Towards Differentially Private Reinforcement Learning with General Function Approximation

Yi He
Wayne State University
yihe@wayne.edu

Xingyu Zhou
Wayne State University
xingyu.zhou@wayne.edu

Abstract

We present the first theoretical guarantees for differentially private online reinforcement learning (RL) with general function approximation, extending beyond prior work restricted to tabular and linear settings. Our approach combines a batched policy update scheme with the exponential mechanism, together with a novel regret analysis. We show that, even under general function approximation, the regret in the model-free setting under differential privacy matches the state of the art for the linear case, scaling as $\widetilde{O}(K^{3/5})$, where K denotes the number of episodes. As an important by-product, we also establish the first regret bound for online RL with batch update that depends on the standard complexity measure of *coverability*, complementing existing results based on a newly introduced Eluder-Condition class. In addition, we uncover fundamental gaps in recent results for private RL with linear function approximation, thereby clarifying its landscape.

1 Introduction

Reinforcement learning (RL) provides a principled framework for learning sequential decision-making strategies in unknown environments through repeated interaction. By observing the consequences of its actions and adjusting its behavior accordingly, an RL agent can progressively improve its performance over time. This paradigm has become a key enabler of modern personalized systems, with successful applications ranging from online recommendation [1] to individualized healthcare interventions [2], social robotics [3], and, more recently, large language models (LLMs) optimized via reinforcement learning from human feedback or verified reward [4, 5]. In these applications, the agent engages with users over multiple rounds, choosing actions based on the current interaction context and receiving feedback—either as per-step process reward or per-trajectory outcome reward—which is then leveraged to refine the agent's decision-making policy.

While highly effective, RL-based personalization inevitably raises privacy concerns, as the feedback collected during interaction often reflects sensitive user information. For example, in personalized medical decision-making, the state may encode private attributes such as a patient's medical history or physiological measurements, while rewards reflect treatment effectiveness or adverse reactions. Similarly, in LLMs, user prompts, preference feedback, and interaction outcomes may reveal personal intentions, beliefs, or proprietary content. Although users may be willing to share such information to obtain better personalized services, they typically do not consent to having their private data inferred by third parties through the outputs or internal updates of the learning algorithm. Indeed, the existing non-private RL approaches have been shown to suffer from privacy leakages [6–8]. Thus, it is essential to develop RL methods that can learn from user interactions while providing formal guarantees on individual privacy.

Motivated by these concerns, a recent line of theoretical work has investigated how to incorporate differential privacy (DP) [9] into RL, with the goal of understanding the fundamental tradeoffs between privacy protection and sample efficiency (e.g., [10–14]). Since standard DP is incompatible

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

with sublinear regret, this literature has largely focused on a relaxed notion named *joint differential privacy* (JDP) [15], under which meaningful guarantees become attainable. They have established sublinear regret bounds for various private RL algorithms, providing the first principled understanding of privacy–utility tradeoffs in RL. While pioneering, existing results are almost exclusively limited to tabular environments or linear function approximation. These structural assumptions are crucial in prior analyses, as they enable private learning through low-dimensional sufficient statistics, simple confidence sets, or carefully controlled policy switching. However, such assumptions substantially limit applicability in modern RL systems, where value functions or policies are typically represented using rich, non-linear function classes. Extending theoretical guarantees beyond the linear regime introduces significant new challenges, as many standard exploration mechanisms, such as confidence-set-based optimism, do not readily admit private implementations in the general function approximation case.

This gap naturally motivates our study of sample-efficient private RL with *general function approximation*. In particular, we ask the following fundamental question:

Can we still enjoy sublinear regret for private RL with general function approximation?

We answer this question in the affirmative by establishing the first sublinear regret bound for RL with general function approximation under JDP privacy constraint. Specifically, our main contributions can be summarized as follows.

1. We study online RL in episodic Markov decision processes (MDPs) with *general function approximation* under formal privacy guarantees via joint differential privacy (JDP). We establish a regret bound of order $K^{3/5}$, where K is the number of episodes, with additional dependence on the *coverability* parameter that characterizes the exploration difficulty. This $K^{3/5}$-regret bound matches the state-of-the-art bound for the linear case in the model-free setting. Our results apply to both general stochastic MDPs (Theorem 3.1) and deterministic MDPs with outcome-based rewards (Theorem 3.4), a modeling framework motivated by LLM applications, where a much weaker representation condition (i.e., realizability) is required.

2. Our results are achieved via a carefully designed algorithmic framework built on several key ingredients (Algorithms 1 and 2). First, instead of the confidence-set-based constrained optimization commonly used for exploration in non-private RL, we adopt an unconstrained optimization objective that explicitly balances exploration and exploitation. This formulation enables the direct application of standard privacy mechanisms, such as the exponential mechanism, to privatize each policy update. Second, to control the cumulative privacy loss incurred by repeated updates, we employ a fixed-size batching strategy with a balanced choice of batch size, updating the policy only at the beginning of each batch.

3. In addition, our analysis yields several by-products of independent interest. First, even without privacy considerations, our regret analysis under fixed-size batching provides new guarantees for RL with adaptivity constraints, where the regret depends directly on *coverability* (cf. (2)), matching the standard exploration complexity measure and avoiding the need for newly introduced conditions in prior work. Second, our generic and modular regret analysis offers a simple recipe for designing private RL algorithms beyond the specific privacy mechanisms used in this paper. Finally, we clarify the state of the art for private RL with linear MDPs by identifying fundamental gaps in recent claims of improved regret bounds.

1.1 Related Work

We mainly discuss the most relevant work here and relegate a detailed discussion to Appendix A. For private RL, Vietri et al. [10], Chowdhury and Zhou [11], and Qiao and Wang [12] established increasingly sharp guarantees in tabular MDPs, while Zhou [13] and Luyo et al. [14] moved to linear function approximation in the model-based and model-free settings, respectively. In the non-private regime, RL with general function approximation has been studied through complexity measures such as coverability [16], as well as through unconstrained optimization frameworks [17, 18] that are particularly relevant to our algorithmic design. A third relevant line studies adaptivity constraints for policy update [19, 20], including both data-independent batching and data-dependent rare policy switching, which will be useful for controlling the privacy composition in online RL.

2 Preliminaries

2.1 RL with General Function Approximation

Markov decision processes (MDPs). We study RL under a finite-horizon episodic Markov decision process specified by a tuple $M = (\mathcal{S}, \mathcal{A}, H, P, R, \rho)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon length H, transition kernel $P = \{P_h\}_{h=1}^H$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R = \{R_h\}_{h=1}^H$, $R_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and initial state distribution $\rho \in \Delta(\mathcal{S})$. A (randomized) policy π consists of H-step functions: $\pi = \{\pi_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^H$. The policy induces a distribution over trajectories $(s_1, a_1, r_1), \ldots, (s_H, a_H, r_H)$ via the following process. The initial state s_1 is first drawn via $s_1 \sim \rho$, and then for each $h \in [H]$, $a_h \sim \pi(\cdot|s_h)$, $r_h = R_h(s_h, a_h)$[1], $s_{h+1} \sim P_h(\cdot|s_h, a_h)$ and s_{H+1} is the terminal state with zero reward. Accordingly, we let $\mathbb{E}_\pi[\cdot]$ and $\mathbb{P}_\pi[\cdot]$ denote expectation and probability under this process, respectively.

The value function and Q-function for a policy π at each step $h \in [H]$ are given by

$$V_h^\pi(s) := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'} \middle| s_h = s \right], \quad Q_h^\pi(s, a) := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'} \middle| s_h = s, a_h = a \right].$$

The total expected reward under a policy π is thus given by

$$J(\pi) := \mathbb{E}_\pi \left[\sum_{h=1}^H r_h \right] = \mathbb{E}_{s_1 \sim \rho} V_1^\pi(s_1).$$

Let $\pi^\star$ denote an optimal policy (i.e., $\pi^\star \in \operatorname{argmax}_\pi J(\pi)$) and $V_h^\star := V_h^{\pi^\star}$, $Q_h^\star = Q_h^{\pi^\star}$ be the corresponding value function and Q-function, which satisfies the well-known Bellman equation for each $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $h \in [H]$:

$$V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s, a), \quad Q_h^\star(s, a) = R_h(s, a) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} V_{h+1}^\star(s').$$

This motivates us to define the so-called Bellman operator $\mathcal{T}_h$ as follows: for any $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, $\mathcal{T}_h f$ is defined as

$$[\mathcal{T}_h f](s, a) := R_h(s, a) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} \max_{a' \in \mathcal{A}} f(s', a').$$

Throughout the paper, as in previous papers (e.g., [16, 18]), we assume that the rewards are normalized such that $\sum_{h=1}^H r_h \in [0, 1]$.

Online RL. We study RL in the online setting where the learner repeatedly interacts with an unknown MDP $M^\star$ via taking a policy for each episode and observing the trajectory, to maximize the total reward. Formally, the learning protocol runs in K episodes, where at each episode $k \in [K]$, the learner (i) selects a policy $\pi_k = \{\pi_{k,h}\}_{h=1}^H$ to run in unknown MDP $M^\star$ and (ii) observes the resultant trajectory $\{(s_h^{(k)}, a_h^{(k)}, r_h^{(k)})\}_{h=1}^H$. The performance of the learner is measured via the regret

$$\mathsf{Reg} := \sum_{k=1}^K J(\pi^\star) - J(\pi_k).$$

General function approximation. To achieve sublinear regret guarantees without dependence on the size of $\mathcal{S}$, we work with model-free value function approximation, where the learner has access to a *general value function class* $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$ that attempts to model the value functions under $M^\star$, with each $\mathcal{F}_h \subseteq (\mathcal{S} \times \mathcal{A} \to [0, 1])$. To derive non-trivial regret bounds for online RL, some structural conditions on the function class $\mathcal{F}$ and its interplay with $M^\star$ are often required. The first set is the so-called *representation conditions*, which require that $\mathcal{F}$ can model some necessary value functions. Two common conditions in this set are *realizability* and *Bellman completeness*.

Assumption 2.1 (Bellman completeness and realizability)**.** For all $h \in [H]$, we have $\mathcal{T}_h f_{h+1} \in \mathcal{F}_h$ for all $f_{h+1} \in \mathcal{F}_{h+1}$. This directly implies realizability that $Q^\star \in \mathcal{F}$.

In contrast to statistical learning, online RL requires another set of conditions, called *exploration conditions*, which roughly limit the amount of exploration to learn the optimal policy. To this end, we adopt the notion of *coverability* [16].

[1] For simplicity, we consider deterministic reward and our result can be easily extended to the random case.

Definition 2.2 (Coverability)**.** The coverability constant $C_{\mathsf{cov}} > 0$ for a given MDP M and policy class Π is

$$C_{\mathsf{cov}}(\Pi; M) := \min_{\mu_1, \cdots, \mu_H \in \Delta(\mathcal{S}\times\mathcal{A})} \max_{h\in[H], \pi\in\Pi} \left\| \frac{d_h^\pi}{\mu_h} \right\|_\infty,$$

where $\left\| \frac{d_h^\pi}{\mu_h} \right\|_\infty := \max_{s\in\mathcal{S}, a\in\mathcal{A}} \frac{d_h^\pi(s,a)}{\mu_h(s,a)}$ and $d_h^\pi(s,a) := \mathbb{P}_\pi[s_h = s, a_h = a]$ is the occupancy measure of π in M. For notation simplicity, we let $C_{\mathsf{cov}} := C_{\mathsf{cov}}(\Pi_{\mathcal{F}}; M^\star)$, where $\Pi_{\mathcal{F}} = \{\pi_f \mid f \in \mathcal{F}\}$ with π_f being the greedy policy with respect to f via $\pi_{f,h}(s) \in \operatorname{argmax}_{a\in\mathcal{A}} f_h(s,a)$.

With Bellman completeness, Xie et al. [16] shows that low coverability is sufficient for sample-efficient online RL in the sense that sublinear regret, i.e., $\widetilde{O}(\sqrt{KC_{\mathsf{cov}}})$ is achieved. Our main goal is to establish the corresponding regret bound under formal privacy guarantees in RL, introduced below.

2.2 Differential Privacy in RL

To adapt standard DP [9] for RL, following prior work [10, 21], the goal here is to roughly ensure that changing one trajectory in the learning process will not impact the outputs (e.g., actions and final policy) too much. More formally, we assume there is a user u_k associated with each episode trajectory (e.g., a multi-step interaction between a user and LLMs) and write $U_K = (u_1, \ldots, u_K) \in \mathcal{U}^K$ to denote a sequence of K unique users in the process, where $\mathcal{U}$ is the set of all users. A k-th neighboring sequence U'_K differs from U_K in only one user at $k \in [K]$, i.e., $u_k \neq u'_k$ and $u_j = u'_j$ for $j \neq k$. The output of a learning mechanism $\mathcal{M}$ on a user sequence U_K is denoted as $\mathcal{M}(U_K)$, which contains all the actions in the K episodes as well as the final policy. With these, a natural adaptation of standard DP is to require that $\mathcal{M}(U_K) \approx_{\varepsilon,\delta} \mathcal{M}(U'_K)$ for any neighboring sequences. However, this is problematic as changing a user u_k will naturally change the corresponding actions recommended to her during episode k (e.g., a different user with a different prompt for LLMs leads to different responses), thus failing to achieve meaningful DP guarantees. In fact, as shown in Shariff and Sheffet [22], even for contextual bandits (i.e., RL with $H = 1$), a linear regret has to be sacrificed for meaningful DP guarantees.

To resolve this conflict, as in prior work on private RL, we adopt the notion of *joint differential privacy* (JDP) [15] from the privacy literature, which slightly relaxes the standard DP by excluding the actions corresponding to the differing user u_k, hence accommodating the intrinsic personalization in RL. Formally, we let $\mathcal{M}_{-k}(U_K) := \mathcal{M}(U_K) \setminus \{a_h^k\}_{h=1}^H$ denote the output obtained by excluding the actions during episode k and define JDP in RL as follows as in Vietri et al. [10], Zhou [21].

Definition 2.3 (JDP in RL)**.** An RL learning algorithm $\mathcal{M}$ is (ε, δ)-jointly differentially private if for all $k \in [K]$, all k-th neighboring user sequences U_K, U'_K, and all events $E \subseteq \mathcal{A}^{H\times[K-1]} \times \Pi$, we have

$$\mathbb{P}\left(\mathcal{M}_{-k}(U_K) \in E\right) \leq e^\varepsilon \cdot \mathbb{P}\left(\mathcal{M}_{-k}(U'_K) \in E\right) + \delta,$$

where Π is the class for the final output policy.

Remark 2.4. We remark that JDP remains a strong privacy guarantee: it protects the information associated with any individual user u_k against arbitrary collusions of other users, provided that u_k does not disclose the actions prescribed to her.

Therefore, our ultimate goal is to design a private online RL algorithm that satisfies JDP while achieving a sublinear regret, which will be the main focus of the next section.

3 Sample Efficient Private Online RL

In this section, we develop model-free private RL algorithms that achieve sublinear regret under the coverability condition.

3.1 General MDPs

We present Algorithm 1 for general MDPs with stochastic transition probability, which is the most standard setup in the non-private case[2]. As mentioned before, we take the unconstrained

[2]As in previous work, we assume that the initial state is fixed.

Algorithm 1 Private RL under General MDPs

Input: Q-function class $\mathcal{F}$, parameter $\eta > 0$, batch size $B \in \mathbb{N}$, privacy parameters ε, δ
Initialize $\mathcal{D} \leftarrow \varnothing$.
for $k = 1, \ldots, K$ **do**
  if $k = i \cdot B + 1$ for some $i \geq 0$ **then**
    Compute the score function for each f using $\mathcal{D}$:

$$S(f) := \eta f_1(s_1) - L_{\mathcal{D}}^{\mathsf{BE}}(f).$$

    Sample $f^{(k)} \in \mathcal{F}$ via the Exponential Mechanism:

$$\mathbb{P}(f) \propto \exp\left(\beta_{\varepsilon,\delta} \cdot S(f)\right).$$

    Update policy $\pi_k \leftarrow \pi_{f^{(k)}}$.
  else
    Reuse: $\pi_k \leftarrow \pi_{k-1}$ and $f^{(k)} \leftarrow f^{(k-1)}$
  end if
  Execute π_k for one episode and collect trajectory: $\tau^{(k)} = \{(s_h^{(k)}, a_h^{(k)}, r_h^{(k)})\}_{h=1}^H$.
  Update dataset: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tau^{(k)}\}$
end for
return $\hat{\pi} = \mathsf{Unif}(\pi_{1:K})$.

optimization rather than the confidence-based constrained optimization. Beyond this, two additional key algorithmic changes due to the privacy constraint are (i) batching policy update and (ii) exponential mechanism, which essentially replaces max with softmax. More specifically, at the beginning of each batch of size B, i.e., $k = i \cdot B + 1$ for some $i \geq 0$, it updates the policy via the following two steps. First, it computes the score function for each $f \in \mathcal{F}$ by combining the Bellman loss $L_{\mathcal{D}}^{\mathsf{BE}}(f)$ with an optimistic term $f_1(s_1) := \max_{a \in \mathcal{A}} f_1(s_1, a)$ (the value of f at the initial state), through a hyper-parameter $\eta > 0$. In particular, the Bellman loss $L_{\mathcal{D}}^{\mathsf{BE}}(f)$ is given by

$$L_{\mathcal{D}}^{\mathsf{BE}}(f) := \sum_{h=1}^{H} \mathcal{E}_{\mathcal{D},h}(f_h, f_{h+1}) - \inf_{f' \in \mathcal{F}} \sum_{h=1}^{H} \mathcal{E}_{\mathcal{D},h}(f'_h, f_{h+1}),$$

where Bellman error $\mathcal{E}_{\mathcal{D},h}(f_h, f_{h+1})$ at each h is given by

$$\sum_{\tau \in \mathcal{D}} \Big(f_h(s_h, a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1}, a') \Big)^2.$$

Note that the subtraction of the inf term in $L_{\mathcal{D}}^{\mathsf{BE}}(f)$ is the standard approach to overcoming the double-sampling issue [23–25]. Leveraging the score function $S(f)$, the second step in the update is to sample $f^{(k)}$ from $\mathcal{F}$ using the exponential mechanism [26], i.e., each f is sampled with probability proportional to $S(f)$ with temperature parameter $\beta_{\varepsilon,\delta}$, which is essentially a softmax rather than the standard max in the non-private case. Then, the updated policy is set to be the greedy policy with respect to $f^{(k)}$. For all other non-update episodes, the policy keeps the same as the previous episode. Finally, it runs the policy π_k in the MDP, receives the new trajectory, consisting of H-step transition information, and updates dataset so far.

Optimistic exploration. One key challenge in online reinforcement learning is strategic exploration. In tabular and linear settings, exploration is typically achieved via upper confidence bound (UCB)–type bonuses (e.g., [27–29]), which are also relatively easy to privatize since the required confidence estimates can be computed from low-dimensional sufficient statistics (e.g., [10, 21, 14]). Under general function approximation, however, a common exploration strategy is to maintain a data-dependent confidence set (or version space) over candidate functions $f \in \mathcal{F}$, as in `GOLF` [25]. Such confidence-set–based approaches are difficult to privatize, as they require repeatedly constructing and updating data-dependent confidence regions. As mentioned before, an alternative is to directly optimize a single objective that balances estimated value and statistical uncertainty, as in `MEX` [17], an idea that dates back to Kumar and Becker [30]. This formulation is much more amenable to differential privacy, as it avoids explicit confidence sets and enables the use of standard private

selection mechanisms such as the exponential mechanism (at least for finite function classes), which is the approach we adopt.

Benefits of batching. The key motivation for batching updates in the private setting is to control the composition of privacy loss arising from repeated operations on the same data. Without batching, data collected in the first episode would participate in all K private updates, whereas batching limits this participation to only $O(K/B)$ updates. With an appropriate choice of the batch size B, this yields a sublinear rather than linear composition of privacy loss. Thus, tuning B plays a central role in our analysis, as it governs the tradeoff between information freshness and cumulative privacy cost.

Unknown number of episodes. When K is unknown, we can use a restart-based doubling schedule with epoch lengths $1, 2, 4, \ldots$. We use a fixed policy in the first epoch and tune B, η, and the other parameters to the planned length of each subsequent epoch. Summing the per-epoch regret bounds preserves the leading $\sqrt{K}$ and $K^{3/5}$ orders. A union bound over epochs adds at most $O(\log \log K)$ to the logarithmic confidence terms. Under our one-user-per-episode model, the epochs use disjoint user data, so parallel composition preserves the overall (ε, δ)-JDP guarantee.

Algorithm 1 enjoys the following regret bound, with its proof given by Appendix F.1.

Theorem 3.1. *Let Assumption 2.1 hold. Fix privacy parameters ε, δ, there exist proper choices of $\beta_{\varepsilon,\delta}$, η and B, such that Algorithm 1 is (ε, δ)-JDP while achieving the following regret bound[3] with probability at least $1 - \alpha$*

$$\mathsf{Reg} \lesssim_\delta H\sqrt{C_{\mathsf{cov}} K \log(|\mathcal{F}|KH/\alpha)\log K} + H\left(C_{\mathsf{cov}} K \log K\right)^{3/5} \cdot \left(\log(|\mathcal{F}|K/\alpha)/\varepsilon\right)^{2/5}.$$

To the best of our knowledge, this is the first regret bound for private online RL. Further, our algorithm is a model-free approach. We also remark that even for a finite function class $\mathcal{F}$, our algorithm and its guarantees are useful in that they allow us to tackle meaningful problems that prior work cannot tackle due to the lack of structure (e.g., our experiment in Appendix E). Moreover, our result can also be extended to the infinite function class, see Appendix C.

Comparison with prior work. The first term in the above bound matches the one in the non-private case (cf. Theorem 1 in Xie et al. [16]) while the second term is additional cost due to privacy, which is on the order of $K^{3/5}$. This privacy cost matches the state-of-the-art regret for RL with linear function approximation in the model-free setting [14], where a linear MDP as in Jin et al. [29] is considered. On the other hand, we note that for a model-based approach under the linear mixture MDPs [28] (a different type of linear function approximation), one can achieve $\sqrt{K}$-regret bound under JDP as in Zhou [21]. The key difference here is that, instead of a $\mathsf{Poly}(K)$ privacy composition in the model-free setting, one can enjoy a total number of only $\log K$ compositions in the model-based linear mixture MDPs, thanks to the applicability of the binary tree mechanism [31] for privately aggregating the sufficient statistics.

Remark 3.2. We remark that some follow-up works [32, 33] claim to achieve an improved $\sqrt{K}$-bound for private RL in the model-free setting under linear MDPs, by reducing the number of compositions to $\log K$ via the technique of rare policy switch [19]. However, as mentioned before, their analysis has gaps, making their results ungrounded, see Section 4.

As in the non-private case, one weakness of Theorem 3.1 is its requirement of Bellman completeness, which is a much stronger condition than realizability. Moreover, the corresponding algorithm (Algorithm 1) requires solving a max-min optimization problem due to the inf operator in the Bellman loss $L_{\mathcal{D}}^{\mathsf{BE}}(f)$. Also, for some interesting applications like LLMs, a general MDP with stochastic transitions is an overkill, as the transition could be a simple deterministic behavior. Finally, for LLMs, the learner typically has access only to an outcome reward, i.e., a single reward over the entire trajectory. All of these facts lead us to the topic of the next section.

3.2 Deterministic MDPs with Outcome Rewards

We show that for deterministic MDPs with outcome rewards (Definition 3.3), one can eliminate both the Bellman completeness condition and the max-min optimization, while still achieving $K^{3/5}$-regret bound under JDP. This can be viewed as the private counterpart of Chen et al. [18].

[3] We use $\lesssim_\delta$ to ignore constant and $\log(1/\delta)$ privacy term for simplicity. See appendix for the detailed bound.

Definition 3.3 (Deterministic MDPs with outcome rewards)**.** An MDP is said to be deterministic if its transition kernel P is deterministic, i.e., for any $h \in [H]$ and $s_h \in \mathcal{S}$, $a_h \in \mathcal{A}$, there is a unique state $s_{h+1} \in \mathcal{S}$ such that $P_h(s_{h+1}|s_h, a_h) = 1$. An outcome reward is a single reward signal $r = \sum_{h=1}^{H} R_h(s_h, a_h)$ for a trajectory.

A key consequence of deterministic transitions is that the Bellman equation holds almost surely, rather than only in expectation. Specifically, for any (s, a) in step h

$$Q_h^\star(s, a) = R_h(s, a) + V_{h+1}^\star(s_{h+1}).$$

This stands in sharp contrast to the stochastic setting, where the Bellman equation involves an expectation over next states. Hence, for any trajectory τ, it holds that

$$r(\tau) = \sum_{h=1}^{H} R_h(s_h, a_h) = \sum_{h=1}^{H} \left[Q_h^\star(s_h, a_h) - V_{h+1}^\star(s_{h+1}) \right].$$

For any $f \in \mathcal{F}$, let $f_h(s) := \max_{a \in \mathcal{A}} f_h(s, a)$ and $f_{H+1} \equiv 0$. We define the outcome reward model $R^{(f)} : (\mathcal{S} \times \mathcal{A})^H \to \mathbb{R}$ by

$$R^{(f)}(\tau) := \sum_{h=1}^{H} \left[f_h(s_h, a_h) - f_{h+1}(s_{h+1}) \right].$$

Motivated by the above observation, we replace the Bellman error loss used in general MDPs with the Bellman residual loss. Given a dataset $\mathcal{D}$ consisting of collected trajectories, the loss is defined as

$$L_{\mathcal{D}}^{\mathsf{BR}}(f) := \sum_{(\tau, r) \in \mathcal{D}} \left(\sum_{h=1}^{H} \left[f_h(s_h, a_h) - f_{h+1}(s_{h+1}) \right] - r \right)^2.$$

Unlike the stochastic case, this loss does not require subtracting an infimum over the function class to address the double-sampling issue. At each batch update, we combine this loss with the distributional optimistic value

$$\bar{f}_1 := \mathbb{E}_{s_1 \sim \rho} \max_{a \in \mathcal{A}} f_1(s_1, a),$$

where ρ is known as part of the MDP specification. This choice aligns the update rule with the regret objective $J(\pi) = \mathbb{E}_{s_1 \sim \rho} V^\pi(s_1)$ and avoids making the score depend on a private sampled initial state.

Algorithm. Compared with Algorithm 1, Algorithm 2 (see Appendix B) has only minor modifications: it replaces the Bellman-error loss $L_{\mathcal{D}}^{\mathsf{BE}}(f)$ with the outcome Bellman-residual loss (i.e., $L_{\mathcal{D}}^{\mathsf{BR}}(f)$ above) computed from trajectory-level rewards, and uses the distributional optimistic term $\bar{f}_1$ in the score function. Consequently, it avoids the infimum correction in $L_{\mathcal{D}}^{\mathsf{BE}}(f)$, and thus removes the max-min optimization while requiring only realizability rather than Bellman completeness. Algorithm 2 enjoys the following regret bound, which scales with the following coverability

$$C'_{\mathsf{cov}} = \mathbb{E}_{s_1 \sim \rho} C_{\mathsf{cov}}(\Pi_{\mathcal{F}}; M_{s_1}^\star),$$

where $M_{s_1}^\star$ is the MDP with deterministic initial state s_1 and the same transition dynamics as $M^\star$. In general, C'_{cov} is always an upper bound on the coverability C_{cov}. The detailed proof is given by Appendix F.2.

Theorem 3.4. *Suppose that realizability holds (s.t. $Q^\star \in \mathcal{F}$), and the initial-state distribution ρ is known as part of the public MDP specification. Fix privacy parameters ε, δ, there exist proper choices of $\beta_{\varepsilon,\delta}$, η and B, such that Algorithm 2 is (ε, δ)-JDP while achieving the following regret bound with probability at least $1 - \alpha$*

$$\mathsf{Reg} \lesssim_\delta H^2 \sqrt{C'_{\mathsf{cov}} K \log(|\mathcal{F}|K/\alpha) \log K} + H^{7/5} \left(C'_{\mathsf{cov}} K \log K \right)^{3/5} \left(\log(|\mathcal{F}|K/\alpha)/\varepsilon \right)^{2/5}$$

The first term matches the one in Chen et al. [18] while the second term is the privacy cost of order $K^{3/5}$.

Remark 3.5. We remark that stochastic initial states are essential in the deterministic setting considered here. With a fixed initial state and deterministic transitions, the interaction can become degenerate; using $s_1 \sim \rho$ leads to a meaningful regret objective and is consistent with the distributional optimistic term used by Algorithm 2.

Remark 3.6 (Unknown initial-state distribution). When ρ is unknown, we can estimate $\bar{f}_1$ from the observed initial states. We use a fixed policy from $\Pi_{\mathcal{F}}$ for the first batch. At batch update $i \geq 1$, we average $\max_{a \in \mathcal{A}} f_1(s_1, a)$ over the iB initial states observed so far. For finite $\mathcal{F}$, uniform concentration gives an additional regret of $\widetilde{O}(B + \sqrt{K})$, including the first batch. The empirical optimism term increases the score sensitivity by at most $\eta/(iB) \leq \eta/B$. Using a public upper bound on C'_{cov} in the original tuning gives $\eta/B = \widetilde{O}(1)$ in K. Recalibrating the exponential mechanism preserves JDP and the same order of regret in K.

Remark 3.7. To illustrate our algorithms and show privacy-utility tradeoff, we include a small proof-of-concept simulation in Appendix E. The experiment is intentionally minimal and is not meant as a comprehensive empirical study; rather, it demonstrates in a controlled deterministic outcome-reward setting that batching and privacy noise introduce the expected slowdown, while larger privacy parameters lead to faster stabilization. We note that even for this simple MDP, prior work on tabular or linear private RL cannot be applied, demonstrating the usefulness of our algorithm.

3.3 Proof Sketch

We present a proof sketch for Theorem 3.1 while also mentioning the key difference in the proof for Theorem 3.4.

Privacy analysis. By the *Billboard Lemma* [34], for (ε, δ)-JDP guarantees, it suffices to show that the computation of the sequence of $f^{(k)}$ satisfies (ε, δ)-DP in the standard sense. Since there is a total of $M = K/B$ updates, by the advanced composition theorem [35], for a final (ε, δ)-DP, it suffices to ensure that each update is $(\varepsilon_0, 0)$-DP with $\varepsilon_0 = \varepsilon/(2\sqrt{2M \log(1/\delta)})$. Thus, setting $\beta_{\varepsilon,\delta} = \frac{\varepsilon_0}{2\Delta_S}$ in the exponential mechanism gives the result, where $\Delta_S = O(H)$ is the sensitivity of our score function $S(f)$.

Regret analysis. We start with the regret decomposition (for the fixed initial state assumed for Theorem 3.1).

$$\mathsf{Reg} = \underbrace{\sum_{k=1}^{K} V_1^\star(s_1) - f_1^{(k)}(s_1)}_{\mathsf{T}_1} + \underbrace{\sum_{k=1}^{K} f_1^{(k)}(s_1) - V_1^{\pi_k}(s_1)}_{\mathsf{T}_2}.$$

To bound T_1, we first *suppose that $f^{(k)}$ is obtained via max rather than softmax*. Then, we have

$$\eta f_1^{(k)}(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) \geq \eta V_1^\star(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star),$$

where we let $[k]$ denote the starting point of the batch that k belongs to and let $\mathcal{D}_{[k]}$ be the dataset at the start of episode $[k]$. That is, if $i \cdot B + 1 \leq k < (i+1) \cdot B + 1$ for some $i \geq 0$, then $[k] = i \cdot B + 1$ and $\mathcal{D}_{[k]}$ contains all episodes from 1 to $i \cdot B$. The above inequality holds directly by the batching schedule and the updated rule, provided $f^{(k)}$ is the maximum of the score function. However, due to the softmax of the exponential mechanism, we only have approximate maximization, thus giving that w.p. $1 - \alpha$,

$$\eta f_1^{(k)}(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) \geq \eta V_1^\star(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star) - \mathsf{Err}_{\mathsf{priv}},$$

where $\mathsf{Err}_{\mathsf{priv}}$ is the privacy cost, and in particular for the exponential mechanism $\mathsf{Err}_{\mathsf{priv}} = O(1/\beta_{\varepsilon,\delta} \log(|\mathcal{F}|K/\alpha))$. Thus, re-arranging the above inequality, we bound T_1 as

$$\mathsf{T}_1 \leq \sum_{k=1}^{K} \frac{1}{\eta} \cdot \left(L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) + \mathsf{Err}_{\mathsf{priv}} \right).$$

Further, by the standard uniform convergence bound (cf. Jin et al. [25]), we can bound the empirical difference by the expectation of the Bellman error plus some statistical error, under the Bellman completeness assumption (Assumption 2.1). With this result, we have the final high probability bound for T_1 as

$$\mathsf{T}_1 \lesssim \frac{1}{\eta}\left(K \cdot \mathsf{Err}_{\mathsf{stat}} + K \cdot \mathsf{Err}_{\mathsf{priv}}\right) - \frac{1}{\eta}\left(\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{t<[k]} \mathbb{E}_{\pi_t}[e_h^{(k)}(s_h, a_h)^2]\right), \tag{1}$$

where $\mathsf{Err}_{\mathsf{stat}}$ is the statistical error, which can be bounded by $H \log(|\mathcal{F}|KH/\alpha)$ for a finite class $\mathcal{F}$ and $e_h^{(k)}(s,a) := f_h^{(k)}(s,a) - [\mathcal{T}_h f_{h+1}^{(k)}(s,a)]$.

We now turn to T_2, i.e., the on-policy error. First, by the standard performance difference lemma [36], we have that

$$\mathsf{T}_2 = \sum_{h=1}^{H}\sum_{k=1}^{K} \mathbb{E}_{\pi_k}[e_h^{(k)}(s_h, a_h)].$$

The ultimate goal is to relate such "test" error above to the "training error", i.e., (1), through the notion of coverability. This is the key challenge in our analysis due to batching, where the standard result (e.g., [18]) cannot be applied. To this end, we derive a new inequality for such a key step in the batching setting. Specifically, we establish that for any $\lambda \geq 1$

$$\sum_{k=1}^{K} \mathbb{E}_{\pi_k}\left|e_h^{(k)}(s_h, a_h)\right| \lesssim C_{\mathsf{cov}} \cdot B \cdot \log K + \sqrt{C_{\mathsf{cov}} \log K \cdot \left(K\lambda + \sum_{k=1}^{K}\sum_{t<[k]} \mathbb{E}_{\pi_t}[e_h^{(k)}(s_h, a_h)^2]\right)},$$

where the first term is the cost due to batch update, and the second term is related to the negative term in T_1 (cf. (1)), allowing for the cancellation effect.

Finally, by Cauchy-Schwarz inequality, plugging the above bound into T_2, and merging with T_1, we have

$$\mathsf{Reg} \lesssim \underbrace{(1/\eta) \cdot K \cdot \mathsf{Err}_{\mathsf{stat}} + \eta \cdot H \cdot C_{\mathsf{cov}} \log K}_{\text{(a)}} + \underbrace{H \cdot C_{\mathsf{cov}} \cdot B \cdot \log K}_{\text{(b)}} + \underbrace{(1/\eta) \cdot K \cdot \mathsf{Err}_{\mathsf{priv}}}_{\text{(c)}} \tag{2}$$

where (a) is the same term as in the non-private case, (b) is the cost due to batching and (c) is the cost due to privacy, which depends on B. Thus, tuning B and η optimally gives us the final regret bound.

Remark 3.8. With the above final generic regret bound, one can see that without privacy, by setting $B = O(\sqrt{K})$, the regret bound still maintains as $\sqrt{K}$ with dependence on the coverability. Hence, as a by-product, our result also gives a $\sqrt{K}$ regret with $\sqrt{K}$ update frequency under the coverability, in contrast to the new notion of Eluder-Condition class in Xiong et al. [20].

Remark 3.9 (Difference in the proof of Theorem 3.4). The proof essentially follows from the same flow as in the general MDPs for Theorem 3.1. The key technical difference is a new inequality to relate the on-policy "test error" to the "training" error, by fully utilizing the deterministic nature, which is our key technique contribution. As in the general MDPs, we believe that this inequality can find broad applications. Another difference is the treatment of the initial state: the algorithm uses the distributional optimistic term $\bar{f}_1 = \mathbb{E}_{s_1 \sim \rho} \max_a f_1(s_1, a)$ rather than the value at a sampled initial state, and the proof conditions on a fixed s_1 before averaging over $s_1 \sim \rho$. Finally, the Bellman-error loss is replaced by the outcome Bellman-residual loss $L_{\mathcal{D}}^{\mathsf{BR}}(f)$, so the proof requires only realizability rather than Bellman completeness, while the score sensitivity becomes $O(H^2)$.

4 Concluding Remarks

General MDPs with outcome rewards. In this setting, we can easily extend our algorithm and analysis to derive a $K^{3/5}$-regret bound under JDP. In particular, following the non-private case [18], we add another loss for reward model estimation to our Algorithm 1 and modify the way of data collection with a helper reference policy. These changes will not impact the final regret order. In addition, if $\mathcal{F}$ is parametric with parameter $\theta \in \mathbb{R}^d$, one can potentially bound the term $\mathsf{Err}_{\mathsf{priv}}$ via DP-ERM for the empirical loss $F_{\mathcal{D}}(\theta) := \eta f_{\theta,1}(s_1) - L_{\mathcal{D}}^{\mathsf{BE}}(f_\theta)$, thus bridging with the literature on private (minimax) optimization (e.g., [37–39]).

Coverability vs Eluder condition. Instead of using coverability, another related approach is based on Bellman-Eluder dimension, such as the Eluder-Condition class of Xiong et al. [20], which provides a

broad framework beyond several standard structural models. However, Bellman-Eluder-dimension-based analyses can be suboptimal in some cases, yielding a $T^{2/3}$ regret rate even when a $\sqrt{T}$ bound is achievable under coverability, see Proposition 11 in Xie et al. [16]. Going beyond coverability may require modifying the SEC of Xie et al. [16], in a way similar to how Xiong et al. [20] modified the Bellman-Eluder dimension; we leave this as an interesting open problem.

Infinite function class. Our general regret bound in Section 3.3 provides us with some guidelines for dealing with an infinite function class $\mathcal{F}$. Specifically, we only need to focus on the two error terms $\mathsf{Err}_{\mathsf{stat}}$ and $\mathsf{Err}_{\mathsf{priv}}$. For $\mathsf{Err}_{\mathsf{stat}}$, one can follow the standard approach in the non-private case to adopt a covering number argument, replacing the $|\mathcal{F}|$ by the covering number (with some approximation error). For $\mathsf{Err}_{\mathsf{priv}}$, one can also potentially *explicitly* construct a covering set in the algorithm and apply the exponential mechanism on the finite covering set. Due to the page limit, we give a formal guarantee in Appendix C for more details.

Alternative private optimizers. Our regret analysis only requires that the output $f^{(k_j)}$ of the private optimizer at the start k_j of batch j satisfies the following inequality for all batches with high probability:

$$S_{\mathcal{D}_{[k_j]}}\left(f^{(k_j)}\right) \geq S_{\mathcal{D}_{[k_j]}}\left(Q^\star\right) - \xi_j.$$

The corresponding contribution to regret in (2) is at most $(B/\eta)\sum_j \xi_j$. Provided that the private updates compose to (ε, δ)-DP, the same Billboard Lemma argument yields JDP. For a parametric function class f_θ, one can use DP-SGD at each batch update and view the score-gap bound as a utility guarantee for private empirical risk minimization. A sufficient guarantee is that the returned parameter $\widehat{\theta}_j$ satisfies $S_{\mathcal{D}_{[k_j]}}\left(f_{\widehat{\theta}_j}\right) \geq \sup_\theta S_{\mathcal{D}_{[k_j]}}\left(f_\theta\right) - \xi_j$. The key challenge is to control this global objective-value gap, since typical guarantees for nonconvex DP-SGD control stationarity. Algorithm 2 is particularly suitable for this direction because its objective is single-level.

Gaps in private RL with linear MDPs. In both works [32, 33], the authors adopt the idea of the rare policy switch via the standard determinant trick argument over the design matrix. This is a *data-dependent adaptive* update rather than the fixed-size batching in our paper, hence enabling only a total of $\log K$ updates, which further reduces the privacy composition. However, due to its data-dependent nature, one also has to privatize the design matrix. This leads to the gap in their analysis, as the standard condition for the determinant trick fails for a private design matrix. In particular, Ngo et al. [32] incorrectly handles monotonicity in both Lemma 31, concerning log-order switches, and Equation 20, when applying the standard determinant trick [19]. These errors subsequently propagate to the follow-up work by Sahu [33], see Appendix D for more detail.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science Foundation (NSF) under Grant Nos. CAREER-2441519, CNS-2312835.

References

[1] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[2] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[3] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. Affective personalization of a social robot tutor for children's second language skills. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[4] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[5] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[6] Xinlei Pan, Weiyao Wang, Xiaoshuai Zhang, Bo Li, Jinfeng Yi, and Dawn Song. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In *Aamas*, volume 19, pages 368–376, 2019.

[7] Qizhang Feng, Siva Rajesh Kasa, Santhosh Kumar Kasa, Hyokun Yun, Choon Hui Teo, and Sravan Babu Bodapati. Exposing privacy gaps: Membership inference attack on preference data for llm alignment. *arXiv preprint arXiv:2407.06443*, 2024.

[8] Yule Liu, Heyi Zhang, Jinyi Zheng, Zhen Sun, Zifan Peng, Tianshuo Cong, Yilong Yang, Xinlei He, and Zhuo Ma. Grpo privacy is at risk: A membership inference attack against reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.14045*, 2025.

[9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[10] 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.

[11] Sayak Ray Chowdhury and Xingyu Zhou. Differentially private regret minimization in episodic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6375–6383, 2022.

[12] Dan Qiao and Yu-Xiang Wang. Near-optimal differentially private reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 9914–9940. PMLR, 2023.

[13] Xingyu Zhou. Differentially private reinforcement learning with linear function approximation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):1–27, 2022.

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

[15] Michael Kearns, Mallesh Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 403–410, 2014.

[16] Tengyang Xie, Dylan J. Foster, Yu Bai, Nan Jiang, and Sham M. Kakade. The role of coverage in online reinforcement learning, 2022. URL https://arxiv.org/abs/2210.04157.

[17] Zhihan Liu, Miao Lu, Wei Xiong, Han Zhong, Hao Hu, Shenao Zhang, Sirui Zheng, Zhuoran Yang, and Zhaoran Wang. Maximize to explore: One objective function fusing estimation, planning, and exploration. *Advances in Neural Information Processing Systems*, 36:22151–22165, 2023.

[18] Fan Chen, Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Outcome-based online reinforcement learning: Algorithms and fundamental limits, 2025. URL https://arxiv.org/abs/2505.20268.

[19] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

[20] Nuoya Xiong, Zhaoran Wang, and Zhuoran Yang. A general framework for sequential decision-making under adaptivity constraints. *arXiv preprint arXiv:2306.14468*, 2023.

[21] Xingyu Zhou. Differentially private reinforcement learning with linear function approximation, 2022. URL https://arxiv.org/abs/2201.07052.

[22] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 4296–4306, 2018.

[23] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

[24] 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.

[25] Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 34:13406–13418, 2021.

[26] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[27] 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.

[28] 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.

[29] 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.

[30] P. Kumar and A. Becker. A new family of optimal adaptive controllers for markov chains. *IEEE Transactions on Automatic Control*, 27(1):137–146, 1982. doi: 10.1109/TAC.1982.1102878.

[31] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

[32] Dung Daniel T Ngo, Giuseppe Vietri, and Steven Wu. Improved regret for differentially private exploration in linear mdp. In *International Conference on Machine Learning*, pages 16529–16552. PMLR, 2022.

[33] Sharan Sahu. Towards optimal differentially private regret bounds in linear mdps. *arXiv preprint arXiv:2504.09339*, 2025.

[34] Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. *SIAM Journal on Computing*, 45(6):1953–1984, January 2016. ISSN 1095-7111. doi: 10.1137/15100271x. URL http://dx.doi.org/10.1137/15100271X.

[35] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

[36] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

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

[38] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[39] Liang Zhang, Kiran K Thekumparampil, Sewoong Oh, and Niao He. Bring your own algorithm for optimal differentially private stochastic minimax optimization. *Advances in Neural Information Processing Systems*, 35:35174–35187, 2022.

[40] Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differential privacy for regret minimization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10561–10573, 2021.

[41] Chonghua Liao, Jiafan He, and Quanquan Gu. Locally differentially private reinforcement learning for linear mixture markov decision processes. In *Asian Conference on Machine Learning*, pages 627–642. PMLR, 2023.

[42] Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

[43] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, volume 26, pages 2256–2264, 2013.

[44] Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, volume 27, pages 1466–1474, 2014.

[45] 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.

[46] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, pages 2898–2933. PMLR, 2019.

[47] Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

[48] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

[49] Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond. *arXiv preprint arXiv:2211.01962*, 2022.

[50] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

[51] Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose Blanchet, Peter W Glynn, and Yinyu Ye. Sequential batch learning in finite-action linear contextual bandits. *arXiv preprint arXiv:2004.06321*, 2020.

[52] Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.

[53] Tianhao Wang, Dongruo Zhou, and Quanquan Gu. Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. *Advances in Neural Information Processing Systems*, 34:13524–13536, 2021.

[54] Dingwen Kong, Ruslan Salakhutdinov, Ruosong Wang, and Lin F Yang. Online sub-sampling for reinforcement learning with general function approximation. *arXiv preprint arXiv:2106.07203*, 2021.

[55] Dan Qiao, Ming Yin, and Yu-Xiang Wang. Logarithmic switching cost in reinforcement learning beyond linear mdps. *arXiv preprint arXiv:2302.12456*, 2023.

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

[57] Xingyu Zhou. When determinants are not enough: Private rare switching. *arXiv preprint arXiv:2605.23131*, 2026.

[58] Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002. URL https://api.semanticscholar.org/CorpusID:31442909.

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

Contents of Appendix

A Related Work **15**

B Private RL under Deterministic MDPs **16**

C Detailed Discussion for Infinite Function Class **16**

D Gaps in Private RL with Linear MDPs **17**

E Experiments **18**

F Proof of Section 3 **20**

F.1 Proof of Theorem 3.1 20

F.2 Proof of Theorem 3.4 24

G Limitations **30**

A Related Work

Private RL. Under the central trust model of joint differential privacy (JDP), the first theoretical guarantees for private reinforcement learning were established by Vietri et al. [10] in the tabular setting, via privatized `UCB-VI`–type algorithms (e.g., [27]). Subsequently, Chowdhury and Zhou [11] improved the regret bounds and introduced the first policy-based (as opposed to value-based) private tabular RL algorithm. The current state of the art in the tabular regime is due to Qiao and Wang [12], which leverages Bernstein-type exploration bonuses. Moving beyond tabular environments, existing theoretical results for private RL are largely restricted to linear function approximation. In particular, Zhou [13] obtained $\sqrt{K}$ regret bounds for private RL with linear function approximation in the model-based setting via linear mixture MDPs [28], while Luyo et al. [14] provided the first model-free result for linear MDPs [29], achieving a regret bound of order $K^{3/5}$. Motivated by this separation, several recent works [32, 33] revisit the model-free setting and claim to close the gap by deriving $\sqrt{K}$ regret bounds. However, we identify a common and fundamental gap in their analyses (cf. Section 4), rendering these improved bounds ungrounded. As a result, for private RL with linear function approximation, the best known regret guarantee in the model-free setting remains $K^{3/5}$. We also note in passing that a parallel line of work studies private RL under the stronger local differential privacy model—without a trusted central learner—which is beyond the scope of this paper [40, 11, 41].

RL with general function approximation. In the non-private setting, substantial recent progress has been made in understanding exploration complexity beyond linear function approximation. Prominent complexity measures include the Eluder dimension [42–44], the Bellman-Eluder dimension [25], Bellman rank [45], witness rank [46], bilinear classes [47], and decision-estimation coefficients (DEC) [48]. Among these, Xie et al. [16] introduced the notion of *coverability* and showed that it suffices to guarantee sample-efficient online RL with general function approximation. On the algorithmic side, while most prior works rely on constrained optimization over confidence sets, Liu et al. [17] proposed an unconstrained optimization framework and established sublinear regret guarantees in terms of the generalized Eluder coefficient (GEC) [49]. This complexity measure is closely related to the sequential extrapolation coefficient (SEC), introduced by Xie et al. [16] as a generalization of coverability. While these works typically assume access to per-step process rewards, Chen et al. [18] recently took the first step toward studying RL with general function approximation under outcome rewards, relying on a similar unconstrained optimization approach in Liu et al. [17] with additional algorithmic changes for outcome reward.

RL with adaptivity constraints. This line of work studies RL under restrictions on how frequently the policy can be updated, motivated by practical concerns such as policy switching costs. A

simple and data-independent approach is fixed-size batching, where the policy is updated only at the beginning of each batch. This setting has been extensively studied in bandits (e.g., [50, 51]) and later extended to RL with linear function approximation [42]. More recently, Xiong et al. [20] analyzed batching under general function approximation and established sublinear regret guarantees for a newly introduced class of MDPs termed the Eluder-Condition. An alternative approach is data-dependent rare policy switching, dating back to Abbasi-Yadkori et al. [19] in the bandit setting. Compared to fixed-size batching, rare policy switching typically enables $\sqrt{K}$ regret with only $O(\log K)$ policy updates, rather than $O(\sqrt{K})$. Such results have been obtained for tabular [52], linear [53], and general function approximation settings [54, 55, 20]. Both batching and rare policy switching have also been employed in private bandits and private RL to control privacy loss composition. For example, Chowdhury and Zhou [56] applies batching to study shuffle privacy in linear contextual bandits, and Luyo et al. [14] used it for private RL with linear function approximation. Meanwhile, rare policy switching with $O(\log K)$ updates is the primary technique used in [32, 33] to claim $\sqrt{K}$ regret bounds for private RL; however, due to additional privacy noise, these analyses are flawed, as discussed in detail in Section 4.

B Private RL under Deterministic MDPs

Algorithm 2 Private RL under Deterministic MDPs

Input: Q-function class $\mathcal{F}$, parameter $\eta > 0$, batch size $B \in \mathbb{N}$, privacy parameters ε, δ
Initialize $\mathcal{D} \leftarrow \varnothing$.
for $k = 1, \ldots, K$ **do**
  if $k = i \cdot B + 1$ for some $i \geq 0$ **then**
    Compute the score function for each f using $\mathcal{D}$

$$S(f) := \eta \bar{f}_1 - L_{\mathcal{D}}^{\mathsf{BR}}(f), \text{ where } \bar{f}_1 := \mathbb{E}_{s_1 \sim \rho} \max_{a \in \mathcal{A}} f_1(s_1, a).$$

    Sample $f^{(k)} \in \mathcal{F}$ with the Exponential Mechanism:

$$\mathbb{P}(f) \propto \exp\left(\beta_{\varepsilon,\delta} \cdot S(f)\right).$$

    Update policy $\pi_k \leftarrow \pi_{f^{(k)}}$.
  else
    Reuse: $\pi_k \leftarrow \pi_{k-1}$ and $f^{(k)} \leftarrow f^{(k-1)}$
  end if
  Execute π_k for one episode and collect trajectory: $\tau^{(k)} = \{(s_h^{(k)}, a_h^{(k)})\}_{h=1}^H$ and outcome reward $r^{(k)}$.
  Update dataset: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\tau^{(k)}, r^{(k)})\}$
end for
return: $\hat{\pi} \leftarrow \mathsf{Unif}(\pi_{1:K})$.

C Detailed Discussion for Infinite Function Class

To handle an infinite function class, we define the covering number as follows:

Definition C.1. Let $\mathcal{F} \subseteq \{f = (f_1, \ldots, f_H) : f_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$, and define

$$\|f - g\|_\infty := \max_{h \in [H]} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |f_h(s,a) - g_h(s,a)|.$$

For $\beta \geq 0$, a set $\mathcal{C}_\beta \subseteq \mathcal{F}$ is a β-cover of $\mathcal{F}$, if for every $f \in \mathcal{F}$, there exists $\tilde{f} \in \mathcal{C}_\beta$ such that $\|f - \tilde{f}\|_\infty \leq \beta$. The β-covering number is

$$\mathcal{N}(\mathcal{F}, \beta) := \min\{|\mathcal{C}_\beta| : \mathcal{C}_\beta \text{ is a } \beta\text{-cover of } \mathcal{F}\}.$$

Algorithm. To handle an infinite function class, we only need to change the **input** of Algorithm 1, specifically, fix $\beta > 0$, explicitly construct a finite β-cover $\mathcal{C}_\beta \subseteq \mathcal{F}$ and replace $\mathcal{F}$ by $\mathcal{C}_\beta$ in

Algorithm 1. Hence, in each update episode $k = iB + 1$, we compute scores on this finite $\mathcal{C}_\beta$ instead of infinite $\mathcal{F}$, which allows us to run the exponential mechanism.

Thus, we can derive a formal corollary:

Corollary C.2 (Infinite class reduction via explicit covering)**.** *Assume the same conditions as Theorem 3.1, except that $\mathcal{F}$ can be infinite. Fix $\beta > 0$ and run the above revised algorithm with an explicit β-cover $\mathcal{C}_\beta$, there exist proper choices of $\beta_{\varepsilon,\delta}$, η and B, such that the revised algorithm is (ε, δ)-JDP while achieving the following regret bound with probability at least $1 - \alpha$*

$$\mathsf{Err}_{\mathsf{stat}} \lesssim \log\left(\frac{\mathcal{N}(\mathcal{F}, \beta)KH}{\alpha}\right) \quad \mathsf{Err}_{\mathsf{priv}} \lesssim \log\left(\frac{\mathcal{N}(\mathcal{F}, \beta)K}{\alpha}\right)$$

Consequently,

$$\begin{aligned}\mathsf{Reg} \lesssim_\delta\ & H\sqrt{C_{\mathsf{cov}} K \log\left(\frac{\mathcal{N}(\mathcal{F}, \beta)KH}{\alpha}\right) \log K} \\ & + H(C_{\mathsf{cov}} K \log K)^{3/5} \cdot \left(\frac{\log(\mathcal{N}(\mathcal{F}, \beta)K/\alpha)}{\varepsilon}\right)^{2/5} + \beta K.\end{aligned}$$

Remark C.3. The first two terms are exactly the finite-class bound with $|\mathcal{F}|$ replaced by $\mathcal{N}(\mathcal{F}, \beta)$, and βK is the additional approximation error over K episodes from restricting the Exponential Mechanism to sample exclusively from $\mathcal{C}_\beta$.

D Gaps in Private RL with Linear MDPs

In this section, we further clarify the gaps in prior work. Yet, before that, we note that a recent work [57] has proposed a fix[4], i.e., a new private rare-switching method different from prior work, which can be extended to address the gap in private linear MDPs.

As discussed in Section 4, because rare policy switching is data-dependent, applying the standard determinant-trick argument requires privatizing the design matrix. This introduces Gaussian privacy noise, which is not a non-negative rank-one update. Consequently, the monotonicity property need not hold, making the follow-up arguments ungrounded.

Monotonicity of the design matrix is required for two claims: (a) *log-order switches* (Lemma 31 in Ngo et al. [32], key for privacy accounting), and (b) *norm factor bound* (Eq. 20 in Ngo et al. [32]), namely that the norms induced by two design matrices are within a constant factor of each other.

For (a), we dive into Lemma 31 in Ngo et al. [32], where the determinant doubling trick is adopted. In particular, without monotonicity, the following step in their proof would fail:

$$\frac{\prod_{h=1}^{H} \det(\tilde{\Lambda}_h^{k_i})}{\prod_{h=1}^{H} \det(\tilde{\Lambda}_h^{k_{i-1}})} = \frac{\det(\tilde{\Lambda}_{h^\star}^{k_i})}{\det(\tilde{\Lambda}_{h^\star}^{k_{i-1}})} \times \prod_{h \neq h^*} \frac{\det(\tilde{\Lambda}_h^{k_i})}{\det(\tilde{\Lambda}_h^{k_{i-1}})} > 2$$

To see this, note that the first term is greater than 2 by definition of the switching design (triggered by a particular $h^\star \in [H]$, see line 6 of Algorithm 1 in Ngo et al. [32]), but the second term may be less than 1 if the design matrix is not monotone. This indeed can happen because their Algorithm 1 will add fresh privacy noise (via Binary tree mechanism) to each design matrix at each $h \in [H]$ (see line 20 of Algorithm 1 in Ngo et al. [32]), which can make the private design matrix non-monotone, even though it is still PSD.

For (b), the norm factor bound (Eq. 20 in Ngo et al. [32]) fails because the standard determinant trick (Lemma 12 in Abbasi-Yadkori et al. [19]) strictly requires monotonicity. Their analysis relies on the following inequality holding for the current private matrix $\tilde{\mathbf{\Lambda}}_h^k$ and the last-updated matrix $\tilde{\mathbf{\Lambda}}_h^{\tilde{k}}$:

$$(\phi_h^k)^\top \left(\tilde{\mathbf{\Lambda}}_h^{\tilde{k}}\right)^{-1} \phi_h^k \leq 2(\phi_h^k)^\top \left(\tilde{\mathbf{\Lambda}}_h^{k}\right)^{-1} \phi_h^k$$

[4]This work is online after our submission.

However, as noted in (a), Algorithm 1 (line 20) in Ngo et al. [32] injects fluctuating noise via the Binary tree mechanism at every episode. This inherently breaks monotonicity ($\tilde{\mathbf{\Lambda}}_h^k \not\succeq \tilde{\mathbf{\Lambda}}_h^{\tilde{k}}$), meaning $\tilde{\mathbf{\Lambda}}_h^k$ can shrink along certain feature directions. Without monotonicity, the inverse matrix relationship breaks down. Instead of being bounded by a constant factor of 2, the ratio can blow up exponentially in dimension d, fundamentally invalidating the claimed $\tilde{O}(\sqrt{K})$ regret bound.

E Experiments

While our main focus is to provide theoretical understandings of DP-RL, rather than extensive empirical studies, we conducted a small **proof-of-concept** simulation in a deterministic outcome-reward setting designed to mirror the structure of Algorithm 2 while remaining simple enough to implement exactly. All experiments were run on a standard CPU machine and took less than a few minutes; no GPU was used.

Environment construction. We consider a finite-horizon deterministic episodic MDP with horizon $H = 4$, setting a binary action space $\mathcal{A} = \{0, 1\}$. At the beginning of each episode, a binary context

$$x = (x_1, \ldots, x_6) \in \{0, 1\}^6$$

is sampled uniformly at random. Thus, the state at stage h is $s_h = (x, a_1, \ldots, a_{h-1}, h)$, so the transition is deterministic, since the next state is obtained by appending the chosen action.

There is no intermediate reward; the environment provides only a terminal outcome reward $r \in \{0, 1\}$ at the end of the episode, which is determined by a hidden target hypothesis.

An episode receives a reward of 1 if the full action sequence matches the target sequence prescribed by the hidden hypothesis for the realized context, and 0 otherwise. Thus, this is a deterministic outcome-reward MDP.

Finite nonlinear hypothesis class. We use a finite nonlinear hypothesis class with $|\mathcal{F}| = 243$. Each hypothesis consists of a binary gate $g(x) \in \{0, 1\}$, chosen from

$$\begin{aligned} g_0(x) &= 1, \\ g_1(x) &= x_5 \oplus x_6, \\ g_2(x) &= x_1 \oplus x_3 \oplus x_5, \end{aligned}$$

and one **stagewise action rule** for each stage $h = 1, 2, 3, 4$, chosen from

$$\begin{aligned} u^{(0)}(x, \text{prefix}) &= x_1 \oplus x_2, \\ u^{(1)}(x, \text{prefix}) &= x_3 \oplus a_{h-1} \quad (a_0 := 0), \\ u^{(2)}(x, \text{prefix}) &= \text{parity}(a_1, \ldots, a_{h-1}) \oplus x_4. \end{aligned}$$

Hence, each hypothesis defines a nonlinear, context-dependent, and history-dependent target action sequence.

Easy and hard instances. We report results on two instances. In the **easy environment**, the hidden target gate is $g_0(x) = 1$, and the stage rules are $(u^{(0)}, u^{(1)}, u^{(0)}, u^{(1)})$. In this case, every context is rewardable, so positive outcomes are relatively dense.

In the **hard environment**, the hidden target gate is $g_2(x) = x_1 \oplus x_3 \oplus x_5$, and the stage rules are $(u^{(2)}, u^{(1)}, u^{(2)}, u^{(1)})$. In this case, only a subset of contexts are rewardable, and the correct sequence depends more strongly on action-history parity, making learning more difficult.

Methods compared. We compare four methods:

- `non_private_no_batch`, a non-private method updated after every episode;
- `non_private_batched`, a non-private method restricted to batched updates;
- `private_eps_8`, a private batched selection method with privacy/noise parameter $\varepsilon = 8$;
- `private_eps_5`, a private batched selection method with privacy/noise parameter $\varepsilon = 5$.

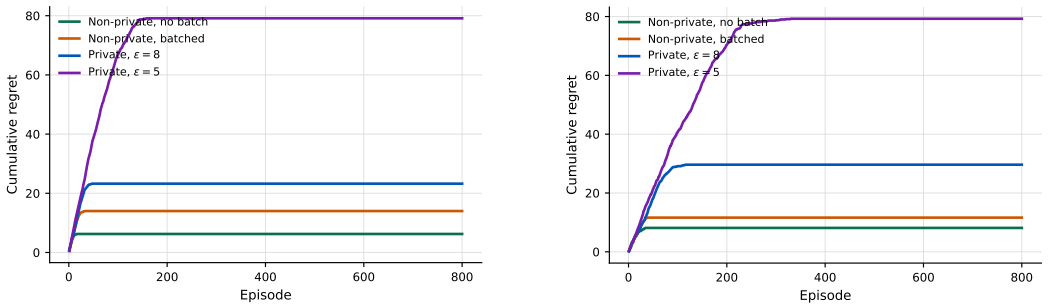
For the private methods, we use the same batched update structure as in the proof-of-concept implementation and replace exact maximization with exponential-mechanism-style sampling over the finite hypothesis class. The batch size is selected according to the theory.

Reporting metric: plateau episode. We summarize each regret curve by its **plateau episode**. We define the plateau episode as the first episode at which **95% of the final cumulative regret** has already been incurred. This gives a robust scalar summary of how quickly the regret curve enters its flat regime.

Plateau results. The resulting plateau episodes are reported in Table 1, and Figure 1 plots the cumulative regret curves for the easy and hard proof-of-concept environments. The curves show the same qualitative pattern as the plateau summary: non-private learning reaches the stable regime fastest, batching introduces a moderate delay, and stronger privacy noise leads to slower stabilization.

Environment	non-private no-batch	non-private batched	private $\varepsilon = 8$	private $\varepsilon = 5$
Easy	13	24	37	127
Hard	31	34	87	218

Table 1: Plateau episode for each method and environment.



(a) Easy environment

(b) Hard environment

Figure 1: Cumulative regret curves in the proof-of-concept deterministic outcome-reward environments.

Interpretation. These results show the expected trend clearly. In both environments, the non-private methods plateau fastest. Among the private methods, the larger privacy parameter, $\varepsilon = 8$, reaches the plateau regime substantially earlier than $\varepsilon = 5$. The hard environment requires more episodes than the easy environment across all methods. In particular, in the easy environment, the plateau time increases from 37 episodes at $\varepsilon = 8$ to 127 episodes at $\varepsilon = 5$. In the hard environment, it increases from 87 episodes at $\varepsilon = 8$ to 218 episodes at $\varepsilon = 5$.

Thus, even in this minimal proof-of-concept simulation, weaker privacy constraints lead to faster convergence, while batching introduces a visible but smaller additional slowdown compared with the non-batched non-private baseline.

Summary. This experiment is intentionally small-scale, but it directly validates the qualitative privacy-utility tradeoff emphasized in the paper: as privacy becomes stronger, more episodes are required to reach the stable learning regime. It also shows that the deterministic outcome-reward setting with general nonlinear finite function approximation already exhibits the expected separation among non-private, batched non-private, and private learning.

F Proof of Section 3

F.1 Proof of Theorem 3.1

Proof. Denote $V^\star(s_1) = Q_1^\star(s_1) := \max_{a\in\mathcal{A}} Q^\star(s_1,a)$, by the Exponential Mechanism, with probability at least $1-\alpha$, we have $S(f^{(k)}) \geq \max_{f\in\mathcal{F}} S(f) - \frac{2\Delta_S}{\varepsilon_0}\log(|\mathcal{F}|K/\alpha)$, which implies

$$\eta f_1^{(k)}(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) \geq \eta Q_1^\star(s_1) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star) - \frac{2\Delta_S}{\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

Dividing both sides by η and rearranging terms yields

$$V^\star(s_1) \leq f_1^{(k)}(s_1) - \frac{1}{\eta}\left(L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star)\right) + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

By subtracting $V^{\pi_k}(s_1)$ from both sides, we obtain

$$V^\star(s_1) - V^{\pi_k}(s_1) \leq f_1^{(k)}(s_1) - V^{\pi_k}(s_1) - \frac{1}{\eta}\left(L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star)\right) + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

For the term $f_1^{(k)}(s_1) - V^{\pi_k}(s_1)$, given by the standard performance difference lemma [58], we have

$$f_1^{(k)}(s_1) - V^{\pi_k}(s_1) = \sum_{h=1}^H \mathbb{E}_{\pi_k}\left[f_h^{(k)}(s_h,a_h) - [\mathcal{T}_h f_{h+1}^{(k)}](s_h,a_h)\right].$$

Thus, denote $e_h^{(k)}(s,a) = f_h^{(k)}(s,a) - [\mathcal{T}_h f_{h+1}^{(k)}](s,a)$, it holds that for $k\in[K]$,

$$V^\star(s_1) - V^{\pi_k}(s_1) \leq \sum_{h=1}^H \mathbb{E}_{\pi_k}[e_h^{(k)}(s_h,a_h)] - \frac{L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BE}}_{\mathcal{D}_{[k]}}(Q^\star)}{\eta} + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

Then, under the success event of Proposition 11 in Chen et al. [18], we have

$$V^\star(s_1) - V^{\pi_k}(s_1) - \frac{2H\kappa}{\eta} \leq \sum_{h=1}^H \left(\mathbb{E}_{\pi_k}[e_h^{(k)}(s_h,a_h)] - \frac{1}{2\eta}\sum_{t<[k]}\mathbb{E}_{\pi_t} e_h^{(k)}(s_h,a_h)^2\right) + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha),$$

where $\kappa = \log\frac{|\mathcal{F}|}{\alpha} + \log\frac{H}{\alpha}$.

Lemma F.1. *Suppose that $p_1,\ldots,p_K$ is a sequence of distributions over $\mathcal{X}$ and there exists $\mu\in\Delta(\mathcal{X})$ such that $\frac{p_k(x)}{\mu(x)} \leq C_{\mathsf{cov}}$ for all $x\in\mathcal{X}$ and $k\in[K]$. Then, for any sequence $f^{(1)},\ldots,f^{(K)}$ of functions $\mathcal{X}\to[0,1]$ and any constant $\lambda\geq 1$, it holds that*

$$\sum_{k=1}^K \mathbb{E}_{x\sim p_k} f^{(k)}(x) \leq C_{\mathsf{cov}} B\log_2\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right) + \sqrt{4C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right)\left[2K\lambda + \sum_{k=1}^K\sum_{t<[k]}\mathbb{E}_{x\sim p_t} f^{(k)}(x)^2\right]}.$$

Consider $\mathcal{X} = \mathcal{S}\times\mathcal{A}$, and we define $p_k := \mathbb{P}^{\pi_k}((s,a)=\cdot) \in (\mathcal{S}\times\mathcal{A}\to[0,1])$, assume that $\frac{p_k(s,a)}{\mu(s,a)} \leq C_{\mathsf{cov}}$, then simply apply the Lemma F.1, we derive the following bound

$$\sum_{k=1}^K \mathbb{E}_{\pi_k}\left|e_h^{(k)}(s_h,a_h)\right| \leq C_{\mathsf{cov}} B\log_2\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right) + \sqrt{4C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right)\left[2K\lambda + \sum_{k=1}^K\sum_{t<[k]}\mathbb{E}_{\pi_t} e_h^{(k)}(s_h,a_h)^2\right]}.$$

Using Cauchy-Schwarz in the form $xy \leq \frac{\eta}{2}x^2 + \frac{1}{2\eta}y^2$, we have

$$\begin{aligned}\sum_{k=1}^K \mathbb{E}_{\pi_k}\left|e_h^{(k)}(s_h,a_h)\right| &\leq C_{\mathsf{cov}} B\log_2\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right) \\ &\quad + 2\eta C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right) + \frac{1}{2\eta}\left[2K\lambda + \sum_{k=1}^K\sum_{t<[k]}\mathbb{E}_{\pi_t} e_h^{(k)}(s_h,a_h)^2\right].\end{aligned}$$

Therefore, combining the inequalities above, set $\kappa = \lambda$ and take summation over $k \in [K]$ gives

$$\sum_{k=1}^{K} (V^\star(s_1) - V^{\pi_k}(s_1)) \le \frac{3KH\kappa}{\eta} + \frac{2K\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha) + 2\eta H C_{\mathsf{cov}} \log\left(1 + \frac{C_{\mathsf{cov}}K}{\kappa}\right) + HC_{\mathsf{cov}}B\log_2\left(1 + \frac{C_{\mathsf{cov}}K}{\kappa}\right).$$

For the non-private case, we set

$$\frac{3KH\kappa}{\eta} = C\eta \implies \eta = \sqrt{\frac{3KH\kappa}{C}}$$

where $C = HC_{\mathsf{cov}}\log\left(1 + \frac{C_{\mathsf{cov}}K}{\kappa}\right)$. Substituting back, the non-private bound is $\frac{3KH\kappa}{\eta} + C\eta = 2\sqrt{3KH\kappa C}$.

For the private case, we write

$$\frac{2K\Delta_S\log(|\mathcal{F}|K/\alpha)}{\eta\varepsilon_0} = C \cdot \eta.$$

Lemma F.2 (Advanced Sequential Composition [35])**.** *Suppose $D = \cup_{i=1}^{k} D_i$ and each $\mathcal{M}_i$ is ε-DP, then M is (ε', δ')-DP for any $\delta' \ge 0$ with*

$$\varepsilon' = \varepsilon\sqrt{2k\ln(1/\delta')} + k\varepsilon(e^\varepsilon - 1)$$

For the privacy parameter ε_0, utilizing the *Advanced Sequential Composition* (Lemma F.2), we have:

$$\varepsilon = \sqrt{2\frac{K}{B}\log(1/\delta)} \cdot \varepsilon_0 \implies \varepsilon_0 = \frac{\varepsilon}{\sqrt{2\frac{K}{B}\log(1/\delta)}},$$

note that our Exponential Mechanism always provides pure-DP $(\varepsilon_0, 0)$-DP, the δ here is given by Advanced Sequential Composition.

Substituting ε_0 back into the expression for η, we obtain

$$\eta = \sqrt{\frac{2K\Delta_S\log(|\mathcal{F}|K/\alpha)}{C\varepsilon_0}} = \sqrt{\frac{2K\Delta_S\log(|\mathcal{F}|K/\alpha)}{C \cdot \frac{\varepsilon\sqrt{B}}{\sqrt{2K\log(1/\delta)}}}} = O\left(\frac{K^{3/4}\Delta_S^{1/2}(\log(|\mathcal{F}|K/\alpha))^{1/2}(\log(1/\delta))^{1/4}}{B^{1/4}\varepsilon^{1/2}C^{1/2}}\right).$$

Finally, substituting back to the regret bound, we balance these two terms:

$$O\left(\frac{K^{3/4}C^{1/2}\Delta_S^{1/2}(\log(|\mathcal{F}|K/\alpha))^{1/2}(\log(1/\delta))^{1/4}}{B^{1/4}\varepsilon^{1/2}}\right) = O(B \cdot C),$$

Thus we solve for $B = \frac{K^{3/5}\Delta_S^{2/5}(\log(|\mathcal{F}|K/\alpha))^{2/5}(\log(1/\delta))^{1/5}}{\varepsilon^{2/5}C^{2/5}}$.

We now turn to bound the sensitivity Δ_S, note that we have $S(f) := \eta f_1(s_1) - L_{\mathcal{D}}^{\mathsf{BE}}(f)$. Since the first term $\eta f_1(s_1)$ does not depend on $\mathcal{D}$, so the sensitivity is entirely from $L_{\mathcal{D}}^{\mathsf{BE}}$.

Recall the definition of $L_{\mathcal{D}}^{\mathsf{BE}}$:

$$L_{\mathcal{D}}^{\mathsf{BE}}(f) := \sum_{h=1}^{H} \mathcal{E}_{\mathcal{D},h}(f_h, f_{h+1}) - \inf_{f' \in \mathcal{F}} \sum_{h=1}^{H} \mathcal{E}_{\mathcal{D},h}(f'_h, f_{h+1}),$$

where Bellman error $\mathcal{E}_{\mathcal{D},h}(f_h, f_{h+1})$ at each h is given by

$$\sum_{\tau \in \mathcal{D}} \left(f_h(s_h, a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1}, a')\right)^2.$$

Since the rewards are normalized ($\sum_{h=1}^{H} r_h \in [0,1]$), then

$$\Big(f_h(s_h,a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1},a')\Big) \in [-2,1]$$

thus changing one trajectory changes $\sum_{h=1}^{H} \mathcal{E}_{\mathcal{D},h}(f_h, f_{h+1})$ by at most $4H$. For the second term, since for a fixed f', the sensitivity has the same bound of $4H$, and the $\inf_{f'}$ cannot increase the sensitivity. Therefore, $\Delta_S \le 8H$.

We now introduce the so-called *billboard lemma*.

Lemma F.3 (Billboard Lemma [34])**.** *Suppose $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is (ε,δ)-differentially private. Consider any set of functions $f_i : \mathcal{D}_i \times \mathcal{R} \to \mathcal{R}'$, where $\mathcal{D}_i$ is the portion of the database containing i's data. The composition $\{f_i(\Pi_i\mathcal{D}, \mathcal{M}(\mathcal{D}))\}$ is ε,δ-joint differentially private, where $\Pi_i : \mathcal{D} \to \mathcal{D}_i$ is the projection to i's data.*

In each episode k, the agent releases a function $f^{(k)}$ selected via the Exponential Mechanism with scoring function $S(f) = \eta f_1(s_1) - L_{\mathcal{D}}^{\mathsf{BE}}(f)$. Given the sensitivity $\Delta_S \le 8H$ and the application of advanced sequential composition, the sequence of functions $\{f^{(k)}\}_{k=1}^{K}$ is (ε,δ)-DP. The action a_h^k recommended to the k-th user is a function of the DP-protected $f^{(k)}$ and the user's private state s_h^k. Since the actions sent to all other users $j \neq k$ depend on $\mathcal{D}_k$ only through the (ε,δ)-DP billboard $f^{(k)}$, the mechanism is (ε,δ)-joint differentially private.

Setting

$$\eta = \max\left\{\sqrt{\frac{3KH\kappa}{C}}, \frac{K^{3/4}\Delta_S^{1/2}(\log(|\mathcal{F}|K/\alpha))^{1/2}(\log(1/\delta))^{1/4}}{B^{1/4}\varepsilon^{1/2}C^{1/2}}\right\}$$
$$B = \frac{K^{3/5}\Delta_S^{2/5}(\log(|\mathcal{F}|K/\alpha))^{2/5}(\log(1/\delta))^{1/5}}{\varepsilon^{2/5}C^{2/5}},$$

we have the final bound

$$\sum_{k=1}^{K}(V^{\star}(s_1) - V^{\pi_k}(s_1)) \le 2\sqrt{3KH\kappa C} + \frac{K^{3/5}C^{3/5}\Delta_S^{2/5}(\log(|\mathcal{F}|K/\alpha))^{2/5}(\log(1/\delta))^{1/5}}{\varepsilon^{2/5}},$$

where $C = HC_{\mathsf{cov}}\log\left(1 + \frac{C_{\mathsf{cov}}K}{\kappa}\right)$, $\kappa = \log\frac{|\mathcal{F}|}{\alpha} + \log\frac{H}{\alpha}$, $\Delta_S \le 8H$.

□

Proof of Lemma F.1. Fix any $x \in \mathcal{X}$, we divide the M batches into two disjoint sets of "good" and "bad" batches, i.e., $[M] = \mathcal{G}_x \cup \mathcal{B}_x$, and $\mathcal{G}_x \cap \mathcal{B}_x = \emptyset$, where both sets depend on the current x. Let $t_m := (m-1)B + 1$ denote the first episode of batch m, with the convention $t_{M+1} := K+1$. To define the "bad" batch, we let $P_m(x) := \lambda\mu(x) + \sum_{t \le mB} p_t(x)$ with $P_0(x) := \lambda\mu(x)$. Then we define

$$\mathcal{B}_x = \{m \in [M] : P_m(x) \ge 2P_{m-1}(x)\}$$

We now write the summation $\sum_{k=1}^{K}\mathbb{E}_{x\sim p_k} f^{(k)}(x)$ as follows.

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{E}_{x\sim p_k} f^{(k)}(x) &= \sum_{x\in\mathcal{X}}\sum_{k=1}^{K} p_k(x) f^{(k)}(x) \\ &= \underbrace{\sum_{x\in\mathcal{X}}\sum_{m\in\mathcal{B}_x}\sum_{k=t_m}^{t_{m+1}-1} p_k(x) f^{(k)}(x)}_{\mathcal{T}_{\mathsf{bad}}} + \underbrace{\sum_{x\in\mathcal{X}}\sum_{m\in\mathcal{G}_x}\sum_{k=t_m}^{t_{m+1}-1} p_k(x) f^{(k)}(x)}_{\mathcal{T}_{\mathsf{good}}}. \end{aligned} \tag{3}$$

For $\mathcal{T}_{\mathsf{bad}}$, we first bound the size of $\mathcal{B}_x$ for any x as follows.

$$|\mathcal{B}_x|\log 2 \le \sum_{m\in[M]} \log\left(\frac{P_m(x)}{P_{m-1}(x)}\right) = \log\left(\frac{\lambda\mu(x) + \sum_{t=1}^{K} p_t(x)}{\lambda\mu(x)}\right) \le \log\left(1 + \frac{C_{\mathsf{cov}}K}{\lambda}\right). \tag{4}$$

Hence, we have for any $x \in \mathcal{X}$,

$$|\mathcal{B}_x| \leq \log_2(1 + \frac{C_{\mathsf{cov}} K}{\lambda}).$$

With this result, we can turn to bound $\mathcal{T}_{\mathsf{bad}}$ below

$$\mathcal{T}_{\mathsf{bad}} \overset{(a)}{\leq} \sum_{x\in\mathcal{X}} \sum_{m\in\mathcal{B}_x} \sum_{k=t_m}^{t_{m+1}-1} p_k(x) \overset{(b)}{\leq} \sum_{x\in\mathcal{X}} \sum_{m\in\mathcal{B}_x} \sum_{k=t_m}^{t_{m+1}-1} C_{\mathsf{cov}}\mu(x) \overset{(c)}{\leq} C_{\mathsf{cov}} \cdot B \cdot \log_2\left(1 + \frac{C_{\mathsf{cov}} K}{\lambda}\right),$$

where (a) holds by the boundedness of $f^{(k)}$; (b) follows from the definition of C_{cov}; (c) is true by (4).

Now, it remains to bound $\mathcal{T}_{\mathsf{good}}$. To this end, we first define $\mathcal{G}_x^k$ to be all the time slots that belong to a good batch. Moreover, for any $x \in \mathcal{X}$ and $k \in [K]$, define

$$\tilde{p}_k(x) := \lambda\mu(x) + \sum_{t<[k]} p_t(x)$$
$$\hat{p}_k(x) := \lambda\mu(x) + \sum_{t\leq k} p_t(x)$$

Thus, we have

$$\begin{aligned}
\mathcal{T}_{\mathsf{good}} &= \sum_{x\in\mathcal{X}} \sum_{k\in\mathcal{G}_x^k} p_k(x) f^{(k)}(x) \\
&= \sum_{x\in\mathcal{X}} \sum_{k\in\mathcal{G}_x^k} \frac{p_k(x)}{\sqrt{\tilde{p}_k(x)}} \cdot f^{(k)}(x)\sqrt{\tilde{p}_k(x)} \\
&\overset{(a)}{\leq} \sum_{x\in\mathcal{X}} \sum_{k\in\mathcal{G}_x^k} \frac{\sqrt{2p_k(x)}}{\sqrt{\hat{p}_k(x)}} \cdot f^{(k)}(x)\sqrt{\tilde{p}_k(x)} \\
&\overset{(b)}{\leq} \sum_{x\in\mathcal{X}} \sum_{k\in[K]} \frac{\sqrt{2p_k(x)}}{\sqrt{\hat{p}_k(x)}} \cdot f^{(k)}(x)\sqrt{\tilde{p}_k(x)} \\
&\overset{(c)}{\leq} \sqrt{2} \sqrt{\underbrace{\sum_{k\in[K]} \sum_{x\in\mathcal{X}} \frac{p_k(x)^2}{\hat{p}_k(x)}}_{\mathcal{T}_1}} \cdot \sqrt{\underbrace{\sum_{k\in[K]} \sum_{x\in\mathcal{X}} \tilde{p}_k(x) f^{(k)}(x)^2}_{\mathcal{T}_2}},
\end{aligned}$$

where (a) holds by the fact that for any k in a good batch under x, $\hat{p}_k(x) \leq 2\tilde{p}_k(x)$ as a result of the definition of the good batch (3); (b) holds by adding more positive terms; (c) holds by Cauchy–Schwarz inequality.

For $\mathcal{T}_2$, by the definition of $\tilde{p}_k(x)$ and boundedness of $f^{(k)}$, we have

$$\begin{aligned}
\sum_{k=1}^{K} \sum_{x\in\mathcal{X}} \tilde{p}_k(x) f^{(k)}(x)^2 &\leq \sum_{k=1}^{K} (\lambda + 1) + \sum_{k=1}^{K} \sum_{t<[k]} \mathbb{E}_{x\sim p_t} f^{(k)}(x)^2 \\
&\leq 2K\lambda + \sum_{k=1}^{K} \sum_{t<[k]} \mathbb{E}_{x\sim p_t} f^{(k)}(x)^2.
\end{aligned}$$

For $\mathcal{T}_1$, we note that

$$\begin{aligned}
\mathcal{T}_1 = \sum_{x\in\mathcal{X}}\sum_{k=1}^{K}\frac{p_k(x)^2}{\hat{p}_k(x)} &\le \sum_{x\in\mathcal{X}} C_{\mathsf{cov}}\mu(x)\sum_{k=1}^{K}\frac{p_k(x)}{\hat{p}_k(x)} \\
&\overset{(a)}{\le} 2\sum_{x\in\mathcal{X}} C_{\mathsf{cov}}\mu(x)\sum_{k=1}^{K}\log\left(1+\frac{p_k(x)}{\hat{p}_k(x)}\right) \\
&\le 2\sum_{x\in\mathcal{X}} C_{\mathsf{cov}}\mu(x)\sum_{k=1}^{K}\log\left(1+\frac{p_k(x)}{\lambda\mu(x)+\sum_{t<k}p_t(x)}\right) \\
&= 2\sum_{x\in\mathcal{X}} C_{\mathsf{cov}}\mu(x)\log\left(\frac{\lambda\mu(x)+\sum_{t\le K}p_t(x)}{\lambda\mu(x)}\right) \\
&\le 2C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right),
\end{aligned}$$

where (a) holds by the fact that $u \le 2\log(1+u)$ for any $u \in [0,1]$. Thus, with results for $\mathcal{T}_1$ and $\mathcal{T}_2$, we have the following bound for $\mathcal{T}_{\mathsf{good}}$

$$\mathcal{T}_{\mathsf{good}} \le \sqrt{4C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right)\cdot\left[2K\lambda+\sum_{k=1}^{K}\sum_{t<[k]}\mathbb{E}_{x\sim p_t}f^{(k)}(x)^2\right]}.$$

Finally, combining the bounds for $\mathcal{T}_{\mathsf{bad}}$ and $\mathcal{T}_{\mathsf{good}}$, we have the final bound

$$\sum_{k=1}^{K}\mathbb{E}_{x\sim p_k}f^{(k)}(x) \le C_{\mathsf{cov}}B\log_2\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right)+\sqrt{4C_{\mathsf{cov}}\log\left(1+\frac{C_{\mathsf{cov}}K}{\lambda}\right)\cdot\left[2K\lambda+\sum_{k=1}^{K}\sum_{t<[k]}\mathbb{E}_{x\sim p_t}f^{(k)}(x)^2\right]}$$

□

F.2 Proof of Theorem 3.4

Proof. For any $f \in \mathcal{F}$, define

$$\bar{f}_1 := \mathbb{E}_{s\sim\rho}\max_a f_1(s,a).$$

In Algorithm 2, the score function is

$$S(f) := \eta\bar{f}_1 - L^{\mathsf{BR}}_{\mathcal{D}}(f).$$

Let $[k]$ denote the beginning of the batch containing episode k, and let $\mathcal{D}_{[k]}$ be the dataset available at that time. By the exponential mechanism and a union bound over all batch updates, with probability at least $1-\alpha$, for every episode k,

$$\eta\bar{f}_1^{(k)} - L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(f^{(k)}) \ge \eta\bar{Q}_1^\star - L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(Q^\star) - \frac{2\Delta_S}{\varepsilon_0}\log(|\mathcal{F}|K/\alpha),$$

where $\bar{Q}_1^\star := \mathbb{E}_{s_1\sim\rho}\max_{a\in\mathcal{A}}Q_1^\star(s_1,a) = J^\star$.

Rearranging, we have

$$J^\star \le \bar{f}_1^{(k)} - \frac{1}{\eta}\left(L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(Q^\star)\right) + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

Thus we subtracting $J(\pi_k) = \mathbb{E}_{s_1\sim\rho}V^{\pi_k}(s_1)$ and obtain

$$J^\star - J(\pi_k) \le \bar{f}_1^{(k)} - J(\pi_k) - \frac{1}{\eta}\left(L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(Q^\star)\right) + \frac{2\Delta_S}{\eta\varepsilon_0}\log(|\mathcal{F}|K/\alpha).$$

Since π_k is greedy with respect to $f^{(k)}$, we have

$$\begin{aligned}\bar{f}_1^{(k)} - J(\pi_k) &= \mathbb{E}_{s_1\sim\rho,\pi_k}\left[\sum_{h=1}^{H}\left(f_h^{(k)}(s_h,a_h) - f_{h+1}^{(k)}(s_{h+1}) - R_h(s_h,a_h)\right)\right] \\ &= \mathbb{E}_{s_1\sim\rho,\pi_k}\left[R^{(f^{(k)})}(\tau) - r(\tau)\right].\end{aligned}$$

Denote $e^{(k)}(\tau) := R^{(f^{(k)})}(\tau) - r(\tau)$, we have

$$\bar{f}_1^{(k)} - J(\pi_k) = \mathbb{E}_{s_1\sim\rho,\pi_k}\left[e^{(k)}(\tau)\right].$$

Recall that

$$L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(f) := \sum_{t<[k]}\left(R^{(f)}(\tau^{(t)}) - r^{(t)}\right)^2,$$

where $(\tau^{(t)}, r^{(t)})$ denotes the data collected in episode t.

By the uniform convergence of the square loss [18, Proposition 19] there exists an event of probability at least $1-\alpha$ such that, for all k,

$$\frac{1}{2}\sum_{t<[k]}\mathbb{E}_{s_1\sim\rho,\pi_t}\left[e^{(k)}(\tau)^2\right] \le L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(f^{(k)}) - L^{\mathsf{BR}}_{\mathcal{D}_{[k]}}(Q^\star) + \kappa, \tag{5}$$

where $\kappa = O\left(H^3\log\frac{|\mathcal{F}|K}{\alpha}\right)$.

Rearranging terms and applying (5), we obtain

$$J^\star - J(\pi_k) \le \frac{\kappa}{\eta} + \frac{2\Delta_S}{\varepsilon_0\eta}\log(|\mathcal{F}|K/\alpha) + \mathbb{E}_{s_1\sim\rho,\pi_k}\left[e^{(k)}(\tau)\right] - \frac{1}{2\eta}\sum_{t<[k]}\mathbb{E}_{s_1\sim\rho,\pi_t}\left[e^{(k)}(\tau)^2\right].$$

Summing over $k = 1,\dots,K$,

$$\begin{aligned}\sum_{k=1}^{K}(J^\star - J(\pi_k)) &\le \frac{K\kappa}{\eta} + \frac{2K\Delta_S}{\varepsilon_0\eta}\log(|\mathcal{F}|K/\alpha) + \sum_{k=1}^{K}\mathbb{E}_{s_1\sim\rho,\pi_k}\left[e^{(k)}(\tau)\right] - \frac{1}{2\eta}\sum_{k=1}^{K}\sum_{t<[k]}\mathbb{E}_{s_1\sim\rho,\pi_t}\left[e^{(k)}(\tau)^2\right] \\ &= \frac{K\kappa}{\eta} + \frac{2K\Delta_S}{\varepsilon_0\eta}\log(|\mathcal{F}|K/\alpha) + \mathbb{E}_{s_1\sim\rho}\left[\sum_{k=1}^{K}\mathbb{E}_{\pi_k}\left[e^{(k)}(\tau)\middle|s_1\right] - \frac{1}{2\eta}\sum_{k=1}^{K}\sum_{t<[k]}\mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)^2\middle|s_1\right]\right].\end{aligned}$$

Lemma F.4. *Let $\lambda \ge 1$. For any initial state $s_1 \in \mathcal{S}$, consider the sequence $\{f^{(k)}, \pi_k\}_{k=1}^K$ generated by Algorithm 2, where $\pi_k = \pi_{f^{(k)}}$ is the greedy policy with fixed deterministic tie-breaking. Then*

$$\begin{aligned}\sum_{k=1}^{K}\left|\mathbb{E}_{\pi_k}\left[e^{(k)}(\tau)\middle|s_1\right]\right| &\le \frac{BN(s_1)}{2\log 2}\log\left(1+\frac{KH}{\lambda}\right) \\ &\quad + 2\sqrt{2N(s_1)\log\left(1+\frac{KH}{\lambda}\right)}\cdot\sqrt{4K\lambda + \sum_{k=1}^{K}\sum_{t<[k]}\left(\mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)\middle|s_1\right]\right)^2},\end{aligned}$$

where $N(s_1)$ is the total number of reachable state-action pairs from s_1 under policy class $\Pi_{\mathcal{F}}$.

For a fixed s_1, applying Lemma F.4, Jensen's inequality $\left(\mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)\middle|s_1\right]\right)^2 \le \mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)^2\middle|s_1\right]$ and Cauchy-Schwarz in the form $xy \le \eta x^2 + \frac{1}{4\eta}y^2$, we have

$$\sum_{k=1}^{K}\mathbb{E}_{\pi_k}\left[e^{(k)}(\tau)\middle|s_1\right] \le \frac{BN(s_1)}{2\log 2}\log\left(1+\frac{KH}{\lambda}\right) + \eta N(s_1)\log\left(1+\frac{KH}{\lambda}\right) + \frac{K\lambda}{\eta} + \frac{1}{2\eta}\sum_{k=1}^{K}\sum_{t<[k]}\mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)^2\middle|s_1\right].$$

Combining the inequalities above and setting $\lambda = \kappa$, we obtain

$$\sum_{k=1}^{K} (J^\star - J(\pi_k)) \le \frac{2K\kappa}{\eta} + \frac{2K\Delta_S}{\varepsilon_0 \eta} \log(|\mathcal{F}|K/\alpha) + \mathbb{E}_{s_1 \sim \rho}\left[\frac{BN(s_1)}{2\log 2} \log\left(1 + \frac{KH}{\kappa}\right) + \eta N(s_1) \log\left(1 + \frac{KH}{\kappa}\right)\right].$$

For each $s \in \mathcal{S}$ and $h \in [H]$, define

$$S_h(s; \Pi_\mathcal{F}) := \{(s', a) : \exists \pi \in \Pi_\mathcal{F}, \text{ under } \pi \text{ and } s_1 = s, \text{ it holds that } s_h = s', a_h = a\}.$$

and let $N_h(s; \Pi_\mathcal{F}) := |S_h(s; \Pi_\mathcal{F})|$ and $N(s) := \sum_{h=1}^{H} N_h(s; \Pi_\mathcal{F})$.

Under deterministic MDPs, given s_1, every greedy policy in $\Pi_\mathcal{F}$ with fixed deterministic tie-breaking reaches a single element in $S_h(s_1; \Pi_\mathcal{F})$ at each layer h. Hence the layer-wise coverability of $M^\star_{s_1}$ is characterized by $N_h(s_1; \Pi_\mathcal{F})$, and therefore

$$N(s_1) \le H \cdot C_{\mathsf{cov}}(\Pi_\mathcal{F}; M^\star_{s_1}).$$

Taking expectation over $s_1 \sim \rho$, we derive

$$\mathbb{E}_{s_1 \sim \rho}[N(s_1)] \le H \cdot C'_{\mathsf{cov}}.$$

Substituting back, with $C := HC'_{\mathsf{cov}} \log\left(1 + \frac{KH}{\kappa}\right)$, we have

$$\sum_{k=1}^{K} (J^\star - J(\pi_k)) \le \frac{2K\kappa}{\eta} + \frac{2K\Delta_S}{\varepsilon_0 \eta} \log(|\mathcal{F}|K/\alpha) + \eta \cdot C + B \cdot C. \tag{6}$$

For the non-private term, we balance the standard regret term and the regularization term:

$$\frac{2K\kappa}{\eta} = C\eta \quad \to \quad \eta = \sqrt{\frac{2K\kappa}{C}}.$$

Substituting this back, the non-private part is

$$\frac{2K\kappa}{\eta} + \eta C = O\left(\sqrt{K\kappa C}\right).$$

For the private case, we balance

$$\frac{K\Delta_S}{\eta\varepsilon_0} \log(|\mathcal{F}|K/\alpha) = C\eta.$$

Given by the *Advanced Sequential Composition* (Lemma F.2), we consider the worst-case privacy loss:

$$\varepsilon = O\left(\sqrt{\frac{K}{B}\log(1/\delta)} \cdot \varepsilon_0\right).$$

Equivalently,

$$\varepsilon_0 = \Omega\left(\frac{\varepsilon\sqrt{B}}{\sqrt{K\log(1/\delta)}}\right).$$

Substituting back, we obtain

$$\eta = O\left(\frac{K^{3/4}\Delta_S^{1/2}\left(\log(|\mathcal{F}|K/\alpha)\right)^{1/2}\left(\log(1/\delta)\right)^{1/4}}{B^{1/4}\varepsilon^{1/2}C^{1/2}}\right).$$

Finally, substituting this choice of η back to (6), the privacy contribution is

$$O\left(\frac{K^{3/4}C^{1/2}\Delta_S^{1/2}\left(\log(|\mathcal{F}|K/\alpha)\right)^{1/2}\left(\log(1/\delta)\right)^{1/4}}{B^{1/4}\varepsilon^{1/2}}\right).$$

We balance this term with the batching term $B \cdot C$:

$$O\left(\frac{K^{3/4}C^{1/2}\Delta_S^{1/2}\left(\log(|\mathcal{F}|K/\alpha)\right)^{1/2}\left(\log(1/\delta)\right)^{1/4}}{B^{1/4}\varepsilon^{1/2}}\right) = O(B \cdot C),$$

which gives

$$B = O\left(\frac{K^{3/5}\Delta_S^{2/5}\left(\log(|\mathcal{F}|K/\alpha)\right)^{2/5}\left(\log(1/\delta)\right)^{1/5}}{\varepsilon^{2/5}C^{2/5}}\right).$$

We now turn to bound the sensitivity Δ_S. Note that

$$S_{\mathcal{D}}(f) = \eta\bar{f}_1 - L_{\mathcal{D}}^{\mathsf{BR}}(f).$$

Since the first term $\eta\bar{f}_1$ does not depend on $\mathcal{D}$, the sensitivity is entirely from $L_{\mathcal{D}}^{\mathsf{BR}}(f)$. Recall

$$L_{\mathcal{D}}^{\mathsf{BR}}(f) = \sum_{(\tau,r)\in\mathcal{D}}\left(R^{(f)}(\tau) - r\right)^2.$$

For each h, we have $f_h(s_h, a_h) \in [0,1]$ and $f_{h+1}(s_{h+1}) \in [0,1]$, so

$$R^{(f)}(\tau) = \sum_{h=1}^{H}\left[f_h(s_h,a_h) - f_{h+1}(s_{h+1})\right] \in [-H, H].$$

Since $r \in [0,1]$, it holds that

$$\left|R^{(f)}(\tau) - r\right| \le H + 1.$$

Consequently, changing one trajectory changes $L_{\mathcal{D}}^{\mathsf{BR}}(f)$ by at most $(H+1)^2$, and we conclude that

$$\Delta_S = O(H^2).$$

Similarly, by applying the *billboard lemma* (Lemma F.3), Algorithm 2 satisfies (ε, δ)-JDP.

Setting

$$\eta = \max\left\{\sqrt{\frac{2K\kappa}{C}}, \frac{K^{3/4}\Delta_S^{1/2}\left(\log(|\mathcal{F}|K/\alpha)\right)^{1/2}\left(\log(1/\delta)\right)^{1/4}}{B^{1/4}\varepsilon^{1/2}C^{1/2}}\right\},$$

and using the above choice of B, we have the final bound

$$\sum_{k=1}^{K}\left(J^\star - J(\pi_k)\right) \le O\left(\sqrt{K\kappa C}\right) + O\left(\frac{K^{3/5}C^{3/5}\Delta_S^{2/5}\left(\log(|\mathcal{F}|K/\alpha)\right)^{2/5}\left(\log(1/\delta)\right)^{1/5}}{\varepsilon^{2/5}}\right).$$

Plugging in

$$C = H \cdot C'_{\mathsf{cov}}\log\left(1 + \frac{KH}{\kappa}\right), \quad \kappa = O\left(H^3\log(|\mathcal{F}|K/\alpha)\right), \quad \Delta_S = O(H^2),$$

we obtain

$$\mathsf{Reg} \le 2H^2\sqrt{2C'_{\mathsf{cov}}K\log(|\mathcal{F}|K/\alpha)\log K} + H^{7/5}\frac{\left(C'_{\mathsf{cov}}K\log K\right)^{3/5}\left(\log(|\mathcal{F}|K/\alpha)\right)^{2/5}\left(\log(1/\delta)\right)^{1/5}}{\varepsilon^{2/5}}$$

□

Proof of Lemma F.4. Fix an initial state $s_1 \in \mathcal{S}$. Define the index set of reachable state-action pairs

$$\mathcal{I} := \{(h, s, a) : h \in [H], (s, a) \in S_h(s_1; \Pi_{\mathcal{F}})\} \subseteq [H] \times \mathcal{S} \times \mathcal{A}, \quad |\mathcal{I}| = N(s_1).$$

For each episode $k \in [K]$, define $\psi^{(k)} \in \mathbb{R}^{|\mathcal{I}|}$ by

$$\psi^{(k)}_{(h,s,a)} := f^{(k)}_h(s, a) - f^{(k)}_{h+1}(s^+_h(s, a)) - R_h(s, a),$$

where $s^+_h(s, a)$ is the unique next state reached from (s, a) at step h under the deterministic transition, and $f^{(k)}_{h+1}(s) := \max_{a' \in \mathcal{A}} f^{(k)}_{h+1}(s, a')$ with the convention $f^{(k)}_{H+1} \equiv 0$.

And define the occupancy vector $\phi^{(k)} \in \mathbb{R}^{|\mathcal{I}|}$ by

$$\phi^{(k)}_{(h,s,a)} := \mathbb{P}_{\pi_k}(s_h = s, a_h = a | s_1).$$

Then, by the definition of $e^{(k)}(\tau)$ and the deterministic transition, the on-policy error in episode k can be written as

$$\mathbb{E}_{\pi_k}\left[e^{(k)}(\tau)\middle| s_1\right] = \langle \psi^{(k)}, \phi^{(k)} \rangle.$$

Thus, our target is to bound $\sum_{k=1}^{K} \left|\langle \psi^{(k)}, \phi^{(k)} \rangle\right|$.

Define the covariance matrices based on historical visitations

$$V_k := \lambda I + \sum_{j=1}^{k} \phi^{(j)} \phi^{(j)\top}.$$

Since the policy is updated only at batch boundaries, the historical covariance matrix available at the beginning of the batch containing episode k is

$$V_{[k]-1} = \lambda I + \sum_{t<[k]} \phi^{(t)} \phi^{(t)\top},$$

where $[k]$ denotes the first episode of the batch containing episode k.

Define the bad-round set Ψ where the covariance matrix at the beginning of the batch significantly underestimates the real-time matrix

$$\Psi := \left\{ k \in [K] : \frac{\|\phi^{(k)}\|_{V^{-1}_{[k]-1}}}{\|\phi^{(k)}\|_{V^{-1}_{k-1}}} > 2 \right\}.$$

We decompose the total sum into bad and good rounds

$$\sum_{k=1}^{K} \left|\langle \psi^{(k)}, \phi^{(k)} \rangle\right| = \sum_{k \in \Psi} \left|\langle \psi^{(k)}, \phi^{(k)} \rangle\right| + \sum_{k \notin \Psi} \left|\langle \psi^{(k)}, \phi^{(k)} \rangle\right|.$$

For good rounds ($k \notin \Psi$), by definition $\|\phi^{(k)}\|_{V^{-1}_{[k]-1}} \leq 2\|\phi^{(k)}\|_{V^{-1}_{k-1}}$. Hence, by the Cauchy–Schwarz inequality,

$$\|\langle \psi^{(k)}, \phi^{(k)} \rangle\| \leq \|\psi^{(k)}\|_{V_{[k]-1}} \cdot \|\phi^{(k)}\|_{V^{-1}_{[k]-1}} \leq 2\|\psi^{(k)}\|_{V_{[k]-1}} \cdot \|\phi^{(k)}\|_{V^{-1}_{k-1}}.$$

Summing over good rounds and applying Cauchy–Schwarz again to the sum

$$\sum_{k \notin \Psi} \left|\langle \psi^{(k)}, \phi^{(k)} \rangle\right| \leq 2\sqrt{\sum_{k=1}^{K} \min\left\{1, \|\phi^{(k)}\|^2_{V^{-1}_{k-1}}\right\}} \cdot \sqrt{\sum_{k=1}^{K} \max\left\{1, \|\psi^{(k)}\|^2_{V_{[k]-1}}\right\}}.$$

By the elliptical potential lemma [59], and using $\|\phi^{(k)}\|_2^2 \leq H$, the first term is bounded by

$$\sum_{k=1}^{K} \min\left\{1, \|\phi^{(k)}\|^2_{V^{-1}_{k-1}}\right\} \leq 2N(s_1) \log\left(1 + \frac{KH}{N(s_1)\lambda}\right).$$

For the second term involving $\psi^{(k)}$, by the definition of $V_{[k]-1}$,

$$\|\psi^{(k)}\|^2_{V_{[k]-1}} = \lambda\|\psi^{(k)}\|_2^2 + \sum_{t<[k]} \langle \phi^{(t)}, \psi^{(k)}\rangle^2.$$

Moreover, for every $t < [k]$,

$$\langle \phi^{(t)}, \psi^{(k)}\rangle = \mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)\middle| s_1\right].$$

By the definition of $\psi^{(k)}$, we have $\|\psi^{(k)}\|_\infty \le 2$ and obtain

$$\sum_{k=1}^{K} \max\left\{1, \|\psi^{(k)}\|^2_{V_{[k]-1}}\right\} \le 4K\lambda N(s_1) + \sum_{k=1}^{K}\sum_{t<[k]} \left(\mathbb{E}_{\pi_t}\left[e^{(k)}(\tau)\middle| s_1\right]\right)^2.$$

Now we bound the bad rounds. For those bad rounds ($k \in \Psi$), since π_k is greedy with respect to $f^{(k)}$, we have

$$\langle \psi^{(k)}, \phi^{(k)}\rangle = \mathbb{E}_{\pi_k}[e^{(k)}(\tau)|s_1] = f_1^{(k)}(s_1) - V^{\pi_k}(s_1).$$

By the normalization of rewards and the range of functions in $\mathcal{F}$, both terms lie in $[0, 1]$. Hence

$$\sum_{k\in\Psi}\left|\langle \psi^{(k)}, \phi^{(k)}\rangle\right| \le |\Psi|.$$

We bound $|\Psi|$ using a determinant argument. Let $t_m := (m-1)B + 1$ denote the first episode of batch m, with the convention $t_{M+1} := K + 1$. For any $k \in \Psi$, suppose k belongs to batch m, so that $t_m = [k]$. By the definition of Ψ,

$$\frac{\|\phi^{(k)}\|_{V^{-1}_{t_m-1}}}{\|\phi^{(k)}\|_{V^{-1}_{k-1}}} > 2.$$

Since $V_{t_m-1} \preceq V_{k-1}$, the standard determinant comparison inequality implies

$$\log\left(\frac{\det(V_{k-1})}{\det(V_{t_m-1})}\right) > 2\log 2.$$

Moreover, since $V_{k-1} \preceq V_{t_{m+1}-1}$, we have

$$\log\left(\frac{\det(V_{t_{m+1}-1})}{\det(V_{t_m-1})}\right) > 2\log 2.$$

Let $\mathcal{M}_\Psi$ be the set of batch-start indices m whose batch contains at least one bad round. Summing the above inequality over these bad batches and bounding it by the total log-determinant growth yields

$$|\mathcal{M}_\Psi| \cdot (2\log 2) \le \sum_{m\in\mathcal{M}_\Psi} \log\frac{\det(V_{t_{m+1}-1})}{\det(V_{t_m-1})} \le \log\frac{\det(V_K)}{\det(V_0)}.$$

Applying the standard determinant bound

$$\log\frac{\det V_K}{\det V_0} \le N(s_1)\log\left(1 + \frac{KH}{N(s_1)\lambda}\right),$$

we can solve for the number of bad batches

$$|\mathcal{M}_\Psi| \le \frac{N(s_1)}{2\log 2}\log\left(1 + \frac{KH}{N(s_1)\lambda}\right).$$

Since each batch has size B, the total number of bad rounds $|\Psi|$ is at most $B \cdot |\mathcal{M}_\Psi|$. Therefore

$$|\Psi| \le \frac{BN(s_1)}{2\log 2} \log\left(1 + \frac{KH}{N(s_1)\lambda}\right).$$

Combining the bounds for good and bad rounds and rescaling $\lambda \leftarrow \frac{\lambda}{N(s_1)}$, we obtain

$$\sum_{k=1}^{K} \left|\mathbb{E}_{\pi_k}\left[e^{(k)}(\tau)\middle|s_1\right]\right| \le \frac{BN(s_1)}{2\log 2} \log\left(1 + \frac{KH}{\lambda}\right) + 2\sqrt{2N(s_1)\log\left(1 + \frac{KH}{\lambda}\right)} \cdot \sqrt{4K\lambda + \sum_{k=1}^{K}\sum_{t<[k]} \left(\mathbb{E}_{\pi_t}[e^{(k)}(\tau)|s_1]\right)^2}.$$

This completes the proof. □

G Limitations

Our results establish the first sublinear JDP regret guarantees for online RL with general function approximation. These guarantees rely on standard exploration and representation conditions, including coverability and, in the general stochastic setting, Bellman completeness. Although Appendix C extends the analysis to infinite function classes via covering arguments, designing more computationally efficient implementations for very rich function classes remains an important direction for future work. Finally, our experiments are intended as proof-of-concept illustrations of the predicted privacy–utility tradeoff, rather than as a large-scale empirical benchmark.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main theoretical contributions, including the JDP setting, general function approximation, the $K^{3/5}$ regret rate, and the dependence on coverability. The scope and assumptions are further specified in the preliminaries and theorem statements.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses its main structural assumptions, including Bellman completeness, realizability, finite function classes, and the use of JDP. Further discussion of infinite function classes and the intentionally limited proof-of-concept experiment appears in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The assumptions are stated in the preliminaries and theorem statements, including Bellman completeness, realizability, privacy parameters, and coverability. Proof sketches are given in the main text and full proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main claims of the paper are theoretical and do not rely on experimental results. The appendix provides the construction, hypothesis class, compared methods, and reporting metric for the small proof-of-concept simulation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper does not currently provide public code. The experiment is a small synthetic proof-of-concept and its construction is described in the appendix, but no executable reproduction package is included.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix E describes the synthetic deterministic outcome-reward MDP environment, finite nonlinear hypothesis class, easy and hard instances, compared methods, privacy parameters, and reporting metric. The experiment is intended only as a proof-of-concept rather than a comprehensive empirical benchmark.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The experiment is only a small proof-of-concept simulation and is not used to support the main theoretical claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E reports the compute resources for the proof-of-concept simulation, including CPU-only execution and runtime. The experiment is small-scale and synthetic, and no GPU is required.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is theoretical and uses only synthetic proof-of-concept experiments, with no human subjects, private datasets, or deployed systems. The paper studies formal privacy guarantees and does not introduce data collection or release risks.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The introduction discusses positive impacts of privacy-preserving RL for sensitive personalization settings such as healthcare and LLM interactions, and also motivates the work through privacy leakage risks in non-private RL. The paper is theoretical and does not propose deployment in a specific high-risk application.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release data, pretrained models, scraped datasets, or other assets with high misuse risk. The only experiment is synthetic and proof-of-concept.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [N/A]

Justification: The paper does not use external datasets, pretrained models, or third-party code assets as part of the reported results. Related methods and prior work are credited through citations.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce or release a new dataset, benchmark, model, or software package. The synthetic proof-of-concept environment is described in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our core methods, theoretical results, and experiments do not use an LLM as an original or non-standard research component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.