Private Reinforcement Learning with PAC and Regret Guarantees

Giuseppe Vietri[1] Borja Balle[2] Akshay Krishnamurthy[3] Zhiwei Steven Wu[1]

Abstract

Motivated by high-stakes decision-making domains like personalized medicine where user information is inherently sensitive, we design privacy preserving exploration policies for episodic reinforcement learning (RL). We first provide a meaningful privacy formulation using the notion of joint differential privacy (JDP)–a strong variant of differential privacy for settings where each user receives their own sets of output (e.g., policy recommendations). We then develop a private optimism-based learning algorithm that simultaneously achieves strong PAC and regret bounds, and enjoys a JDP guarantee. Our algorithm only pays for a moderate privacy cost on exploration: in comparison to the non-private bounds, the privacy parameter only appears in lower-order terms. Finally, we present lower bounds on sample complexity and regret for reinforcement learning subject to JDP.

1. Introduction

Privacy-preserving machine learning is critical to the deployment of data-driven solutions in applications involving sensitive data. Differential privacy (DP) (Dwork et al., 2006) is a de-facto standard for designing algorithms with strong privacy guarantees for individual data. Large-scale industrial deployments – e.g. by Apple (Team, 2017), Google (Erlingsson et al., 2014) and the US Census Bureau (Abowd, 2018) – and general purpose DP tools for machine learning (Andrew et al., 2019) and data analysis (Holohan et al., 2019; Wilson et al., 2019) exemplify that existing methods are well-suited for simple data analysis tasks (e.g. averages, histograms, frequent items) and batch learning problems where the training data is available beforehand. While these techniques cover a large number of applications in the central and (non-interactive) local models, they are often insufficient to tackle machine learning applications involving other threat models. This includes federated learning problems (Kairouz et al., 2019; Li et al., 2019) where devices cooperate to learn a joint model while preserving their individual privacy, and, more generally, interactive learning in the spirit of the reinforcement learning (RL) framework (Sutton & Barto, 2018).

In this paper we contribute to the study of reinforcement learning from the lens of differential privacy. We consider sequential decision-making tasks where users interact with an agent for the duration of a fixed-length episode. At each time-step the current user reveals a state to the agent, which responds with an appropriate action and receives a reward generated by the user. Like in standard RL, the goal of the agent is to learn a policy that maximizes the rewards provided by the users. However, our focus is on situations where the states and rewards that users provide to the agent might contain sensitive information. While users might be ready to reveal such information to an agent in order to receive a service, we assume they want to prevent third parties from making unintended inferences about their personal data. This includes external parties who might have access to the policy learned by the agent, as well as malicious users who can probe the agent's behavior to trigger actions informed by its interactions with previous users. For example, Pan et al. (2019) recently showed how RL policies can be probed to reveal information about the environment where the agent was trained.

The question we ask in this paper is: how should the learnings an agent can extract from an episode be balanced against the potential information leakages arising from the behaviors of the agent that are informed by such learnings? We answer the question by making two contributions to the analysis of the privacy-utility trade-off in reinforcement learning: (1) we provide the first privacy-preserving RL algorithm with formal accuracy guarantees, and (2) we provide lower bounds on the regret and number of sub-optimal episodes for any differentially private RL algorithm. To measure the privacy provided by episodic RL algorithms we introduce a notion of episodic joint differential privacy (JDP) under continuous observation, a variant of joint differ-

[*]Equal contribution [1]Department of Computer Science and Engineering, University of Minnesota [2]Now at Deepmind [3]Microsoft Research. Correspondence to: Giuseppe Vietri <vietr002@umn.edu>, Zhiwei Steven Wu <zstevenwu@cmu.edu>, Akshay Krishnamurthy <akshaykr@microsoft.com>, Borja Balle <borja.balle@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

ential privacy (Kearns et al., 2014) that captures the potential information leakages discussed above.

Overview of our results. We study reinforcement learning in a fixed-horizon episodic Markov decision process with S states, A actions, and episodes of length H. We first provide a meaningful privacy formulation for this general learning problem with a strong relaxation of differential privacy: joint differential privacy (JDP) under continual observation, controlled by a privacy parameter $\varepsilon \geq 0$ (larger ε means less privacy). Under this formulation, we give the first known RL sample complexity and regret upper and lower bounds with formal privacy guarantees. First, we present a new algorithm, `PUCB`, which satisfies ε-JDP in addition to two utility guarantees: it finds an α-optimal policy with a sample complexity of

$$\tilde{O}\left(\frac{SAH^4}{\alpha^2} + \frac{S^2AH^4}{\varepsilon\alpha}\right) \quad ,$$

and achieves a regret rate of

$$\tilde{O}\left(H^2\sqrt{SAT} + \frac{SAH^3 + S^2AH^3}{\varepsilon}\right)$$

over T episodes. In both of these bounds, the first terms $\frac{SAH^4}{\alpha^2}$ and $H^2\sqrt{SAT}$ are the non-private sample complexity and regret rates, respectively. The privacy parameter ε only affects the lower order terms – for sufficiently small approximation α and sufficiently large T, the "cost" of privacy becomes negligible.

We also provide new lower bounds for ε-JDP reinforcement learning. Specifically, by incorporating ideas from existing lower bounds for private learning into constructions of hard MDPs, we prove a sample complexity bound of

$$\tilde{\Omega}\left(\frac{SAH^2}{\alpha^2} + \frac{SAH}{\varepsilon\alpha}\right)$$

and a regret bound of

$$\tilde{\Omega}\left(\sqrt{HSAT} + \frac{SAH}{\varepsilon}\right) \quad .$$

As expected, these lower bounds match our upper bounds in the dominant term (ignoring H and polylogarithmic factors). We also see that necessarily the utility cost for privacy grows linearly with the state space size, although this does not match our upper bounds. Closing this gap is an important direction for future work.

1.1. Related Work

Most previous works on differentially private interactive learning with partial feedback concentrate on bandit-type problems, including on-line learning with bandit feedback (Thakurta & Smith, 2013; Agarwal & Singh, 2017), multi-armed bandits (Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016; 2017; 2018), and linear contextual bandits (Neel & Roth, 2018; Shariff & Sheffet, 2018). These works generally differ on the assumed reward models under which utility is measured (e.g. stochastic, oblivious adversarial, adaptive adversarial) and the concrete privacy definition being used (e.g. privacy when observing individual actions or sequences of actions, and privacy of reward or reward and observation in the contextual setting). Basu et al. (2019) provides a comprehensive account of different privacy definitions used in the bandit literature.

Much less work has addressed DP for general RL. For policy evaluation in the batch case, Balle et al. (2016) propose regularized least-squares algorithms with output perturbation and bound the excess risk due to the privacy constraints. For the control problem with private rewards and public states, Wang & Hegde (2019) give a differentially private Q-learning algorithm with function approximation.

On the RL side, as we are initiating the study of RL with differential privacy, we focus on the well-studied tabular setting. While a number of algorithms with utility guarantees and lower bound constructions are known for this setting (Kakade, 2003; Azar et al., 2017; Dann et al., 2017), we are not aware of any work addressing the privacy issues that are fundamental in high-stakes applications.

2. Preliminaries

2.1. Markov Decision Processes

A fixed-horizon *Markov decision process* (MDP) with time-dependent dynamics can be formalized as a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, p_0, H)$. $\mathcal{S}$ is the state space with cardinality S. $\mathcal{A}$ is the action space with cardinality A. $\mathcal{R}(s_h, a_h, h)$ is the reward distribution on the interval $[0, 1]$ with mean $r(s_h, a_h, h)$. $\mathcal{P}$ is the transition kernel, given time step h, action a_h, and state s_h the next state is sampled from $s_{h+1} \sim \mathcal{P}(.|s_h, a_h, h)$. Let p_0 be the initial state distribution at the start of each episode, and let H be the number of time steps in an episode.

In our setting, an agent interacts with an MDP by following a (deterministic) policy $\pi \in \Pi$, which maps states s and time steps h to actions, i.e., $\pi(s, h) \in \mathcal{A}$. The *value function* in time step $h \in [H]$ for a policy π is defined as:

$$\begin{aligned} V_h^\pi(s) &= \mathbb{E}\left[\sum_{i=h}^{H} r(s_i, a_i, i) \middle| s_h = s, \pi\right] \\ &= r(s, \pi(s, h), h) + \sum_{s' \in \mathcal{S}} V_{h+1}^\pi(s')\mathcal{P}(s'|s, \pi(s, h), h) \quad . \end{aligned}$$

The *expected total reward* for policy π during an entire episode is:

$$\rho^\pi = \mathbb{E}\left[\sum_{i=1}^{H} r(s_i, a_i, i) \middle| \pi\right] = p_0^\top V_1^\pi \ .$$

The *optimal value function* is given by $V_h^*(s) = \max_{\pi\in\Pi} V_h^\pi(s)$. Any policy π such that $V_h^\pi(s) = V_h^*(s)$ for all $s \in \mathcal{S}$ and $h \in [H]$ is called optimal. It achieves the optimal expected total reward $\rho^* = \max_{\pi\in\Pi} \rho^\pi$.

The goal of an RL agent is to learn a near-optimal policy after interacting with an MDP for a finite number of episodes T. During each episode $t \in [T]$ the agent follows a policy π_t informed by previous interactions, and after the last episode it outputs a final policy π.

Definition 1. An agent is (α, β)-*probably approximately correct* (PAC) with sample complexity $f(S, A, H, \frac{1}{\alpha}, \log(\frac{1}{\beta}))$, if with probability at least $1 - \beta$ it follows an α-optimal policy π such that $\rho^* - \rho^\pi \le \alpha$ except for at most $f(S, A, H, \frac{1}{\alpha}, \log(\frac{1}{\beta}))$ episodes.

Definition 2. The (expected cumulative) *regret* of an agent after T episodes is given by

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} (\rho^* - \rho^{\pi_t}) \ ,$$

where $\pi_1, \ldots \pi_T$ are the policies followed by the agent on each episode.

2.2. Privacy in RL

In some RL application domains such as personalized medical treatments, the sequence of states and rewards received by a reinforcement learning agent may contain sensitive information. For example, individual users may interact with an RL agent for the duration of an episode and reveal sensitive information in order to obtain a service from the agent. This information affects the final policy produced by the agent, as well as the actions taken by the agent in any subsequent interaction. Our goal is to prevent damaging inferences about a user's sensitive information in the context of the interactive protocol in Algorithm 1 summarizing the interactions between an RL agent $\mathcal{M}$ and T distinct users.

Throughout the execution of this protocol the agent observes a collection of T state-reward trajectories of length H. Each user u_t gets to observe the actions chosen by the agent during the t-th episode, as well as the final policy π. To preserve the privacy of individual users we enforce a (joint) differential privacy criterion: upon changing one of the users in the protocol, the information observed by the other $T-1$ participants will not change substantially. This criterion must hold even if the $T-1$ participants collude adversarially, by e.g., crafting their states and rewards to induce the agent to reveal information about the remaining user.

Algorithm 1 Episodic RL Protocol

```
input Agent $\mathcal{M}$ and users $u_1, \ldots, u_n$
  for all $t \in [n]$ do
    for all $h \in [H]$ do
      $u_t$ sends state $s_h^{(t)}$ to $\mathcal{M}$
      $\mathcal{M}$ sends action $a_h^{(t)}$ to $u_t$
      $u_t$ sends reward $r_h^{(t)}$ to $\mathcal{M}$
    end for
  end for
  $\mathcal{M}$ releases policy $\pi$
```

Formally, we write $U = (u_1, \ldots, u_T)$ to denote a sequence of T users participating in the RL protocol. Technically speaking a user can be identified with a tree of depth H encoding the state and reward responses they would give to all the A^H possible sequences of actions the agent can choose. During the protocol the agent only gets to observe the information along a single root-to-leaf path in each user's tree. For any $t \in [T]$, we write $\mathcal{M}_{-t}(U)$ to denote all the outputs excluding the output for episode t during the interaction between $\mathcal{M}$ and U. $\mathcal{M}_{-t}(U)$ captures all the outputs which might leak information about the t-th user in interactions after the t-th episode, as well as all the outputs from earlier episodes where other users could be submitting information to the agent adversarially to condition its interaction with the t-th users. We also say that two user sequences U and U' are t-neighbors if they only differ in their t-th user.

Definition 3. A randomized RL agent $\mathcal{M}$ is *ε-jointly differentially private under continual observation* (JDP) if for all $t \in [T]$, all t-neighboring user sequences U, U', and all events $E \subseteq \mathcal{A}^{H\times[T-1]} \times \Pi$ we have

$$\Pr\left[\mathcal{M}_{-t}(U) \in E\right] \le e^{\varepsilon} \Pr\left[\mathcal{M}_{-t}(U') \in E\right] \ .$$

This definition extends to the RL setting the one used in Shariff & Sheffet (2018) for designing privacy-preserving algorithms for linear contextual bandits. The key distinctions is that in our definition each user interacts with the agent for H time-steps (in bandit problems one has $H = 1$), and we also allow the agent to release the learned policy at the end of the learning process.

Another distinction is that our definition holds for all past and future outputs. In contrast, the definition of JDP in (Shariff & Sheffet, 2018) only captures future episodes; hence, it only protects against collusion from future users. To demonstrate that our definition gives a stronger privacy protection, we use a simple example.

Consider an online process that takes as input a stream of binary bits $u = (u_1, \ldots, u_T)$, where $u_t \in \{0, 1\}$ is the data of user t, and on each round t the mechanism outputs the partial sum $m_t(u) = \sum_{i=1}^{t} u_i$. Then the following trivial mechanism satisfies JDP for m's future outputs (as in the JDP definition of (Shariff & Sheffet, 2018)): First, sample once from the Laplace mechanism $\xi \sim \text{Lap}(\varepsilon)$ before the rounds begin, and on each round output $\tilde{m}_t(u) = m_t(u) + \xi$. Note that the view of any future user $t' > t$ is $\tilde{m}_{t'}(u)$. Now let u be a binary stream with user t bit on and let w be identical to u but with user t bit off. Then, by the differential-privacy guarantee of the Laplace mechanism, a user $t' > t$ cannot distinguish between $\tilde{m}_{t'}(u)$ and $\tilde{m}_{t'}(w)$. Furthermore, any coalition of future users cannot provide more information about user t. Therefore this simple mechanism satisfies the JDP definition from (Shariff & Sheffet, 2018).

However the simple counting mechanism with one round of Laplace noise does not satisfy JDP for past and future outputs as in our JDP definition 3. To see why, suppose that user $t-1$ and user $t+1$ collude in the following way: For input u, the view of user $t-1$ is $\tilde{m}_{t-1}(u)$ and the view of user $t+1$ is $\tilde{m}_{t+1}(u)$. They also know their own data u_{t-1}, u_{t+1}. Then they can recover the data of the t-th user as follows

$$\begin{aligned} &\tilde{m}_{t+1}(u) - u_{t+1} - \tilde{m}_{t-1}(u) \\ &= m_{t+1}(u) + \xi - u_{t+1} - m_{t-1}(u) - \xi \\ &= \sum_{i=1}^{t+1} u_i - u_{t+1} - \sum_{i=1}^{t-1} u_i = u_t \end{aligned}$$

Remark. *1. would the algorithm leak more info for the returning user? yes, but we could bound using group privacy. 2. would other users be affected? no, because JDP prevents arbitrary collusion*

2.3. Counting Mechanism

The algorithm we describe in the next section maintains a set of counters to keep track of events that occur when interacting with the MDP. We denote by $\widehat{n}_t(s,a,h)$ the count of visits to state tuple (s,a,h) right before episode t, where $a \in \mathcal{A}$ is the action taken on state $s \in \mathcal{S}$ and time-step $h \in [H]$. Likewise $\widehat{m}_t(s,a,s',h)$ is the count of going from state s to s' after taking actions a before episode t. Finally, we have the counter $\widehat{r}_t(s,a,h)$ for the total reward received by taking action a on state s and time h before episode t. Then, on episode t, the counters are sufficient to create an estimate of the MDP dynamics to construct a policy for episode t. The challenge is that the counters depend on the sequence of states and actions, which is considered sensitive data in this work. Therefore the algorithm must release the counts in a privacy-preserving way, and we do this the private counters proposed by Chan et al. (2011) and Dwork et al. (2010).

A private counter mechanism takes as input a stream $\sigma = (\sigma_1 \dots, \sigma_T) \in [0,1]^T$ and on any round t releases and approximation of the prefix count $c(\sigma)(t) = \sum_{i=1}^t \sigma_i$. In this work we will use PC to denote the binary mechanism of Chan et al. (2011) and Dwork et al. (2010) with parameters ε and T. This mechanism produces a monotonically increasing count and satisfies the following accuracy guarantee: Let $\mathcal{M} := \text{PC}(T, \varepsilon)$ be a private counter and $c(\sigma)(t)$ be the true count on episode t, then given a stream σ, with probability at least $1-\beta$, simultaneously for all $1 \le t \le T$, we have

$$|\mathcal{M}(\sigma)(t) - c(\sigma)(t)| \le \frac{4}{\varepsilon} \ln(1/\beta) \log(T)^{5/2} \quad .$$

While the above bound holds for a single ε-DP counter, our algorithm needs to maintain more than S^2AH many counters. A naive allocation of the privacy budget across all these counters will require noise that scales polynomially with S, A, and H. However, we will leverage the fact that a single user can in total influence all counters by at most the episode length H, which allows us to add a much smaller amount of noise that scales linearly in H.

3. The PUCB Algorithm

In this section, we introduce the Private Upper Confidence Bound algorithm (PUCB), a JDP algorithm with both PAC and regret guarantees. The pseudo-code for PUCB is in algorithm 2. At a high level, the algorithm is a private version of the UBEV algorithm (Dann et al., 2017). UBEV keeps track of three types of statistics about the history, including (a) the average empirical reward for taking action a in state s at time h, denoted $\widehat{r}_t(s,a,h)$, (b) the number of times the agent has taken action a in state s at time h, denoted $\widehat{n}_t(s,a,h)$, and (c) the number of times the agent has taken action a in state s at time h and transitioned to s', denoted $\widehat{m}_t(s,a,s',h)$. In each episode t, UBEV uses these statistics to compute a policy via dynamic programming, executes the policy, and updates the statistics with the observed trajectory. Dann et al. (2017) compute the policy using an optimistic strategy and establish both PAC and regret guarantees for this algorithm.

Of course, as the policy depends on the statistics from the previous episodes, UBEV as is does not satisfy JDP. On the other hand, the policy executed only depends on the previous episodes *only* through the statistics $\widehat{r}_t, \widehat{n}_t, \widehat{m}_t$. If we maintain and use private versions of these statistics, and we set the privacy level appropriately, we can ensure JDP.

To do so PUCB initializes one private counter mechanism for each $\widehat{r}_t, \widehat{n}_t, \widehat{m}_t$ ($2SAH + S^2AH$ counters in total). At episode t, we compute the policy using optimism as in UBEV, but we use only the private counts $\widetilde{r}_t, \widetilde{n}_t, \widetilde{m}_t$ released from the counter mechanisms. We require that each set of counters is $(\varepsilon/3)$ JDP, and so with

$$E_\varepsilon = \frac{3}{\varepsilon} H \log\left(\frac{2SAH + HAS^2}{\beta}\right) \log(T)^{5/2} ,$$

Algorithm 2 Private Upper Confidence Bound (`PUCB`)

```
Require: Privacy parameter $\varepsilon$, target failure probability $\beta$
input Maximum number of episodes $T$, horizon $H$, state space $\mathcal{S}$, action space $\mathcal{A}$
  $\varepsilon' := \varepsilon/(3H)$
  for all $s, a, s', h \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times [H]$ do
    Initialize private counters:
    $\widetilde{r}(s,a,h), \widetilde{n}(s,a,h), \widetilde{m}(s,a,s',h) :=$ PC $(T, \varepsilon')$
  end for
  for $t \leftarrow 1$ to $T$ do
    Private planning: $\widetilde{Q}_t^+ :=$ PrivQ$(\widetilde{r}, \widetilde{n}, \widetilde{m}, \varepsilon)$
    for $h \leftarrow 1$ to $H$ do
      Let $s$ denote the state during step $h$ and episode $t$
      Execute $a := \arg\max_{a'} \widetilde{Q}_t^+(s, a', h)$
      Observe $r \sim \mathcal{R}(s,a,h)$ and $s' \sim \mathcal{P}(.|s,a,h)$
      Feed $r$ to $\widetilde{r}(s,a,h)$
      Feed $1$ to $\widetilde{n}(s,a,h)$ and $\widetilde{m}(s,a,s',h)$ and $0$ to all other counters $\widetilde{n}(\cdot,\cdot,h)$ and $\widetilde{m}(\cdot,\cdot,\cdot,h)$
    end for
  end for
```

we can ensure that with probability at least $1 - \beta$:

$$\forall t \in [T] : |\widetilde{n}_t(s,a,h) - \widehat{n}_t(s,a,h)| < E_\varepsilon \ ,$$

where $\widehat{n}_t, \widetilde{n}_t$ are the count and release at the beginning of the t-th episode. The guarantee is uniform in (s,a,h) and also holds simultaneously for $\widetilde{r}$ and $\widetilde{m}$.

To compute the policy, we define a bonus function $\widetilde{\text{conf}}(s,a,h)$ for each (s,a,h) tuple, which can be decomposed into two parts $\widetilde{\phi}_t(s,a,h)$ and $\widetilde{\psi}_t(s,a,h)$, where

$$\widetilde{\phi}_t(s,a,h) = \sqrt{\frac{2\ln\left(\widetilde{n}_t(s,a,h) + E_\varepsilon\right) + 2\ln\left(\frac{SAH}{\beta}\right)}{\max(\widetilde{n}_t(s,a,h) - E_\varepsilon, 1)}} \ ,$$

$$\widetilde{\psi}_t(s,a,h) = (1 + SH)\left(\frac{3E_\varepsilon}{\widetilde{n}_t(s,a,h)} + \frac{2E_\varepsilon^2}{\widetilde{n}_t(s,a,h)^2}\right) \ .$$

The term $\widetilde{\phi}_t(\cdot)$ roughly corresponds to the sampling error, while $\widetilde{\psi}_t(\cdot)$ corresponds to errors introduced by the private counters. Using this bonus function, we use dynamic programming to compute an optimistic private Q-function in Algorithm 3. The algorithm here is a standard batch Q-learning update, with $\widetilde{\text{conf}}(\cdot)$ serving as an optimism bonus. The resulting Q-function, called $\widetilde{Q}^U$, encodes a greedy policy, which we use for the t-th episode.

4. Privacy Analysis of `PUCB`

We show that releasing the sequence of actions by algorithm `PUCB` satisfies JDP. Formally,

Theorem 1. *Algorithm (2) `PUCB` is ε-JDP.*

Algorithm 3 `PrivQ`$(\widetilde{r}, \widetilde{n}, \widetilde{m}, \varepsilon)$

```
input Private counters $\widetilde{r}, \widetilde{n}, \widetilde{m}$ and privacy parameter $\varepsilon$
  $E_\varepsilon := \frac{3}{\varepsilon} H \log\left(\frac{2SAH + HAS^2}{\beta}\right) \log(T)^{5/2}$
  $\widetilde{V}_{H+1}(s) := 0 \quad \forall s \in \mathcal{S}$
  for $h \leftarrow H$ to $1$ do
    for all $s, a \in \mathcal{S} \times \mathcal{A}$ do
      if $\widetilde{n}_t(s,a,h) \geq 2E_\varepsilon$ then
        $\widetilde{\text{conf}}_t(s,a,h) := (H+1)\widetilde{\phi}_t(s,a,h) + \widetilde{\psi}_t(s,a,h)$
      else
        $\widetilde{\text{conf}}_t(s,a,h) := H$
      end if
      $\widetilde{Q}_t := \frac{\widetilde{r}_t(s,a,h) + \sum_{s' \in \mathcal{S}} \widetilde{V}_{h+1}(s')\widetilde{m}_t(s,a,s',h)}{\widetilde{n}_t(s,a,h)}$
      $\widetilde{Q}_t^+(s,a,h) := \min\left\{H, \widetilde{Q}_t + \widetilde{\text{conf}}_t(s,a,h)\right\}$
    end for
    $\widetilde{V}_h(s) := \max_a \widetilde{Q}_t^+(s,a,h) \quad \forall s \in \mathcal{S}$
  end for
output $\widetilde{Q}_t^+$
```

To prove theorem 1, we use the *billboard lemma* due to Hsu et al. (2016) which says that an algorithm is JDP if the output sent to each user is a function of the user's private data and a common signal computed with standard differential privacy. We state the formal lemma:

Lemma 2 (Billboard lemma (Hsu et al., 2016)). *Suppose $\mathcal{M} : U \to \mathcal{R}$ is ε-differentially private. Consider any set of functions $f_i : U_i \times \mathcal{R} \to \mathcal{R}'$ where U_i is the portion of the database containing the i's user data. The composition $\{f_i(\Pi_i U, \mathcal{M}(U))\}$ is ε-joint differentially private, where $\Pi_i : U \to U_i$ is the projection to i's data.*

Let $U_{<t}$ denote the data of all users before episode t and u_t denote the user's data during episode t. Algorithm `PUCB` keeps track of all events on users $U_{<t}$ in a differentially-private way with private counters $\widetilde{r}_t, \widetilde{n}_t, \widetilde{m}_t$. These counters are given to the procedure `PrivQ` which computes a Q-function $\widetilde{Q}_t^+$, and induces the policy $\pi_t(s,h) := \max_a \widetilde{Q}_t^+(s,a,h)$ to be used by the agent during episode t.

Then the output during episode t is generated by the policy π_t and the private data of the user u_t according to protocol 1, the output on a single episode is: $\left(\pi_t\left(s_1^{(t)}, 1\right), \ldots, \pi_t\left(s_H^{(t)}, H\right)\right)$. By the billboard lemma, the composition of the output of all T episodes, and the final policy $\left(\left\{\left(\pi_t(s_1^{(t)}, 1), \ldots, \pi_t(s_H^{(t)}, H)\right)\right\}_{t \in [T]}, \pi_T\right)$ satisfies ε-JDP if the policies $\{\pi_t\}_{t \in [T]}$ are computed with a ε-DP mechanism.

Then it only remains to show that the noisy counts satisfy ε-DP. First, consider the counters for the number of visited states. The algorithm `PUCB` runs SAH parallel private counters, one for each state tuple (s, a, h). Each counter is instantiated with a $\varepsilon/(3H)$-differentially private mechanism which takes an input an event stream $\widehat{n}(s, a, h) = \{0, 1\}^T$ where the i-th bit is set to 1 if a user visited the state tuple (s, a, h) during episode i and 0 otherwise. Hence each stream $\widehat{n}(s, a, h)$ is the data for a private counter. The next claim says that the total ℓ_1 sensitivity over all streams is bounded by H:

Claim 1. *Let U, U' be two t-neighboring user sequences, in the sense that they are only different in the data for episode t. For each $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, let $\widehat{n}(s, a, h)$ be the event stream corresponding to user sequence U and $\widehat{n}'(s, a, h)$ be the event stream corresponding to U'. Then the total ℓ_1 distances of all streams is:*

$$\sum_{(s,a,h)\in\mathcal{S}\times\mathcal{A}\times[H]} \|\widehat{n}(s, a, h) - \widehat{n}'(s, a, h)\|_1 \leq H$$

Proof. The proof follows from the fact that on any episode t a user visits at most H states. □

Finally we use a result from Hsu et al. (2016, Lemma 34) which states that the composition of the SAH $(\varepsilon/3H)$-DP counters for $\widehat{n}(\cdot)$ satisfy $(\varepsilon/3)$-DP as long as the ℓ_1 sensitivity of the counters is H as shown in claim 1. We can apply the same analysis to show that the counters corresponding to the empirical reward $\widehat{r}(\cdot)$ and the transitions $\widehat{m}(\cdot)$ are both also $\varepsilon/3$-differentially private. Thus, releasing all the noisy counters is ε-differentially private.

5. PAC and Regret Analysis of `PUCB`

Now that we have established that `PUCB` is JDP, we turn to utility guarantees. We establish two forms of utility guarantee namely a PAC sample complexity bound and a regret bound. In both cases, comparing to `UBEV`, we show that the price for JDP is quite mild. In both bounds the privacy parameter interacts quite favorably with the "error parameter."

We first state the PAC guarantee.

Theorem 3 (PAC guarantee for `PUCB`)**.** *Let T be the maximum number of episodes and ε the JDP parameter. Then for any $\alpha \in (0, H]$ and $\beta \in (0, 1)$, algorithm `PUCB` with parameters (ε, β) follows a policy that with probability at least $1 - \beta$ is α-optimal on all but*

$$O\left(\left(\frac{SAH^4}{\alpha^2} + \frac{S^2AH^4}{\varepsilon\alpha}\right) \operatorname{polylog}\left(T, S, A, H, \tfrac{1}{\alpha}, \tfrac{1}{\beta}, \tfrac{1}{\varepsilon}\right)\right)$$

episodes.

The theorem states that if we run `PUCB` for many episodes, it will act near-optimally in a large fraction of them. The number of episodes where the algorithm acts suboptimally scales polynomially with all the relevant parameters. In particular, notice that in terms of the utility parameter α, the bound scales as $1/\alpha^2$. In fact the first term here matches the guarantee for the non-private algorithm `UBEV` up to polylogarithmic factors. On the other hand, the privacy parameter ε appears only in the term scaling as $1/\alpha$. In the common case where α is relatively small, this term is typically of a lower order, and so the price for privacy here is relatively low.

Analogous to the PAC bound, we also have a regret guarantee.

Theorem 4 (Regret bound for `PUCB`)**.** *With probability at least $1 - \beta$, the regret of `PUCB` is at most*

$$O\left(\left(H^2\sqrt{SAT} + \tfrac{S^2AH^4}{\varepsilon}\right) \operatorname{polylog}\left(T, S, A, H, \tfrac{1}{\beta}, \tfrac{1}{\varepsilon}\right)\right) \quad .$$

A similar remark to the PAC bound applies here: the privacy parameter only appears in the $\operatorname{polylog}(T)$ terms, while the leading order term scales as $\sqrt{T}$. In this guarantee it is clear that as T gets large, the utility price for privacy is essentially negligible.

We also remark that both bounds have "lower order" terms that scale with S^2. This is quite common for tabular reinforcement algorithms (Dann et al., 2017; Azar et al., 2017). We find it quite interesting to observe that the privacy parameter ε interacts with this term, but not with the so-called "leading" term in these guarantees.

Proof Sketch. The proofs for both results parallel the arguments in Dann et al. (2017) for the analysis of `UBEV`. The main differences arises from the fact that we have adjusted the confidence interval $\widetilde{\text{conf}}$ to account for the noise in the releases of $\widetilde{r}, \widetilde{n}, \widetilde{m}$. In Dann et al. (2017) the bonus is crucially used to establish optimism, and the final guarantees are related to the over-estimation incurred by these bonuses. We focus on these two steps in this sketch, with a full proof deferred to the appendix.

First we verify optimism. Fix episode t and (s, a, h), and let us abbreviate the latter simply by x. Assume that $\widetilde{V}_{h+1}$ is private and optimistic in the sense that $\widetilde{V}_{h+1}(s) \geq V^*_{h+1}(s)$, for all $s \in \mathcal{S}$. First define the empirical Q-value

$$\widehat{Q}_t(\text{x}) := \frac{\widehat{r}_t(\text{x}) + \sum_{s'\in\mathcal{S}} \widetilde{V}_{h+1}(s')\widehat{m}_t(\text{x}, s')}{\widehat{n}_t(\text{x})} \quad .$$

The optimistic Q-function, which is similar to the one used by Dann et al. (2017), is given by

$$\widehat{Q}^+_t(\text{x}) := \widehat{Q}_t(\text{x}) + (H + 1)\widehat{\phi}_t(\text{x}) \quad ,$$

where $\widehat{\phi}_t(\text{x}) := \sqrt{\frac{2\ln \widehat{n}_t(\text{x}) + 2\ln\left(\frac{SAH}{\beta}\right)}{\widehat{n}_t(\text{x})}}$. A standard concentration argument shows that $\widehat{Q}^+_t \geq Q^\star$, assuming that $\widetilde{V}_{h+1} \geq V^\star_{h+1}$.

Of course, both $\widehat{Q}_t$ and $\widehat{Q}_t^+$ involve the non-private counters $\widehat{r}, \widehat{n}, \widehat{m}$, so they are *not* available to our algorithm. Instead, we construct a surrogate for the empirical Q-value using the private releases:

$$\widetilde{Q}_t(\mathrm{x}) := \frac{\widetilde{r}_t(\mathrm{x}) + \sum_{s' \in \mathcal{S}} \widetilde{V}_{h+1}(s')\widetilde{m}_t(\mathrm{x}, s')}{\widetilde{n}_t(\mathrm{x})} \quad .$$

Our analysis involves relating $\widetilde{Q}_t$ which the algorithm has access to, with the non-private $\widehat{Q}_t$. To do this, note that by the guarantee for the counting mechanism, we have

$$\widehat{Q}_t(\mathrm{x}) \le \frac{\widetilde{r}_t(\mathrm{x}) + E_\varepsilon + \sum_{s' \in \mathcal{S}} \widetilde{V}_{h+1}(s')(\widetilde{m}_t(\mathrm{x}, s') + E_\varepsilon)}{\widetilde{n}_t(\mathrm{x}) - E_\varepsilon} \quad . \tag{1}$$

Next, we use the following elementary fact.

Claim 2. *Let $y \in \mathbb{R}$ be any positive real number. Then for all $x \in \mathbb{R}$ with $x \ge 2y$ it holds that $\frac{1}{x-y} \le \frac{1}{x} + \frac{2y}{x^2}$.*

If $\widetilde{n}_t(\mathrm{x}) \ge 2E_\varepsilon$, then we can use claim 2 in (1), along with the facts that $\widetilde{V}_{h+1}(s') \le H$ and $\widetilde{r}_t(\mathrm{x}) \le \widetilde{n}_t(\mathrm{x}) + 2E_\varepsilon \le 2\widetilde{n}_t(\mathrm{x})$, to upper bound $\widehat{Q}_t$ by $\widetilde{Q}_t$. This gives:

$$\begin{aligned}\widehat{Q}_t(\mathrm{x}) &\le \widetilde{Q}_t(\mathrm{x}) + \left(\frac{1}{\widetilde{n}_t(\mathrm{x})} + \frac{2E_\varepsilon}{\widetilde{n}_t(\mathrm{x})^2}\right) \cdot (1 + SH)E_\varepsilon \\ &= \widetilde{Q}_t(\mathrm{x}) + \widetilde{\psi}_t(\mathrm{x}) \quad .\end{aligned}$$

Therefore, we see that $\widetilde{Q}_t(\mathrm{x}) + \widetilde{\psi}_t(\mathrm{x})$ dominates $\widehat{Q}_t(\mathrm{x})$. Accordingly, if we inflate by $\widetilde{\phi}_t(\mathrm{x})$ – which is clearly an upper bound on $\widehat{\phi}_t(\mathrm{x})$ – we account for the statistical fluctuations and can verify optimism. In the event that $\widetilde{n}_t(\mathrm{x}) \le 2E_\varepsilon$, we simply upper bound $Q^* \le H$.

For the over-estimation, the bonus we have added is $\widetilde{\phi}_t(\mathrm{x}) + \widetilde{\psi}_t(\mathrm{x})$, which is closely related to the original bonus $\widehat{\phi}_t(\mathrm{x})$. The essential property for our bonus is that it is not significantly larger than the original one $\widehat{\phi}_t(\mathrm{x})$. Indeed, $\widehat{\phi}_t(\mathrm{x})$ scales as $1/\sqrt{\widehat{n}_t(\mathrm{x})}$ while $\widetilde{\psi}_t(\mathrm{x})$ scales roughly as $E_\varepsilon/\widehat{n}_t(\mathrm{x}) + E_\varepsilon^2/\widehat{n}_t(\mathrm{x})^2$, which is lower order in the dependence on $\widehat{n}_t(\mathrm{x})$. Similarly, the other sources of error here only have lower order effects on the over-estimation.

In detail, there are three sources of error. First, $\widetilde{\phi}_t(\mathrm{x})$ is within a constant factor of $\widehat{\phi}_t(\mathrm{x})$ since we are focusing on rounds where $\widetilde{n}_t(\mathrm{x}) \ge 2E_\varepsilon$. Second, as the policy suboptimality is related to the bonuses on the states and actions we are likely to visit, we cannot have many rounds where $\widetilde{n}_t(\mathrm{x}) \le 2E_\varepsilon$, since all of the private counters are increasing. A similar argument applies for $\widetilde{\psi}_t(\mathrm{x})$: we can ignore states that we visit infrequently, and the private counters $\widetilde{n}_t(\mathrm{x})$ for states that we visit frequently increase rapidly enough to introduce minimal additional error. Importantly, in the latter two arguments, we have terms of the form $E_\varepsilon/\widetilde{n}_t(\mathrm{x})$, while $\widehat{\phi}_t(\mathrm{x})$ itself scales as $\sqrt{1/\widehat{n}_t(\mathrm{x})}$, which dominates in terms of the accuracy parameter α or the number of episodes T. As such we obtain PAC and regret guarantees where the privacy parameter ε does not appear in the dominant terms.

6. Lower Bounds

In this section we prove the following lower bounds on the sample complexity and regret for any PAC RL agent providing joint differential privacy.

Theorem 5 (PAC Lower Bound)**.** *Let $\mathcal{M}$ be an RL agent satisfying ε-JDP. Suppose that $\mathcal{M}$ is (α, β)-PAC for some $\beta \in (0, 1/8)$. Then, there exists a fixed-horizon episodic MDP where the number of episodes until the algorithm's policy is α-optimal with probability at least $1 - \beta$ satisfies*

$$\mathbb{E}\left[n_{\mathcal{M}}\right] \ge \Omega\left(\frac{SAH^2}{\alpha^2} + \frac{SAH}{\alpha\varepsilon}\ln\left(\frac{1}{\beta}\right)\right) \quad .$$

Theorem 6 (Private Regret Lower Bound)**.** *For any ε JDP-algorithm $\mathcal{M}$ there exist an MDP M with S states A actions over H time steps per episode such that the expected regret after T steps is*

$$\mathbb{E}\left[\mathrm{Regret}(T)\right] = \Omega\left(\sqrt{HSAT} + \frac{SAH\log(T)}{\varepsilon}\right)$$

Due to space constraints, we will present the proof steps for the sample complexity lower bound in Theorem 5. The proof for the regret lower bound in Theorem 6 follows from a similar argument and is deferred to the appendix.

To obtain Theorem 5, we go through two intermediate lower bounds: one for private best-arm identification in multi-armed bandits problems (Lemma 8), and one for private RL in a relaxed scenario where the initial state of each episode is considered public information (Lemma 10). At first glance our arguments look similar to other techniques that provide lower bounds for RL in the non-private setting by leveraging lower bounds for bandits problems, e.g. (Strehl et al., 2009; Dann & Brunskill, 2015). However, getting this strategy to work in the private case is significantly more challenging because one needs to ensure the notions of privacy used in each of the lower bounds are compatible with each other. Since this is the main challenge to prove Theorem 5, we focus our presentation on the aspects that make the private lower bound argument different from the non-private one, and defer details to the appendix.

6.1. Lower Bound for Best-Arm Identification

The first step is a lower bound for best-arm identification for differentially private multi-armed bandits algorithms. This considers mechanisms $\mathcal{M}$ interacting with users via the MAB protocol described in Algorithm 4, where we assume arms $a^{(t)}$ come from some finite space $\mathcal{A}$ and rewards are binary, $r^{(t)} \in \{0, 1\}$. Our lower bound applies to mechanisms for this protocol that satisfy standard DP in the sense that the adversary has access to all the outputs $\mathcal{M}(U) = (a^{(1)}, \ldots, a^{(T)}, \hat{a})$ produced by the mechanism.

Definition 4. A MAB mechanism $\mathcal{M}$ is ε-DP if for any neighboring user sequences U and U' differing in a single user, and all events $E \subseteq \mathcal{A}^{T+1}$ we have

$$\Pr[\mathcal{M}(U) \in E] \leq e^{\varepsilon} \Pr[\mathcal{M}(U') \in E] \ .$$

To measure the utility of a mechanism for performing *best-arm identification* in MABs we consider a stochastic setting with independent arms. In this setting each arm $a \in \mathcal{A}$ produces rewards following a Bernoulli distribution with expectation $\bar{P}_a$ and the goal is to identify with high probability an optimal arm a^* with expected reward $\bar{P}_{a^*} = \max_{a \in \mathcal{A}} \bar{P}_a$. A problem instance can be identified with the vector of expected rewards $\bar{P} = (\bar{P}_a)_{a \in \mathcal{A}}$.

Algorithm 4 MAB Protocol for Best-Arm Identification

```
input  Agent $\mathcal{M}$ and users $u_1, \ldots, u_T$
  for all $t \in [T]$ do
    $\mathcal{M}$ sends arm $a^{(t)}$ to $u_t$
    $u_t$ sends reward $r^{(t)}$ to $\mathcal{M}$
  end for
  $\mathcal{M}$ releases arm $\hat{a}$
```

The lower bound result relies on an adaptation of the coupling lemma from Karwa & Vadhan (2017, Lemma 6.2).

Lemma 7. *Fix any arm $a \in [k]$. Now consider any pair of MAB instances $\mu, \nu \in [0, 1]^k$ both with k arms and time horizon T, such that $\|\mu_a - \nu_a\|_{tv} < \alpha$ and $\|\mu_{a'} - \nu_{a'}\|_{tv} = 0$ for all $a' \neq a$. Let $R \sim B(\mu)^T$ and $Q \sim B(\nu)^T$ be the sequence of T rounds of rewards sampled under μ and ν respectively, and let $\mathcal{M}$ be any ε-DP multi-armed bandit algorithm. Then, for any event E such that under event E arm a is pulled less than t times,*

$$Pr_{\mathcal{M},R}\left[E\right] \leq e^{6\varepsilon t\alpha} Pr_{\mathcal{M},Q}\left[E\right]$$

Lemma 8 (Private MAB Lower Bound)**.** *Let $\mathcal{M}$ be a MAB best-arm identification algorithm satisfying ε-DP that succeeds with probability at least $1 - \beta$, for some $\beta \in (0, 1/4)$. For any MAB instance $\bar{P}$ and any α-suboptimal arm a with $\alpha > 0$ (i.e. $\bar{P}_a = \bar{P}_{a^*} - \alpha$), the number of times that $\mathcal{M}$ pulls arm a during the protocol satisfies*

$$\mathbb{E}\left[n_a\right] > \frac{1}{24\varepsilon\alpha} \ln\left(\frac{1}{4\beta}\right) \ .$$

6.2. Lower Bound for RL with Public Initial State

To leverage the lower bound for private best-arm identification in the RL setting we first consider a simpler setting where the initial state of each episode is public information. This means that we consider agents $\mathcal{M}$ interacting with a variant of the protocol in Algorithm 1 where each user t releases their first state $s_1^{(t)}$ in addition to sending it to the agent. We model this scenario by considering agents whose inputs (U, S_1) include the sequence of initial states $S_1 = \left(s_1^{(1)}, \ldots, s_1^{(T)}\right)$, and define the privacy requirements in terms of a different notion of neighboring inputs: two sequences of inputs (U, S_1) and (U', S_1') are t-neighboring if $u_{t'} = u'_{t'}$ for all $t \neq t'$ and $S_1 = S_1'$. That is, we do not expect to provide privacy in the case where the user that changes between U and U' also changes their initial state, since in this case making the initial state public already provides evidence that the user changed. Note, however, that u_t and u'_t can provide different rewards for actions taken by the agent on state $s_1^{(t)}$.

Definition 5. A randomized RL agent $\mathcal{M}$ is ε-JDP under continual observation in the *public initial state* setting if for all $t \in [T]$, all t-neighboring user-state sequences (U, S_1), (U', S_1'), and all events $E \subseteq A^{H \times [T-1]} \times \Pi$ we have

$$\Pr\left[\mathcal{M}_{-t}(U, S_1) \in E\right] \leq e^{\varepsilon} \Pr\left[\mathcal{M}_{-t}(U', S_1') \in E\right] \ .$$

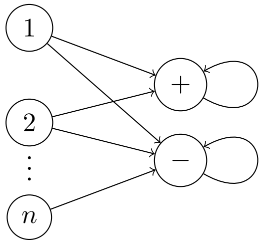

Figure 1. Class of hard MDP instances used in the lower bound.

We obtain a lower bound on the sample complexity of PAC RL agents that satisfy JDP in the public initial state setting by constructing a class of hard MDPs shown in Figure 1. An MDP in this class has state space $\mathcal{S} := [n] \cup \{+, -\}$ and action space $\mathcal{A} := \{0, \ldots, m\}$. On each episode, the agent starts in one of the initial states $\{1, \ldots, n\}$ chosen uniformly at random. In each of the initial states the agent has $m + 1$ possible actions and transitions can only take it to one of two possible absorbing states $\{+, -\}$. If the current state is either one of $\{+, -\}$ then the only possible transition is a self loop, hence the agent will remain in that state until the end of the episode. We assume in these absorbing states the agent can only take a fixed action. Every action which transitions to state $+$ provides reward 1 while actions transitioning to state $-$ provide reward 0. Thus, in each episode the agent either receives reward H or 0.

Such an MDP can be seen as consisting of n parallel MAB problems. Each MAB problem determines the transition probabilities between the initial state $s \in \{1, \dots, n\}$ and the absorbing states $\{+, -\}$. We index the possible MAB problems in each initial state by their optimal arm, which is always one of $\{0, \dots, m\}$. We write $I_s \in \{0, \dots, m\}$ to denote the MAB instance in initial state s, and define the transition probabilities such that $\Pr[+|s, 0] = 1/2 + \alpha'/2$ and $\Pr[+|s, a'] = 1/2$ for $a' \neq I_s$ for all I_s, and for $I_s \neq 0$ we also have $\Pr[+|s, I_s] = 1/2 + \alpha'$. Here α' is a free parameter to be determined later. We succinctly represent an MDP in the class by identifying the optimal action (i.e. arm) in each initial state: $I \coloneqq (I_1, \dots, I_n)$.

To show that our MAB lower bounds imply lower bounds for an RL agent interacting with MDPs in this class we prove that collecting the first action taken by the agent in all episodes t with a fixed initial state $s_1^{(t)} = s \in [n]$ simulates the execution of an ε-DP MAB algorithm.

Let $\mathcal{M}$ be an RL agent and (U, S_1) a user-state input sequence with initial states from some set $\mathcal{S}_1$. Let $\mathcal{M}(U, S_1) = (\vec{a}^{(1)}, \dots, \vec{a}^{(T)}, \pi) \in \mathcal{A}^{H \times T} \times \Pi$ be the collection of all outputs produced by the agent on inputs U and S_1. For every $s \in \mathcal{S}_1$ we write $\mathcal{M}_{1,s}(U, S_1)$ to denote the restriction of the previous trace to contain just the first action from all episodes starting with s together with the action predicted by the policy at states s:

$$\mathcal{M}_{1,s}(U, S_1) \coloneqq \left(a_1^{(t_{s,1})}, \dots, a_1^{(t_{s,T_s})}, \pi(s)\right) \quad ,$$

where T_s is the number of occurrences of s in S_1 and $t_{s,1}, \dots, t_{s,T_s}$ are the indices of these occurrences. Furthermore, given $s \in \mathcal{S}_1$ we write $U_s = (u_{t_{s,1}}, \dots, u_{t_{s,T_s}})$ to denote the set of users whose initial state equals s.

Lemma 9. *Let (U, S_1) be a user-state input sequence with initial states from some set $\mathcal{S}_1$. Suppose $\mathcal{M}$ is an RL agent that satisfies ε-JDP in the public initial state setting. Then, for any $s \in \mathcal{S}_1$ the trace $\mathcal{M}_{1,s}(U, S_1)$ is the output of an (ε)-DP MAB mechanism on input U_s.*

Using Lemmas 8 and 9 and a reduction from RL lower bounds to bandits lower bounds yields the second term in the following result. The first terms follows directly from the non-private lower bound in (Dann & Brunskill, 2015).

Lemma 10. *Let $\mathcal{M}$ be an RL agent satisfying ε-JDP in the public initial state setting. Suppose that $\mathcal{M}$ is (α, β)-PAC for some $\beta \in (0, 1/8)$. Then, there exists a fixed-horizon episodic MDP where the number of episodes until the algorithm's policy is α-optimal with probability at least $1 - \beta$ satisfies*

$$\mathbb{E}[n_{\mathcal{M}}] \geq \Omega\left(\frac{SAH^2}{\alpha^2} + \frac{SAH}{\alpha\varepsilon} \ln\left(\frac{1}{\beta}\right)\right) \quad .$$

Finally, Theorem 5 follows from Lemma 10 by observing that any RL agent $\mathcal{M}$ satisfying ε-JDP also satisfies ε-JDP in the public state setting (see appendix for a formal statement).

7. Conclusion

In this paper, we initiate the study of differentially private algorithms for reinforcement learning. On the conceptual level, we formalize the privacy desiderata via the notion of joint differential privacy, where the algorithm cannot strongly base future decisions off sensitive information from previous interactions. Under this formalism, we provide a JDP algorithm and establish both PAC and regret utility guarantees for episodic tabular MDPs. Our results show that the utility cost for privacy is asymptotically negligible in the large accuracy regime. We also establish the first lower bounds for reinforcement learning with JDP.

A natural direction for future work is to close the gap between our upper and lower bounds. A similar gap remains open for tabular RL *without* privacy considerations, but the setting is more difficult with privacy, so it may be easier to establish a lower bound here. We look forward to pursuing this direction, and hope that progress will yield new insights into the non-private setting.

Beyond the tabular setup considered in this paper, we believe that designing RL algorithms providing state and reward privacy in non-tabular settings is a promising direction for future work with considerable potential for real-world applications.

8. Acknowledgements

We thank Matthew Joseph, whose comments corrected our definition of joint differential privacy. Giuseppe Vietri is supported by the GAANN fellowship from the U.S. Department of Education.

References

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

Agarwal, N. and Singh, K. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 32–40. JMLR. org, 2017.

Andrew, G., Chien, S., and Papernot, N. Tensorflow privacy. https://github.com/tensorflow/privacy, 2019.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Balle, B., Gomrokchi, M., and Precup, D. Differentially private policy evaluation. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 2130–2138, 2016.

Basu, D., Dimitrakakis, C., and Tossou, A. Differential privacy for multi-armed bandits: What is it and what is its cost? *arXiv preprint arXiv:1905.12298*, 2019.

Chan, T.-H. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):26, 2011.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5713–5723, 2017.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

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. ACM, 2010.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

Holohan, N., Braghin, S., Aonghusa, P. M., and Levacher, K. Diffprivlib: The IBM differential privacy library. *CoRR*, abs/1907.02444, 2019.

Hsu, J., Huang, Z., Roth, A., Roughgarden, T., and Wu, Z. S. Private matchings and allocations. *SIAM Journal on Computing*, 45(6):1953–1984, 2016.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning, 2019.

Kakade. On the sample complexity of reinforcement learning. *Diss. University of London*, 2003.

Karwa, V. and Vadhan, S. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

Kearns, M. J., Pai, M. M., Roth, A., and Ullman, J. Mechanism design in large games: incentives and privacy. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pp. 403–410, 2014. doi: 10.1145/2554797.2554834.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions, 2019.

Mishra, N. and Thakurta, A. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 592–601, 2015.

Neel, S. and Roth, A. Mitigating bias in adaptive data gathering via differential privacy. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 3717–3726, 2018.

Pan, X., Wang, W., Zhang, X., Li, B., Yi, J., and Song, D. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pp. 368–376, 2019.

Shariff, R. and Sheffet, O. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pp. 4296–4306, 2018.

Strehl, A. L., Li, L., and Littman, M. L. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Team, A. D. P. Learning with privacy at scale. https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html, 2017.

Thakurta, A. G. and Smith, A. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pp. 2733–2741, 2013.

Tossou, A. C. and Dimitrakakis, C. Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Tossou, A. C. and Dimitrakakis, C. On the differential privacy of thompson sampling with gaussian prior. *arXiv preprint arXiv:1806.09192*, 2018.

Tossou, A. C. Y. and Dimitrakakis, C. Achieving privacy in the adversarial multi-armed bandit. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Wang, B. and Hegde, N. Privacy-preserving q-learning with functional noise in continuous spaces. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 11323–11333. 2019.

Wilson, R. J., Zhang, C. Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., and Gipson, B. Differentially private sql with bounded user contribution, 2019.