for the local model without any further assumption. In our current result, we still rely heavily on the fact that $\mathbb{E}_t[x_t x_t^T] \succeq \lambda_{min} I_d$ to get $\tilde{O}\left(\sqrt{T}\right)$ regret. If that is not possible, what are other settings that allow us to achieve $\tilde{O}\left(\sqrt{T}\right)$ regret? One could hope that by further exploiting the properties of specific online learners from the rich literature of online optimization, we can find other favorable domains as well.

References

Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/e1d5be1c7f2f456670de3d53c7b54f4a-Paper.pdf.

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In Neil D. Lawrence and Mark Girolami (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 1–9, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL https://proceedings.mlr.press/v22/abbasi-yadkori12.html.

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pp. 12–38. PMLR, 2017.

T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3), nov 2011. ISSN 1094-9224. doi: 10.1145/2043621.2043626. URL https://doi.org/10.1145/2043621.2043626.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 375–403. Springer, 2019.

Sayak Ray Chowdhury and Xingyu Zhou. Shuffle private linear contextual bandits. *arXiv preprint arXiv:2202.05567*, 2022.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 208–214, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/chu11a.html.

Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pp. 1655–1658, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450347037. doi: 10.1145/3183713.3197390. URL https://doi.org/10.1145/3183713.3197390.

Ashok Cutkosky, Abhimanyu Das, and Manish Purohit. Upper confidence bounds for combining stochastic bandits. *arXiv preprint arXiv:2012.13115*, 2020.

Ashok Cutkosky, Christoph Dann, Abhimanyu Das, Claudio Gentile, Aldo Pacchiano, and Manish Purohit. Dynamic balancing for model selection in bandits and rl. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2276–2285. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/cutkosky21a.html.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pp. 715–724, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300506. doi: 10.1145/1806689.1806787. URL https://doi.org/10.1145/1806689.1806787.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

Evrard Garcelon, Kamalika Chaudhuri, Vianney Perchet, and Matteo Pirotta. Privacy amplification via shuffling for linear contextual bandits. In *International Conference on Algorithmic Learning Theory*, pp. 381–407. PMLR, 2022.

Y. Gordon, Alexander Litvak, Carsten Schuett, and Elisabeth Werner. On the minimum of several random variables. *Proceedings of the American Mathematical Society*, 134:3665–3675, 12 2006. doi: 10.2307/4098204.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. 2021.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Yingkai Li, Yining Wang, and Yuan Zhou. Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*, pp. 2173–2174. PMLR, 2019.

Aldo Pacchiano, My Phan, Yasin Abbasi Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. Model selection in contextual stochastic bandit problems. *Advances in Neural Information Processing Systems*, 33:10328–10337, 2020.

Aldo Pacchiano, Christoph Dann, and Claudio Gentile. Data-driven regret balancing for online model selection in bandits. *arXiv preprint arXiv:2306.02869*, 2023.

Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. *Advances in Neural Information Processing Systems*, 31, 2018.

Roman Vershynin. High-dimensional probability. 2018.

Guanghui Wang, Shiyin Lu, and Lijun Zhang. Adaptivity and optimality: A universal algorithm for online convex optimization. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 659–668. PMLR, 22–25 Jul 2020a. URL `https://proceedings.mlr.press/v115/wang20e.html`.

Guanghui Wang, Shiyin Lu, and Lijun Zhang. Adaptivity and optimality: A universal algorithm for online convex optimization. In *Uncertainty in Artificial Intelligence*, pp. 659–668. PMLR, 2020b.

Jiujia Zhang and Ashok Cutkosky. Parameter-free regret in high probability with heavy tails. *arXiv preprint arXiv:2210.14355*, 2022.

Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. Locally differentially private (contextual) bandits learning. *Advances in Neural Information Processing Systems*, 33:12300–12310, 2020.