Bridging The Gap between Local and Joint Differential Privacy in RL

Evrard Garcelon [1 2] **Vianney Perchet** [2] **Ciara Pike-Burke** [3] **Matteo Pirotta** [1]

Abstract

[1] In this paper, we study privacy in the context of finite-horizon Markov Decision Processes. Two notions of privacy have been investigated in this setting: joint differential privacy (JDP) and local differential privacy (LDP). We show that it is possible to achieve a smooth transition in terms of privacy and regret (i.e., utility) between JDP and LDP. By leveraging shuffling techniques, we present an algorithm that, depending on the provided parameter, is able to attain any privacy/utility value in between the pure JDP and LDP guarantee.

1. Introduction

The practical successes of Reinforcement Learning (RL) algorithms have led to them becoming ubiquitous in many settings such as digital marketing, healthcare and finance, where it is desirable to provide a personalized service (e.g. Mao et al., 2020; Wang & Yu, 2021). However, users are becoming increasingly wary of the amount of personal information that these services require. This is particularly pertinent in many of the aforementioned domains where the data obtained by the RL algorithm are highly sensitive. For example, in healthcare, the state encodes personal information such as gender, age, vital signs, etc. In advertising, it is normal for states to include browser history, geolocalized information, etc. In response, the literature has started investigated differential privacy (DP) (e.g., Dwork et al., 2010; Dwork & Roth, 2014) guarantees both in bandits (e.g., Tossou & Dimitrakakis, 2015; Shariff & Sheffet, 2018; Zheng et al., 2020) and in RL (e.g., Vietri et al., 2020; Garcelon et al., 2021).

The RL literature has studied joint DP (Vietri et al., 2020) and local DP (Garcelon et al., 2021) in the context of tabular finite-horizon Markov Decision Processes (MDPs). Informally, JDP requires the algorithm to not expose sensitive information through its decisions, i.e., an observer should not be able to infer sensitive information by observing the output of the algorithm. On the other hand, LDP prevents the algorithm to observe sensitive information by requiring data to be privatized before being sent to the algorithm. These definitions have different impact on the regret (i.e., utility) of an algorithm. While ε-JDP guarantees can be obtained by paying only an additional logarithmic term in the regret ($\Omega(\sqrt{K} + \log(K)/\varepsilon)$ in (Vietri et al., 2020)), LDP comes with stronger requirements and higher impact on the regret. In fact, Garcelon et al. (2021) have shown that ε-LDP has a multiplicative impact ($\Omega(\sqrt{K}/\varepsilon)$). While LDP poses stronger requirements, it is currently unclear whether it is possible to achieve some form of privacy/utility trade-off between these two models.

In this paper, we address this question by leveraging the shuffling 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). In particular, we show that it is possible to achieve a smooth transition between JPD and LDP guarantees in RL via shuffling, always preserving a minimal LDP level. Although, it is not to get strict JDP but only approximated JDP. At the end of the spectrum, we nearly recover the results in (Vietri et al., 2020) and (Garcelon et al., 2021).[2] This provides a comprehensive understanding of the connections between the privacy models used in RL.

2. Preliminaries

We consider a finite-horizon Markov Decision Process (MDP) (Puterman, 1994, Chp. 4) $M = (\mathcal{S}, \mathcal{A}, p, r, H)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, and horizon $H \in \mathbb{N}^+$. Every state-action pair is characterized by a reward distribution with mean $r(s,a)$ supported in $[0,1]$ and a transition distribution $p(\cdot|s,a)$ over next state.[3] We denote by $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$ the number of states and actions. A deterministic

[1]Facebook AI Research [2]CREST, ENSAE [3]Imperial College London. Correspondence to: Evrard Garcelon <evrard@fb.com>.

ICML 2021 Workshop on Reinforcement Learning Theory.

[1]Extended abstract. Full version to appear.

[2]First, in this extended abstract we put the focus on the dependence in the length of interaction K and privacy level ε. We ignore S, A and H factors in the discussion since we believe the analysis can be improved. Second, we do not recover the $\log(K)/\varepsilon$ dependence in (Vietri et al., 2020) for JDP but a $K^{1/3}/\varepsilon$ term that we believe can be improved by a more careful analysis.

[3]We can modify the algorithm to handle step dependent transitions and rewards. The regret is then multiplied by a factor $H\sqrt{H}$.

policy is defined as a collection $\pi = (\pi_1, \dots, \pi_H) \in \Pi$ of policies $\pi_h : \mathcal{S} \to \mathcal{A}$. For any $h \in [H] := \{1, \dots, H\}$ and state $s \in \mathcal{S}$, the value functions of a policy π are defined as $Q_h^\pi(s,a) = r(s,a) + \mathbb{E}_\pi \left[\sum_{i=h+1}^H r(s_i, a_i)\right]$ and $V_h^\pi(s) = Q^\pi(s, \pi_h(s))$. There exists an optimal deterministic policy $\pi^\star \in \Pi$ (Puterman, 1994, Sec. 4.4) such that $V_h^\star(s) = V_h^{\pi^\star}(s) = \max_\pi V_h^\pi(s)$. The Bellman equations at stage $h \in [H]$ are defined as $Q_h^\star(s,a) = r_h(s,a) + \max_{a'} \mathbb{E}_{s' \sim p_h(s,a')} \left[V_{h+1}^\star(s')\right]$. The optimal policy is simply the greedy policy: $\pi_h^\star(s) = \operatorname{argmax}_a Q_h^\star(s,a)$.

We consider the standard *learning protocol* for finite horizon. The learning agent (e.g., a personalization service) interacts with an unknown MDP in a sequence of episodes $k \in [K]$ of fixed length H. We consider each episode as the interaction with a different user $u_k \in \mathcal{U}$, $u_j \neq u_i, \forall i, j$. Following (Vietri et al., 2020), a user u is characterized by a starting state distribution $\rho_{0,u}$ (i.e., for user u, $s_1 \sim \rho_{0,u}$) and a tree of depth H, describing all the possible sequence of states and rewards corresponding to all possible sequences of actions. For each episode $k \in [K]$, let $s_{1,k} \sim \rho_{0,u_k}$ be the initial state for user u_k. The learner selects a policy π_k that is sent to the user u_k for execution. The outcome of the execution, i.e., a trajectory, $X_k = (s_{kh}, a_{kh}, r_{kh})_{h \in [H]}$ is sent to the learner to update the policy. We evaluate the performance of a learning algorithm $\mathfrak{A}$ which plays policies $\pi_1, \dots, \pi_K$ by its cumulative regret after K episodes

$$\Delta(K) = \sum_{k=1}^K (V_1^\star(s_{1,k}) - V_1^{\pi_k}(s_{1,k})). \tag{1}$$

2.1. Differential Privacy in RL

Exploration with privacy guarantees has only been recently studied in RL (Vietri et al., 2020; Garcelon et al., 2021). Vietri et al. (2020) propose a regret minimization algorithm able to guarantee *joint differential privacy* (JDP). Intuitively, JDP requires that when a user changes, the actions computed by the algorithm for the other $K-1$ users stay the same, hence the other users can not infer the sequence of states, actions and rewards observed by the changed user. The algorithm has access to all the information about the users (i.e., trajectories) containing sensitive data. This approach to privacy requires the user to trust the RL algorithm to privately handle the data and not to expose or share sensitive information to an external user observing its behavior. Formally, JDP is defined as:

Definition 1. *For $\varepsilon > 0$ and $\delta_0 > 0$, a randomized RL agent $\mathfrak{A}$ is (ε, δ_0)-*joint differentially *private if for every $k \in \{1, \dots, K\}$, two sequences of users, $U = \{u_1, \dots, u_K\}$ and $U' = \{u'_1, \dots, u'_K\}$, that differs only for the k-th user and for all events $E \subset \mathcal{A}^{H \times [K-1]}$ then:*

$$\mathbb{P}(\mathfrak{A}_{-k}(U) \in E) \leq e^\varepsilon \mathbb{P}(\mathfrak{A}_{-k}(U) \in E) + \delta_0 \tag{2}$$

where $\mathfrak{A}_{-k}(U)$ denotes all the outputs of algorithm $\mathfrak{A}$, i.e., all actions $(a_{i,h})_{i \neq k, h \leq H}$ excluding the output of episode k for the sequence of users U.

Garcelon et al. (2021) studied the stronger *local differential privacy* (LDP) notion and proposed a model-based exploration algorithm called LDP-OBI. Opposite to JDP, LDP prevents the RL algorithm from observing the true sensitive data. Indeed, LDP requires that an algorithm has access to user information (i.e., trajectories) only through samples that have been privatized before being passed to the learning agent. The appeal of this local model is that *privatization can be done locally on the user-side* using a private randomizer $\mathcal{M}$. Since nobody other than the user has ever access to any piece of non private data, this local setting is far more private. Formally, we write $\mathcal{M}(X_{u_k})$ to denote the privatized data generated by the randomizer from a trajectory X_{u_k}. The goal of mechanism $\mathcal{M}$ is to privatize sensitive information while encoding sufficient knowledge for learning. With these notions in mind, LDP in RL can be defined as follows:

Definition 2. *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 $u, u' \in \mathcal{U}$, trajectories $(X_u, X_{u'}) \in \mathcal{X}_u \times \mathcal{X}_{u'}$ and all $O \subset \{\mathcal{M}(\mathcal{X}_u) \mid u \in \mathcal{U}\}$:*

$$\mathbb{P}\left(\mathcal{M}(X_u) \in O\right) \leq e^\varepsilon \mathbb{P}\left(\mathcal{M}(X_{u'}) \in O\right) + \delta \tag{3}$$

where $\mathcal{X}_u$ is the space of trajectories associated to user u.

We end this section with the definition of differential privacy.

Definition 3. *For any $\varepsilon \geq 0$ and $\delta \geq 0$, a privacy preserving mechanism $\mathcal{M}$ is said to be (ε, δ)-*differential *private if and only if there exists $n \in \mathbb{N}$, for all inputs of $\mathcal{M}$, $(x_1, \dots, x_n)$ and $(x'_1, \dots, x'_n)$ there exists $i^\star \leq n$ with $x_{i^\star} \neq x'_{i^\star}$ for all $j \neq i^\star$, $x_j = x'_j$ and for all events O:*

$$\mathbb{P}\left(\mathcal{M}(x_1, \dots, x_n) \in O\right) \leq e^\varepsilon \mathbb{P}\left(\mathcal{M}(x'_1, \dots, x'_n) \in O\right) + \delta$$

where $\mathcal{X}_u$ is the space of trajectories associated to user u.

3. Shuffling in RL

The shuffling 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) has attracted a lot of interest in machine learning, because it allows to build (ε, δ)-DP algorithm with an additional LDP guarantees. The most attractive feature of this privacy model is that it offers a smooth transition in terms of privacy/utility trade-off between the stringent LDP requirements and the differential privacy requirements (see Feldman et al., 2020, for an example of this transition in the problem of estimating a discrete distribution).

The interaction protocol in the shuffle model is as follows. Similarly to LDP, the user u computes a private version

Algorithm 1 Shuffling Protocol

Input: number of episodes K, horizon H, failure probability $\delta \in (0,1)$, bias $\alpha > 1$, private randomizer $\mathcal{M}_{\text{sh}}$ with LDP parameters (ϵ_0, δ_0), RL algorithm $\mathfrak{A}$
for $k = 1$ **to** K **do**
  Shuffler $\mathcal{R}$ sends $(\mathcal{M}_{\text{sh}}(X_{u_{\sigma_k(l)}}))_{l<k}$ (σ_k a random permutation)
  $\mathfrak{A}$ computes policy π_k based on $(\mathcal{M}_{\text{sh}}(X_{u_{\sigma_k(l)}}))_{l<k}$
  User u_k executes policy π_k in the environment, collects trajectory $X_k = \{(s_{k,h}, a_{k,h}, r_{k,h})_{h\le H}\}$ and sends the privatized trajectory $\mathcal{M}_{\text{sh}}(X_k)$ to $\mathcal{R}$
end for

of their trajectory X_u by means of a private randomizer $\mathcal{M}_{\text{sh}}$. This private information $\mathcal{M}_{\text{sh}}(X_u)$ is passed to a shuffler $\mathcal{R}$. At each episode $k \in [K]$, the shuffler $\mathcal{R}$ has thus access to all private information $(\mathcal{M}_{\text{sh}}(X_{u_l}))_{l<k}$ up to episode k. It computes a random permutation σ_k of $[k-1]$ and sends the permuted set of privatized statistics, $(\mathcal{M}_{\text{sh}}(X_{u_{\sigma_k(l)}}))_{l<k}$ to a regret minimizing algorithm. The protocol is detailed in Alg. 1. *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 through subsampling and, overall, it improves the standard LDP protocol with virtually no cost.

3.1. Privacy-preserving mechanism $\mathcal{M}_{\text{sh}}$

Similarly to (Garcelon et al., 2021), we take a *model-based approach* to the privacy problem. The output of the randomizer is thus any succinct information that can be used by the algorithm to build an estimate of the MDP. We thus construct private estimates of the number of visits to (s,a) and (s,a,s'), and of the cumulative reward. We do not use the same approach as in (Garcelon et al., 2021) but we adapt the algorithm for bit-sum protocol by Cheu et al. (2019) to MDPs. The first step of the process $\mathcal{M}_{\text{sh}}$ is to apply a one-hot encoding for each state-action of the trajectory. That is to say, let $x \in \{0,1\}^{H\times S\times A}$ and $y \in \{0,1\}^{(H-1)\times S\times A\times S}$ such that for each (s,a,s') and h

$$x_{h,s,a} = \mathbb{1}_{\left\{\substack{s_h=s,\\ a_h=a}\right\}}, \text{ and } y_{h,s,a,s'} = \mathbb{1}_{\left\{\substack{s_h=s,\\ a_h=a,\\ s_{h+1}=s'}\right\}} \tag{4}$$

To encode rewards, we first compute the reward for each state-action pair, $\left(r_h \mathbb{1}_{\{s_h=s,a_h=a\}}\right)_{(h,s,a)\in[\![1,H]\!]\times\mathcal{S}\times\mathcal{A}}$ then given a parameter $m \in \mathbb{N}^\star$ for each state-action pair (s,a) we compute $b_{h,s,a} \in \{0,1\}^m$ such that for $j \in [\![1,m]\!]$:

$$(b_{h,s,a})_j = \begin{cases} 1 & \text{if } j < \mu_{h,s,a} \\ \text{Ber}\,(p_{h,s,a}) & \text{if } j = \mu_{h,s,a} \\ 0 & \text{if } j > \mu_{h,s,a} \end{cases} \tag{5}$$

Algorithm 2 Local randomizer $R_p^{0/1}$

Input: randomization probability: $p \in [0,1]$, $x \in \{0,1\}$
Sample $b \sim \text{Ber}(p)$
If $b = 0$ **then** return x **else** return $\text{Ber}(1/2)$

Algorithm 3 Privacy-preserving mechanism $\mathcal{M}_{\text{sh}}$

Input: trajectory $\tau = \{(s_h, a_h, r_h)_{h\le H}\}$, privacy parameter $\varepsilon > 0$, parameter $m \in \mathbb{N}^\star$
Compute x, y and $(b_{h,s,a})_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ as in Eqs. (4), (5)
Return $(R_p^{0/1}(x_{h,s,a}))_{(h,s,a)}$, $(R_p^{0/1}(y_{h,s,a,s'}))_{(h,s,a,s')}$ and $((R_p^{0/1}((b_{h,s,a})_j)_{j\le m})_{(h,s,a)}$ with $p = 2/(e^\varepsilon + 1)$

with $\mu_{h,s,a} = \lceil m\, r_h\, \mathbb{1}_{\{s_h=s,a_h=a\}} \rceil$ and $p_{h,s,a} = m\, r_h\, \mathbb{1}_{\{s_h=s,a_h=a\}} - \mu_{h,s,a} + 1$.

Using those encodings, Alg. 2 with parameter p guarantees $\ln(2/p - 1)$ local differential privacy. The final privacy-preserving mechanism $\mathcal{M}_{\text{sh}}$ is described by Alg. 3.

Lemma 4. *For any $\varepsilon_0 > 0$ and trajectory X we have that $\mathcal{M}_{sh}$ with parameters $\varepsilon = \frac{\varepsilon_0}{(4+2m)H}$ satisfies ε_0-LDP.*

Here, we use Theorem 4.1 in (Cheu et al., 2019), we consider a shuffle model where all users use the same level of privacy when using the mechanism $\mathcal{M}_{\text{sh}}$. However, for the first users the amplification of the privacy level by shuffling is not noticeable. Next, we state the privacy of the resulting estimator with shuffling:

Proposition 5. *For any $\delta_0 > 0$, any users $(u_l)_{l\le k}$ with trajectories $(X_{u_l})_{l\le k}$ using the mechanism $\mathcal{M}_{sh}$ with privacy parameter $0 < \varepsilon_l \le \log(k/(7\log(4/\delta_0)) - 1)$ and $m \ge 1$, for σ a permutation of $\{1,\dots,k\}$ chosen uniformly at random then $\left(\sum_{i\le k, h\le H} R_p^{0/1}(x_{i,h,s,a})\right)_{s,a}$, $\left(\sum_{i\le k, h\le H} R_p^{0/1}(y_{i,h,s,a,s'})\right)_{s,a,s'}$ and $\left(\sum_{i\le k, h\le H, j\le m} R_p^{0/1}((b_{i,h,s,a})_j\right)_{s,a}$ is $(\varepsilon_{c,k}, \delta)$-DP with*

$$\varepsilon_{c,k} = \frac{256\log\left(\frac{8m}{\delta_0}\right)\sqrt{m\log\left(\frac{2}{\delta_0}\right)}}{\sqrt{(k-1)H\left(p-\sqrt{\frac{2p\log\left(\frac{4m}{\delta_0}\right)}{(k-1)H}}\right)}}\left(1-p+\sqrt{\frac{2p\log\left(\frac{4m}{\delta_0}\right)}{(k-1)H}}\right)$$
$$+\frac{64\log\left(\frac{4}{\delta_0}\right)}{\sqrt{(k-1)H\left(p-\sqrt{\frac{2p\log\left(\frac{2}{\delta_0}\right)}{(k-1)H}}\right)}}\left(1-p+\sqrt{\frac{2p\log\left(\frac{2}{\delta_0}\right)}{(k-1)H}}\right)$$

and for each $l \le k$, $\mathcal{M}_{sh}(X_{u_l})$ is ε_l-LDP.

3.2. Utility Analysis

Upon receiving the shuffled privatized, the algorithm SHUFFLED-OBI computes the different counts $(\widetilde{N}_k^p(s,a,s'))_{(s,a,s')}$, $(\widetilde{N}_k^r(s,a))_{(s,a)}$ and $(\widetilde{R}_k(s,a))_{(s,a)}$.

For any $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, we define counters as:

$$\begin{aligned}
\tilde{N}_k^r(s,a) &= \frac{1}{1-p}\sum_{l=1}^{k-1}\sum_{h=1}^{H} R_p^{0/1}(x_{l,h,s,a}) - \frac{p}{2} \\
\tilde{N}_k^p(s,a,s') &= \frac{1}{1-p}\sum_{l=1}^{k-1}\sum_{h=1}^{H} R_p^{0/1}(y_{l,h,s,a,s'}) - \frac{p}{2} \\
\tilde{R}_k^r(s,a) &= \frac{1}{m(1-p)}\sum_{j=1}^{m}\sum_{l=1}^{k-1}\sum_{h=1}^{H} R_p^{0/1}((b_{l,h,s,a})_j) - \frac{p}{2}
\end{aligned} \tag{6}$$

Thanks to Hoeffding concentration inequality, for every state-action-next state $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}'$ and for any $\delta > 0$, with probability at least $1-\delta$:

$$\begin{aligned}
\left|\sum_{l=1}^{k-1}\sum_{h=1}^{H} x_{l,h,s,a} - \tilde{N}_k^r(s,a)\right| &\le \frac{2\ln(1/\delta)}{3(1-p)} \\
&\quad + \frac{\sqrt{(k-1)Hp(1-\frac{p}{2})\ln\left(\frac{1}{\delta}\right)}}{1-p} \\
\left|\sum_{l=1}^{k-1}\sum_{h=1}^{H-1} y_{l,h,s,a,s'} - \tilde{N}_k^p(s,a,s')\right| &\le \frac{2\ln(1/\delta)}{3(1-p)} \\
&\quad + \frac{\sqrt{(k-1)Hp\left(1-\frac{p}{2}\right)\ln(1/\delta)}}{1-p}
\end{aligned} \tag{7}$$

Using Claim C.4 in (Cheu et al., 2019), we have that with probability at least $1-\delta$ that:

$$\begin{aligned}
\left|\sum_{l=1}^{k-1}\sum_{h=1}^{H} r_{l,h}\mathbb{1}_{\left\{\substack{s_{l,h}=s,\\ a_{l,h}=a}\right\}} - \tilde{R}_k^r(s,a)\right| &\le \frac{\sqrt{2Hk\log(\frac{2}{\delta})}}{m} \\
&\quad + \frac{\sqrt{kHmp(1-\frac{p}{2}))\ln(1/\delta)}}{m(1-p)} + \frac{2\ln(1/\delta)}{3(1-p)}
\end{aligned} \tag{8}$$

4. Trade-off between Privacy and Regret

The results introduced in the previous section can be used to build a regret minimizing algorithm which interpolates between LDP and JDP.

Our algorithm, called Shuffled-OBI (see Alg. 4), requires a burn-in phase of length τ to ensure some level of privacy amplification for the first users. Indeed, at the beginning, the privacy amplification properties of the shuffling mechanism are not strong enough to guarantee any privacy. For the first τ users the algorithm chooses a policy at random for those users, ensuring strong privacy. Then, for episode $k > \tau$, the algorithm computes an optimistic estimate by leveraging the private counters to build a model of the MDP. Compared to (Garcelon et al., 2021), our algorithm recomputes the estimate of the model at each episode based on shuffled data, instead of updating it incrementally. *This requires that the algorithm discards the computed estimates at the end of each episode.* Due to the similarities with LDP-OBI, the regret bound for SHUFFLED-OBI can be seen as a corollary of Thm. 2 in (Garcelon et al., 2021), by using the functions $\{c_{k,i}\}_{i\le 4}$ defined in Eq. (7) and (8).[4] This immediately give us Thm. 6.

Algorithm 4 SHUFFLED-OBI

Input: number of episodes K, horizon H, failure probability $\delta \in (0,1)$, bias $\alpha > 1$, private randomizer $\mathcal{M}_{\text{sh}}$ with LDP parameters (ϵ_0, δ_0), burn-in parameter: τ
for $k = 1$ **to** τ **do**
  Select policy π_k uniformly at random
end for
for $k = 1$ **to** K **do**
  Compute $\widetilde{p}_k$ and $\widetilde{r}_k$ as in Eq. (16) using $\{\mathcal{M}(X_{u_{\sigma_k(l)}})\}_{l\in[K-1]}$, β_k^r and β_k^p as in Prop. 7 using $\{c_{k,i}(\varepsilon_0, \delta_0, \frac{3\delta}{2k^2\pi^2})\}_i$, and $b_{h,k}$
  Compute π_k as in Eq. (17) and send it to user u_k
  User u_k executes policy π_k, collects trajectory X_k and sends back privatized value $\mathcal{M}(X_k)$
end for

Theorem 6. *For any users $K \in \mathbb{N}^\star$, $\delta > 0$ using the mechanism $\mathcal{M}_{sh}$ with parameter $\varepsilon > 0$ and $m = 1$ using Alg. 4, with probability at least $1-\delta$:*

$$\Delta(K) \le \mathcal{O}\left(\sqrt{K} + \tau H + \frac{\sqrt{2K}e^{\varepsilon/2}}{e^\varepsilon - 1}\right) \tag{9}$$

In addition, Alg. 4 is ε-LDP and $(\varepsilon_{c,\tau}, \delta_0)$-JDP for any $\delta_0 > 0$ and $\varepsilon_{c,\tau}$ defined in Prop. 5.

In Thm. 6, τ is the minimal number of users needed to ensure a level of privacy with $\varepsilon_{c,\tau}$ (see Prop. 5). Thm. 6 also shows that, for a fixed $\varepsilon > 0$, the regret of Alg. 4 is bounded by $\mathcal{O}\left(\frac{\sqrt{K}}{e^\varepsilon - 1}\right)$ which is the optimal LDP rate shown by Garcelon et al. (2021). However, under the constraint that Alg. 4 only needs to be $(\varepsilon_0, \delta_0)$-JDP for a certain $\varepsilon_0 > 0$ and $\delta_0 > 0$, it is enough to choose $\varepsilon = \mathcal{O}\left(\ln(1+\varepsilon_0\sqrt{\tau})\right)$ in Thm. 6, and the regret of Alg. 4 is bounded by $\mathcal{O}\left(\sqrt{K} + \frac{K^{1/3}}{\varepsilon_0}\right)$ for $\tau = \mathcal{O}\left(K^{1/3}\right)$. Vietri et al. (2020) showed that ϵ-JDP can be achieved by paying only an additional $O(\log(K)/\varepsilon)$ cost in the regret. As a consequence, our result is slightly suboptimal as it has a $O(K^{1/3}/\varepsilon)$. This is just an artifact of our analysis where we require at least a privacy level ε_0 in the initial phase. One way to resolve this issue would be to use an adaptive privacy-preserving mechanism for shuffling as the one presented in (Feldman et al., 2020).

[4] For ease of presentation, in this extended abstract we remove dependencies on S, A and H. Refer to the full version of the paper for a complete presentation.

Algorithm	τ	LDP	JDP	Regret
SHUFFLED-OBI (our)	$O(K^{1/3})$	$(O(\ln(1+\varepsilon_0 K^{1/6})), 0)$	$(\varepsilon_0, \delta_0)$	$\widetilde{O}(\sqrt{K} + \frac{K^{1/3}}{\varepsilon_0})$
	τ	$(\varepsilon_0, 0)$	$(\varepsilon_{c,\tau}, \delta_0)$	$\widetilde{O}\left(\sqrt{K} + \tau H + \frac{\sqrt{2K}e^{\varepsilon_0/2}}{e^{\varepsilon_0}-1}\right)$
	0	$(\varepsilon_0, 0)$	$(\varepsilon_0, \delta_0)$	$\widetilde{O}(\frac{\sqrt{K}}{e^{\varepsilon_0}-1})$
PUCB (Vietri et al., 2020)		N/A	$(\varepsilon_0, 0)$	$\widetilde{O}(\sqrt{K} + \frac{1}{\varepsilon_0})$
LDP-OBI (with Laplace Mechanism) (Garcelon et al., 2021)		$(\varepsilon_0, 0)$	$(\varepsilon_0, 0)$	$\widetilde{O}(\frac{\sqrt{K}}{\varepsilon_0})$

Table 1. Regret and Privacy guarantee for different algorithms. SHUFFLED-OBI interpolates between JDP and LDP. For $\tau = 0$, we retrieve $(\varepsilon_0, 0)$-LDP guarantee and the regret of (Garcelon et al., 2021). While for $\tau = \mathcal{O}(\ln(1+\varepsilon_0 K^{1/6}))$, SHUFFLED-OBI is $(\varepsilon_0, \delta_0)$-JDP.

5. Conclusion

In this paper, we showed it is possible to design an algorithm that, based on the input parameters, offers a trade-off between JDP and LDP. While we recover the optimal $O(\sqrt{K}/(e^{\varepsilon}-1))$ rate in K and ε for LDP, we are slightly suboptimal for JDP due to a shortcoming of the analysis (i.e., we obtain $O(\sqrt{K} + \frac{K^{1/3}}{\varepsilon})$ while the optimal rate is $O(\sqrt{K} + \frac{\log(K)}{\varepsilon})$). We think it is possible to improve this result by leveraging adaptive privacy for shuffling as done in (Feldman et al., 2020).

References

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 263–272. PMLR, 2017.

Balle, B., Bell, J., Gascón, A., and Nissim, K. The privacy blanket of the shuffle model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pp. 638–667. Springer, 2019.

Chen, L., Ghazi, B., Kumar, R., and Manurangsi, P. On distributed differential privacy and counting distinct elements. In *ITCS*, volume 185 of *LIPIcs*, pp. 56:1–56:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

Cheu, A., Smith, A., Ullman, J., Zeber, D., and Zhilyaev, M. Distributed differential privacy via shuffling. *Lecture Notes in Computer Science*, pp. 375–403, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-17653-2_13.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 715–724, 2010.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Song, S., Talwar, K., and Thakurta, A. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618, 2020.

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

Garcelon, E., Perchet, V., Pike-Burke, C., and Pirotta, M. Local differentially private regret minimization in reinforcement learning, 2021.

Mao, H., Chen, S., Dimmery, D., Singh, S., Blaisdell, D., Tian, Y., Alizadeh, M., and Bakshy, E. Real-world video adaptation with reinforcement learning, 2020.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994. ISBN 0471619779.

Qian, J., Fruit, R., Pirotta, M., and Lazaric, A. Exploration bonus for regret minimization in discrete and continuous average reward mdps. In *NeurIPS*, pp. 4891–4900, 2019.

Shariff, R. and Sheffet, O. Differentially private contextual linear bandits. In *NeurIPS*, pp. 4301–4311, 2018.

Tossou, A. and Dimitrakakis, C. Differentially private multi-agent multi-armed bandits. In *European Workshop on Reinforcement Learning (EWRL-15)*, 2015.

Vietri, G., de Balle Pigem, B., Krishnamurthy, A., and Wu, S. Private reinforcement learning with pac and regret guarantees. In *ICML*, 2020.

Wang, H. and Yu, S. Robo-advising: Enhancing investment with inverse optimization and deep reinforcement learning, 2021.

Zheng, K., Cai, T., Huang, W., Li, Z., and Wang, L. Locally differentially private (contextual) bandits learning. *CoRR*, abs/2006.00701, 2020.

A. Proof of Lem. 4

Thanks to group privacy it suffices to show that the mechanism $\mathcal{M}_{\text{sh}}$ is ε-DP for two trajectories $x = (s_h, a_h, r_h)_{h\leq H}$ and $x' = \{(s'_h, a'_h, r'_h)_{h\leq H}\}$ such that there exists a unique $i \in [\![1, H]\!]$ such that, for all $j \neq i$, $s_j = s'_j$, $a_j = a'_j$ and $r_j = r'_j$ but $(s_i, a_i, r_i) \neq (s'_i, a'_i, r'_i)$. Now we show the differential result for each component of $\mathcal{M}_{\text{sh}}$:

1. Because x and x' only differ in one element, we have that for a $q \in \{0,1\}^{HSA}$:

$$\frac{\mathbb{P}\left((R_p^{0/1}(x_{h,s,a}))_{(h,s,a)} = (q_{h,s,a})_{(h,s,a)}\right)}{\mathbb{P}\left((R_p^{0/1}(x'_{h,s,a}))_{(h,s,a)} = (q_{h,s,a})_{(h,s,a)}\right)} = \prod_{s,a} \frac{\mathbb{P}\left(R_p^{0/1}(x_{i,s,a}) = q_{i,s,a}\right)}{\mathbb{P}\left(R_p^{0/1}(x'_{i,s,a}) = q_{i,s,a}\right)} \tag{10}$$

$$= \prod_{s,a} \frac{p/2 + (1-p)\mathbb{1}_{\{q_{i,s,a}=x_{i,s,a}\}}}{p/2 + (1-p)\mathbb{1}_{\{q_{i,s,a}=x'_{i,s,a}\}}} \tag{11}$$

$$= \prod_{(s,a)\in\{(s_i,a_i),(s'_i,a'_i)\})} \frac{p/2 + (1-p)\mathbb{1}_{\{q_{i,s,a}=x_{i,s,a}\}}}{p/2 + (1-p)\mathbb{1}_{\{q_{i,s,a}=x'_{i,s,a}\}}} \tag{12}$$

$$\leq \exp(2\varepsilon) \tag{13}$$

   using the definition of p and the fact that $x \in \{0,1\}^{HSA}$.

2. With the same reasoning as above we have that:

$$\frac{\mathbb{P}\left((R_p^{0/1}(y_{h,s,a,s'}))_{(h,s,a,s')} = (q_{h,s,a,s'})_{(h,s,a,s')}\right)}{\mathbb{P}\left((R_p^{0/1}(y'_{h,s,a,s'}))_{(h,s,a,s')} = (q_{h,s,a,s'})_{(h,s,a,s')}\right)} \leq \exp(2\varepsilon) \tag{14}$$

3. Finally for the reward we have that:

$$\frac{\mathbb{P}\left(((R_p^{0/1}((b_{h,s,a})_j)_{j\leq m})_{(h,s,a)} = (q_{j,h,s,a})_{(j,h,s,a)}\right)}{\mathbb{P}\left(((R_p^{0/1}((b'_{h,s,a})_j)_{j\leq m})_{(h,s,a)} = (q_{j,h,s,a})_{(j,h,s,a)}\right)} = \prod_{j,s,a} \frac{\mathbb{P}\left(R_p^{0/1}((b_{i,s,a})_j = q_{j,i,s,a}\right)}{\mathbb{P}\left(R_p^{0/1}(b'_{i,s,a})_j = q_{j,i,s,a}\right)} \quad \leq \exp(2m\varepsilon) \tag{15}$$

B. Confidence Intervals

In this section, we recall how Garcelon et al. (2021) build the confidence intervals around the rewards and the transition probabilities. First, given a parameter $\alpha > 0$, we compute an estimator for each state-action pair as follow:

$$\widetilde{r}_k(s,a) = \frac{\widetilde{R}_k(s,a)}{\widetilde{N}_k^r(s,a) + \alpha c_{k,2}(\varepsilon_0, \delta_0, \delta)}, \qquad \widetilde{p}_k(s' \mid s,a) = \frac{\widetilde{N}_k^p(s,a,s')}{\widetilde{N}_k^p(s,a) + \alpha c_{k,3}(\varepsilon_0, \delta_0, \delta)} \tag{16}$$

with $\widetilde{N}_k^p(s,a) = \sum_{s'} \widetilde{N}^p(s,a,s')$. The next proposition then shows how to build confidence intervals thanks to deviation estimation around $(\widetilde{N}^p(s,a,s'))_{(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}}$, $(\widetilde{N}^r(s,a))_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ and $(\widetilde{R}(s,a))_{(s,a)\in\mathcal{S}\times\mathcal{A}}$.

The private randomizer $\mathcal{M}$ satisfies $(\varepsilon_0, \delta_0)$-LDP, Def. 2, with $\varepsilon_0, \delta_0 \geq 0$. Moreover,

Proposition 7. *For any $\varepsilon_0 > 0$, $\delta_0 \geq 0$, $\delta > 0$, $\alpha > 1$ and episode k, using mechanism $\mathcal{M}$ such that for any $k \geq 0$, there exist four finite strictly positive function, $c_{k,1}(\varepsilon_0, \delta_0, \delta), c_{k,2}(\varepsilon_0, \delta_0, \delta), c_{k,3}(\varepsilon_0, \delta_0, \delta), c_{k,4}(\varepsilon_0, \delta_0, \delta) \in \mathbb{R}_+^\star$ such that with probabilty at least $1-\delta$ for all $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$:*

$$\left|\widetilde{R}_k(s,a) - R_k(s,a)\right| \leq c_{k,1}(\varepsilon_0, \delta_0, \delta), \qquad \left|\widetilde{N}_k^r(s,a) - N_k^r(s,a)\right| \leq c_{k,2}(\varepsilon_0, \delta_0, \delta)$$

$$\left|\sum_{s'} N_k^p(s,a,s') - \widetilde{N}_k^p(s,a,s')\right| \leq c_{k,3}(\varepsilon_0, \delta_0, \delta), \qquad \left|N_k^p(s,a,s') - \widetilde{N}_k^p(s,a,s')\right| \leq c_{k,4}(\varepsilon_0, \delta_0, \delta)$$

with $R_k(s,a) = \sum_{l<k}\sum_{h=1}^{H} r_{l,h}\mathbb{1}_{\{s_{l,h}=s,a_{l,h}=a\}}$, $N_k^r(s,a) = \sum_{l<k}\sum_{h=1}^{H}\mathbb{1}_{\{s_{l,h}=s,a_{l,h}=a\}}$ *and* $N_k^p(s,a,s') = \sum_{h=1}^{H-1}\mathbb{1}_{\{s_{l,h}=s,a_{l,h}=a,s_{l,h+1}=s'\}}$. *Then with probability at least* $1-2\delta$*, for any* $(s,a)\in\mathcal{S}\times\mathcal{A}$

$$|r(s,a)-\widetilde{r}_k(s,a)| \le \beta_k^r(s,a) = \sqrt{\frac{2\ln\left(\frac{4\pi^2 SAHk^3}{3\delta}\right)}{\widetilde{N}_k^r(s,a)+\alpha c_{k,2}(\varepsilon_0,\delta_0,\delta)}} + \frac{(\alpha+1)c_{k,2}(\varepsilon_0,\delta_0,\delta)+c_{k,1}(\varepsilon_0,\delta_0,\delta)}{\widetilde{N}_k^r(s,a)+\alpha c_{k,2}(\varepsilon_0,\delta_0,\delta)}$$

$$\|p(\cdot|s,a)-\widetilde{p}_k(\cdot|s,a)\|_1 \le \beta_k^p(s,a) = \sqrt{\frac{14S\ln\left(\frac{4\pi^2 SAHk^3}{3\delta}\right)}{\widetilde{N}_k^p(s,a)+\alpha c_{k,3}(\varepsilon_0,\delta_0,\delta)}} + \frac{Sc_{k,4}(\varepsilon_0,\delta_0,\delta)}{\widetilde{N}_k^p(s,a)+\alpha c_{k,3}(\varepsilon_0,\delta_0,\delta)} + \frac{(\alpha+1)c_{k,3}(\varepsilon_0,\delta_0,\delta)}{\widetilde{N}_k^p(s,a)+\alpha c_{k,3}(\varepsilon_0,\delta_0,\delta)}$$

As commonly done in the literature (e.g., Azar et al., 2017; Qian et al., 2019), we use these concentration results to define a bonus function $b_{h,k}(s,a) := (H-h+1)\cdot\beta_k^p(s,a)+\beta_k^r(s,a)$ which is used to define an optimistic value function and policy by running the following backward induction procedure:

$$Q_{h,k}(s,a)s = \widetilde{r}_k(s,a)+b_{h,k}(s,a)+\widetilde{p}_k(\cdot|s,a)^{\mathsf{T}}V_{h+1,k}, \quad \pi_{h,k}(s) = \operatorname*{argmax}_a Q_{h,k}(s,a) \tag{17}$$

where $V_{h,k}(s) = \min\{H-h+1, \max_a Q_{h,k}(s,a)\}$ and $V_{H+1,k}(s)=0$.