Privacy Amplification via Shuffling for Linear Contextual Bandits

Evrard Garcelon EVRARD@FB.COM
Meta AI and CREST, ENSAE

Kamalika Chaudhuri
Meta AI

Vianney Perchet
CREST, ENSAE

Matteo Pirotta
Meta AI

Editors: Sanjoy Dasgupta and Nika Haghtalab

Abstract

Contextual bandit algorithms are widely used in domains where it is desirable to provide a personalized service by leveraging contextual information, that may contain sensitive information that needs to be protected. Inspired by this scenario, we study the contextual linear bandit problem with differential privacy (DP) constraints. While the literature has focused on either centralized (joint DP) or local (local DP) privacy, we consider the shuffle model of privacy and we show that it is possible to achieve a privacy/utility trade-off between JDP and LDP. By leveraging shuffling from privacy and batching from bandits, we present an algorithm with regret bound $\widetilde{\mathcal{O}}(T^{2/3}/\varepsilon^{1/3})$, while guaranteeing both central (joint) and local privacy. Our result shows that it is possible to obtain a trade-off between JDP and LDP by leveraging the shuffle model while preserving local privacy.

Keywords: Differential Privacy, Shuffling, Linear Contextual Bandits, Joint Differential Privacy, Local Differential Privacy

1. Introduction

In a *contextual bandit algorithm*, at each time $t \in [T] := \{1, \dots, T\}$, a learner first observes a set of features $(x_{t,a})_{a\in[K]} \subset \mathbb{R}^d$, selects an action $a_t \in [K]$ out of a set of K actions, and observes a reward $r_t = r(x_{t,a_t}) + \eta_t$ where η_t is a conditionally independent zero-mean noise (r is not known beforehand). Consequently, the learning algorithm has to balance exploration of the environment with exploitation of the current knowledge to maximize the cumulative reward. The performance of the the learner is measured by the cumulative regret, which is the difference between its own cumulative reward, and the cumulative reward it would have received had it always played the best action. Contextual bandit algorithms have achieved great practical success, and have been used for many sensitive applications such as personalization, digital marketing, healthcare and finance (e.g., Mao et al., 2020; Wang and Yu, 2021). With these applications in mind, the literature has started investigated privacy guarantees both in bandits (e.g., Shariff and Sheffet, 2018; Zheng et al., 2020) and in RL (e.g., Vietri et al., 2020; Garcelon et al., 2020). In this paper, we focus on privacy-preserving contextual bandits.

For a contextual bandit problem on sensitive data, we assume that a single user enters the system at time t, and hence the context at time t is their private information. To measure privacy, we use differential privacy (Dwork et al., 2006) – a privacy definition introduced by cryptographers that

has emerged as the gold standard for privacy-preserving data analysis (e.g., Erlingsson et al., 2014; Dwork et al., 2014; Abowd, 2018; Chaudhuri et al., 2011; Abadi et al., 2016; Boursier and Perchet, 2020). The standard differential privacy framework applies to static data in a batch setting, but two extensions have been proposed to address online problems. The first is Joint Differential Privacy (JDP) (e.g., Shariff and Sheffet, 2018), an analogue of central differential privacy, where the users trust the bandit algorithm. JDP ensures that changing a single user's private information in the data does not change the probability of any future outcome (namely, actions taken and rewards received by any other user) by much.

Definition 1 (Joint DP) *For $\varepsilon > 0$ and $\delta_0 > 0$, a randomized bandit agent $\mathfrak{A}$ is* (ε, δ_0)-joint differentially *private if for every $t \in [T]$, two sequences of users, $U = \{u_1, \dots, u_T\}$ and $U' = \{u'_1, \dots, u'_T\}$, that differs only for the t-th user and for all events $E \subset \mathcal{A}^{[T-1]}$ then:*

$$\mathbb{P}(\mathfrak{A}_{-t}(U) \in E) \leq e^{\varepsilon}\mathbb{P}(\mathfrak{A}_{-t}(U') \in E) + \delta_0 \tag{1}$$

where $\mathfrak{A}_{-t}(U)$ denotes all the outputs of algorithm $\mathfrak{A}$, i.e., all actions $(a_i)_{i\neq t}$ excluding the output of time t for the sequence of users U.

A second, stronger concept is Local Differential Privacy (LDP) (e.g., Zheng et al., 2020), where the users do not trust the bandit algorithm, and transmit only sanitized versions (using a private randomizer $\mathcal{M}$) of their contexts and rewards to the algorithm. Here, LDP ensures that user information is sanitized in such a manner that changing a single user's private value does not alter the distribution of the sanitized value by much.

Definition 2 (Local DP) *For any $\varepsilon \geq 0$ and $\delta \geq 0$, a privacy preserving mechanism $\mathcal{M}$ is said to be* (ε, δ)-locally differential *private if and only if for all users $u, u' \in \mathcal{U}$, contexts/rewards $((x_u, r_u), (x_{u'}, r_{u'})) \in (\mathbb{R}^d \times \mathbb{R})^2$ and all $O \subset \{\mathcal{M}(\mathcal{B}(0, L) \times [0, 1]) \mid u \in \mathcal{U}\}$:*

$$\mathbb{P}\left(\mathcal{M}((x_u, r_u)) \in O\right) \leq e^{\varepsilon}\,\mathbb{P}\left(\mathcal{M}((x_{u'}, r_{u'})) \in O\right) + \delta \tag{2}$$

where $\mathcal{B}(0, L) \times [0, 1]$ is the space of context/reward associated to user u.

Just like the standard batch setting, while LDP offers a strong notion of privacy, its utility is often much lower. Specifically, for contextual linear bandit algorithms, while ε-JDP guarantees can be obtained by paying a multiplicative factor in the regret, LDP comes with a much higher impact on the regret. In fact, Zheng et al. (2020) have shown that ε-LDP regret scales with $\widetilde{\mathcal{O}}(T^{3/4}/\sqrt{\varepsilon})$ instead of $\widetilde{\mathcal{O}}(T^{1/2}/\sqrt{\varepsilon})$ for a ε-JDP algorithm (see Tab. 1 for more details.)

Real applications are gradually moving away from the *centralized* model of privacy, favoring the simpler and stronger notion of local privacy. This change is illustrated by the rise of on-device computation for mobile application (e.g., Apple). The natural question we address in this paper is:

Is it possible to design a bandit algorithm with guarantees akin to local privacy but better utility?

To address this question, we consider the shuffle model of privacy (e.g., Cheu et al., 2019; Feldman et al., 2020; Chen et al., 2021; Balle et al., 2019; Erlingsson et al., 2020) that, in supervised learning settings, allow to achieve a trade-off between central and local DP through a shuffler. The shuffler receives users' reports and permutes them before sending them to the server. This setting was first introduced in Bittau et al. (2017), named the *ESA* model (Encode-Shuffle-Analyze) and

motivated by the need for anonymous data collection. Erlingsson et al. (2019) later provided an analysis of the amplification of privacy thanks to the combined use of shuffling and local differential privacy showing that the shuffling model of privacy is able to strike a middle ground between the totally decentralized but somewhat sample inefficient *local* model and the centralized but more sample efficient central model of privacy. It is currently unclear whether it is possible to achieve some form of privacy/utility trade-off between these two models in the contextual bandit setting.

1.1. Our Contributions

In this paper, we investigate the linear contextual bandit problem under the shuffle model of privacy, for the first time considering this privacy model in contextual bandit. Compared to the standard shuffle model (e.g., in supervised learning), there are several challenges introduced by the sequential nature of the problem. First, the shuffler is executed continuously and not only once as normally considered in supervised learning. Second, the number of samples available grows with time and depends on the decisions of the learning agent. This makes the design of the algorithm non-trivial, in particular for efficiently trading-off privacy amplification and regret.

We address these challenges in two ways. First, we carefully design separate asynchronous batch schedules for the shuffler and the bandit algorithm (i.e., LINUCB); here, batching at the shuffler is used to ensure privacy, and not just improved regret. Second, we leverage the martingale structure of the problem to analyze these batching schedules and provide privacy guarantees on the entire sequence of outputs generated by the shuffler and bandit algorithm. We summarize our main contributions as follows (see also Tab. 1):

- If there is no adversary in between the shuffler and the algorithm (i.e., the communication channel is secure), we show that it is possible to achieve a regret bound of $\widetilde{\mathcal{O}}\left(dT^{2/3}/\varepsilon^{1/3}\right)$ with a fixed batch size for the shuffler and dynamic batch for the bandit algorithm.
- In the case of adversary in between the shuffler and the users, our algorithm achieves a regret bound of $\widetilde{\mathcal{O}}\left(T^{3/4}/\sqrt{\varepsilon}\right)$ with a fixed batch size for the shuffler and dynamic batch for the bandit algorithm.

Algorithm	Regret Bound	Privacy Model	
		Joint DP	Local DP
Shariff and Sheffet (2018)	$\widetilde{\mathcal{O}}\left(T^{1/2}/\varepsilon^{1/2}\right)$	(ε, δ)	N/A
Zheng et al. (2020)	$\widetilde{\mathcal{O}}\left(T^{3/4}/\varepsilon^{1/2}\right)$	(ε, δ)	(ε, δ)
Our Cor. 8 (*LDP optimization*)	$\widetilde{\mathcal{O}}\left(T^{3/4}/\varepsilon^{1/2}\right)$	$(\frac{\varepsilon^{3/2}}{T^{1/4}}, \delta)$	$(\varepsilon, 0)$
Our Cor. 9 (*regret optimization*)	$\widetilde{\mathcal{O}}\left(T^{2/3}/\varepsilon^{1/3}\right)$	(ε, δ)	$(\varepsilon^{2/3}T^{1/6}, 0)$

Table 1: Regret and privacy for algorithms in *linear contextual bandits* for $T \geq 1/(27\varepsilon)^4$.

2. Preliminaries

We consider linear contextual bandit problems, where rewards are linearly representable in the features, i.e., for any feature vector $x_{t,a}$, it writes as $r(x_{t,a}) = \langle x_{t,a}, \theta^\star \rangle$, where $\theta^\star \in \mathbb{R}^d$ is unknown. We do not pose any assumption on the context generating process but we rely on the following standard assumptions.

Assumption 3 *There exist $S > 0$ and $L > 0$ such that $\|\theta^\star\|_2 \leq S$ and, for all time $t \in [T]$, arm $a \in [K]$, $\|x_{t,a}\|_2 \leq L$. Furthermore, the noisy reward is $r_t = \langle x_{t,a}, \theta^\star \rangle + \eta_t \in [0,1]$ with η_t being σ-subGaussian for some $\sigma > 0$. These parameters, L, S and σ, are known.*

The performance of the learner $\mathfrak{A}$ over T steps is measured by the regret $R_T = \sum_{t=1}^{T} r(x_{t,a_t^\star}) - r(s_{t,a_t})$, which represents the cumulative difference between playing the optimal action $a_t^\star = \arg\max_{a \in [K]} r(x_{t,a})$ and a_t the action selected by the algorithm.

2.1. Shuffle-model in Contextual Bandits

In this section, we introduce the generic shuffle-model for contextual bandit, inspired by the ESA model. In Sec. 3, we will provide the details for instantiating it in linear contextual bandits. In the standard shuffle model, a shuffler is introduced in between the data and the algorithm. The shuffler enables privacy amplification by permuting information of l users. The larger the batch, the higher the privacy amplification but also the degradation of the utility (see e.g., Cheu et al., 2019), leading to some fundamental trade-off between privacy amplification and utility loss. In online learning, we observe users sequentially and it is natural to assume that, in order to achieve privacy amplification, the shuffler builds a batch of consecutive users before communicating with the bandit algorithm. The bandit algorithm can then behave synchronously or asynchronously w.r.t. the shuffler. In other words, it can update its internal statistics with the same frequency of the shuffler or use an independent batch schedule.

More formally, the shuffle-model for contextual bandit is described by the following interaction protocol (see also Fig. 1). At each time $t \in [T]$,

❶ A new user x_t receives model information from the bandit algorithm (e.g., estimated rewards and confidence intervals) that are used to *locally* compute the action to play. Then, the user plays the prescribed action a_t which generates the associated reward r_t.

❷ The user sends its own privatized version of the data $\mathcal{M}_{\mathrm{LDP}}(x_{t,a_t}, r_t)$ to the shuffler. This new data is added to the shuffler batch $B_{k_t}^S := \bigcup_{i=t_{k_t^S}}^{t} \{\mathcal{M}_{\mathrm{LDP}}(x_{i,a_i}, r_i)\}$, where k_t^S denotes the shuffler batch at time t and t_k is the starting time of batch k.

❸ The bandit algorithm queries statistics from the shuffler. If the shuffler is ready to send data (e.g., enough samples has been collected for privacy amplification), it computes a statistic u on a permutation of the data (i.e., $u(\sigma(B_{k_t}^S))$) and sends it to the bandit algorithm. Otherwise no information is provided. The bandit algorithm adds the new statistic to its batch (i.e., $B_{k_t^A}^A := \bigcup_{i=t_{k_t^A}}^{t} \{u(\sigma(B_{k_i^S}^S))\}$) and may then decide to update the model as soon as data is received (i.e., synchronously) or use an independent batch schedule (i.e., asynchronous).

The objective is to minimize the (pseudo) regret and simultaneously guarantee privacy of the data and of the statistics. To this extent, we assume all users (including the shuffler and the bandit algorithms) behaves in an *honest but curious* manner (Oded, 2009), i.e., the users and the algorithm behaves as prescribed by the protocol. We consider different threat models for privacy, including an adversary in between **a** the user and the shuffler, **b** the shuffler and the bandit algorithm, and **c** the bandit algorithm and the user. We will show that different privacy/regret guarantees can be achieved in the different settings.

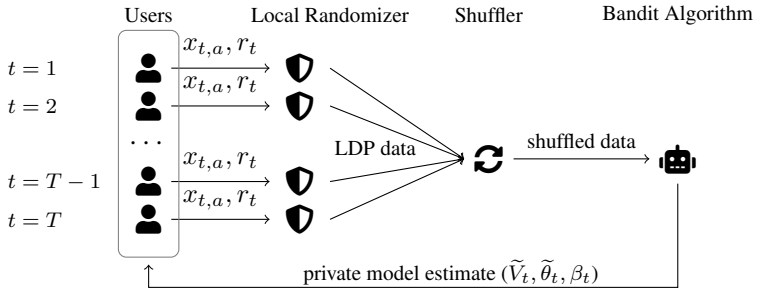

Figure 1: Illustration of the shuffle model for linear contextual bandits.

Algorithm 1: SBLB

Input: LDP parameter: ε_0, privacy parameters: ε, δ_0, regularizer: λ, context bound: L, failure probability: δ, low switching parameter: η, encoding parameter: m, dimension: d, fix batch size: ℓ

Initialize $j^S = j^A = 0$, $\widetilde{\theta}_0 = 0$, $\widetilde{V}_0 = \lambda I_d$ and $p = 2\big(\exp\left(\frac{2\varepsilon_0}{md(d+3)}\right)+1\big)^{-1}$

for $t = 0, 1, \ldots$ **do**

c Communication with the user

User receives $\widetilde{\theta}_{j^A}$, $\widetilde{V}_{j^A}$ and β_{j^A} and selects $a_t \in \operatorname{argmax}_{a\in[K]} \langle x_{t,a}, \widetilde{\theta}_{j^A}\rangle + \beta_{j^A} \| x_{t,a}\|_{\widetilde{V}_{j^A}^{-1}}$

Observe reward r_t and compute private statistics $(\widetilde{b}_t, \widetilde{w}_t) = \mathcal{M}_{\mathrm{LDP}}(x_{t,a_t}, r_t, L, \varepsilon_0, m)$ (Alg. 2)

a Communication with the shuffler

$B^S_{j^S} = B^S_{j^S} \cup (\widetilde{b}_t, \widetilde{w}_t)$

if $|B^S_{j^S}| = l$ **then**

Set $t_{j^S+1} = t$, compute a permutation σ of $[\![t_{j^S}+1, t_{j^S+1}]\!]$ and compute aggregate statistics

$$\forall i \le d, k \le i, \qquad Z_{j^S,i} = \sum_{n=1}^{l}\sum_{q=1}^{m} \tilde{b}_{\sigma(n),i,q} \quad \text{and} \quad U_{j^S,i,k} = \sum_{n=1}^{l}\sum_{q=1}^{m} \widetilde{w}_{\sigma(n),i,k,q}$$

Set $U_{j^S,i,k} = U_{j^S,k,i}$, $B_{j^S+1} = \emptyset$ and $j^S = j^S + 1$

b Communication with the bandit algorithm

Receives (Z_{j^S-1}, U_{j^S-1}) and compute candidate statistics

$$\widetilde{B}_{j^A+1} = \widetilde{B}_{j^A+1} + \frac{Z_{j^S-1}}{m(1-p)} - \frac{l^S}{2(1-p)}$$
$$\widetilde{V}_{j^A+1} = \widetilde{V}_{j^A+1} + \frac{U_{j^S-1}}{m(1-p)} - \frac{l^S}{2(1-p)} + 2(\lambda_{j^A+1} - \lambda_{j^A})I_d$$

if $det(\widetilde{V}_{j^A+1}) \ge (1+\eta) det(\widetilde{V}_{j^A})$ **then**

Compute $\widetilde{\theta}_{j^A+1} = \frac{1}{L}\widetilde{V}^{-1}_{j^A+1}\widetilde{B}_{j^A+1}$

Set $t_{j^A+1} = t$, β_{j^A+1} and λ_{j^A+1} as in Eq. (7) and Eq. (8)

Set $j^A = j^A + 1$, $\widetilde{B}_{j^A+1} = \widetilde{B}_{j^A}$ and $\widetilde{V}_{j^A+1} = \widetilde{V}_{j^A}$

end

end

end

3. Shuffle Model with Fixed-Batch Shuffler

In this section, we provide an instantiation of the shuffle model for linear contextual bandit. We base our algorithm on the non-private low-switching LINUCB (Abbasi-Yadkori et al., 2011), that incrementally builds an estimate $\widehat{\theta}_j$ of the unknown parameter $\theta^\star$. Since the algorithm leverages sum of statistics received from the users, we consider the binary sum mechanism inspired by Cheu et al. (2019) as building block for achieving privacy in the shuffle model. While this scheme allows us to obtain standard LDP guarantees on users information, the shuffler is responsible to provide privacy amplification via batching and shuffling. The main challenge is to combine these elements with the low-switching scheme of LINUCB. As we will explain later, adaptive batching at the level of LINUCB is not for computational efficiency but it is rather fundamental for obtaining a good privacy/regret trade-off.

3.1. Algorithmic Design

In this section, we provide a full description of the Shuffle-Batched Linear Bandit (SBLB) algorithm. Intuitively, the algorithm relies on a shuffler with fixed batch size to achieve privacy amplification from LDP data, and a variation of LINUCB with dynamic batch schedule based on the determinant condition. The pseudo-code is reported in Alg. 1.

❶ Action Selection. At each time t, the user x_t receives, from the bandit algorithm, an estimate of the model composed by a parameter $\widetilde{\theta}_{k_t^A} \in \mathbb{R}^d$, a design matrix $\widetilde{V}_{k_t^A} \in \mathbb{R}^{d\times d}$ and confidence width $\beta_{k_t^A}$. Notice that these are parameters computed at the beginning of the batch k_t^A of the bandit algorithm. Then, the action is selected by maximizing the following standard optimistic problem:

$$a_t \in \operatorname*{argmax}_{a\in[K]} \left\{ \langle x_{t,a}, \widetilde{\theta}_{k_t^A} \rangle + \beta_{k_t^A} \| x_{t,a} \|_{\widetilde{V}^{-1}_{k_t^A}} \right\}$$

where β_t is the size of the confidence ellipsoid, defined in Lem. 10, which roughly scales as $\widetilde{O}\left(t_{k_t^A}^{1/4}\right)$. Note that it is possible to directly access the features $x_{t,a}$ of the user since this computation happens locally. The action is played and a reward r_t is observed.

❷ Local Privacy and Shuffler. Users' information is then protected through a local private mechanism $\mathcal{M}_{\text{LDP}}$. As noticed in (Shariff and Sheffet, 2018), only the information required by the algorithm to compute $\widetilde{\theta}$ through ridge regression and the associated confidence interval must be privatized. We are thus interested in privatizing the quantities $x_{t,a_t} r_t$ and $x_{t,a_t} x_{t,a_t}^\top$. To obtain LDP quantities, we leverage a variation of the private mechanism introduce by Cheu et al. (2019). We independently privatize each component of the vector $x_{t,a_t} r_t$ and of the upper triangular part of the matrix $x_{t,a_t} x_{t,a_t}^\top$, the rest follows from the symmetric structure. Each entry is normalized to $[0, 1]$ and approximated by a truncated 0/1-bit representation, which length is controlled by the parameter $m \in \mathbb{N}^*$. The full procedure is reported in Alg. 2 in the appendix.

The shuffler receives the privatize data $\mathcal{M}_{\text{LDP}}(x_{t,a_t}, r_t)$ and adds it to the current batch. The role of the shuffler is to provide additional privacy by sending data in a random order compared to what it has received. At a high-level this provides an additional privacy guarantee because it breaks the link between a given user and their data. Indeed for an algorithm receiving data from the shuffler, the t-th row of data has little chance to come from user t. If the shuffler has access to a batch of size l, it can provide a privacy amplification of level $l^{-1/2}$ (see e.g., Cheu et al., 2019, Thm.

5.4). Ideally, we would like to shuffle all the data at each time t, achieving a privacy amplification of $t^{-1/2}$. However, this approach would not provide enough privacy due to the fact an adversary would have multiple observations of the same data, thus greatly decreasing the advantage of using the shuffling mechanism. To avoid this issue, we need to force the shuffler to use batches and discard samples after each batch. Let's denote by l^S the fix batch size of the shuffler. At time t, if the batch $B^S_{k_t}$ is of size l^S, the shuffler permutes the data and compute the statistics required by the bandit algorithm. To compute those statistics, the shuffler uses a secure and trusted third-party different that the shuffler. This third-party is assumed to be secure with for example the use of encrypted communication between the shuffler and it, like in (Cheu et al., 2019). When $|B^S_{k_t}| < l^S$, the shuffler do not provide any information to the bandit algorithm. The shuffling setting is not fundamentally different than the LDP one, but it allows to achieve a large gain in privacy in the high data regime from multiple users. Shuffling allows to achieve better privacy guarantees and, overall, it improves the standard LDP protocol with virtually no cost.

❸ **Model Estimation (the bandit algorithm).** As last step, the bandit algorithm queries new data to the shuffler which replies only if the batch is full. If no data is received, the bandit algorithm does nothing. Otherwise, the bandit algorithm receives summary statistics $Z_{k^S_t}$ and $U_{k^S_t}$ corresponding to the sum over the shuffled batch $B^S_{k^S_t}$ of the LDP data associated to xr and $xx^\top$. The algorithm could behave synchronously with the batch schedule of the shuffler and update the model by updating the design matrix $\widetilde{V}_{k^S_t+1}$ and parameter $\widetilde{\theta}_{k^S_t+1}$. However, this behavior would lead to a worse privacy/regret trade-off than an asynchronous data-adaptive schedule. Although it is possible to achieve the same regret bound in non-private settings with static and dynamic batch schedules, in the private case it is no more the case because of required inflation of the confidence intervals by a factor $t^{1/4}$ to deal with concentrations of private statistics. In App. C, we provide a more formal support to this claim.

As a consequence, we shall leverage the determinant-based condition introduced by Abbasi-Yadkori et al. (2011). Upon receiving the data at time t, the bandit algorithm has access to the following set of private statistics $\left\{(Z_i, U_i), i \in [k^S_t]\right\}$, which is further divided into batches of various lengths. Denote by $j = k^A_t$ the bandit batch at time t with associated parameters $\widetilde{V}_j$, $\widetilde{B}_j$ and $\widetilde{\theta}_j$ computed at the beginning of the batch. Then, we denote by $\widetilde{V}_t$ the new design matrix obtained by updating the matrix $\widetilde{V}_j$ with all the statistics received from the shuffler after t_j. If $\det(\widetilde{V}_t) \geq (1+\eta)\det(\widetilde{V}_j)$, then a new batch is started and the model is updated, i.e., $\widetilde{\theta}_{j+1} = \frac{1}{L}\widetilde{V}^{-1}_{j+1}\widetilde{B}_{j+1}$ is computed through ridge-regression. In a LinUCB fashion, the last step for the algorithm is to compute the size of a confidence intervals around $\widetilde{\theta}_{j+1}$ containing the true parameter $\theta^\star$. Contrary to the non-private setting (Abbasi-Yadkori et al., 2011), the algorithm uses wider confidence intervals to account for the noise added to ensure privacy. This increase is quite significant as the confidence intervals grow at a $t^{1/4}$ rate compared to $\log(t)$ in the non private setting. Refer to Lem. 10 for the explicit definition.

4. Analysis of The Shuffle Model with Fixed-Batch Shuffler

In this section, we provide the privacy and regret guarantees of SBLB. We first begin to describe which privacy guarantees are attainable in the different attack scenarios outlined in the introduction. Then we show how the regret of SBLB is impacted by the these attack models.

For sake of clarity, we recall the parameters that regulates the privacy/regret analysis of our algorithm. The first parameter ε_0 regulates the level of local differential privacy introduced by the local randomizer $\mathcal{M}_{\text{LDP}}$. However, to simplify the analysis, we often use the alternative parameter $p := 2\big(\exp\left(\frac{2\varepsilon_0}{md(d+3)}\right)+1\big)^{-1}$ derived from ϵ_0 (see Alg. 2). The other two parameters (ε, δ_0) controls the level of joint differential privacy that SBLB should attain.

4.1. Privacy Analysis of SBLB

As discussed in Sec. 2, the shuffling model encompasses all the multiple scenarios in which the privacy of users can be threatened.

a Compromised communication between the user and the shuffler. In the first and most harmful scenario, the communication between the users and the shuffler is not secured and the data can be observed by an adversary. This is the standard LDP setting in linear contextual bandit. In this case, the use of the local randomizer $\mathcal{M}_{\text{LDP}}$ guarantees that the data sent by the user to the shuffler are ε_0-LDP. That is to say the most stringent privacy guarantees in the differential privacy model.

Proposition 4 (LDP guarantee) *For any $\varepsilon_0 > 0$ and $m \in \mathbb{N}^\star$, $\mathcal{M}_{LDP}(.,.,\varepsilon_0, L, m)$ is ε_0-LDP.*

This particular scenario corresponds to a decentralized setting where the users do not trust the algorithm or the communication channel between them to be secure and they have to protect the privacy of their data at a individual level, that is to say to guarantee that the data sent could have been sent by anyone else. This setting (i.e., the "pure" LDP scenario) is also the one studied in (Zheng et al., 2020). We will show that we can recover their result when we want to guarantee the highest level of LDP privacy. However, at the cost of sacrificing a portion of LDP level, we can obtain a better regret bound, closing the gap with the less stringent JDP setting.

b Compromised communication between the shuffler and the bandit algorithm. In another privacy loss scenario, an adversary can observe the same data as the bandit algorithm. Stated otherwise, the adversary has access to the output of the shuffler. In that case, SBLB is still ε_0-LDP but stronger differential privacy guarantees can be achieved thanks to privacy amplification. In this scenario, the adversary observes the different outputs of the shuffler, that are statistics computed on a number of different users. The question, in the differential privacy setting, is whether it is possible to know that one particular user (i.e., user's data) was involved in the computation of those statistics.

Tenenbaum et al. (2021) studies a weaker version of this question in the multi-armed bandit setting where an adversary *only observes the output of the shuffler for one time step*, while we focus on the more challenging case where the adversary observes all the history. Technically, this is the same difference as ensuring event-level privacy in the continual observation model compared to a differential privacy on a single query. Note that it would be possible to obtain a better regret bound if we consider the adversary model in (Tenenbaum et al., 2021) since a smaller level of privacy is required (see Remark 12).

The complicated aspect is to guarantee that the whole sequence of M_S vectors and matrices $(Z_{j^S}, U_{j^S})_{j^S=1}^{M_S}$ is private, and not a single output at a given time. This issue is solved by leveraging batching. Formally, we can show in this scenario that the sequence $(Z_{j^S}, U_{j^S})_{j^S}$ is $(\varepsilon, \delta_0 + \delta)$-DP for any $\delta_0, \delta \in (0,1)$ and $\varepsilon \in (0,1)$.

Theorem 5 *For any $\varepsilon \in (0,1)$, $\delta_0, \delta \in (0,1)$, encoding parameter m and LDP parameter $\varepsilon_0 > 0$, let $p = 2(e^{2\varepsilon_0/md(d+3)} + 1)^{-1}$. Then if $l^\star$, the length of a shuffler batch, satisfies $l^\star p \geq 14 \log(8mT/\delta_0)$ and:*

$$\sqrt{\left(2 + \left(\frac{\varepsilon l^\star}{32d(d+3)\log(8mT/\delta_0)\sqrt{2T\ln(2T/\delta_0)}}\right)^2\right)^2 - 4} \geq 1 - 2p + 2\sqrt{\frac{2\log(2mT/\delta_0)}{l}} + \left(\frac{\varepsilon l^\star}{32d(d+3)\log(8mT/\delta_0)\sqrt{2T\ln(2T/\delta_0)}}\right)^2, \tag{3}$$

the sequence $(Z_{j^S}, U_{j^S})_{j^S}$ is central $(\varepsilon, \delta_0 + \delta)$-DP.[1]

The result of Thm. 5 is a consequence of the advanced composition theorem (Dwork et al., 2010). Indeed, thanks to shuffling, for any batch j^S, the statistics (Z_{j^S}, U_{j^S}) are $\left(\frac{\sqrt{\varepsilon(1-p)}}{T^{1/4}}, \frac{\delta_0 \varepsilon}{\sqrt{T}(1-p)}\right)$-DP, since the batch length l is approximately $\frac{\sqrt{T}(1-p)}{\varepsilon}$. As a consequence, when composing them together we get that the central DP level of each batch is $\widetilde{\mathcal{O}}\left(\varepsilon\sqrt{\frac{l^\star}{T}}\right)$. Therefore by advanced composition, since we have a total number of batches $M_S \approx \sqrt{T}$, the total privacy over the sequence of $(Z_{j^S}, U_{j^S})_{j^S}$ is of order $\widetilde{\mathcal{O}}\left(\varepsilon\sqrt{\frac{l^\star}{T}} \times \sqrt{\frac{T}{l^\star}}\right)$ that is to say of order $\widetilde{\mathcal{O}}(\varepsilon)$.

c Compromised Communication between the bandit algorithm and the users. Similarly to Shariff and Sheffet (2018), in the final scenario we consider, an adversary can observe the same data coming from SBLB as the users, i.e., the stream of estimates $(\widetilde{\theta}_{k_t^A}, \widetilde{V}_{k_t^A}, \beta_{k_t^A})_{t\in[T]}$. Recall that the bandit algorithm uses a dynamic batch schedule based on the determinant technique and it is asynchronous w.r.t. the shuffler. This leads to a number of bandit batches roughly of order $\log(T)$. While we have to guarantee privacy on a smaller number of element ($\log(T)$ compared to $\sqrt{T}$ in the shuffler), we are technically limited by the former scenario **b**. As shown in Prop. 6, SBLB is $(\varepsilon, \delta_0 + \delta)$-JDP w.r.t. the sequence $(\widetilde{\theta}_{j^A}, \widetilde{V}_{j^A}, \beta_{j^A})_{j^A}$ since $(Z_{j^S}, U_{j^S})_{j^S}$ is $(\varepsilon, \delta_0 + \delta)$-DP.

Proposition 6 (JDP guarantee) *For any $\varepsilon \in (0,1)$, $\varepsilon_0 > 0$, $\delta, \delta_0 \in (0,1)$, $m \in \mathbb{N}^\star$, selecting the length of a shuffler like in Thm. 5 ensures that the sequence of $(\widetilde{\theta}_{j^A}, \widetilde{V}_{j^A}, \beta_{j^A})_{j^A}$ is $(\varepsilon, \delta + \delta_0)$-DP. In other words* SBLB *is $(\varepsilon, \delta + \delta_0)$-JDP.*

Since we are directly leveraging advance composition, we cannot get any privacy amplification when we consider **b** and **c** together. Scenario **c** is indeed the most stringent adversary model in the shuffle-model, limiting the gain in the privacy/regret we can obtain compared to the pure LDP setting. It is however possible to achieve a better privacy/utility trade-off when considering only scenario **c** (and not **b**), but we believe it is a much weaker attack scenario. Although both scenario **b** and **c** aim to ensure JDP, model **c** deals with the issue when attackers can submit potentially false contexts to the bandit algorithm and observes the action recommended with the objective to learn the context/reward of a target user. Guaranteeing that this task is difficult is the objective of

1. We provide the definition of central DP in Def. 13 in App. B. Note that the concept of central DP is at the core for proving JDP results, in fact thanks to Claim 7 in (Shariff and Sheffet, 2018) having a sequence $(\widetilde{V}_t, B_t)_t$ is (ε, δ)-DP implies that a bandit algorithm based on this sequence is (ε, δ)-DP.

JDP. In this paper, we use a deterministic bandit algorithm therefore in terms of privacy scenarios b and c are the same (thanks to the post-processing lemma). However, one could think of using a randomized algorithm and therefore improve the privacy of the whole scheme.

Remark 7 *In online learning, JDP and central-DP are not equivalent definitions. A DP constraint on the actions selected implies that the probability of selecting any action is strictly positive thus hindering the algorithm to select the optimal action. Indeed, as noted in (Shariff and Sheffet, 2018) (see Claim 13) any central-DP linear contextual bandit algorithm must incur linear regret, whereas in the weaker definition of JDP it is possible to attain a sublinear regret. The fact that the computation of the action is local is key to achieve a sublinear regret.*

4.2. Regret Analysis of SBLB

In the previous section, we stated several privacy guarantees of SBLB with different attack models. We shall now show the impact of those privacy guarantees on the regret. As mentioned, shuffling allows to regulate the level and type of privacy by trading-off the regret guarantee. In SBLB, this trade-off is regulated by the parameter ε_0 which has impact on all the main elements in the privacy and regret analysis (e.g., batch size, privacy p, etc.).

The first result we provide is a validation of our algorithm. The following proposition shows that SBLB recovers the results in (Zheng et al., 2020), providing the highest possible *local* DP level at the expense of the regret bound.

Corollary 8 *For any $\varepsilon_0 > 0$ and $\delta \in (0,1)$ then choosing $\varepsilon = \sqrt{\exp(\varepsilon_0) - 1}$ and $\delta_0 = \delta$ we have that* SBLB *is ε_0-LDP and with probability at least $1-\delta$ is bounded by:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{T^{3/4}\sqrt{e^{\varepsilon_0}+1}}{\sqrt{e^{\varepsilon_0}-1}} + \frac{\log(T)\left(e^{\varepsilon_0}+1\right)^2}{4} + \frac{\sqrt{T}}{\sqrt{e^{\varepsilon_0}-1}}\right) \tag{4}$$

On the other hand, Cor. 9 shows that SBLB interpolates between the regret in (Zheng et al., 2020) (LDP setting studied under scenario a) and (Shariff and Sheffet, 2018) (JDP setting studied under scenario c). The structure of the shuffle-model requires to also consider scenario b that, as mentioned before, poses the highest restriction on the regret bound we can achieve.

Corollary 9 *For any $\varepsilon \leq \frac{1}{27T^{1/4}}$ and $\delta, \delta_0 \in (0,1)$, the choices of $\eta = 0.5$, $\lambda = \sqrt{T}$, $m = 1$ and $\varepsilon_0 = \frac{d(d+3)}{2}\ln\left(\frac{2}{1-\varepsilon^{2/3}T^{1/6}} - 1\right)$ ensures that with probability at least $1-\delta$ the regret of* SBLB *is bounded by:*

$$R_T \leq \frac{4T^{2/3}}{\varepsilon^{1/3}}\left(S + d + \frac{1}{T^{1/4}}\widetilde{\mathcal{O}}(1)\right), \tag{5}$$

where $\widetilde{\mathcal{O}}(\cdot)$ hides poly-log factor (in T, δ, δ_0) and polynomial factors (in d, L). In addition SBLB *is $(\varepsilon, \delta_0 + \delta)$-JDP and $6d^2\varepsilon^{2/3}T^{1/6}$-LDP.*

For the complete regret bound refer to the end of App. B. This shows that the regret bound of SBLB is of order $\mathcal{O}\left(dT^{2/3}/\varepsilon^{1/3}\right)$, while being (ε, δ)-JDP and approximately $(2\varepsilon^{2/3}T^{1/6}, 0)$-LDP. As expected, this indicates the regret bound can be improved by sacrificing some level of LDP. However, the $\sqrt{T}$ regret bound of Shariff and Sheffet (2018) cannot be recovered directly. While the search for a better upper-bound or a lower-bound is an interesting future direction, we think it

would be hard to match such JDP result. Indeed, shuffling allows to interpolate between JDP (where the best worst-case upper bound is $\sqrt{T}$) and LDP (where the best known worst-case upper bound is $T^{3/4}$). Since we will always have a non-zero LDP level of privacy in the considered ESA shuffle model, we are not sure it is possible to achieve a $\sqrt{T}$ regret.[2]

4.2.1. PROOF SKETCH

The proof of this theorem is presented in details in App. B. To understand this result however we present how we build the confidence intervals around the parameter $\theta^\star$. As noticed in (Shariff and Sheffet, 2018), the estimator $\widetilde{\theta}_j$ is the result of a ridge regression computed by a design matrix regularized by a regularizer which is a function of the time. Therefore in order to apply Prop. 4 in (Shariff and Sheffet, 2018) we need to ensure that our estimator $\widetilde{V}_j$ of the design matrix, $\sum_t x_{t,a_t} x_{t,a_t}^\top$, is unbiased and to bound with high probability the deviation with respect to the design matrix. We also need the same type of guarantees with respect to the vector $\widetilde{B}_j$ and $\sum_t r_t x_{t,a_t}$.

Computation of our Estimators. The bandit algorithm receives the estimate (Z_{j^S}, U_{j^S}) from the shuffler but given the data those estimates are biased. For a couple of vector and reward, x and r, let us note $\mathcal{M}_{\text{LDP}}(x,r) = (b,w)$, so that

$$\mathbb{E}\left(b_{k,q} \mid x, r\right) = \frac{p}{2} + (1-p)\left[\mathbb{1}_{\{q<\lceil rx_k m\rceil\}} + \mathbb{1}_{\{q=\lceil rx_k m\rceil\}}(mrx_k - \lceil rx_k m\rceil + 1)\right]$$
$$\mathbb{E}\left(w_{k,l} \mid x, r\right) = \frac{p}{2} + (1-p)\left[\mathbb{1}_{\{q<\lceil x_l x_k m\rceil\}} + \mathbb{1}_{\{q=\lceil x_l x_k m\rceil\}}(mx_l x_k - \lceil x_l x_k m\rceil + 1)\right]$$

for all $k, l \leq d$ and $q \leq m$. Therefore, we introduce a debiased estimator for computing the estimators of SBLB, written as follows:[3]

$$\widetilde{V}_{j^A} = \sum_{t=1}^{t_{j^A}} \frac{x_{t,a_t} x_{t,a_t}^\top}{2L^2} + H_{j^A} + \lambda_{j^A} I_d \quad \text{and} \quad \widetilde{B}_{j^A} = \sum_{l=1}^{t_{j^A}} \frac{r_l x_{l,a_l}}{2L} + h_{j^A}, \tag{6}$$

where, for all batches, $H_{j^A} + \lambda^{j^A} I_d$ is with high probability a symmetric positive definite matrix decomposed as the sum of zero mean noise and a regularization λ_{j^A}, and h_{j^A} is a vector of zero mean noise. Both noises are due to the noise introduced by the local randomizer $\mathcal{M}_{\text{LDP}}$. In addition, as we show in App. B, controlling the eigenvalues of the regularizer $H_{j^A} + \lambda^{j^A} I_d$ and the noise h_{j^A} leads to a factor of roughly $\sqrt{t_{j^A}}$. Therefore thanks to Prop. 4 in (Shariff and Sheffet, 2018), the following proposition holds.

Lemma 10 (Confidence Ellipsoid) *For any $\delta \in (0,1)$, $\varepsilon_0 > 0$, $p = \frac{2}{e^{2\varepsilon_0/(md(d+3))}+1}$ and $\lambda > 0$, we have with probability at least $1-\delta$ that:*

$$\forall j^A \leq M_S, \qquad \|\theta^\star - \widetilde{\theta}_{j^A}\|_{\widetilde{V}_{j^A}^{-1}} \leq \beta_{j^A} := \sigma\sqrt{8\log\left(\frac{2t_{j^A}}{\delta}\right) + d\log\left(3 + \frac{t_{j^A}L^2}{\lambda_{j^A}}\right)} + S\sqrt{3\lambda_{j^A}}$$
$$+ \frac{d}{\sqrt{\lambda_{j^A}}}\left(2\sqrt{p\left(1-\frac{p}{2}\right)t_{j^A} m \log\left(\frac{2t_{j^A}}{\delta}\right)} + \frac{8\log(2t_{j^A}/\delta)}{3} + \frac{\sqrt{8}}{m}\sqrt{t_{j^A}\log\left(\frac{2t_{j^A}}{\delta}\right)}\right) \tag{7}$$

2. Note that in multi-armed bandit (MAB), it is possible to achieve a regret bound of order $\sqrt{T}$ both in central DP and LDP (Ren et al., 2020; Basu et al., 2019). We think this is an important aspect leveraged by Tenenbaum et al. (2021) for shuffling in MAB. In addition, as already mentioned, they considered a weaker attack model.
3. Note that this is an alternative but equivalent form to the one used in Alg. 1.

where $M_S = {}^T/_{l^\star}$ *is the number of shuffler batch and for all* $j^A \leq M_S$,

$$\lambda_{j^A} = \frac{\sqrt{8t_{j^A}\ln(2t_{j^A}/\delta)}}{m} + \frac{2\sqrt{8t_{j^A}\ln(2t_{j^A}/\delta)}}{(1-p)\sqrt{m}} + \lambda \tag{8}$$

Given the definition of the confidence ellipsoid above, we can analyze the regret using a standard regret analysis for algorithms using the optimism-in-the-face-of-uncertainty principle. For a generic set of privacy parameters ε_0, ε and δ_0, the regret bound of SBLB is given in the following theorem.

Theorem 11 *For any* $\delta, \delta_0 \in (0,1)$, $\varepsilon, \varepsilon_0 \in (0,1)$ *and* $T \geq 1$, *let* $p = 2(e^{2\varepsilon_0/md(d+3)}+1)^{-1}$ *then with probability at least* $1-\delta$, *the regret of Alg. 2 is bounded by:*

- *If* $p^2(1-p) \leq \frac{7T^{-1/2}\varepsilon}{64\sqrt{2\ln(2T/\delta_0)d(d+1)}}$:

$$\begin{aligned} R_T \leq{}& \frac{2\sqrt{3}(S+md)T^{3/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)} \\ &+ \frac{dLm}{\sqrt{\lambda}}\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\frac{14\log(8mT/\delta_0)}{p^2} \end{aligned} \tag{9}$$

- *If* $p^2(1-p) \geq \frac{7T^{-1/2}\varepsilon}{64\sqrt{2\ln(2T/\delta_0)d(d+1)}}$:

$$\begin{aligned} R_T \leq{}& \frac{2\sqrt{3}(S+md)T^{3/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)} \\ &+ \frac{264}{\sqrt{\lambda}}\sqrt{2}d^3\log\left(\frac{8mT}{\delta_0}\right)^{3/2} Lm\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\frac{\sqrt{T}(1-p)}{\varepsilon} \end{aligned} \tag{10}$$

The first term of the regret in Thm. 11 highlights the regret coming from the local privacy guarantees whereas the second term is coming from the mismatch between the batch of the shuffler and the batch of the bandit algorithm. Indeed, when the bandit algorithm updates its batch it means that during the last shuffler batch the determinant condition was satisfied at some point. However, the impact on the regret during this shuffler batch can only be bounded by the length of a shuffler batch times the maximum reward possible. Given that from Thm. 5 the length of a shuffler batch scales with $\widetilde{\mathcal{O}}\left({}^{\sqrt{T}}/_{\varepsilon}\right)$, the final regret scales with $\widetilde{\mathcal{O}}\left({}^{T^{3/4}}/_{\sqrt{1-p}} + {}^{\sqrt{T}}/_{\varepsilon}\right)$. As a consequence, Cor. 8 and Cor. 9 are obtained by optimizing for the highest privacy level and smaller regret bound, respectively.

Remark 12 *A better regret bound can be obtained in the setting of (Tenenbaum et al., 2021), where the adversary only observes the output of the shuffler for one time step. In particular, this allows to improve the privacy analysis and obtain a generic regret bound of order* $\mathcal{O}\left({}^{T^{3/4}}/_{\sqrt{1-p}} + {}^{\log(T)}/_{\varepsilon^2}\right)$ *that once optimized leads to a regret bound of* $T^{3/5}/\varepsilon^{2/5}$ *which is much closer to the best JDP regret bound. However, we think this setting is less of practical interest than the one considered in this paper.*

5. Conclusion

We introduced SBLB, an algorithm for linear contextual bandits that achieves a trade-off between joint and local differential privacy. Our algorithm is a variant of batched LINUCB with dynamic schedule using a variant of the binary sum method to achieve privacy. Thanks to an asynchronous batch schedule between shuffler and bandit algorithm, it is able to take advantage of the privacy amplification through shuffling to reduce the gap between JDP and LDP regret bounds.

An interesting question raised by our paper is whether it is possible to use a synchronous schedule between the shuffler and the bandit algorithm, e.g., by making the shuffler batch data dependent. We believe this would require to use some private technique (e.g., sparse vector technique by Dwork et al., 2009) to guarantee privacy at the output of the shuffler. Another direction inspired by our paper is to gain a better understanding about the intrinsic limitations of differential privacy in linear contextual bandits by studying lower-bounds for these setting.

Acknowledgments

V. Perchet acknowledges support from the French National Research Agency (ANR) under grant number #ANR-19-CE23-0026 as well as the support grant, as well as from the grant "Investissements d'Avenir" (LabEx Ecodec/ANR-11-LABX-0047).

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

Apple. Learning with privacy at scale. https://machinelearning.apple.com/research/learning-with-privacy-at-scale.

Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 638–667. Springer, 2019.

Debabrota Basu, Christos Dimitrakakis, and Aristide C. Y. Tossou. Differential privacy for multi-armed bandits: What is it and what is its cost? *CoRR*, abs/1905.12298, 2019.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, pages 441–459. ACM, 2017.

Etienne Boursier and Vianney Perchet. Utility/privacy trade-off through the lens of optimal transport. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pages 591–601. PMLR, 2020.

Alexandra Carpentier, Claire Vernade, and Yasin Abbasi-Yadkori. The elliptical potential lemma revisited. *CoRR*, abs/2010.10182, 2020.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Lijie Chen, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On distributed differential privacy and counting distinct elements. In *ITCS*, volume 185 of *LIPIcs*, pages 56:1–56:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. *Lecture Notes in Computer Science*, page 375–403, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-17653-2_13.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390. ACM, 2009.

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*, pages 715–724, 2010.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

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 *SODA*, pages 2468–2479. SIAM, 2019.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618, 2020.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling, 2020.

Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differentially private regret minimization in reinforcement learning. *CoRR*, abs/2010.07778, 2020.

Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose H. Blanchet, Peter W. Glynn, and Yinyu Ye. Sequential batch learning in finite-action linear contextual bandits. *CoRR*, abs/2004.06321, 2020.

Hongzi Mao, Shannon Chen, Drew Dimmery, Shaun Singh, Drew Blaisdell, Yuandong Tian, Mohammad Alizadeh, and Eytan Bakshy. Real-world video adaptation with reinforcement learning. *CoRR*, abs/2008.12858, 2020.

Goldreich Oded. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, USA, 1st edition, 2009. ISBN 052111991X.

Wenbo Ren, Xingyu Zhou, Jia Liu, and Ness B Shroff. Multi-armed bandits with local differential privacy. *arXiv preprint arXiv:2007.03121*, 2020.

Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *NeurIPS*, pages 4301–4311, 2018.

Jay Tenenbaum, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Differentially private multi-armed bandits in the shuffle model. *CoRR*, abs/2106.02900, 2021.

Giuseppe Vietri, Borja de Balle Pigem, Akshay Krishnamurthy, and Steven Wu. Private reinforcement learning with pac and regret guarantees. In *ICML*, 2020.

Haoran Wang and Shi Yu. Robo-advising: Enhancing investment with inverse optimization and deep reinforcement learning. In *ICMLA*, pages 365–372. IEEE, 2021.

Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. Locally differentially private (contextual) bandits learning. In *NeurIPS*, 2020.

Appendix A. Local Privatizer $\mathcal{M}_{\text{LDP}}$

In this appendix, we present the privacy-preserving mechanism $\mathcal{M}_{\text{LDP}}$ used in this paper.

Algorithm 2: Local Privatizer $\mathcal{M}_{\text{LDP}}$

Input: context: $x \in \mathbb{R}^d$, reward: $r \in [0,1]$, context bound: L, privacy parameter: ε_0, encoding parameter: m

//*Encoder*//

Set $\tilde{y} = \frac{rx}{2L} + \frac{1}{2}$ and $\tilde{z} = \frac{xx^\top}{2L^2} + \frac{\mathbb{1}\mathbb{1}^\top}{2}$

for $j = 1, \dots, d$ **do**

  Compute $\mu_j = \lceil \tilde{y}_j \cdot m \rceil$ and $p_j = m \cdot \tilde{y}_j - \mu_j + 1$

  for $k = 1, \dots, m$ **do**

    Let $b_{j,k} = \begin{cases} 1 & \text{if } k < \mu_j \\ \text{Ber}(p_j) & \text{if } k = \mu_j \\ 0 & \text{if } k > \mu_j \end{cases}$

  end

end

for $i = 1, \dots, d$ **do**

  for $j = 1, \dots, i$ **do**

    Compute $\kappa_{i,j} = \lceil \tilde{z}_{i,j} \cdot m \rceil$ and $q_{i,j} = m \cdot \tilde{z}_{i,j} - \kappa_{i,j} + 1$

    for $k = 1, \dots, m$ **do**

      Let $w_{i,j,k} = \begin{cases} 1 & \text{if } k < \kappa_{i,j} \\ \text{Ber}(q_{i,j}) & \text{if } k = \kappa_{i,j} \\ 0 & \text{if } k > \kappa_{i,j} \end{cases}$

      Let $w_{j,i,k} = w_{i,j,k}$

    end

  end

end

//*Local Randomizer*//

Set probabilities $p = \frac{2}{\exp(2\varepsilon_0/md(d+3))+1}$ and compute private values

$\tilde{b}_j = \left(R_p^{0/1}(b_{j,1}), \dots, R_p^{0/1}(b_{j,m}) \right)$ for all $j \in [\![1, d]\!]$,

$\tilde{w}_{i,j} = \left(R_p^{0/1}(w_{i,j,1}), \dots, R_p^{0/1}(w_{i,j,m}) \right)$ for all $i \in [\![1, d]\!]$, $j \le i$

Algorithm 3: Local Randomizer $R_p^{0/1}$

Input: probability: p, $x \in \{0, 1\}$

Let $\mathbf{b} \sim \text{Ber}(p)$;

if $\mathbf{b} = 0$ **then**

  Return x

else

  Return $\text{Ber}(1/2)$

end

Appendix B. Proofs

In this appendix, we provide the full derivation of the results stated in the main text. We start introducing the notion of central (ε, δ)-DP that is widely used in the proofs.

Definition 13 *A randomized mechanism, $\mathcal{M} : \mathbb{R}^d \to \mathcal{Z}$, is said to be central (ε, δ) differential private (DP) if for all sequence of values $z \in \mathcal{R}^d$ and z' such that there exists a unique $i \leq t$ for which $z_i \neq z'_i$ and for all $j \neq i$, $z_j = z'_j$ then*

$$\mathbb{P}\left(\mathcal{M}(z) \in A \mid z\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathcal{M}(z) \in A \mid z'\right) + \delta$$

for any $A \subset Range(\mathcal{M})$.

Note that the concept of central DP is at the core for proving JDP results, in fact thanks to Claim 7 in (Shariff and Sheffet, 2018) having a sequence $(\widetilde{V}_t, B_t)_t$ is (ε, δ)-DP implies that a bandit algorithm based on this sequence is (ε, δ)-DP.

B.1. Proof of Lem. 10

Here, we detail how to obtain the confidence intervals around $\theta^\star$ using the privatized estimator $\widetilde{\theta}_j$ for any batch $j^S \leq M_S$ (with $M_S = Tl^\star$ the total number of batches from the shuffler side). First, let's define the sequence of random variables $(Y_{t,k,l,q})_{t\leq T,k,l\leq d,q\leq m}$, $(Z_{t,k,l,q})_{t\leq T,k,l\leq d,q\leq m}$ two independent sequences of i.i.d. Bernoulli distributed random variable with parameters $p = 2/(\exp(2\varepsilon_0/md(d+3))+1)$ and $1/2$ and such that for all $k, l \leq d$, $Y_{t,k,l,q} = Y_{t,l,k,q}$ and $Z_{t,k,l,q} = Z_{t,l,k,q}$. For every $(t,k,l,q) \in [T] \times [d] \times [d] \times [m]$, $Y_{t,k,l,q}$ is sampled by Alg. 3 if $Y_{t,k,l,q} = 1$ then it return the random variable $Z_{t,k,l,q}$ otherwise it returns the true data.

In addition, let's define $(A_{t,k,l} = w_{t,k,l,\kappa_{t,k,l}})_{t\leq T,k,l\leq d}$ a sequence of Bernoulli random variable with parameter $(q_{t,k,l})_{t\leq T,k,l\leq d}$ defined by the two sequences $(\widetilde{z}_{t,k,l})_{t\leq T,k,l\leq d}$ and $(\kappa_{t,k,l})_{t\leq T,k,l\leq d}$ in the mechanism $\mathcal{M}_{\text{LDP}}$, Alg. 2. Finally, let's note the sequence of data computing by the encoding part of Alg. 2.

For any batch $j^A \leq M_S$, we can write the approximate design matrix and vector B_j as follows for every coordinate $k, l \leq d$:

$$\begin{aligned}\widetilde{V}_{j,k,l} =& \frac{1}{m(1-p)}\sum_{t=1}^{t_j}\sum_{q=1}^{m} Y_{t,k,l,q}Z_{t,k,l,q} - \frac{p}{2} + \sum_{t=1}^{t_j}\frac{x_{t,a_t}x_{t,a_t}^\top}{2L^2} + +2\lambda_j \mathbb{1}_{\{k=l\}} \\ &+ \frac{1}{m}\sum_{t=1}^{t_j} A_{t,k,l} - (m\widetilde{z}_{t,k,l} - \kappa_{t,k,l} + 1) + \frac{1}{m(1-p)}\sum_{t=1}^{t_j}\sum_{q=1}^{m}(p - Y_{t,k,l,q})w_{t,k,l,q}\end{aligned} \tag{11}$$

where λ_j is defined in Eq. (8).

$$\widetilde{B}_{j,k} = \frac{1}{m(1-p)}\sum_{l=1}^{t_j}\sum_{q=1}^{m}\left(\tilde{b}_{l,i,q} - \frac{p}{2}\right) - \frac{t_j}{2} \tag{12}$$

Now, given an well-chosen regularization λ_j the approximate design matrix $\widetilde{V}_j$ can be written as the sum of the true design matrix $\sum_t x_{t,a_t}x_{t,a_t}^\top$ and a time-varying regularizer similar to (Shariff

and Sheffet, 2018). We just need to bound with high probability the deviation of the eigenvalues of $\widetilde{V}_j - \sum_{t=1}^{t_j} \frac{x_{t,a_t} x_{t,a_t}^\top}{2L^2} - \lambda_j I_d$.

Let's consider a vector $v \in \mathbb{R}^d$ such that $\|v\|_2 = 1$ then for any time $t_j \leq T$ and $\delta \in (0,1)$ we have with probability at least $1-\delta$:

$$\left| \left\langle v, \left(\sum_{t=1}^{t_j} \sum_{q=1}^{m} Y_{t,.,.,q} Z_{t,.,.,q} - \frac{p \mathbb{1}\mathbb{1}^\top}{2} \right) v \right\rangle \right| \leq 2\sqrt{2 t_j m \ln(2/\delta)} \tag{13}$$

Therefore because the matrix $\left(\sum_{t=1}^{t_j} \sum_{q=1}^{m} Y_{t,.,.,q} Z_{t,.,.,q} - \frac{p \mathbb{1}\mathbb{1}^\top}{2} \right)$ is symmetric we have that with high probability:

$$\max \left\{ \left| \lambda_{\min} \left(\sum_{t,q} Y_{t,.,.,q} Z_{t,.,.,q} - \frac{p \mathbb{1}\mathbb{1}^\top}{2} \right) \right|, \lambda_{\max} \left(\sum_{t,q} Y_{t,.,.,q} Z_{t,.,.,q} - \frac{p \mathbb{1}\mathbb{1}^\top}{2} \right) \right\} \leq 2\sqrt{2 t_j m \ln(2/\delta)}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues. Similarly, using the martingale difference structure,we have that for any $v \in \mathbb{R}^d$, $\|v\|_2 \leq 1$ and $\delta \in (0,1)$, we have with probability at least $1-\delta$:

$$\left| \left\langle v, \left(\sum_{t=1}^{t_{j+1}} \sum_{q=1}^{m} (p \mathbb{1}\mathbb{1}^\top - Y_{t,.,.,q}) w_{t,.,.,q} \right) v \right\rangle \right| \leq 2\sqrt{2 t_{j+1} m \ln(2/\delta)} \tag{14}$$

and

$$\left| \left\langle v, \left(\sum_{t=1}^{t_{j+1}} A_t - (m\widetilde{z}_t - \kappa_t + 1) \right) v \right\rangle \right| \leq 2\sqrt{2 t_{j+1} \ln(2/\delta)} \tag{15}$$

Indeed, for every $t \leq T$, let's define the filtration $\mathcal{F}_t$ which is the filtration generated by all the history up to time t included except for the noise added by the mechanism $\mathcal{M}_{\text{LDP}}$ that is to say $\mathcal{F}_t = \sigma((x_{l,a_l}, r_l)_{l \leq t}, (Y_{l,i,j,q})_{l<t-1, i,j \leq d, q \leq m}, (Z_{l,i,j,q})_{l<t-1,i,j \leq d, q \leq m}, (w_{t,k,l,q})_{l<t-1,i,j \leq d, q \leq m})$. Therefore, we have that:

$$\begin{aligned} &\mathbb{E}\left((p - Y_{t,k,l,q}) w_{t,k,l,q} \mid \mathcal{F}_t\right) = \mathbb{E}(p - Y_{t,k,l,q}) \mathbb{E}\left(w_{t,k,l,q} \mid \mathcal{F}_t\right) = 0 \\ &\mathbb{E}\left(A_{t,k,l} - (m\widetilde{z}_{t,k,l} - \kappa_{t,k,l} + 1) \mid \mathcal{F}_t\right) = \mathbb{E}(A_{t,k,l} \mathcal{F}_t) - (m\widetilde{z}_{t,k,l} - \kappa_{t,k,l} + 1) = 0 \end{aligned} \tag{16}$$

because Y_t is independent of $\mathcal{F}_t$ and w_t. The second equality comes from the fact that given $\mathcal{F}_t$, $A_{t,k,l}$ is a Bernoulli random variable with parameter $m\widetilde{z}_{t,k,l} - \kappa_{t,k,l} + 1$.

Hence, when choosing $\lambda_j = \frac{\sqrt{8 t_j \ln(2t_j/\delta)}}{m} + \frac{2\sqrt{8 t_j \ln(2t_j/\delta)}}{(1-p)\sqrt{m}}$, we have that with probability at least $1-\delta$:

$$\forall j \leq M_S, \qquad \lambda_{\min} \left(\widetilde{V}_j - \sum_{t=1}^{t_j} \frac{x_{t,a_t} x_{t,a_t}^\top}{2L^2} \right) \geq \frac{\sqrt{8 t_j \ln\left(\frac{2t_j}{\delta}\right)}}{m} + \frac{2\sqrt{8 t_j \ln\left(\frac{2t_j}{\delta}\right)}}{(1-p)\sqrt{m}} + \tag{17}$$

$$\lambda_{\max} \left(\widetilde{V}_j - \sum_{t=1}^{t_j} \frac{x_{t,a_t} x_{t,a_t}^\top}{2L^2} \right) \leq \frac{2\sqrt{8 t_j \ln\left(\frac{2t_j}{\delta}\right)}}{m} + \frac{4\sqrt{8 t_j \ln\left(\frac{2t_j}{\delta}\right)}}{(1-p)\sqrt{m}} \tag{18}$$

In addition, with the same reasoning, we have with probability at least $1-\delta$:

$$\left\|\sum_{l=1}^{t_{j+1}} \frac{r_l x_{l,a_l}}{2L} - B_j\right\| \leq 2\sqrt{dp\left(1-\frac{p}{2}\right)t_j m \log\left(\frac{2t_j}{\delta}\right)} + \frac{4}{3}\sqrt{d}\log\left(\frac{2t_j}{\delta}\right) + \frac{2}{m}\sqrt{dt_j \log\left(\frac{2t_j}{\delta}\right)} \tag{19}$$

Therefore, using Prop. 5 in (Shariff and Sheffet, 2018), we have that the result.

B.2. Proof of Prop. 4

We now move to prove the following proposition which implies Prop. 4;

Proposition 14 *For any encoding parameter $m \in \mathbb{N}^\star$ and LDP parameter $\varepsilon_0 > 0$, $\mathcal{M}_{LDP}(x,r)$ is ε_0-LDP for any $\|x\| \leq L$ and $r \in [0,1]$.*

Proof For any $x, x' \in \mathbb{R}^d$ and $r, r' \in [0,1]$ such that $\|x\| \leq L$ and $\|x'\| \leq L$ let's note $\mathcal{M}_{\text{LDP}}(x,r) = \left((\tilde{w}_{i,j})_{i,j\leq d}, (\tilde{b}_j)_{j\leq d}\right) \in \{0,1\}^{d^2m\times dm}$ and $\mathcal{M}_{\text{LDP}}(x',r') = \left((\tilde{w}'_{i,j})_{i,j\leq d}, (\tilde{b}'_j)_{j\leq d}\right) \in \{0,1\}^{d^2m\times dm}$. Therefore, let's consider a tuple $(W_0, B_0) \in \{0,1\}^{d^2m\times dm}$ then we want to show that:

$$\mathbb{P}\left(\mathcal{M}_{\text{LDP}}(x,r) = (W_0,B_0)\right) \leq e^{\varepsilon_0}\mathbb{P}\left(\mathcal{M}_{\text{LDP}}(x',r') = (W_0,B_0)\right) \tag{20}$$

But we have:

$$\mathbb{P}\left(\forall i,j\leq d, \tilde{w}_{i,j} = W_{0,i,j}, \tilde{b}_j = B_{0,j}\right) = \mathbb{P}\left(\forall i,j\leq d, \tilde{w}_{i,j} = W_{0,i,j}\right)\mathbb{P}\left(\forall j\leq d, \tilde{b}_j = B_{0,j}\right) \tag{21}$$

In addition, because the mechanism $R_p^{0/1}$ is an example of a randomized response mechanism (Dwork et al., 2010), we have that for all $j \leq d, q \leq m$, $\mathbb{P}(R_p^{0/1}(b_{j,m}) \mid b_{j,m}) \leq (2/p-1)\mathbb{P}(R_p^{0/1}(b'_{j,m}) \mid b'_{j,m})$. Therefore, because of the independence of the sequence $(\tilde{b}_j)_j$:

$$\begin{aligned}\mathbb{P}\left(\forall j\leq d, \tilde{b}_j = B_{0,j}\right) &= \prod_{j,q}\mathbb{P}\left(\tilde{b}_{j,q} = B_{0,j,q}\right) \\ &\leq \prod_{j,q}\mathbb{P}\left(\tilde{b}'_{j,q} = B_{0,j,q}\right)\left(\frac{2}{p}-1\right) = \left(\frac{2}{p}-1\right)^{dm}\mathbb{P}\left(\forall j, \tilde{b}'_j = B_{0,j}\right)\end{aligned} \tag{22}$$

For all $i,j \leq d$, we have that $\tilde{w}_{i,j} = \tilde{w}_{j,i}$ therefore:

$$\begin{aligned}\mathbb{P}\left(\forall i,j\leq d, \tilde{w}_{i,j} = W_{0,i,j}\right) &= \prod_{i,j\leq i,q}\mathbb{P}\left(\tilde{w}_{i,j,q} = W_{0,i,j,q}\right) \\ &\leq \prod_{i,j\leq i,q}\left(\frac{2}{p}-1\right)\mathbb{P}\left(\tilde{w}'_{i,j,q} = W_{0,i,j,q}\right) \\ &= \left(\frac{2}{p}-1\right)^{md(d+1)/2}\mathbb{P}\left(\forall i,j\leq d, \tilde{w}'_{i,j} = W_{0,i,j}\right)\end{aligned} \tag{23}$$

Hence the resulting when setting $p = \frac{2}{\exp\left(\frac{\varepsilon_0}{md(d+3)/2}\right)+1}$. ■

B.3. Proof of Thm. 5

Before proving the JDP guarantees of our algorithm, that is to say Thm. 5. We first prove the following proposition that is a consequence of Thm. 5.4 in (Cheu et al., 2019).

Proposition 15 *For any $\delta_0, \delta \in (0,1)$, number of batch M_S and length l, encoding parameter m, LDP parameter $0 < \varepsilon_0 \leq \ln\left(\frac{l}{(7\ln(8m/\delta_0))} - 1\right)$ and for all batch $j \leq M_S$ of length l, the statistics (Z_j, U_j) computed by the shuffler (with $p = 2/(e^{2\varepsilon_0/md(d+3)} + 1)$)) are $(\varepsilon_{j,c}, \delta + \delta_0)$-DP with*

$$\frac{\varepsilon_{j,c}}{2d(d+3)\sqrt{8m\log(8m/\delta_0)}} = \left(1 - \left(p - \sqrt{\frac{2p\log\left(\frac{2m}{\delta_0}\right)}{l}}\right)\right)\sqrt{\frac{32\log(8m/\delta_0)}{l\left(p - \sqrt{\frac{2p\log(8\delta_0/m)}{l}}\right)}} \tag{24}$$

Proof [of Prop. 15] Let's consider $\delta \in (0,1)$ and define

$$E_\delta = \bigcap_{T=1}^{+\infty}\left\{\left\|\frac{1}{m(1-p)}\sum_{t=1}^{T}\sum_{q=1}^{m} Y_{t,\dots,q}Z_{t,\dots,q} - \frac{p}{2}\mathbb{1}\mathbb{1}^\top\right\| + \left\|\frac{1}{m}\sum_{t=1}^{T} A_t - (m\widetilde{z}_t - \widetilde{\theta}_t + 1)\right\| + \left\|\frac{1}{m(1-p)}\sum_{t=1}^{T}\sum_{q=1}^{m}(p - Y_{t,\dots,q})w_{t,\dots,q}\right\| \leq \frac{\sqrt{8T\ln(2T/\delta)}}{m} + \frac{2\sqrt{8T\ln(2T/\delta)}}{(1-p)\sqrt{m}}\right\}$$

This event is such that $\mathbb{P}(E_\delta) \geq 1 - \delta$. Therefore for a batch j and any event A, we have that:

$$\begin{aligned}
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A\Big) = \\
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A, \mathcal{E}_\delta\Big) \\
&+ \mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A, \mathcal{E}_\delta^c\Big)
\end{aligned}$$

Therefore, we have that:

$$\begin{aligned}
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A\Big) \leq \\
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A, \mathcal{E}_\delta\Big) + \delta
\end{aligned}$$

And thanks to the definition of privacy with shuffling we have that:

$$\begin{aligned}
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x_{\sigma_j(t)}x_{\sigma_j(t)}^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A\Big) \leq \\
&\mathbb{P}\left(\left(\mathcal{M}_{LDP}(x'_{\sigma_j(t)}(x'_{\sigma_j(t)})^\top, r_{\sigma_j(t)}x_{\sigma_j(t)})\right)\right)_{t\in[\![t_j+1,t_{j+1}]\!]} \in A\Big)\exp(\varepsilon_{j,c}) + \delta_0 + \delta
\end{aligned}$$

where $\varepsilon_{j,c}$ is such that:

$$\frac{\varepsilon_{j,c}}{2d(d+3)\sqrt{8m\log\left(\frac{8m}{\delta_0}\right)}} = \left(1 - \left(p - \sqrt{\frac{2p\log\left(\frac{2m}{\delta_0}\right)}{l}}\right)\right)\sqrt{\frac{32\log(8m/\delta_0)}{l\left(p - \sqrt{\frac{2p\log(8\delta_0/m)}{l}}\right)}} \tag{25}$$

according to Thm. 5.4 in (Cheu et al., 2019). ■

Now let's consider a set of parameters $\delta_0, \delta \in (0,1)$ and $\varepsilon, \varepsilon_0 \in (0,1)$ and a length l that satisfies Eq. (3). Such length l exists for any $p \in [0,1]$ as

$$\lim_{l\to+\infty} 2\sqrt{\frac{2\log(2m/\delta_0)}{l}} + \left(\frac{l\varepsilon}{2^5 d(d+3)\log(8m/\delta_0)\sqrt{2T\ln(1/\delta_0)}}\right)^2 - \sqrt{\left(2+\left(\frac{l\varepsilon}{2^5 d(d+3)\log(8m/\delta_0)\sqrt{2T\ln(1/\delta_0)}}\right)^2\right)^2 - 4} = -2 \tag{26}$$

Therefore, thanks to Prop. 15, we have that each update to the design matrix is $\left(\frac{\varepsilon\sqrt{l}}{\sqrt{T}}, \delta_0 + \delta\right)$-DP. Therefore, using advanced composition yields the result.

B.4. Proof of Thm. 11

Let's now move on to the proof of the main theorem, Thm. 11. Let's note $l^\star = T/M_S$ where M_S is the number of batch from the shuffler point of view, this parameter is given to the shuffler. Now let's consider a shuffler batch $j \leq M_S$, sent to the bandit algorithm, let's note then $q_j < j$ the last shuffler batch where Alg. 1 has updated the estimate $\widetilde{\theta}$. Therefore, if Alg. 1 decides to update the parameter $\widetilde{\theta}$ after receiving the data from the shuffler batch j, we have that:

$$\det(\tilde{V}_j) \geq (1+\eta)\det(\tilde{V}_{q_j}) \tag{27}$$

Let's consider any bandit batch r, between time $t_r + 1$ and t_{r+1} we can then decompose the interval $\{t_r+1, \dots, t_{r+1}\}$ into successive shuffler batches and we note the last of them j_r. That is to say, upon receiving the shuffler batch j_r and t_{j_r} the time step at which this batch begins, Alg. 1 updates the parameter $\widetilde{\theta}$, so increasing the bandit batch from r to $r+1$. Therefore, for all shuffler batch $j \leq j_r - 1$, we have that $\det(\widetilde{V}_j) \leq (1+\eta)\det(\widetilde{V}_r)$ therefore for any vector $x \in \mathbb{R}^d$, $\left|\langle \theta^\star - \widetilde{\theta}_r, x\rangle\right| \leq \sqrt{1+\eta}\beta_r \|x\|_{\widetilde{V}_r^{-1}}$ (see App. D in (Abbasi-Yadkori et al., 2011)). In addition, for any time step t during a batch j, $\|x\|_{\widetilde{V}_j^{-1}} \leq \|x\|_{\widetilde{V}_t^{-1}}$ where $\widetilde{V}_t$ is the design matrix computed with only data from the first t time steps. In addition for $t \in \{t_{j_r}, \dots, t_{r+1}\}$, we have that the norm $\|x\|_{\widetilde{V}_r^{-1}}$ can not be related to the norm of $\|x\|_{\widetilde{V}_t^{-1}}$ but we have that:

$$\left|\langle \theta^\star - \widetilde{\theta}_r, x\rangle\right| \leq \beta_r \|x\|_{\widetilde{V}_r^{-1}} \leq \frac{\beta_r \|x\|_2}{\sqrt{\lambda_{\min}(V_r)}} \tag{28}$$

Therefore, we can write the regret as:

$$R_T = \sum_{t=1}^{T} \langle \theta^\star, x_{t,a_t^\star} - x_{t,a_t}\rangle = \sum_{p=0}^{M_R} \sum_{t=t_p+1}^{t_{p+1}} \langle \theta^\star, x_{t,a_t^\star} - x_{t,a_t}\rangle \tag{29}$$

where M_R is the number of batch of Alg. 1. Using the reasoning above, we have:

$$R_T \leq \sum_{p=0}^{M_R-1} \sum_{t=t_p+1}^{t_{j_p}} 2\beta_p \sqrt{1+\eta} \|x_{t,a_t}\|_{\widetilde{V}_t^{-1}} + \sum_{t=t_{j_p}+1}^{t_{p+1}} 2\beta_p \|x_{t,a_t}\|_{\widetilde{V}_{t_p}^{-1}} \tag{30}$$

$$\leq 2\beta_T \sum_{t=1}^{T} \sqrt{1+\eta} \|x_{t,a_t}\|_{\widetilde{V}_t^{-1}} + \sum_{p=0}^{M_R-1} \frac{2\beta_p L l^\star}{\sqrt{\lambda_{\min}(\widetilde{V}_p)}} \tag{31}$$

where $l^\star$ is the length of a shuffler batch. In addition, the design matrix $\widetilde{V}_p$ is regularized to ensure that its minimum eigenvalues grows at a rate of $\sqrt{t_p}$. Therefore we have that for any bandit algorithm batch r:

$$\frac{2\beta_r L l^\star}{\sqrt{\lambda_{\min}(\widetilde{V}_r)}} \leq 2Ll^\star \left(\frac{\sigma\sqrt{2\log\left(\frac{2T}{\delta}\right) + d\log\left(3+\frac{TL^2}{\lambda}\right)}}{\sqrt{\lambda_r}} + S\sqrt{3} \right.$$
$$\left. + \frac{d\left(\sqrt{t_r m \log\left(\frac{2}{\delta}\right)} + \frac{2\log(2/\delta)}{3} + \frac{\sqrt{2}}{m}\sqrt{t_r \log\left(\frac{2}{\delta}\right)}\right)}{\lambda_r} \right)$$

Therefore using (Carpentier et al., 2020), the regret can be bounded by:

$$R_T \leq 2\beta_T \sqrt{(1+\eta)\, T \log\left(1+\frac{T}{d\lambda}\right)} + \sum_{r=0}^{M_R-1} 2Ll^\star \left(\frac{\sigma\sqrt{2\log\left(\frac{2T}{\delta}\right) + d\log\left(3+\frac{TL^2}{\lambda}\right)}}{\sqrt{\lambda_r}} \right.$$
$$\left. + S\sqrt{3} + \frac{d\left(\sqrt{t_r m \log\left(\frac{2}{\delta}\right)} + \frac{2\log(2/\delta)}{3} + \frac{\sqrt{2}}{m}\sqrt{t_r \log\left(\frac{2}{\delta}\right)}\right)}{\lambda_r} \right) \tag{32}$$

We now proceed to bound each term individually. First, we have:

$$\sum_{r=0}^{M_R-1} 2\sqrt{3} L l^\star S \leq 2\sqrt{3} L S l^\star M_R \tag{33}$$

This is because the shuffler sends data on a fix length schedule. Also, we have:

$$\sum_{r=0}^{M_R-1} 2Ll^\star \frac{\sigma\sqrt{2\log\left(\frac{2T}{\delta}\right) + d\log\left(3+\frac{TL^2}{\lambda}\right)}}{\sqrt{\lambda_r}} \leq \frac{2M_R L l^\star \sigma}{\sqrt{\lambda}} \sqrt{2\log\left(\frac{2}{\delta}\right) + d\log\left(3+\frac{TL^2}{\lambda}\right)} \tag{34}$$

Finally,

$$\sum_{r=0}^{M_R-1} \frac{2Ll^\star d}{\lambda_r} \left(\sqrt{t_r m \log\left(\frac{2}{\delta}\right)} + \frac{2\log(2T/\delta)}{3} + \frac{\sqrt{2}}{m}\sqrt{t_r \log\left(\frac{2}{\delta}\right)} \right) \leq \frac{4Ll^\star d M_R}{3} \log\left(\frac{2T}{\delta}\right)$$
$$+ \frac{\sqrt{2} L l^\star d M_R m}{4} \tag{35}$$

Therefore with probability at least $1-\delta$ the regret is bounded by:

$$R_T \leq \underbrace{2\beta_T\sqrt{(1+\eta)\,T\log\left(1+\frac{T}{d\lambda}\right)}}_{:=ⓐ} + \frac{2M_R L l^\star \sigma}{\sqrt{\lambda}}\sqrt{2\log\left(\frac{2}{\delta}\right)+d\log\left(3+\frac{TL^2}{\lambda}\right)} + \frac{4Ll^\star d M_R}{3}\log\left(\frac{2T}{\delta}\right) + \frac{\sqrt{2}Ll^\star d M_R m}{4} + 2\sqrt{3}LSl^\star M_R \tag{36}$$

Bounding ⓐ. Given the expression of β_T, we have that:

$$\begin{aligned} ⓐ \leq \sigma\sqrt{\left(8\log\left(\frac{2T}{\delta}\right)+d\log\left(3+\frac{TL^2}{\lambda}\right)\right)(1+\eta)\,T\log\left(1+\frac{T}{d\lambda}\right)} \\ + \frac{2\sqrt{3}ST^{1/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\,T\log\left(1+\frac{T}{d\lambda}\right)} \\ + \left(4dmT^{1/4}+\frac{8\log(2T/\delta)\sqrt{m}}{3}\right)\sqrt{(1+\eta)\,T\log\left(1+\frac{T}{d\lambda}\right)} \end{aligned} \tag{37}$$

Now, we are left with bounding the remaining of the right hand part of Eq. (36). The first step to do so is to notice that the number of bandit algorithm batch is bounded by roughly $\mathcal{O}\left(\log(T)\right)$, more precisely:

$$M_R \leq 1 + \frac{d\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)}{\log(1+\eta)} \tag{38}$$

In addition, if $l^\star$ satisfies Eq. (3) then we have that:

$$l^\star \leq \max\left\{\frac{8\log(2m/\delta_0)}{p^2}, \frac{128\sqrt{2T\ln(2/\delta_0)}d(d+1)\log(8m/\delta_0)(1-p)}{\varepsilon}, \frac{14\log(2m/\delta_0)}{p}\right\} \tag{39}$$

In Eq. (39) we have that the regret is bounded with probability at least $1-\delta$:

$$\begin{aligned} R_T \;\leq\; & \frac{2\sqrt{3}(S+md)T^{3/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)} \\ & + \frac{dLm}{\sqrt{\lambda}}\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\times \\ & \times\max\left\{\frac{14\log(8m/\delta_0)}{p^2}, \frac{128\sqrt{2T\ln\left(\frac{2}{\delta_0}\right)}d(d+1)\log\left(\frac{8m}{\delta_0}\right)(1-p)}{\varepsilon}\right\} \end{aligned} \tag{40}$$

Therefore, we can differentiate two different scenarios:

- If $p^2(1-p) \leq \frac{7T^{-1/2}\varepsilon}{64\sqrt{2\ln(2/\delta_0)}d(d+1)}$:

$$R_T \leq \frac{2\sqrt{3}(S+md)T^{3/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)} \tag{41}$$

$$+\frac{dLm}{\sqrt{\lambda}}\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\frac{14\log(8m/\delta_0)}{p^2} \tag{42}$$

- If $p^2(1-p) \geq \frac{7T^{-1/2}\varepsilon}{64\sqrt{2\ln(2/\delta_0)}d(d+1)}$:

$$R_T \leq \frac{2\sqrt{3}(S+md)T^{3/4}}{\sqrt{1-p}}\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)}$$
$$+\frac{264}{\sqrt{\lambda}}\sqrt{2}d^3\log\left(\frac{8m}{\delta_0}\right)^{3/2}Lm\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\frac{\sqrt{T}(1-p)}{\varepsilon} \tag{43}$$

The last step now is to choose the parameter ε_0 to optimize the regret. Therefore, if $\varepsilon \leq \frac{1}{27T^{1/4}}$, so in a high privacy regime, when choosing $p = 1-\varepsilon^{2/3}T^{1/6}$ we are in the second scenario above and:

$$R_T \leq \frac{T^{2/3}}{\varepsilon^{1/3}}\left[\frac{264}{\sqrt{\lambda}}\sqrt{2}d^3\log\left(\frac{8m}{\delta_0}\right)^{3/2}Lm\left(1+\frac{d^{3/2}\log\left(\frac{L^2T}{d}+\frac{16\sqrt{T}\log(2T/\delta)}{(1-p)}\right)^{3/2}}{\log(1+\eta)}\right)\right.$$
$$\left.+2\sqrt{3}(S+md)\sqrt{(1+\eta)\log\left(1+\frac{T}{d\lambda}\right)}\right]$$

Appendix C. Regret with Scheduled Update Algorithm

In this appendix, we present a bandit algorithm using a fixed schedule update instead of the determinant based condition used in Alg. 1. The main consequence of using a fixed batch bandit algorithm is a worse regret compared to Alg. 1. That is a consequence of the inflated bonus needed by the use of the local randomizer algorithm $\mathcal{M}_{\text{LDP}}$. Let's consider the batched algorithm described in Alg. 4.

In terms of privacy Alg. 4 enjoys the same guarantees as Alg. 1. For any $\delta \in (0,1)$, we have that with probability at least $1-\delta$:

$$R_T = \sum_{t=1}^{T}\langle\theta^\star, x_{t,a_t^\star} - x_{t,a_t}\rangle = \sum_{j=1}^{M}\sum_{t=t_j+1}^{t_{j+1}}\langle\theta^\star, x_{t,a_t^\star} - x_{t,a_t}\rangle \tag{44}$$

Algorithm 4: FixedBatchedShuffling-LinUCB

Input: LDP parameter: ε_0, privacy parameter: ε, δ', regularization parameter:λ, context bound:L, failure probability:δ, low switching parameter: η, encoding parameter:m, dimension: d

Initialize $j^S = j^A = 0$, $\widetilde{\theta}_0 = 0$, $\widetilde{V}_0 = \lambda I_d$, $p = \frac{2}{\exp(2\varepsilon_0/(md(d+3)))+1}$

for $t = 0, 1, \ldots$ **do**

User receives $\widetilde{\theta}_{j^A}$, $\widetilde{V}_{j^A}$ and β_{j^A} and selects $a_t \in \operatorname{argmax}_{a\in[K]} \langle x_{t,a}, \widetilde{\theta}_{j^A}\rangle + \beta_{j^A} \| x_{t,a}\|_{\widetilde{V}_{j^A}^{-1}}$

Observe reward r_t and compute private statistics $(\widetilde{b}_t, \widetilde{w}_t) = \mathcal{M}_{\mathrm{LDP}}((x_{t,a_t}, r_t), p, m, L)$

Communication with the shuffler

$B^S_{j^S} = B^S_{j^S} \cup (\widetilde{b}_t, \widetilde{w}_t)$

if $|B^S_{j^S}| = l$ **then**

Set $t_{j^S+1} = t$, compute a permutation σ of $[\![t_{j^S}+1, t_{j^S+1}]\!]$ and compute aggregate statistics

$$\forall i \le d, k \le i, \qquad Z_{j^S,i} = \sum_{n=1}^{l}\sum_{q=1}^{m} \tilde{b}_{\sigma(n),i,q} \quad \text{and} \quad U_{j^S,i,k} = \sum_{n=1}^{l}\sum_{q=1}^{m} \widetilde{w}_{\sigma(n),i,k,q}$$

Set $U_{j^S,i,k} = U_{j^S,k,i}$, $B_{j^S+1} = \emptyset$ and $j^S = j^S + 1$

Communication with the bandit algorithm

Receives (Z_{j^S-1}, U_{j^S-1}) and compute candidate statistics

$$\widetilde{B}_{j^A+1} = \widetilde{B}_{j^A+1} + \frac{Z_{j^S-1}}{m(1-p)} - \frac{l^S}{2(1-p)}$$
$$\widetilde{V}_{j^A+1} = \widetilde{V}_{j^A+1} + \frac{U_{j^S-1}}{m(1-p)} - \frac{l^S}{2(1-p)} + 2(\lambda_{j^A+1} - \lambda_{j^A})I_d$$

Compute $\theta_{j^A+1} = \frac{1}{L}\widetilde{V}^{-1}_{j^A+1}\widetilde{B}_{j^A+1}$

Set $t_{j^A+1} = t$, β_{j^A+1} and λ_{j^A+1} as in Eq. (7) and Eq. (8)

Set $j^A = j^A + 1$, $\widetilde{B}_{j^A+1} = \widetilde{B}_{j^A}$ and $\widetilde{V}_{j^A+1} = \widetilde{V}_{j^A}$

end

end

But using Lem.3 in (Han et al., 2020), we have that for any batch j:

$$\begin{aligned}\sum_{j=1}^{M}\sum_{t=t_j+1}^{t_{j+1}}\|x_{t,a_t}\|_{\widetilde{V}_j^{-1}} &\leq \sqrt{\frac{T}{M}}\sum_{j=1}^{M}\sqrt{\mathrm{Tr}\left(\widetilde{V}_j^{-1}\sum_{t=t_j+1}^{t_{j+1}}x_{t,a_t}x_{t,a_t}^\top\right)} \\ &\leq \sqrt{\frac{10T}{M}}\log(T+1)\left(\sqrt{Md}+d\sqrt{\frac{T}{M}}\right)\end{aligned} \tag{45}$$

where M is the total number of batch. Therefore, the regret is bounded with high probability by:

$$R_T \leq 2\beta_M\sqrt{\frac{10T}{M}}\log(T+1)\left(\sqrt{Md}+d\sqrt{\frac{T}{M}}\right) = \mathcal{O}\left(\frac{T^{3/4}}{\sqrt{1-p}}+\frac{T^{3/4}\sqrt{1-p}}{\varepsilon}\right) \tag{46}$$

Where we used the fact that M is defined in Eq. (39). Therefore, using a fixed schedule algorithm the trade-off highlighted in Thm. 11 does not appear.