# Locally Differentially Private Reinforcement Learning for Linear Mixture Markov Decision Processes

**Chonghua Liao*** LCH18@MAILS.TSINGHUA.EDU.CN
*Department of Automation, Tsinghua University, Beijing, China, 100084*

**Jiafan He*** JIAFANHE19@UCLA.EDU
*Department of Computer Science, University of California, Los Angeles, CA 90095, USA*

**Quanquan Gu** QGU@CS.UCLA.EDU
*Department of Computer Science, University of California, Los Angeles, CA 90095, USA*

**Editors:** Emtiyaz Khan and Mehmet Gönen

## Abstract

Reinforcement learning (RL) algorithms can be used to provide personalized services, which rely on users' private and sensitive data. To protect the users' privacy, privacy-preserving RL algorithms are in demand. In this paper, we study RL with linear function approximation and local differential privacy (LDP) guarantees. We propose a novel $(\varepsilon, \delta)$-LDP algorithm for learning a class of Markov decision processes (MDPs) dubbed linear mixture MDPs, and obtains an $\widetilde{\mathcal{O}}(d^{5/4}H^{7/4}T^{3/4}\left(\log(1/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$ regret, where $d$ is the dimension of feature mapping, $H$ is the length of the episodes, and $T$ is the number of interactions with the environment. We also prove a lower bound $\Omega(dH\sqrt{T}/\left(e^{\varepsilon}(e^{\varepsilon}-1)\right))$ for learning linear mixture MDPs under $\varepsilon$-LDP constraint. Experiments on synthetic datasets verify the effectiveness of our algorithm. To the best of our knowledge, this is the first provable privacy-preserving RL algorithm with linear function approximation.

**Keywords:** Machine learning, Reinforcement learning, Differential privacy, Linear mixture MDPs

## 1. Introduction

Reinforcement learning (RL) algorithms have been studied extensively in the past decade. When the state and action spaces are large or even infinite, traditional tabular RL algorithms (e.g., Watkins, 1989; Jaksch et al., 2010; Azar et al., 2017) become computationally inefficient or even intractable. To overcome this limitation, modern RL algorithms with function approximation are proposed, which often make use of feature mappings to map states and actions to a low-dimensional space. This greatly expands the application scope of RL. While RL can provide personalized service such as online recommendation and personalized advertisement, existing algorithms rely heavily on user's sensitive data. Recently, how to protect sensitive information has become a central research problem in machine learning. For example, in online recommendation systems, users want accurate recommendation from the online shopping website to improve their shopping experience while preserving their personal information such as demographic information and purchase history.

* Equal Contribution

Differential privacy (DP) is a solid and highly successful notion of algorithmic privacy introduced by Dwork et al. (2006), which indicates that changing or removing a single data point will have little influence on any observable output. However, DP is vulnerable to membership inference attacks (Shokri et al., 2017) and has the risk of data leakage. To overcome the limitation of DP, a stronger notion of privacy, *local differential privacy* (LDP), was introduced by Kasiviswanathan et al. (2011); Duchi et al. (2013). Under LDP, users will send privatized data to the server and each individual user maintains its own sensitive data. The server, on the other hand, is totally agnostic about the sensitive data.

(Local) differential privacy has been extensively studied in multi-armed bandit problems, which can be seen as a special case of MDPs with unit episode length and without state transition. Nevertheless, Shariff and Sheffet (2018) proved that the standard DP is incompatible in the contextual bandit setting, which will yield a linear regret bound in the worst case. Therefore, they studied a relaxed version of DP named *joint differential privacy* (JDP), which basically requires that changing one data point in the collection of information from previous users will not have too much influence on the decision of the future users. Recently, LDP has attracted increasing attention in multi-armed bandits. Gajane et al. (2018) are the first to study LDP in stochastic multi-armed bandits (MABs). Chen et al. (2020) studied combinatorial bandits with LDP guarantees. Zheng et al. (2020) studied both MABs and contextual bandits, and proposed a locally differentially private algorithm for contextual linear bandits. However, differentially private RL is much less studied compared with bandits, even though MDPs are more powerful since state transition is rather common in real applications. For example, a user may click a link provided by the recommendation system to visit a related webpage, which can be viewed as state transition. In tabular RL, Vietri et al. (2020) proposed a $\varepsilon$-JDP algorithm and proved an $\widetilde{\mathcal{O}}(\sqrt{H^4SAT} + SAH^3(S+H)/\varepsilon)$ regret, where $H$ is the episode length, $S$ and $A$ are the number of states and actions respectively, $K$ is the number of episodes, and $T = KH$ is the number of interactions with the MDP. Recently, Garcelon et al. (2020) designed the first LDP tabular RL algorithm with an $\widetilde{\mathcal{O}}(\max\{H^{3/2}S^2A\sqrt{T}/\varepsilon, HS\sqrt{AT}\})$ regret. However, as we mentioned before, tabular RL algorithms suffer from computational inefficiency when applied to large state and action spaces. Therefore, a natural question arises:

*Can we design a privacy-preserving RL algorithm with linear function approximation while maintaining the statistical utility of data?*

In this paper, we answer this question affirmatively. More specifically, we propose a locally differentially private algorithm for learning linear mixture MDPs (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021b) (See Definition 3.1 for more details.), where the transition probability kernel is a linear function of a predefined $d$-dimensional feature mapping over state-action-next-state triple. The key idea is to inject Gaussian noises into the sensitive information in the UCRL-VTR backbone, and the main challenge is how to balance the tradeoff between the Gaussian perturbations for privacy preservation and the utility to learn the optimal policy.

Our contributions are summarized as follows.

- We propose a novel algorithm named LDP-UCRL-VTR to learn the optimal value function while protecting the sensitive information. We show that our algorithm guarantees

$(\varepsilon, \delta)$-LDP and enjoys an $\widetilde{\mathcal{O}}(d^{5/4}H^{7/4}T^{3/4}\left(\log(1/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$ regret bound, where $T$ is the number of rounds and $H$ is the length of episodes. To our knowledge, this is the first locally differentially private algorithm for RL with linear function approximation.

- We prove a $\Omega(dH\sqrt{T}/\left(e^{\varepsilon}(e^{\varepsilon}-1)\right))$ lower bound for learning linear mixture MDPs under $\varepsilon$-LDP constraints. Our lower bound suggests that the aforementioned upper bound might be improvable in some parameters (i.e., $d, H, T$). As a byproduct, our lower bound also implies $\Omega(d\sqrt{T}/\left(e^{\varepsilon}(e^{\varepsilon}-1)\right))$ lower bound for $\varepsilon$-LDP contextual linear bandits. This suggests that the algorithms proposed in Zheng et al. (2020) might be improvable as well.

**Notation** We use lower case letters to denote scalars, lower and upper case bold letters to denote vectors and matrices. For a vector $\mathbf{x} \in \mathbb{R}^d$ , we denote by $\|\mathbf{x}\|_1$ the Manhattan norm and denote by $\|\mathbf{x}\|_2$ the Euclidean norm. For a semi-positive definite matrix $\mathbf{\Sigma}$ and any vector $\mathbf{x}$, we define $\|\mathbf{x}\|_{\mathbf{\Sigma}} := \left\|\mathbf{\Sigma}^{1/2}\mathbf{x}\right\|_2 = \sqrt{\mathbf{x}^\top \mathbf{\Sigma}\mathbf{x}}$. $\mathbb{1}(\cdot)$ is used to denote the indicator function. For any positive integer $n$, we denote by $[n]$ the set $\{1, \ldots, n\}$. For any finite set $\mathcal{A}$, we denote by $|\mathcal{A}|$ the cardinality of $\mathcal{A}$. We also use the standard $\mathcal{O}$ and $\Omega$ notations, and the notation $\widetilde{\mathcal{O}}$ is used to hide logarithmic factors. We denote $D_{1:h} = \{D_1, \ldots, D_h\}$. For two distributions $p$ and $p'$, we define the Kullback–Leibler divergence (KL-divergence) between $p$ and $p'$ as follows: $\mathrm{KL}(p, p') = \int p(\mathbf{z}) \log\left(p(\mathbf{z})/p'(\mathbf{z})\right) \mathrm{d}\mathbf{z}$.

## 2. Related Work

**Reinforcement Learning with Linear Function Approximation** Recently, there have been many advances in RL with function approximation, especially in the linear case. Jin et al. (2020) considered linear MDPs where the transition probability and the reward are both linear functions with respect to a feature mapping $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, and proposed an efficient algorithm for linear MDPs with $\widetilde{\mathcal{O}}(\sqrt{d^3H^3T})$ regret. Yang and Wang (2019a) assumed the probabilistic transition model has a linear structure. They also assumed that the features of all state-action pairs can be written as a convex combination of the anchoring features. Wang et al. (2019) designed a statistically and computationally efficient algorithm with generalized linear function approximation, which attains an $\widetilde{\mathcal{O}}(H\sqrt{d^3T})$ regret bound. Zanette et al. (2020) proposed RLSVI algorithm with $\widetilde{\mathcal{O}}(d^2\sqrt{H^4T})$ regret bound under the linear MDPs assumption. Jiang et al. (2017) studied a larger class of MDPs with low Bellman rank and proposed an OLIVE algorithm with polynomial sample complexity. Another line of work considered linear mixture MDPs (a.k.a., linear kernel MDPs) (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021b), which assumes the transition probability function is parameterized as a linear function of a given feature mapping on a triplet $\boldsymbol{\psi} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$. Jia et al. (2020) proposed a model-based RL algorithm, UCRL-VTR, which attains an $\widetilde{\mathcal{O}}(d\sqrt{H^3T})$ regret bound. Ayoub et al. (2020) considered the same model but with general function approximation, and proved a regret bound depending on Eluder dimension (Russo and Van Roy, 2013). Zhou et al. (2021a) proposed an improved algorithm which achieves the nearly minimax optimal regret. He et al. (2021) showed that logarithmic regret is attainable for learning both linear MDPs and linear mixture MDPs.

**Differentially Private Bandits** The notion of *differential privacy* (DP) was first introduced in Dwork et al. (2006) and has been extensively studied in both MAB and contextual linear bandits. Basu et al. (2019) unified different privacy definitions and proved an

$\Omega(\sqrt{KT}/(e^{\varepsilon}(e^{\varepsilon}-1)))$ regret lower bound for locally differentially private MAB algorithms, where $K$ is the number of arms. Shariff and Sheffet (2018) derived an impossibility result for learning contextual bandits under DP constraint by showing an $\Omega(T)$ regret lower bound for any $(\varepsilon, \delta)$-DP algorithms. Hence, they considered the relaxed *joint differential privacy* (JDP) and proposed an algorithm based on Lin-UCB (Abbasi-Yadkori et al., 2011) with $\widetilde{\mathcal{O}}(\sqrt{T}/\varepsilon)$ regret while preserving $\varepsilon$-JDP. Recently, a stronger definition of privacy, *local differential privacy* (Duchi et al., 2013; Kasiviswanathan et al., 2011), gained increasing interest in bandit problems. Intuitively, LDP ensures that each collected trajectory is differentially private when observed by the agent, while DP requires the computation on the entire set of trajectories to be DP. Zheng et al. (2020) proposed an LDP contextual linear bandit algorithm with $\widetilde{\mathcal{O}}(d^{3/4}T^{3/4})$ regret.

**Differentially Private RL** In RL, Balle et al. (2016) is the first to propose a private algorithm for policy evaluation with linear function approximation. In the tabular setting, Vietri et al. (2020) designed a $\varepsilon$-JDP algorithm for regret minimization which attains an $\widetilde{\mathcal{O}}(\sqrt{H^4SAT}+SAH^3(S+H)/\varepsilon)$ regret. Later, Garcelon et al. (2020) presented an optimistic algorithm with LDP guarantees, which enjoys an $\widetilde{\mathcal{O}}(\max\{H^{3/2}S^2A\sqrt{T}/\varepsilon, HS\sqrt{AT}\})$ regret upper bound. They also provided a $\widetilde{\Omega}(\sqrt{HSAT}/\min\{\exp(\varepsilon)-1, 1\})$ regret lower bound. However, all these private RL algorithms are in the tabular setting, and private RL algorithms with linear function approximation remain understudied. Recently, an independent (concurrent) work by Luyo et al. (2021) also studied (locally) differentially private reinforcement learning and proposed an almost same algorithm as ours with slightly different parameter choice. Later, Zhou (2022) studied the joint differential privacy guarantee for linear mixture MDPs, and their result can be potentially extended to the locally differential privacy. Nevertheless, neither Luyo et al. (2021) or Zhou (2022) provided a lower bound for regret.

## 3. Preliminaries

In this paper, we study locally differentially private RL with linear function approximation for episodic MDPs. In the following, we will introduce the necessary background and definitions.

### 3.1. Markov Decision Processes

**Episodic Markov Decision Processes** We consider the setting of an episodic time-inhomogeneous Markov decision process (Puterman, 2014), denoted by the following tuple $M = M(\mathcal{S}, \mathcal{A}, H, \{r_h\}_{h=1}^H, \{\mathbb{P}_h\}_{h=1}^H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $H$ is the length of each episode, $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the deterministic reward function and $\mathbb{P}_h(s'|s, a)$ is the transition probability function which denotes the probability for state $s$ to transfer to state $s'$ given action $a$ at stage $h$. A policy $\pi = \{\pi_h\}_{h=1}^H$ is a collection of $H$ functions, where $\pi_h(s)$ denote the action that the agent will take at stage $h$ and state $s$. Moreover, for each $h \in [H]$, we define the value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ that maps state $s$ to the expected value of cumulative rewards received under policy $\pi$ when starting from state $s$ at the $h$-th stage. We also define the action-value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which maps a state-action pair $(s, a)$ to the expected value of cumulative rewards when the agent

starts from state-action pair $(s, a)$ at the $h$-th stage and follows policy $\pi$ afterwards. More specifically, for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have

$$Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}\left[\sum_{h'=h+1}^{H} r_{h'}\big(s_{h'}, \pi_{h'}(s'_h)\big)\right], V_h^\pi(s) = Q_h^\pi\big(s, \pi_h(s)\big),$$

where $s_h = s, a_h = a$ and $s_{h'+1} \sim \mathbb{P}_{h'}(\cdot \,|\, s_{h'}, a_{h'})$.

For each function $V : \mathcal{S} \to \mathbb{R}$, we further denote $[\mathbb{P}_h V](s,a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot \,|\, s,a)} V(s')$. Using this notation, the Bellman equation with policy $\pi$ can be written as

$$Q_h^\pi(s,a) = r_h(s,a) + \left[\mathbb{P}_h V_{h+1}^\pi\right](s,a), \qquad V_h^\pi(s) = Q_h^\pi(s, \pi_h(s)), \qquad V_{H+1}^\pi(s) = 0,$$

We define the optimal value function $V_h^*$ as $V_h^*(s) = \max_\pi V_h^\pi(s)$ and the optimal action-value function $Q_h^*$ as $Q_h^*(s,a) = \max_\pi Q_h^\pi(s,a)$. With this notation, the Bellman optimality equation can be written as follows

$$Q_h^*(s,a) = r_h(s,a) + \left[\mathbb{P}_h V_{h+1}^*\right](s,a), \qquad V_{h+1}^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s,a),$$

where $V_{H+1}^*(s) = 0$. In the setting of an episodic MDP, an agent aims to learn the optimal policy by interacting with the environment and observing the past information. At the beginning of the $k$-th episode, the agent chooses the policy $\pi_k$ and the adversary picks the initial state $s_1^k$. At each stage $h \in [H]$, the agent observes the state $s_h^k$, chooses an action following the policy $a_h^k = \pi_h^k(s_h^k)$ and observes the next state with $s_{h+1}^k \sim \mathbb{P}_h(\cdot | s_h^k, a_h^k)$. The difference between $V_1^*(s_1^k)$ and $V_1^{\pi^k}(s_1^k)$ represents the expected regret in the $k$-th episode. Thus, the total regret in first $K$ episodes can be defined as

$$\text{Regret}(K) = \sum_{k=1}^{K} \left(V_1^*(s_1^k) - V_1^{\pi_k}(s_1^k)\right).$$

**Linear Function Approximation** In this work, we consider a class of MDPs called *linear mixture MDPs* (Modi et al., 2020; Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021b), where the transition probability function can be represented as a linear function of a given feature mapping $\boldsymbol{\phi}(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ satisfying that for any bounded function $V : S \to [0, H]$ and any tuple $(s,a) \in S \times A$, we have

$$\|\boldsymbol{\phi}_V(s,a)\|_2 \leqslant H, \text{ where } \boldsymbol{\phi}_V(s,a) = \sum_{s' \in \mathcal{S}} \boldsymbol{\phi}(s' \,|\, s,a) V(s'). \tag{3.1}$$

Formally, we have the following definition:

**Definition 3.1** (Jia et al. 2020; Ayoub et al. 2020; Zhou et al. 2021b)**.** An Markov Decision Processe $(\mathcal{S}, \mathcal{A}, H, \{r_h\}_{h=1}^H, \{\mathbb{P}_h\}_{h=1}^H)$ is an inhomogeneous, episodic bounded linear mixture MDP if there exist vectors $\boldsymbol{\theta}_h \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}_h^*\|_2 \leqslant \sqrt{d}$ and a feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ satisfying (3.1) such that $\mathbb{P}_h(s' \,|\, s,a) = \langle \boldsymbol{\phi}(s' \,|\, s,a), \boldsymbol{\theta}_h^* \rangle$ for any state-action-next-state triplet $(s,a,s') \in S \times A \times S$ and stage $h$.

Therefore, learning the underlying $\boldsymbol{\theta}_h^*$ can be regarded as solving a "linear bandit" problem (Part V, Lattimore and Szepesvári (2020)), where the context is $\boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k) \in \mathbb{R}^d$, and the noise is $V_{k,h+1}(s_{h+1}^k) - [\mathbb{P}_h V_{k,h+1}](s_h^k, a_h^k)$.

**Remark 3.2.** Linear mixture MDPs have been widely studied in the literature (Modi et al., 2020; Ayoub et al., 2020; Zhou et al., 2021b; Cai et al., 2020; Zhou et al., 2021a; He et al., 2021; Wu et al., 2022; He et al., 2022) for reinforcement learning with linear function approximation, and it contains several important MDPs models such as tabular MDP and feature embedding of transition model (Yang and Wang, 2019b).

**Example 3.3** (Tabular MDPs)**.** In particular, for any tabular Markov Decision Process $M(\mathcal{S}, \mathcal{A}, H, \{r_h\}_{h=1}^H, \{\mathbb{P}_h\}_{h=1}^H)$ with finite state space $\mathcal{S}$ and finite action space $\mathcal{A}$, it can be represented by a linear mixture MDP with dimension $d = |\mathcal{S}|^2|\mathcal{A}|$. More specifically, the transition probability function can be written as the inner product between the one-hot feature mapping $\boldsymbol{\phi}(s'|s,a) = \mathbf{e}_{(s,a,s')}$, and the unknown parameter vector $\boldsymbol{\theta}_h = [\mathbb{P}_h(s'|s,a)]_{s'\in\mathcal{S},s\in\mathcal{S},a\in\mathcal{A}}$.

**Example 3.4** (Feature embedding of transition model Yang and Wang 2019b)**.** For any feature embedding of transition model, it assumes that the transition probability function can be denoted by $\mathbb{P}_h(s'|s,a) = \boldsymbol{\phi}(s,a)^\top \mathbf{M}_h \boldsymbol{\psi}(s')$, where $\boldsymbol{\phi}(s,a) : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^{d_1}$, $\boldsymbol{\psi}(s') : \mathcal{S} \to \mathbb{R}^{d_2}$ are feature mappings based on the state and action, and $\mathbf{M}_h$ is the unknown parameter matrix. This model can be represented as a linear mixture MDP with the dimension $d = d_1 d_2$. More precisely, the transition probability function can be written as the inner product between the feature mapping $\boldsymbol{\phi}(s'|s,a) = \mathrm{vec}\big(\boldsymbol{\psi}(s')\boldsymbol{\phi}(s,a)^\top\big)$ and the unknown parameter vector $\boldsymbol{\theta}_h = \mathrm{vec}(\mathbf{M}_h)$.

### 3.2. (Local) Differential Privacy

In this subsection, we introduce the standard definition of differential privacy (Dwork et al., 2006) and local differential privacy (Kasiviswanathan et al., 2011; Duchi et al., 2013). We also present the definition of Gaussian mechanism.

**Differential Privacy** Differential privacy is a mathematically rigorous notion of data privacy. In our setting, DP considers that the information collected from all the users can be observed and aggregated by a server. It ensures that the algorithm's output renders neighboring inputs indistinguishable. Thus, we formalize the definition as follows:

**Definition 3.5** (Differential Privacy)**.** For any user $k \in [K]$, let $D_k$ be the information sent to a privacy-preserving mechanism from user $k$ and the collection of data from all the users can be written as $\{D_k\}_{1:K} = \{D_1, \ldots, D_k, \ldots, D_K\}$. For any $\varepsilon \geqslant 0$ and $\delta \geqslant 0$, a randomized mechanism $\mathcal{M}$ preserves $(\varepsilon, \delta)$-differential privacy if for any two neighboring datasets $\{D_k\}_{1:K}, \{D'_k\}_{1:K} \subseteq \mathcal{Z}$ which only differ at one entry, and for any measurable subset $U \in \mathcal{Z}$, it satisfies

$$\mathbb{P}\left(\mathcal{M}\left(\{D_k\}_{1:K}\right) \in U\right) \leqslant e^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\left\{D'_k\right\}_{1:K}\right) \in U\right) + \delta\,,$$

where $\{D_k\}_{1:K} = \{D_1, \ldots, D_k, \ldots, D_K\}, \{D'_k\}_{1:K} = \{D_1, \ldots, D'_k, \ldots, D_K\}$.

**Local Differential Privacy** In online RL, we view each episode $k \in [K]$ as a trajectory associated to a specific user. A natural way to conceive LDP in RL setting is to guarantee that for any user, the information sent to the server has been privatized. Thus, LDP ensures that the server is totally agnostic to the sensitive data, and we are going to state the following definition:

**Definition 3.6** (Local Differential Privacy)**.** For any $\varepsilon \geqslant 0$ and $\delta \geqslant 0$, a randomized mechanism $\mathcal{M}$ preserves $(\varepsilon, \delta)$-local differential privacy if for any two users $u$ and $u'$ and their corresponding data $D_u, D_{u'} \in \mathcal{U}$, it satisfies:

$$\mathbb{P}\left(\mathcal{M}\left(D_u\right) \in U\right) \leqslant e^{\varepsilon} \mathbb{P}\left(\mathcal{M}\left(D_{u'}\right) \in U\right) + \delta, \quad U \in \mathcal{U}\,.$$

**Remark 3.7.** The dataset $\{D_k\}_{1:K}$ in DP is a collection of information from users $1, \ldots, K$, where the subscript indicates the $k$-th user. Post-processing theorem implies that LDP is a more strict notion of privacy than DP.

Now we are going to introduce the Gaussian mechanism which is widely used as a privacy-preserving mechanism to ensure DP/LDP property.

**Lemma 3.8.** (The Gaussian Mechanism, Dwork et al. 2014). Let $f : \mathcal{X} \mapsto \mathbb{R}^d$ be an arbitrary d-dimensional function (a query), and define its $\ell_2$ sensitivity as $\Delta_2 f = \max_{\text{adjacent(x,y)}} \|f(x) - f(y)\|_2$, where adjacent$(x, y)$ indicates that $x, y$ are different only at one entry. For any $0 < \varepsilon < 1$ and $c^2 > 2\log(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geqslant c\Delta_2 f/\varepsilon$ is $(\varepsilon, \delta)$-differentially private.

## 4. Algorithm

We propose LDP-UCRL-VTR algorithm as displayed in Algorithm 1, which can be regarded as a variant of UCRL-VTR algorithm proposed in Jia et al. (2020) with $(\varepsilon, \delta)$-LDP guarantee. Algorithm 1 takes the privacy parameters $\varepsilon, \delta$ as input (Line 1). For the first user $k = 1$, we simply have $\mathbf{\Lambda}_{1,h} = \mathbf{\Sigma}_{1,h} = \lambda \mathbf{I}$ and $\widehat{\boldsymbol{\theta}}_{1,h} = \mathbf{0}$ (Line 4). For local user $k$ and received information $\mathbf{\Lambda}_h^k, \mathbf{u}_h^k$, the optimistic estimator of the optimal action-value function is constructed with an additional UCB bonus term (Line 6),

$$Q_{k,h}(\cdot,\cdot) \leftarrow \min\left\{H, r_h(\cdot,\cdot) + \left\langle \widehat{\boldsymbol{\theta}}_{k,h}, \boldsymbol{\phi}_{V_{k,h+1}}(\cdot,\cdot)\right\rangle + \beta \left\|\mathbf{\Sigma}_{k,h}^{-1/2} \boldsymbol{\phi}_{V_{k,h+1}}(\cdot,\cdot)\right\|_2\right\},$$

and $\beta$ is specified as $cd^{3/4}(H-h+1)^{3/2}k^{1/4}\log(dT/\alpha)\left(\log((H-h+1)/\delta)\right)^{1/4}\sqrt{1/\varepsilon}$, where $c$ is an absolute constant. From the previous sections, we know that learning the underlying $\boldsymbol{\theta}_h^*$ can be regarded as solving a "linear bandit" problem, where the context is $\boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k) \in \mathbb{R}^d$, and the noise is $V_{k,h+1}(s_{h+1}^k) - [\mathbb{P}_h V_{k,h+1}](s_h^k, a_h^k)$. Therefore, to estimate $Q_h^*$, it suffices to estimate the vector $\boldsymbol{\theta}_h^*$ by ridge regression with input $\boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k)$ and output $V_{k,h+1}(s_{h+1}^k)$. In order to implement the ridge regression, the server should collect the information of $\boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k)\boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k)^\top$ and $V_{k,h+1}(s_{h+1}^k)$ from each user $k$ (Line 15). Thus, we need to add noises to privatize the data before sending these information to the server in order to kept user's information private. In LDP-UCRL-VTR, we attain LDP by adding independent Gaussian noises: symmetric Gaussian matrix and $d$-dimensional Gaussian noise (Line 12 and Line 13). For simplicity, we denote the original information (without noise) $\Delta\widetilde{\mathbf{\Lambda}}_h^k = \boldsymbol{\phi}_{V_{k,h+1}}(s_h, a_h)\boldsymbol{\phi}_{V_{k,h+1}}(s_h, a_h)^\top, \Delta\widetilde{\mathbf{u}}_h^k = \boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k)V_{k,h+1}(s_{h+1}^k)$, where $k$ indicates the user and $h$ indicates the stage. Since the input information to the server is kept private by the user, it is easy to show that LDP-UCRL-VTR algorithm satisfies $(\varepsilon, \delta)$-LDP. After receiving the information from user 1 to user $k$, the server aggregates information $\mathbf{\Lambda}_{k,h}, \mathbf{u}_{k,h}$, and maintains them for $H$ stages separately (Line 18). Besides, since

**Algorithm 1** LDP-UCRL-VTR

**Require:** privacy parameters $\varepsilon, \delta$, failure probability $\alpha$, parameter $\lambda$

1: Set $\sigma = 4H^3\sqrt{2\log(2.5H/\delta)/\varepsilon}$
2: **for** user $k = 1, \ldots, K$ **do**
3: **For the local user $k$:**
4: Receive $\{\mathbf{\Sigma}_{k,1}, \ldots, \mathbf{\Sigma}_{k,H}, \widehat{\boldsymbol{\theta}}_{k,1}, \ldots, \widehat{\boldsymbol{\theta}}_{k,H}\}$ from the server
5: **for** $h = H, \ldots, 1$ **do**
6: $Q_{k,h}(\cdot,\cdot) \leftarrow \min\{H-h+1, r_h(\cdot,\cdot) + \left\langle \widehat{\boldsymbol{\theta}}_{k,h}, \boldsymbol{\phi}_{V_{k,h+1}}(\cdot,\cdot)\right\rangle + \beta_h \left\|\mathbf{\Sigma}_{k,h}^{-1/2}\boldsymbol{\phi}_{V_{k,h+1}}(\cdot,\cdot)\right\|_2\}$
7: $V_{k,h}(\cdot) \leftarrow \max_a Q_{k,h}(\cdot, a)$.
8: **end for**
9: Receive the initial state $s_1^k$
10: **for** $h = 1, \ldots, H$ **do**
11: Take action $a_h^k \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} Q_{k,h}(s_h^k)$, and observe $s_{h+1}^k$
12: Set $\Delta\mathbf{\Lambda}_h \leftarrow \boldsymbol{\phi}_{V_{k,h+1}}(s_h, a_h)\boldsymbol{\phi}_{V_{k,h+1}}(s_h, a_h)^\top + \mathbf{W}_{k,h}$, where $\mathbf{W}_{k,h}(i,j) = \mathbf{W}_{k,h}(j,i)$ and $\mathbf{W}_{k,h}(i,j) \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,\sigma^2), \forall i \leqslant j$
13: Set $\Delta\mathbf{u}_h \leftarrow \boldsymbol{\phi}_{V_{k,h+1}}(s_h^k, a_h^k)V_{k,h+1}(s_{h+1}^k) + \boldsymbol{\xi}_h$, where $\boldsymbol{\xi}_h \sim \mathcal{N}(\mathbf{0}_d, \sigma^2\mathbf{I}_{d\times d})$
14: **end for**
15: Send $D^k = \{\Delta\mathbf{\Lambda}_1, \ldots, \Delta\mathbf{\Lambda}_H, \Delta\mathbf{u}_1, \ldots, \Delta\mathbf{u}_H\}$ to the server
16: **For the server:**
17: **for** $h = 1, \ldots, H$ **do**
18: $\mathbf{\Lambda}_{k+1,h} \leftarrow \mathbf{\Lambda}_{k,h} + \Delta\mathbf{\Lambda}_h, \mathbf{u}_{k+1,h} \leftarrow \mathbf{u}_{k,h} + \Delta\mathbf{u}_h$
19: $\mathbf{\Sigma}_{k+1,h} \leftarrow \mathbf{\Lambda}_{k+1,h} + r\mathbf{I}$
20: $\widehat{\boldsymbol{\theta}}_{k+1,h} \leftarrow (\mathbf{\Sigma}_{k+1,h})^{-1}\mathbf{u}_{k+1,h}$
21: **end for**
22: Send $\{\mathbf{\Sigma}_{k+1,1}, \ldots, \mathbf{\Sigma}_{k+1,H}, \widehat{\boldsymbol{\theta}}_{k+1,1}, \ldots, \widehat{\boldsymbol{\theta}}_{k+1,H}\}$ to user $k+1$
23: **end for**

the Gaussian matrix may not preserve the PSD (Positive semi-definite) property, we adapt the idea of shifted regularizer in Shariff and Sheffet (2018) and shift this matrix $\mathbf{\Lambda}_{k,h}$ by $r\mathbf{I}$ to guarantee PSD (Line 19). We then calculate $\widehat{\boldsymbol{\theta}}_{k+1,h}$ and send $\mathbf{\Sigma}_{k+1,h}, \widehat{\boldsymbol{\theta}}_{k+1,h}$ back to $k+1$-th user in order to get a more precise estimation of $\boldsymbol{\theta}_h^*$ for better exploration.

**Comparison with related algorithms.** We would like to comment on the difference between our LDP-UCRL-VTR and other related algorithm. The key difference between our LDP-UCRL-VTR and UCRL-VTR (Jia et al., 2020), which is the most related algorithm, is that we add additive noises to the contextual vectors and the optimistic value functions in order to guarantee privacy. Then the server collects privatized information from different users and update $\mathbf{\Lambda}, \mathbf{u}$ for ridge regression. A shifted regularizer designed in Shariff and Sheffet (2018) is used to guarantee PSD property of the matrix. It is easy to show that if we add no noise to user's information, our LDP-UCRL-VTR algorithm degenerates to inhomogeneous UCRL-VTR. Another related algorithm is the Contextual Linear Bandits with LDP in Zheng et al. (2020), which is an algorithm designed for contextual linear bandits. Setting $H = 1$, our LDP-UCRL-VTR will degenerate to Contextual Linear Bandits with LDP in Zheng et al. (2020).

## 5. Main Results

In this section, we provide both privacy and regret guarantees for Algorithm 1. The detailed proofs of the main results are deferred to the appendix.

### 5.1. Privacy Guarantees

Recall that in Algorithm 1, we use Gaussian mechanism to protect the private information of the contextual vectors and the optimistic value functions. Based on the property of Gaussian mechanism, we can show that our algorithm is $(\varepsilon, \delta)$-LDP.

**Theorem 5.1.** Algorithm 1 preserves $(\varepsilon, \delta)$-LDP.

The privacy analysis relies on the fact that if the information from each user satisfies $(\varepsilon, \delta)$-DP, then the whole algorithm is $(\varepsilon, \delta)$-LDP.

### 5.2. Regret Upper Bound

The following theorem states the regret upper bound of Algorithm 1.

**Theorem 5.2.** For any fixed $\alpha \in (0, 1)$, for any privacy parameters $\varepsilon > 0$ and $\delta > 0$, if we set the parameters $\lambda = 1$ and $\beta_h = \widetilde{\mathcal{O}}(d^{3/4}(H-h+1)^{3/2}k^{1/4}\log(dT/\alpha)\left(\log((H-h+1)/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$ for user $k$, with probability at least $1-\alpha$, the total regret of Algorithm 1 in the first $T$ steps is at most $\widetilde{O}(d^{5/4}H^{7/4}T^{3/4}\log(dT/\alpha)\left(\log(H/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$, where $T = KH$ is the number of interactions with the MDP.

**Remark 5.3.** By setting the failure probability $\alpha = \delta$, our regret bound can be written as $\widetilde{\mathcal{O}}(d^{5/4}H^{7/4}T^{3/4}\left(\log(H/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$. Compared with UCRL-VTR, which enjoys an upper bound of $\widetilde{\mathcal{O}}(d\sqrt{H^3T})$, our bound suggests that learning the linear mixture MDP under the LDP constraint is inherently no easier than learning it non-privately.

### 5.3. Regret Lower Bounds

In this subsection, we present a lower bound for learning linear mixture MDPs under the $\varepsilon$-LDP constraint. We follow the idea firstly developed in Zhou et al. (2021a), which basically shows that learning a linear mixture MDP is no harder than learning $H/2$ linear bandit problems. As a byproduct, we also derived the regret lower bound for learning $\varepsilon$-LDP contextual linear bandits.

In detail, in order to prove the regret lower bound for MDPs under $\varepsilon$-LDP constraint, we first prove the lower bound for learning $\varepsilon$-LDP linear bandit problems. We adapted the proof techniques in Lattimore and Szepesvári (2020, Theorem 24.1) and Basu et al. (2019). In the non-private setting, the observed history of a contextual bandit algorithm in the first $T$ rounds can be written as $\mathcal{H}_T = \{\mathbf{x}_t, y_t\}_{t=1}^T$. Given history $\mathcal{H}_{t-1}$, the contextual linear bandit algorithm chooses action $\mathbf{x}_t$, and the reward is generated from a distribution $f_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_t)$, which is conditionally independent of the previously observed history. We use $\mathbb{P}_{\pi,\boldsymbol{\theta}}^T$ to denote the distribution of observed history up to time $T$, which is induced by $\pi$ and $f_{\boldsymbol{\theta}}$. Hence, we have

$$\mathbb{P}_{\pi,\boldsymbol{\theta}}^T = \mathbb{P}_{\pi,\boldsymbol{\theta}}(\mathcal{H}_T) = \prod_{t=1}^{T} \pi(\mathbf{x}_t \,|\, \mathcal{H}_{t-1}) f_{\boldsymbol{\theta}}(y_t \,|\, \mathbf{x}_t)\,,$$

where $\pi$ is the stochastic policy (the distribution over an action set induced by a bandit algorithm) and $f_{\boldsymbol{\theta}}(\cdot \,|\, \mathbf{x}_t)$ is the reward distribution given action $\mathbf{x}_t$, which is conditionally independent of the previously observed history $\mathcal{H}_{t-1}$.

In the LDP setting, the privacy-preserving mechanism $\mathcal{M}$ generates the privatized version of the context $\mathbf{x}_t$, denoted by $\widetilde{\mathbf{x}}_t = \mathcal{M}(\mathbf{x}_t)$, to the contextual linear bandit algorithm. For simplicity, we denote $\mathcal{M}_\pi$ as the distribution (stochastic policy) by imposing a locally differentially private mechanism $\mathcal{M}$ on the distribution (policy) $\pi$. Also, we use $f_{\boldsymbol{\theta}}^{\mathcal{M}}$ to denote the conditional distribution of $\widetilde{y}_t$ parameterized by $\boldsymbol{\theta}$, where $\widetilde{y}_t$ is the privatized version of $y_t$ obtained by the privacy-preserving mechanism $\mathcal{M}$. We denote the observed history by $\widetilde{\mathcal{H}}_T := \{(\widetilde{\mathbf{x}}_t, \widetilde{y}_t)\}_{t=1}^T$, where $\widetilde{\mathbf{x}}_t$, $\widetilde{y}_t$ are the privatized version of contexts and rewards. Similarly, we have

$$\widetilde{\mathbb{P}}_{\pi,\boldsymbol{\theta}}^T := \mathbb{P}_{\pi,\boldsymbol{\theta}}(\widetilde{\mathcal{H}}_T) = \prod_{t=1}^T \mathcal{M}_\pi(\widetilde{\mathbf{x}}_t \,|\, \widetilde{\mathcal{H}}_{t-1}) f_{\boldsymbol{\theta}}^{\mathcal{M}}(\widetilde{y}_t \,|\, \mathbf{x}_t)\,. \tag{5.1}$$

With the formulation above, we proved the following key lemma for $\varepsilon$-LDP contextual linear bandits.

**Lemma 5.4.** (Locally Differentially Private KL-divergence Decomposition) We denote the reward generated by user $t$ for action $\mathbf{x}_t$ as $y_t = \mathbf{x}_t^\top \boldsymbol{\theta} + \eta_t$, where $\eta_t$ is a zero-mean noise. If the reward generation process is $\varepsilon$-locally differentially private for both the bandits with parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, we have,

$$\mathrm{KL}(\widetilde{\mathbb{P}}_{\pi,\boldsymbol{\theta}_1}^T, \widetilde{\mathbb{P}}_{\pi,\boldsymbol{\theta}_2}^T) \leqslant 2\min\left\{4, e^{2\varepsilon}\right\}(e^\varepsilon - 1)^2 \cdot \sum_{t=1}^T \mathbb{E}_{\pi,\boldsymbol{\theta}_1}\left[\mathrm{KL}(f_{\boldsymbol{\theta}_1}^{\mathcal{M}}(\widetilde{y}_t \,|\, \mathbf{x}_t), f_{\boldsymbol{\theta}_2}^{\mathcal{M}}(\widetilde{y}_t \,|\, \mathbf{x}_t))\right],$$

where $\widetilde{\mathbf{x}}_t$, $\widetilde{y}_t$ are the privatized version of contexts and rewards.

Lemma 5.4 can be seen as an extension of Lemma 3 in Basu et al. (2019) from multi-armed bandits to contextual linear bandits.

Equipped with Lemma 5.4, the KL-divergence of privatized history distributions can be decomposed into the distributions of rewards. We construct a contextual linear bandit with Bernoulli reward. In detail, for an action $\mathbf{x}_t \in \mathcal{A} \subseteq \mathbb{R}^d$, the reward follows a Bernoulli distribution $y_t \sim B(\langle \boldsymbol{\theta}, \mathbf{x}_t \rangle + \delta)$, where $0 \leqslant \delta \leqslant 1/3$. We first derive a regret lower bound of learning contextual bandits under the LDP constraint in the following lemma.

**Lemma 5.5** (Regret Lower Bound for LDP Contextual Linear Bandits)**.** Given an $\varepsilon$-locally differentially private reward generation mechanism with $\varepsilon$ and a time horizon $T$, for any environment with finite variance, if then time horizon $T$ satisfies that $T \geqslant 4d^2/\big(\min\{4, e^{2\epsilon}\}(e^\epsilon - 1)^2\big)$, then the pseudo regret of any algorithm $\pi$ satisfies

$$\mathrm{Regret}(T) \geqslant \frac{c}{\min\left\{2, e^\varepsilon\right\}(e^\varepsilon - 1)} d\sqrt{T}\,.$$

Since the distribution of rewards will only influence the KL-divergence by an absolute constant, the lower bound we obtained is similar to that in Lattimore and Szepesvári (2020, Theorem 24.1), which assumes that the reward follows a normal distribution.

According to the proof of Lemma 5.5, the only difference between our hard-to-learn MDP instance and that in Zhou et al. (2021a) is that we need to specify the parameter $\Delta$ as $\Delta = \sqrt{\delta}/\left(\min\{2, e^{\varepsilon}\}(e^{\varepsilon}-1)\sqrt{T}\right)$. We then utilize the hard-to-learn MDPs constructed in Zhou et al. (2021a) and obtain the following lower bound for learning linear mixture MDPs with $\varepsilon$-LDP guarantee:

**Theorem 5.6.** For any $\varepsilon$-LDP algorithm, if the number of interactions with the environment $T$ satisfies that $T \geqslant 4d^2H/\left(\min\{4, e^{2\epsilon}\}(e^{\epsilon}-1)^2\right)$, then there exists a linear mixture MDP parameterized by $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_H)$ such that the expected regret is lower bounded as follows:

$$\mathbb{E}_{\boldsymbol{\Theta}}\text{Regret}(M_{\boldsymbol{\Theta}}, K) \geqslant \Omega\left(\frac{1}{\min\{2, e^{\varepsilon}\}(e^{\varepsilon}-1)} dH\sqrt{T}\right),$$

where $T = KH$ and $\mathbb{E}_{\boldsymbol{\Theta}}$ denotes the expectation over the probability distribution generated by the interaction of the algorithm and the MDP.

**Remark 5.7.** Compared with the upper bound $\widetilde{\mathcal{O}}(d^{5/4}H^{7/4}T^{3/4}(\log(H/\delta))^{1/4}\sqrt{1/\varepsilon})$ in Theorem 5.2, it can be seen that there is a $d^{1/4}T^{1/4}H^{3/4}$ gap between our upper bound and lower bound if treating $\varepsilon$ as a constant. It is unclear if the upper bound and/or the lower bound are not tight.

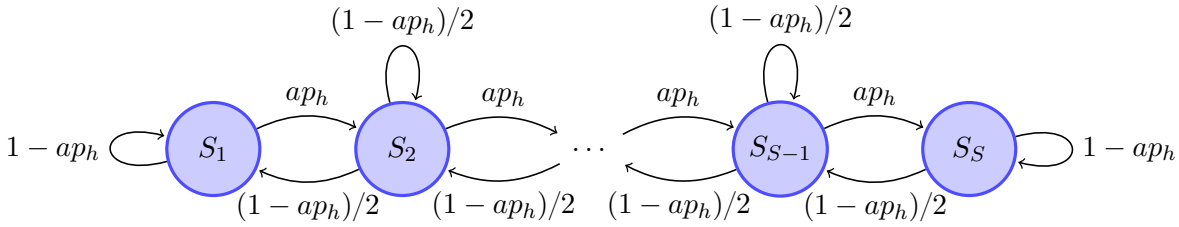


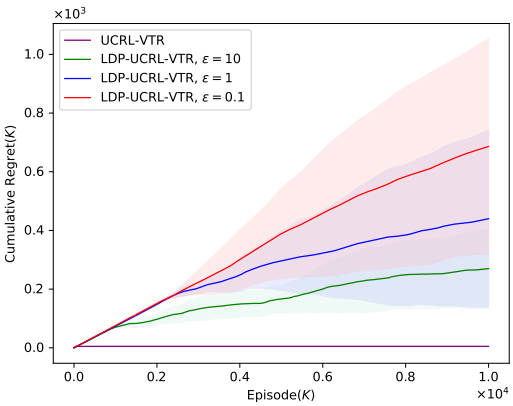

Figure 1: The transition kernel $\mathbb{P}_h$ of inhomogeneous "RiverSwim" MDP instance.

## 6. Experiments

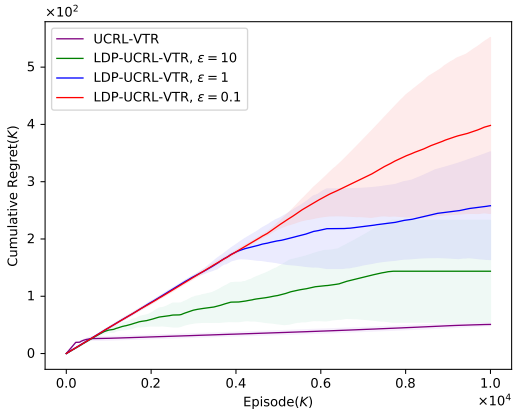
In this section, we carry out experiments to evaluate the performance of LDP-UCRL-VTR, and compare with its non-private counterpart UCRL-VTR (Jia et al., 2020).

### 6.1. Experimental Setting

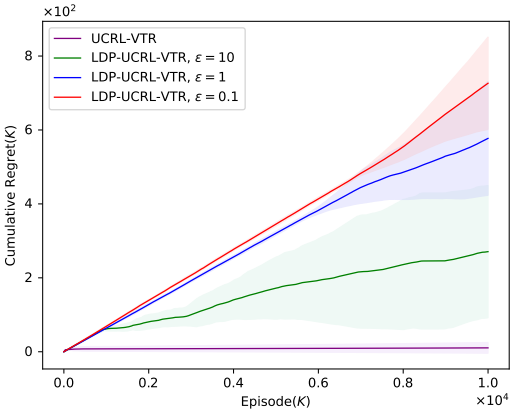
We tested LDP-UCRL-VTR on a benchmark MDP instance named "RiverSwim" (Strehl and Littman, 2008; Ayoub et al., 2020), where we model this instance as a linear mixture model by defining the feature mapping as $\boldsymbol{\phi}(s' \mid s, a) = \mathbf{e}_{s,a,s'}$, which is a one-hot vector with value 1 in the $(s, a, s')$-th entry. The purpose of this MDP is to tempt the agent to go left while it is hard for a short sighted agent to go right since $r(0,0) \neq 0, r(s,1) = 0, 0 \leqslant s \leqslant |\mathcal{S}|-1$. Therefore, it is hard for the agent to decide which direction to choose. In our experiment, the reward in each stage is normalized by $H$, e.g., $r(0,0) = 5/(1000H), r(S,1) = 1/H$. Our LDP-UCRL-VTR is also tested on the time-inhomogeneous "RiverSwim", where for each



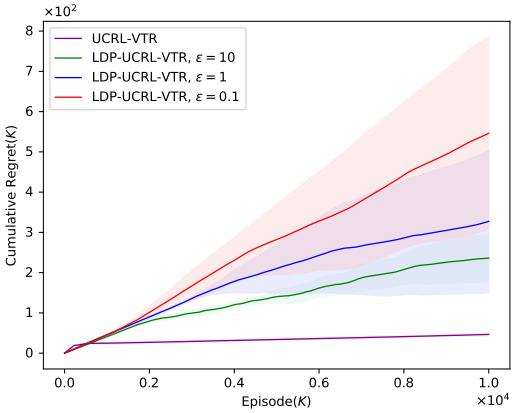

(a) $S = 3, A = 2, H = 6, d = 18$, homogeneous




(b) $S = 5, A = 2, H = 10, d = 50$, homogeneous




(c) $S = 3, A = 2, H = 6, d = 18$, inhomogeneous




(d) $S = 5, A = 2, H = 10, d = 50$, inhomogeneous

Figure 2: Evaluation of the algorithms in two "RiverSwim" MDPs. Results are averaged over 10 runs and the standard deviations are calculated to plot the confidence band. These results show that the cumulative regret of LDP-UCRL-VTR is sublinear in $K$, and its performance is getting closer to that of UCRL-VTR while the privacy guarantee becomes weaker, i.e., choosing a larger $\varepsilon$.

$h \in [H]$, the transition probability $p_h$ is sampled from a uniform distribution $U(0.8, 1)$. We also choose $H = 2S$. Figure 1 shows the state transition graph of this MDP.

### 6.2. Results and Discussion

We evaluate LDP-UCRL-VTR with different privacy budget $\varepsilon$ and compare it with UCRL-VTR on both homogeneous and inhomogeneous "RiverSwim". For UCRL-VTR, we set $\sqrt{\beta} = c\sqrt{d_1} + (H-h+1)\sqrt{2\log(1/K) + \log\det \mathbf{M}}$, where $d_1 = SA$ with $S$ being the number of states and $A$ being the number of actions, $\mathbf{M}$ is the covariance matrix in Algorithm 3 (Jia et al., 2020). We fine tune the hyper parameter $c$ for different experiments. For LDP-

UCRL-VTR, we choose $\delta = 0.1, \alpha = 0.01$. Since $\delta$ and $\alpha$ are prefixed for all experiments, they can be treated as constants. Thus, we can choose $\beta$ in the form $cd^{3/4}(H-h+1)^{3/2}k^{1/4}$ and only fine tune $c$. The results for each $\varepsilon$ are averaged over 10 runs.

In our experiments, since the reward is normalized by $H$, we need to recompute $\sigma$ for the Gaussian mechanism. Recall that $\sigma = 2\Delta f H\sqrt{2\log(2.5H/\delta)}/\varepsilon$, where $\Delta f$ represents the $\ell_2$ sensitivity of $\Delta\mathbf{u}$ in Algorithm 1. In our setting, $|Q| \leqslant 1$ and therefore $\Delta f \leqslant 1$. Thus, we set $\sigma = 4H\sqrt{2\log(2.5H/\delta)}/\varepsilon$. In addition, we set $K = 10000$ for all experiments. To fine tune the hyper parameter $c$, we use grid search and select the one which attains the best result. The experiment results are shown in Figure 2.

From Figure 2, we can see that the cumulative regret of LDP-UCRL-VTR is indeed subliear in $K$. In addition, it is not surprising to see that LDP-UCRL-VTR incurs a larger regret than UCRL-VTR. The performance of LDP-UCRL-VTR with larger $\varepsilon$ is closer to that of UCRL-VTR as the privacy guarantee becomes weaker. Our results are also greatly impacted by $H$ and $d$, as the convergence (learning speed) slows down as we choose larger $H$ and smaller $\varepsilon$. The experiments are consistent with our theoretical results.

## 7. Conclusion and Future Work

In this paper, we studied RL with linear function approximation and LDP guarantee. To the best of our knowledge, our designed algorithm is the first provable privacy-preserving RL algorithm with linear function approximation. We proved that LDP-UCRL-VTR satisfies $(\varepsilon, \delta)$-LDP. We also show that LDP-UCRL-VTR enjoys an $\widetilde{\mathcal{O}}(d^{5/4}H^{7/4}T^{3/4}\left(\log(1/\delta)\right)^{1/4}\sqrt{1/\varepsilon})$ regret. Besides, we proved a lower bound $\Omega(dH\sqrt{T}/\left(e^{\varepsilon}(e^{\varepsilon}-1)\right))$ for $\varepsilon$-LDP linear mixture MDPs. We also provide experiments on synthetic datasets to corroborate our theoretical findings. In our current results, there is still a gap between the regret upper bound and the lower bound. We conjecture the gap to be a fundamental difference between learning linear mixture MDPs and tabular MDPs. In the future, it remains to study if this gap could be eliminated.

## Acknowledgement

We thank the anonymous reviewers for their helpful comments. JH and QG are partially supported by the National Science Foundation CAREER Award 1906169 and the Sloan Research Fellowship. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

## 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 *NIPS*, volume 11, pages 2312–2320, 2011.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

Borja Balle, Maziar Gomrokchi, and Doina Precup. Differentially private policy evaluation. In *International Conference on Machine Learning*, pages 2130–2138. PMLR, 2016.

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

Jean Bretagnolle and Catherine Huber. Estimation des densités: risque minimax. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 47(2):119–137, 1979.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

Xiaoyu Chen, Kai Zheng, Zixin Zhou, Yunchang Yang, Wei Chen, and Liwei Wang. (locally) differentially private combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 1757–1767. PMLR, 2020.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv preprint arXiv:1302.3203*, 2013.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521): 182–201, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

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.

Pratik Gajane, Tanguy Urvoy, and Emilie Kaufmann. Corrupt bandits for preserving local privacy. In *Algorithmic Learning Theory*, pages 387–412. PMLR, 2018.

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

Jiafan He, Dongruo Zhou, and Quanquan Gu. Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pages 4171–4180. PMLR, 2021.

Jiafan He, Dongruo Zhou, and Quanquan Gu. Nearly optimal regret for learning adversarial mdps with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

Zeyu Jia, Lin Yang, Csaba Szepesvari, and Mengdi Wang. Model-based reinforcement learning with value-targeted regression. In *Learning for Dynamics and Control*, pages 666–686. PMLR, 2020.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

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

Paul Luyo, Evrard Garcelon, Alessandro Lazaric, and Matteo Pirotta. Differentially private exploration in reinforcement learning with linear representation. *arXiv preprint arXiv:2112.01585*, 2021.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 2010–2020, 2020.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pages 2256–2264. Citeseer, 2013.

Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. *arXiv preprint arXiv:1810.00068*, 2018.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

Giuseppe Vietri, Borja Balle, Akshay Krishnamurthy, and Steven Wu. Private reinforcement learning with pac and regret guarantees. In *International Conference on Machine Learning*, pages 9754–9764. PMLR, 2020.

Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Yue Wu, Dongruo Zhou, and Quanquan Gu. Nearly minimax optimal regret for learning infinite-horizon average-reward mdps with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019a.

Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019b.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. Locally differentially private (contextual) bandits learning. *arXiv preprint arXiv:2006.00701*, 2020.

Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021a.

Dongruo Zhou, Jiafan He, and Quanquan Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*. PMLR, 2021b.

Xingyu Zhou. Differentially private reinforcement learning with linear function approximation. *arXiv preprint arXiv:2201.07052*, 2022.