Privacy Preserving Off-Policy Evaluation

Tengyang Xie Philip S. Thomas Gerome Miklau
College of Information and Computer Sciences, UMass Amherst
{txie,pthomas,miklau}@cs.umass.edu

Abstract

Many reinforcement learning applications involve the use of data that is sensitive, such as medical records of patients or financial information. However, most current reinforcement learning methods can leak information contained within the (possibly sensitive) data on which they are trained. To address this problem, we present the first differentially private approach for off-policy evaluation. We provide a theoretical analysis of the privacy-preserving properties of our algorithm and analyze its utility (speed of convergence). After describing some results of this theoretical analysis, we show empirically that our method outperforms previous methods (which are restricted to the on-policy setting).

1 Introduction

Many proposed applications of *reinforcement learning* (RL) involve the use of data that could contain sensitive information. For example, Raghu et al. [2017] proposed an application of RL and off-policy evaluation methods that uses peoples' medical records, and Theocharous et al. [2015] applied off-policy evaluation methods to user data collected by a bank in order to improve the targeting of advertisements. In examples like these, the data used by the RL systems is sensitive, and one should ensure that the methods applied to the data do not leak any sensitive information.

Recently, Balle et al. [2016] showed how techniques from *differential privacy* can be used to ensure that (with high probability) policy evaluation methods for RL do not leak (much) sensitive information. In this paper we extend their work in two ways. First, RL methods are often applied to batches of data collected from the use of a currently deployed policy. The goal of these RL methods is not to evaluate the performance of the current policy, but to improve upon it. Thus, policy evaluation methods must be *off-policy*—they must use the data from the behavior policy to reason about the performance of newly proposed policies. This is the problem of *off-policy evaluation*, and both of the previous medical and banking examples require these methods. Whereas Balle et al. [2016] consider the *on-policy* setting (evaluating the deployed policy), we focus on the off-policy setting.

Second, Balle et al. [2016] achieve their privacy guarantee using output perturbation: they first run an existing (non-private) least-squares policy evaluation method, resulting in a real-valued vector; then they add random noise to each element of the vector. Although this approach was one of the first and most simple methods for ensuring that guarantees of privacy hold [Dwork et al., 2006b], more sophisticated methods for ensuring privacy have since been developed. We show how one of these newer approaches to differential privacy, which adds noise to stochastic gradient descent updates [Song et al., 2013; Bassily et al., 2014], rather than to the least squares solution, can be combined with GTD2, the dominant off-policy evaluation algorithm [Sutton et al., 2009].

After presenting our new privacy preserving off-policy evaluation algorithm, which we call *gradient perturbed off-policy evaluation* (GPOPE) to differentiate it from the previous output-

perturbation methods, we provide proofs of privacy and convergence rate. We use the properties of Rényi differential privacy and its amplification via subsampling [Bun and Steinke, 2016; Mironov, 2017; Balle et al., 2018; Wang et al., 2018] together with the moments accountant technique [Abadi et al., 2016] to effectively keep track of (ε, δ)-differential privacy parameters through all steps of our algorithm. The convergence rate analysis quantifies the trade-off between the strength of the privacy guarantees that our algorithms provide and the accuracy of their off-policy predictions.

Since the on-policy setting is a special case of the off-policy setting (where the policy being evaluated happens to be the same as the currently deployed policy), we can compare our algorithm directly to the output-perturbation methods of Balle et al. [2016] in the on-policy setting. We show empirically that our algorithm offers greater utility, i.e., using the same data, our algorithm can provide stronger guarantees of privacy for the same degree of prediction error. We also conduct experiments in the off-policy setting, where prior work is not applicable, and the results support the conclusions of our analytic analysis.

The rest of the paper is organized as follows. We review the relevant background on off-policy evaluation in Section 2 and background on differential privacy in Section 3. We present our algorithm in Section 4. In Section 5 we analyze the privacy preserving properties of our algorithm, and in Section 6 we provide an analysis of the utility of our algorithm. We provide an empirical case study in Section 7, using a synthetic MDP that mimics characteristics of a medical application, the standard Mountain Car domain, and a more challenging HIV simulator. We conclude in Section 8 with a discussion of future work.

2 Background: Off-Policy Evaluation

This section offers a brief overview of off-policy evaluation, including the definition of Markov decision processes, mean squared projected Bellman error, and the saddle-point formulation of the gradient temporal-difference (GTD2) off-policy evaluation method [Sutton et al., 2009].

A *Markov decision process* (MDP) [Sutton and Barto, 1998; Puterman, 2014] is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the finite set of possible states, $t \in \{0, 1, 2, \dots\}$ is the *time step*, S_t is the state at time t (a random variable), $\mathcal{A}$ is the finite set of possible actions, A_t is the action at time t, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the *transition function*, defined such that $\mathcal{P}(s, a, s') \coloneqq \Pr(S_{t+1} = s' | S_t = s, A_t = a)$, R_t is the scalar reward at time t, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined such that $R(s, a) \coloneqq \mathbf{E}[R_t | S_t = s, A_t = a]$, and $\gamma \in [0, 1]$ is a parameter that characterizes how rewards are discounted over time. A policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, describes one way that actions can be chosen: $\pi(s, a) \coloneqq \Pr(A_t = a | S_t = s)$.

A key step in many RL algorithms is to estimate the state-value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of a given policy π, which is defined as $V^\pi(s) \coloneqq \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, \pi]$. The process of estimating a state-value function is known as *policy evaluation*. In this paper we consider the problem of *off-policy* evaluation, wherein we estimate V^π given data (states, actions, and rewards) sampled from applying a different policy, π_b, called the *behavior policy*, which may be different from π (i.e., the policy being evaluated). Furthermore, we consider the setting where a linear function approximator, $\widehat{V}^\pi$, is used. That is $\widehat{V}^\pi$ can be written as $\widehat{V}^\pi(s) \coloneqq \theta^\intercal \phi(s)$, where $\theta \in \mathbb{R}^n$ is a set of weights, and $\phi(s) \in \mathbb{R}^n$ is a feature vector associated with state s.

Let $H \coloneqq \{(S_t, A_t, R_t, S_{t+1})\}_{t=0}^{\tau}$ be a *trajectory* with length τ. Often each trajectory contains data pertaining to a single individual over time. In real-world applications, states often describe people: their bank balance when the MDP models automatic selection of online credit card ads [Theocharous et al., 2015], or medical conditions when selecting between treatments [Raghu et al., 2017]. Similarly, actions can include drug prescriptions and rewards can encode medical outcomes.

Recent work has shown that optimizing the weight vector, θ, can be phrased as a saddle point problem [Liu et al., 2015]: $\min_\theta \max_w \mathcal{L}(\theta, w)$, where

$$\begin{aligned}
\mathcal{L}(\theta, w) &\coloneqq w^\intercal(b - A\theta) - \frac{1}{2}\|w\|_C^2 \\
A &\coloneqq \mathbf{E}\left[\frac{1}{\tau}\sum_{t=0}^{\tau} \rho_t \phi_t(\phi_t - \gamma\phi_{t+1})^\intercal\right], \\
b &\coloneqq \mathbf{E}\left[\frac{1}{\tau}\sum_{t=0}^{\tau} \rho_t \phi_t R_t\right], \; C \coloneqq \mathbf{E}\left[\frac{1}{\tau}\sum_{t=0}^{\tau} \phi_t \phi_t^\intercal\right],
\end{aligned} \tag{2.1}$$

where $w \in \mathbb{R}^n$ is introduced by duality [Boyd et al., 2011; Sutton et al., 2009; Liu et al., 2015], the expected values in (2.1), are over states, S_t, actions, A_t, and rewards, R_t, produced by running the behavior policy, π_b, τ is the (finite) length of the trajectory, ϕ_t is shorthand for $\phi(S_t)$, and $\rho_t \coloneqq \pi(S_t, A_t)/\pi_b(S_t, A_t)$.

Liu et al. [2015] proposed using a stochastic gradient method to optimize this saddle-point problem. This algorithm uses the following unbiased estimates of A, b, and C, produced using the states, actions, and rewards from the i^{th} trajectory (which is of length τ_i):

$$\begin{aligned}
\widehat{A}_i &= \frac{1}{\tau_i}\sum_{t=0}^{\tau_i} \rho_t \phi_t(\phi_t - \gamma\phi_{t+1})^\intercal, \\
\widehat{b}_i &= \frac{1}{\tau_i}\sum_{t=0}^{\tau_i} \rho_t \phi_t R_t, \; \widehat{C}_i = \frac{1}{\tau_i}\sum_{t=0}^{\tau_i} \phi_t \phi_t^\intercal.
\end{aligned} \tag{2.2}$$

The resulting stochastic gradient algorithm proposed by Liu et al. [2015] is identical to the GTD2 algorithm, and is given by the following update equations:[1]

$$\begin{aligned}
\theta_{i+1} &= \theta_i + \beta_i \widehat{A}_i^\intercal w_i, \\
w_{i+1} &= w_i + \beta_i(\widehat{b}_i - \widehat{A}_i\theta_i - \widehat{C}_i w_i),
\end{aligned} \tag{2.3}$$

where $\beta_1, \beta_2, \ldots$ is a sequence of positive step sizes.

3 Background: Differential Privacy

In this section we define *differential privacy* (DP) and its application to the data underlying off-policy evaluation. We also describe some tools that aid in analyzing the privacy loss when using gradient methods.

A *data set*, d, consists of a set of m points, $\{x_1, \ldots, x_m\}$, where each point is an element of universe $\mathcal{D}$ (for RL, a point will correspond to a trajectory, H, containing data associated with one person). For RL applications to human data, each point typically describes a trajectory consisting of a finite sequence of transitions of a single individual, i.e., $x_i = \{(S_t, A_t, R_t, S_{t+1})\}_{t=0}^{\tau_i}$, and the length of trajectory may vary across individuals. We assume each trajectory is generated by running a behavior policy, π_b, and that states, actions, and rewards may all be potentially sensitive and therefore worthy of privacy protection. We denote by $\mathcal{D}$ the set of all possible data sets.

The privacy condition our algorithm provides constrains the treatment of pairs of *adjacent* datasets:

[1] Although Sutton et al. [2009] were the first to derive GTD2, they did not derive it as a stochastic gradient algorithm. Liu et al. [2015] were the first to show that GTD2 can be phrased as presented here—as a stochastic gradient algorithm for a saddle-point problem.

Definition 1 (**Adjacent Data Set**). *Two data sets, $d, d' \in \mathcal{D}$ are adjacent if they differ by exactly one point.*

Differential privacy is a formal notion of privacy, which guarantees that the output of a computation on a sensitive data set cannot reveal too much about any one individual. Formally, consider a *randomized mechanism*, $\mathcal{M}$, which takes as input a data set and produces as output an element of some set, $\mathcal{Y}$.

Definition 2 (**Differential Privacy**). *Let $\mathcal{M}$ denote a randomized mechanism that has domain $\mathcal{D}$ and range $\mathcal{Y}$. $\mathcal{M}$ satisfies (ε, δ)-differential privacy for some $\varepsilon, \delta > 0$, if for every pair of adjacent data sets, $d, d' \in \mathcal{D}$, and for every $S \subseteq \mathcal{Y}$ the following holds:*

$$\Pr\left(\mathcal{M}(d) \in S\right) \leq e^{\varepsilon} \Pr\left(\mathcal{M}(d') \in S\right) + \delta.$$

This definition requires that the difference in output probabilities resulting from changing the database by altering any one individual's contribution will be small. Note that adjacent databases differ in an individual's *full* trajectory, not merely one transition.

Applied to our reinforcement learning problem, a differentially private training mechanism allows the public release of a parameter vector of the value function with a strong guarantee: by analyzing the output, an adversary is severely limited in what they can learn about any individual, even if they have access to arbitrary public information.

Standard ε-differential privacy [Dwork et al., 2006b] corresponds to $\delta = 0$; we use the common relaxation, (ε, δ)-differential privacy [Dwork et al., 2006a, 2014].

4 Differentially Private Off-Policy Evaluation Algorithms

In this section we provide the details of our differentially private off-policy evaluation algorithms.

We construct our differentially private off-policy evaluation algorithm by using the Gaussian mechanism [Dwork et al., 2006b] and the moments accountant [Abadi et al., 2016] to privatize the stochastic gradient off-policy evaluation algorithm presented in (2.3). This involves three steps. First, a trajectory of data is collected from running the behavior policy, π_b. Second, a primal-dual stochastic gradient estimate is generated from this data, and its l_2 norm is clipped to ensure that it is bounded below a positive constant, h. Third, we add normally distributed noise to each term of the gradient before updating the weights using the (clipped and noisy) stochastic gradient estimate. In subsequent sections we show that the amount of noise that we introduce provides the desired privacy preserving guarantees, regardless of the value chosen for h.

Before providing pseudocode for our algorithms, we first define the primal-dual gradient at the i-th step, $B_i(\theta, w)$, which is obtained by stacking the estimated primal and negative dual gradients:

$$\begin{aligned} B_i(\theta, w) &:= \left[\frac{\partial \widehat{\mathcal{L}}_i(\theta, w)}{\partial \theta}^{\intercal}, -\frac{\partial \widehat{\mathcal{L}}_i(\theta, w)}{\partial w}^{\intercal}\right]^{\intercal} \\ &= \begin{bmatrix} 0 & -\widehat{A}_i^{\intercal} \\ \widehat{A}_i & \widehat{C}_i \end{bmatrix} \begin{bmatrix} \theta \\ w \end{bmatrix} - \begin{bmatrix} 0 \\ \widehat{b}_i \end{bmatrix}, \end{aligned} \tag{4.1}$$

where $\widehat{\mathcal{L}}_i(\theta, w) := w^{\intercal}(\widehat{b}_i - \widehat{A}_i\theta) - 0.5\|w\|^2_{\widehat{C}_i}$, and $\widehat{A}_i, \widehat{b}_i, \widehat{C}_i$ are defined in (2.2). Let $B(\theta, w)$ denote the true primal-dual gradient, $B(\theta, w) := \mathbf{E}[B_i(\theta, w)]$, where the expected values are over states, actions, and rewards produced by running the behavior policy.

Pseudocode for our new privacy preserving off-policy evaluation algorithm, which we call *gradient Perturbed off-policy evaluation* (GPOPE), is provided in Algorithm 1.

Algorithm 1 Gradient Perturbed Off-Policy Evaluation (GPOPE)

Input: Initial point, (θ_1, w_1), step size sequence $\{\beta_i\}$, clipping bound, $h \in \mathbb{R}$, private dataset, d, with m trajectories, number of iterations, N, and a noise scale $\sigma \in \mathbb{R}$.

1: **for** $i = 1$ **to** N **do**
2: Randomly choose a trajectory from private dataset, d.
3: Compute $\widehat{A}_i$, $\widehat{b}_i$, $\widehat{C}_i$ using all transitions from sampled trajectory by (2.2).
4: Compute $B_i(\theta_i, w_i)$ as in (4.1).
5: Clip $\bar{B}_i(\theta_i, w_i) = B_i(\theta_i, w_i)/\max(1, \|B_i(\theta_i, w_i)\|_2/h)$.
6: Sample vector $\zeta \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, of length $2n$, and compute $\tilde{B}_i(\theta_i, w_i) = \bar{B}_i(\theta_i, w_i) + h\zeta$.
7: $[\theta_{i+1}^\intercal, w_{i+1}^\intercal]^\intercal = [\theta_i^\intercal, w_i^\intercal]^\intercal - \beta_i \tilde{B}_i(\theta_i, w_i)$.
8: **end for**

Notice that in GPOPE we use all of the transitions from trajectory i to create the unbiased estimates of A, b, and C. Alternate algorithms could use data from a single trajectory to create multiple estimates of A, b, and C, and thus could perform multiple gradient updates given one trajectory. However, in preliminary experiments we found that the episodic approach taken by GPOPE (where we use all of the data from the trajectory for one update) performed the best. This is supported by our theoretical analysis, which shows that the trade-offs between number of updates, the variance of updates, and the amount of noise that must be added to updates, favors this episodic approach.

We use σ^2 to denote the variance of the Gaussian noise in our algorithm. The choice of σ depends on the desired privacy level of the algorithm, as discussed in the next section.

5 Privacy Analysis

In this section we provide a formal privacy analysis for our algorithm. We adapt the moments accounting introduced by Abadi et al. [2016] and the recent privacy amplification properties of subsampling mechanisms [Bun and Steinke, 2016; Balle et al., 2018; Wang et al., 2018] to bound the privacy loss of a sequence of adaptive mechanisms, and we show that our algorithm is (ε, δ)-differentially private.

Theorem 1. *Given a data set consisting of m points and fixing the number of iterations, N, there exist constants c_1 and c_2, such that for any $\varepsilon < c_1 N/m^2$, Algorithm 1 is (ε, δ)-differentially private for $\delta > 0$ if*

$$\sigma \geq \frac{c_2\sqrt{N\log(1/\delta)}}{m\varepsilon}.$$

The detailed proof of Theorem 1 is in the appendix. In the remainder of this section we provide an outline of the proof of Theorem 1, which proceeds as follows. We first define the *privacy loss* and the *privacy loss random variable*. We use privacy loss to measure the difference in the probability distribution resulting from running $\mathcal{M}$ on d and d'. Bounds on the tails of the privacy loss random variable then imply the privacy condition.

Definition 3 (**Privacy Loss**). *Let $\mathcal{M}$ be a randomized mechanism with domain $\mathcal{D}$ and range $\mathcal{Y}$, and* aux *be auxiliary input, $d, d' \in \mathcal{D}$ be a pair of adjacent data sets. For an outcome $o \in \mathcal{Y}$, the*

privacy loss at o is:

$$l(o; \mathcal{M}, \mathsf{aux}, d, d') \coloneqq \log \left(\frac{\Pr[\mathcal{M}(\mathsf{aux}, d) = o]}{\Pr[\mathcal{M}(\mathsf{aux}, d') = o]} \right).$$

The auxiliary information, aux, could be any additional information available to the adversary. We use aux here to model the composition of adaptive mechanisms, where we have a sequence of mechanisms and the i-th mechanism, $\mathcal{M}_i$, could use the output of previous mechanisms, $\mathcal{M}_1, \ldots, \mathcal{M}_{i-1}$, as its input.

We define the privacy loss random variable using the outcome sampled from $\mathcal{M}(d)$, as $L(\mathcal{M}, \mathsf{aux}, d, d') = l(\mathcal{M}(d); \mathcal{M}, \mathsf{aux}, d, d')$.

In order to more precisely analyze the privacy cost of sequences of mechanisms, we use a recent advance in privacy cost accounting called the moments accountant, introduced by Abadi et al. [2016] and which builds on prior work [Bun and Steinke, 2016; Dwork and Rothblum, 2016].

Definition 4 (Moments Accountant). *Let $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ be a randomized mechanism and d, d' a pair of adjacent databases. The λ^{th} moment of the privacy loss random variable $L(\mathcal{M}, \mathsf{aux}, d, d')$ is:*

$$\alpha_{\mathcal{M}}(\lambda; \mathsf{aux}, d, d') \coloneqq \log \mathbf{E}[\exp(\lambda L(\mathcal{M}, \mathsf{aux}, d, d'))].$$

The moments accountant is defined as

$$\alpha_{\mathcal{M}}(\lambda) \coloneqq \max_{\mathsf{aux}, d, d'} \alpha_{\mathcal{M}}(\lambda; \mathsf{aux}, d, d'),$$

which bounds the λ-moment for all possible inputs (i.e., all possible d, d', aux).

In the following lemma we provide an upper bound on the moments accountant for each iteration in our algorithm. This upper bound on the moments accountant is the key for proving Theorem 1.

Lemma 1. *Let sensitive dataset d contain m trajectories, and x_i be the sampled trajectory in the i^{th} iteration. Then the randomized mechanism $\mathcal{M}(d) = \bar{B}_t(x_i) + h \cdot \mathcal{N}(0, \sigma^2 \mathbf{I})$ satisfies*

$$\alpha_{\mathcal{M}}(\lambda) \leq \frac{\lambda(\lambda+1)}{2m^2} \min \left\{ 4 \left(e^{1/\sigma^2} - 1 \right), 2e^{1/\sigma^2} \right\},$$

where $\bar{B}_t(x_i)$ denotes $\bar{B}_t(\theta_i, w_i)$ defined in step 5 of Algorithm 1, and $\mathcal{M}$ returns the noised gradient.

We prove Lemma 1 using use the amplification properties for Rényi differential privacy via subsampling [Bun and Steinke, 2016; Mironov, 2017; Wang et al., 2018]. We provide a detailed proof in the appendix. The results in Lemma 1 are similar to a result of Abadi et al. [2016] when σ^2 is large (if $\sigma^2 \geq 1/\ln 2$, $\alpha_{\mathcal{M}}(\lambda) \leq 4\lambda(\lambda+1)/m^2\sigma^2$), but our Lemma 1 also covers the regime of small σ Abadi et al. [2016] does not cover. Also note that our definition of adjacent data sets is different from that of Abadi et al. [2016]. Our approach avoids the need to specify a discrete list of moments ahead of time as required in the moments accountant method of Abadi et al. [2016].

Note that our algorithm can guarantee (ε, δ)-differential privacy when each update only uses data from one trajectory. This is because the length of trajectories are not always the same, and so using data from multiple trajectories would cause Lemma 1 to not hold. However, our privacy analysis holds with the same privacy guarantee for the case when a subset of the transitions of the sampled trajectory are used. Intuitively, the best choice is to use all transitions of the sampled trajectory; we will justify this in the next section.

6 Utility Analysis

In this section we present the convergence analysis (utility analysis) of our algorithm. For this analysis, we assume that h is selected to be sufficiently large so that the l_2 norm of the gradient estimate is not clipped, i.e., the gradient estimates are sufficiently small (empirically, we found this assumption held across all of our experiments). Also, without loss of generality, we can avoid gradient clipping by scaling the objective function [Wang et al., 2017], i.e., changing the basis used for approximation.

Let (ε, δ) be the privacy parameters, N be the total number of iterations of the loop in Algorithm 1, and m be the number of trajectories in the data set. The noise added to the gradient of θ is $\mathcal{N}(0, \sigma^2\mathbf{I})$, where $\mathbf{I}$ is the $2n \times 2n$ identity matrix and σ is the noise scale chosen according to Theorem 1. Let c be a constant defined as $c \coloneqq m^2\|h^2\sigma^2\mathbf{I}\|_F^2/N$, where $\|\cdot\|_F$ is the Frobenius norm. Note that we choose σ according to Theorem 1, i.e., $\sigma \geq c_2\sqrt{N\log(1/\delta)}/m\varepsilon$, so that we have

$$c \geq 2nc_2^2\log(1/\delta)/\varepsilon^2,$$

which does not depend on m and N.

First, let the optimal solution be expressed as

$$\begin{aligned} \theta^* &\coloneqq (A^\intercal C^{-1}A)^{-1}A^\intercal C^{-1}b, \\ w^* &\coloneqq C^{-1}(b - A^\intercal\theta^*). \end{aligned} \tag{6.1}$$

In order to analyze the convergence of the algorithm, we examine the difference between the current parameters and the optimal solution. We define a residual vector ξ_j, at each iteration j, and a useful parameter Q, as:

$$\xi_i \coloneqq \begin{bmatrix} \theta_i - \theta^* \\ w_i - w^* \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & -A^\intercal \\ A & C \end{bmatrix}. \tag{6.2}$$

Note that the optimal solution can be expressed as (6.1). The first order optimally condition is obtained by setting the gradient to zero, which is satisfied by (θ^*, w^*), such that

$$\begin{bmatrix} 0 & -A^\intercal \\ A & C \end{bmatrix}\begin{bmatrix} \theta^* \\ w^* \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}.$$

We have defined $B_i(\theta_i, w_i)$ to be the stochastic approximate gradient at iteration i, which is stacking of the approximate primal and negative dual gradient using the $\widehat{A_i}, \widehat{b_i}, \widehat{C_i}$ at iteration i, and $B(\theta_i, w_i)$ using the true gradient at iteration i. Also let $\tilde{B}_i(\theta_i, w_i)$ be the perturbed approximate gradient, which is defined in step 7 of Algorithm 1.

We also define Δ_i to be the approximation error of the primal-dual gradient at iteration i, which is $\Delta_i \coloneqq \tilde{B}_i(\theta_i, w_i) - B(\theta_i, w_i)$. Note that $\mathbf{E}[\Delta_i] = 0$, since it is an unbiased stochastic approximation. We introduce an assumption, which ensures that the variance of Δ_i is bounded:

Assumption 1. *There exists a constant, G^2, such that for any t,*

$$\mathbf{E}[\|\Delta_i\|_2^2] \leq G^2 + cN/m^2.$$

Remark 1. *Note that bounded variance of the stochastic approximation is a standard assumption in the literature of stochastic gradient methods. In our differentially private case, the variance bound should be in terms of the privacy guarantee (i.e. ε and δ), since Algorithm 1 adds normally distributed noise to each term of the gradient. The term cN/m^2 in the assumption above follows $cN/m^2 = \|h^2\sigma^2\mathbf{I}\|_F^2$, where $\|h^2\sigma^2\mathbf{I}\|_F^2$ is the Frobenius norm of the covariance matrix of added noise. In the non-private case, c should be 0, i.e., $\mathbf{E}[\|B_i(\theta_i, w_i) - B(\theta_i, w_i)\|_2^2] \leq G^2$.*

Thus, we obtain the key properties of each iteration in our algorithm.

Lemma 2. *Let ξ_{i+1} be generated by the non-private algorithm at iteration i, if we define Q as (6.2), and we use $\lambda_{\min}(Q)$ to denote the minimum eigenvalue of Q, $\lambda_{\max}(Q)$ to denote the maximum eigenvalue of Q. If we choose $\beta_i \leq 1/\lambda_{\max}(Q)$, we then have*

$$\begin{aligned} &\mathbf{E}[\|\xi_{t+1}\|_2^2] \\ &\leq (1-\beta_i\lambda_{\min}(Q))^2\,\mathbf{E}[\|\xi_i\|_2^2] + \beta_i^2(G^2 + cN/m^2), \end{aligned} \tag{6.3}$$

where $\lambda_{\min}(Q) \geq \frac{8}{9}\lambda_{\min}(A^\intercal C^{-1}A) > 0$.

The detailed proof of Lemma 2 is in the appendix. Note that, in the stochastic programming literature, similar results rely on the assumption of a strongly convex (concave) objective function [Nesterov, 2013]. However, our results show that we do not need both the primal variable, θ, and dual variable, w, to be strongly convex (concave). This is because of the special form our objective function, i.e., our objective function is a quadratic optimization problem [Bertsekas, 1999].

Next, we provide the utility analysis in terms of different step size approaches. We first provide the utility bound when using a constant step size.

Theorem 2. *Let ξ_{N+1} be generated from Algorithm 1. If step size $\{\beta_i\}_{i=1}^{\infty}$ is constant, i.e., $\beta_i = \eta/N^k < 1/\lambda_{\min}(Q)$, where $k \in (0,1)$ and η is any positive real number, then*

$$\begin{aligned} &\mathbf{E}[\|\xi_{N+1}\|_2^2] \leq \\ &\underbrace{\left(1-\frac{\eta}{N^k}\lambda_{\min}(Q)\right)^{2N}\left(\mathbf{E}[\|\xi_1\|_2^2] - C_0\right)}_{term\ I} + \underbrace{C_0}_{term\ II}, \end{aligned} \tag{6.4}$$

where $C_0 = \dfrac{\eta(G^2 + cN/m^2)}{2N^k\lambda_{\min}(Q) - \eta\lambda_{\min}^2(Q)}$ and $\lambda_{\min}(Q) \geq \frac{8}{9}\lambda_{\min}(A^\intercal C^{-1}A) > 0$.

The detailed proof of Theorem 2 is in the appendix. Theorem 2 shows that there is a strong trade-off between accuracy and privacy when using a constant step size. Increasing privacy requires c to become larger, which increases the right side of (6.4). Furthermore, notice that term I has a linear rate of convergence, since we have $k \in (0,1)$, so that

$$\begin{aligned} &\left(1-\frac{\eta}{N^k}\lambda_{\min}(Q)\right)^{2N} \\ &= \left(1-\frac{\eta\lambda_{\min}(Q)}{N^k}\right)^{(N^k/(\eta\lambda_{\min}(Q)))\cdot(2\eta\lambda_{\min}(Q)N^{1-k})} \\ &\to e^{(-2\eta\lambda_{\min}(Q)N^{1-k})}, \end{aligned}$$

as N goes to infinity, while term II *diverges* since it has N in the numerator. Thus, initially term I dominates and we would expect rapid convergence. However, for large N, term II will eventually dominate, and the algorithm will diverge.

Next, we consider the convergence rate (utility analysis) when using a diminishing step size sequence. Theorem 3 shows that in this setting the divergent term is not present.

Theorem 3. *Let ξ_{N+1} be generated by Algorithm 1. If $\{\beta_i\}_{i=1}^{\infty}$ is a sequence of diminishing step sizes defined as $\beta_i = \frac{\eta}{\lambda_{\min}(Q)i}$, where $\eta > 1$, then:*

$$\begin{aligned}\mathbf{E}[\|\xi_{N+1}\|_2^2] \leq & \frac{\max\left\{\|\xi_1\|_2^2, \frac{\eta^2(G^2+cN/m^2)}{(\eta-1)\lambda_{\min}^2(Q)}\right\}}{N} \\ \leq & \underbrace{\frac{1}{N}\max\left\{\|\xi_1\|_2^2, \frac{\eta^2 G^2}{(\eta-1)\lambda_{\min}^2(Q)}\right\}}_{term\ III} \\ & + \underbrace{\frac{\eta^2 c}{m^2(\eta-1)\lambda_{\min}^2(Q)}}_{term\ IV},\end{aligned}$$

where $\lambda_{\min}(Q) \geq \frac{8}{9}\lambda_{\min}(A^\intercal C^{-1}A) > 0$.

The detailed proof of Theorem 3 is in the appendix. First, notice that Theorem 3 has the same accuracy-privacy trade-off as Theorem 2 due to its dependence on c. However, Theorem 3 shows that using diminishing step sizes results in a sublinear (i.e., worse than term I) convergence rate (term III), up to the information-theoretic limit (term IV). Since constant step sizes provide a better initial convergence rate (before term II dominates) than diminishing step sizes, initially using a constant step size would be preferable. However, after enough iterations, the noise in the gradient prevents the constant step size algorithm from converging to an optimal solution due to term II. Thus, in the long-term (when running many iterations), using a diminishing step size will produce a better solution. It should be also noted that Bassily et al. [2014] gave the optimal lower bound of utility for the problem of Empirical Risk Minimization (ERM) for both general convex and strongly convex loss function. Theorem 4 shows that for large N (i.e., $N = O(m^2\varepsilon^2/d)$), our method can attain that optimal lower bound, i.e., $\Omega(d/m^2\varepsilon^2)$.

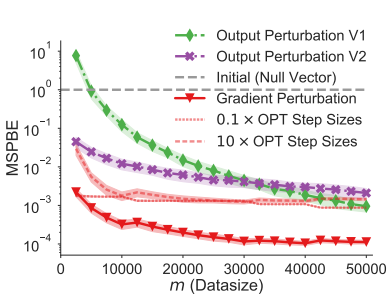
We now consider the influence of the mini-batch size (i.e., the number of transitions used in the sampled trajectory). Let τ_i be the number of transition samples which are used in iteration i. The approximation error according to the definition of primal-dual gradient in (4.1), can be written as $\Delta_i = \tau_i^{-1}\sum_{t=1}^{\tau_i}\Delta_t$, where Δ_i is the approximation error for only using one transition. Thus, if we replace Assumption 1 with the assumption $\mathbf{E}[\|\Delta_i\|_2^2] \leq G^2$, then we have that:

$$\mathbf{E}[\|\Delta_i\|_2^2] = \mathbf{E}\left[\left\|\frac{1}{\tau_i}\sum_{t=1}^{\tau_i}\Delta_t\right\|_2^2\right] \leq \frac{G^2}{\tau_i}.$$

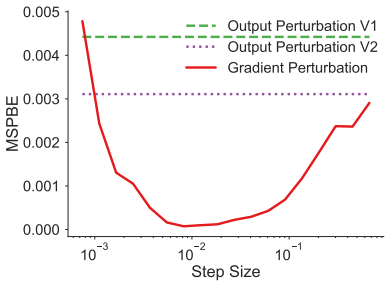
Thus, the variance bound is inversely proportion to the number of transitions used, and tighter variance bounds provide faster rates of convergence (as shown in Theorems 2 and 3). Therefore, the best choice is to use all of the transitions in the ith trajectory when computing $B_i(\theta_i, w_i)$.

7 Experimental Results

In this section we compare the performance of our proposed algorithm, *gradient perturbed off-policy evaluation* (GPOPE) (called gradient perturbation in this section to emphasize its difference from prior methods), with two prior methods, DP-LSW and DP-LSL [Balle et al., 2016] on two on-policy evaluation tasks. For clarity, we use output perturbation V1 to denote DP-LSW, and output perturbation V2 to denote DP-LSL. We then illustrate the behaviour of gradient perturbation on off-policy task, a common benchmark control tasks and on a more challenging HIV simulator.

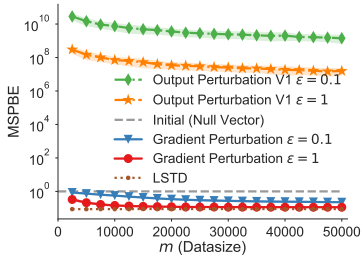


(a) Empirical comparison

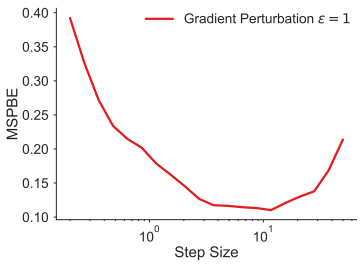


(b) Sensitivity of step sizes ($m = 3 \times 10^5$)

Figure 1: Experiments in on-policy chain

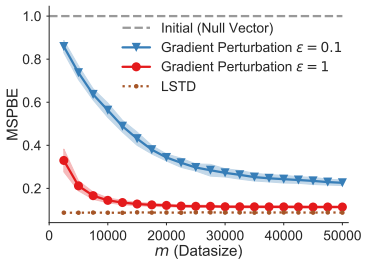


(a) Empirical comparison

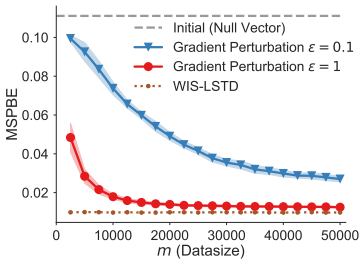


(b) Sensitivity of step sizes ($m = 3 \times 10^5$)

Figure 2: Experiments in on-policy mountain car



(a) On-policy mountain car



(b) Off-policy mountain car

Figure 3: Detailed empirical results in mountain car

The results we show in the following figures are all averaged over 100 trials and include standard deviation error bars, and we fix $\delta = 10^{-5}$ in all our experiments.

Synthetic chain domain In the first on-policy task, we consider a chain domain that consists of 40 states. The agent begins at a uniformly random state on the chain. In each state the agent has probability $p = 0.5$ to stay and probability $1 - p$ of advancing to the right. The agent receives a reward of 1 when reaching the final absorbing state, and 0 for all other states. We use $\gamma = 0.99$, $\varepsilon = 0.1$. We compared our algorithm, gradient perturbation, with output perturbation V1 and output perturbation V2 for on-policy evaluation in the tabular setting. This toy example illustrates one typical case in medical applications [Balle et al., 2016], where patients tend to progress through stages of recovery at different speeds, and past states are not typically revisited (partly because in the medical domain, states contain historical information about past treatments). The main result is shown in Figure 1(a), where MSPBE denotes *mean squared projected Bellman error* (a common measure of inaccuracy for policy evaluation in reinforcement learning [Sutton et al., 2009]), and where the datasize, m, is the number of trajectories used.

We use different step sizes (a hyper-parameter) for different m (amounts of data). Since the choice of step size cannot depend on the private data (this choice could leak information not captured by our analysis), we assume that the step size was tuned using similar public data—a common approach in differential privacy [Papernot et al., 2016]. For Figure 1(a), we assume that this method was used to obtain optimal step sizes. Our proposed method outperforms output perturbation V1 and output perturbation V2 in terms of accuracy by an order of magnitude.

In practice there may not always be public data similar to the private data, or the public data may differ slightly from the private data. This means that the optimal step size for the public data may not be the optimal step size for the private data. Therefore, it is necessary to test the robustness of our algorithm to changing hyper-parameters. Since step size is the only variable hyper-parameter for different m, Figure 1(a) also shows the results of using $0.1\times$ and $10\times$ the optimal step sizes. Even using these imprecise optimal step sizes, our proposed approach usually achieves better accuracy than prior methods. Figure 1(b) shows the accuracy when the step size varies, but the amount of data, m, is fixed. This shows that accuracy is stable for a very wide range of step sizes. We provide additional experiments in the appendix to further show the robustness of our algorithm to the step size parameter.

Mountain Car Next we performed these same experiments using the mountain car domain [Sutton and Barto, 1998] for on-policy policy evaluation. Mountain car is a popular RL benchmark problem with a two dimensional continuous state space, three discrete actions, and deterministic dynamics. We first used Q-learning with the fifth order Fourier basis [Konidaris et al., 2011] to obtain a decent policy to evaluate. We ran this policy to collect the trajectories that comprise the data set, and used our gradient perturbation algorithm and the output perturbation algorithms to estimate the value function for the learned policy.

Figure 2(a) shows the accuracy of our algorithm and compares with output perturbation V1 and *least squares temporal difference* [Bradtke and Barto, 1996, LSTD]. LSTD does not provide any privacy guarantees, and is presented here to show how close our algorithm is in accuracy to non-private methods. Note that output perturbation V2 fails to guarantee differential privacy for MDPs with continuous states or actions. While Figure 2(a) shows that our proposed gradient perturbation algorithm improves upon existing methods by orders of magnitude, Figure 3(a) provides a zoomed in view of the same plot to show the speed with which our algorithm converges when using different privacy settings. Similar to the chain domain, we show the robustness of our algorithm to step sizes in Figure 2(b), and present additional experiments in the appendix.

We also tested our algorithm on the mountain car domain for *off-policy* evaluation. Since LSTD is an on-policy algorithm, here we compare to a non-private off-policy variant of LSTD, called WIS-LSTD [Mahmood et al., 2014]. Note that output perturbation methods fail to guarantee differential privacy for off-policy evaluation, and so we only evaluate our algorithm in this part. Figure 3(b) shows the result of gradient perturbation for off-policy evaluation for the mountain car domain with different privacy settings. The behavior policy, π_b, is the policy learned by Q-learning, and the evaluated policy is the uniform policy. Despite being off-policy (which usually increases data requirements relative to on-policy problems), our algorithm's performances in Figures 3(a) and 3(b) are remarkably similar.

HIV simulator We also evaluate our approach on an HIV treatment simulation domain. This simulator was first introduced by Ernst et al. [2006], and consists of six features describing the state of the patient and four possible actions. Compared with the two domains above, this simulator is much closer to the practical medical treatment design, and its dynamics are more complex.

Figure 4 shows the results on the HIV simulator. We obtain the policy that is evaluated using Q-learning, and use a policy that is softmax w.r.t. the optimal Q function as the behavior policy. We use relative MSPBE in Figure 4, which normalizes MSPBE using the average reward ($\sim 10^5$) of the evaluated policy.

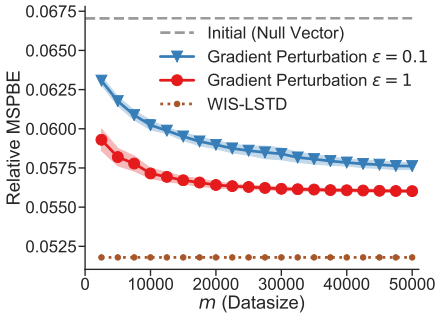


Figure 4: Off-policy HIV domain

8 Discussion and Conclusion

To protect individual privacy when applying reinforcement learning algorithms to sensitive training data, we present the first differentially private algorithm for off-policy evaluation. Our approach extends on the TD methods and comes with a privacy analysis and a utility (convergence rate) analysis. The utility guarantee shows that the privacy cost can be diminished by increasing the size of training batches, and the privacy/utility trade-off can be optimized by using a decaying step size sequence. In our experiments, our algorithm, gradient Perturbed off-policy evaluation (GPOPE), outperforms the previous methods in the restricted on-policy setting that prior work considers, can work well for both discrete and continuous domains, and guarantees differential privacy for both on-policy and off-policy evaluation problems. We also demonstrate the effectiveness of our approach in both common benchmark tasks and on a more challenging HIV simulator. Since our approach is based on gradient computations, it can be extended easily to more advanced first-order optimization methods, such as stochastic variance reduction methods [Du et al., 2017; Palaniappan and Bach, 2016], and momentum methods [Nesterov, 2013].

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6280–6290.

Balle, B., Gomrokchi, M., and Precup, D. (2016). Differentially private policy evaluation. In *International Conference on Machine Learning*, pages 2130–2138.

Bassily, R., Smith, A., and Thakurta, A. (2014). Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. *arXiv preprint arXiv:1405.7085*.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3):33–57.

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. (2017). Stochastic variance reduction methods for policy evaluation. *arXiv preprint arXiv:1702.07944*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer.

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

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

Ernst, D., Stan, G.-B., Goncalves, J., and Wehenkel, L. (2006). Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Decision and Control, 2006 45th IEEE Conference on*, pages 667–672. IEEE.

Konidaris, G., Osentoski, S., and Thomas, P. S. (2011). Value function approximation in reinforcement learning using the fourier basis. In *AAAI*, volume 6, page 7.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. (2015). Finite-sample analysis of proximal gradient td algorithms. In *UAI*, pages 504–513.

Mahmood, A. R., Hasselt, H., and Sutton, R. S. (2014). Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems 27*.

Meyer, C. D. (2000). *Matrix analysis and applied linear algebra*, volume 71. Siam.

Mironov, I. (2017). Renyi differential privacy. *arXiv preprint arXiv:1702.07476*.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Palaniappan, B. and Bach, F. (2016). Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, pages 1416–1424.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. (2016). Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

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

Raghu, A., Komorowski, M., Ahmed, I., Celi, L., Szolovits, P., and Ghassemi, M. (2017). Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*.

Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 245–248. IEEE.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM.

Theocharous, G., Thomas, P. S., and Ghavamzadeh, M. (2015). Personalized ad recommendation systems for life-time value optimization with guarantees. In *IJCAI*, pages 1806–1812.

Wang, D., Ye, M., and Xu, J. (2017). Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2719–2728.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. (2018). Subsampled r\'enyi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*.

Appendix

A Relationships Among Parameters

Table 1 shows the relationships among privacy parameters $\varepsilon, \delta, \sigma$, the total number of iterations N, and the size of dataset m. We use the color of red to denote negatively related, green to denote positively related. For example, if ε is decreased, and only δ is changed, then δ must be increased. Similarly, if size of dataset m is increased, and only ε changed, then ε must be decreased.

	ε	δ	N	σ	m
ε					
δ					
N					
σ					
m					

Table 1: Relationship Matrix among ε, δ, N, σ, m (red denotes negatively related, green denotes positively related)

B Proofs in Privacy Analysis

In this section we provide a detailed analysis of the privacy guarantee of our algorithm. We first introduce the following key definitions and properties we will use.

Definition 5 (Rényi Divergence)**.** *Let P and Q be probability distributions on ω. For $\alpha \in (1, \infty)$, we define the Rényi divergence of order α between P and Q as*

$$\begin{aligned} D_\alpha(P\|Q) =& \frac{1}{\alpha-1}\log\left(\int_\Omega P(x)^\alpha Q(x)^{1-\alpha}\mathrm{d}x\right) \\ =& \frac{1}{\alpha-1}\log\left(\mathbf{E}_{x\sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^\alpha\right]\right) \\ =& \frac{1}{\alpha-1}\log\left(\mathbf{E}_{x\sim P}\left[\left(\frac{P(x)}{Q(x)}\right)^{\alpha-1}\right]\right), \end{aligned}$$

where $P(\cdot)$ and $Q(\cdot)$ are the probability density functions of P and Q respectively.

Definition 6 (Rényi Differential Privacy)**.** *We say that a mechanism $\mathcal{M}$ is (α, ε)-Rényi Differential Privacy (RDP) with order $\alpha \in (1, \infty)$ if for all neighboring dataset d, d'*

$$D_\alpha(\mathcal{M}(d)\|\mathcal{M}(d')) \coloneqq \frac{1}{\alpha-1}\log\left(\mathbf{E}_{x\sim\mathcal{M}(d')(x)}\left[\left(\frac{\mathcal{M}(d)(x)}{\mathcal{M}(d')(x)}\right)^\alpha\right]\right) \leq \varepsilon.$$

Lemma 3 (Lemma 2.5 in [Bun and Steinke, 2016])**.** *Let $\nu, \mu \in \mathbb{R}^d$, $\sigma \in \mathbb{R}$, and $\alpha \in [1, \infty)$. Then,*

$$D_\alpha(\mathcal{N}(\mu, \sigma I_d)\|\mathcal{N}(\nu, \sigma I_d)) = \frac{\alpha\|\mu-\nu\|_2^2}{2\sigma^2}.$$

B.1 Proof of Lemma 1

Proof. Let fixed d' and let $d = d' \cup x_m$, where x_m denotes trajectory m with length τ_m. Without loss of generality, let $\bar{B}_t(x_m) = \mathbf{e_1}$ and $\sum_{x_i \in d'} \bar{B}_t(x_m) = \mathbf{0}$. Thus $\mathcal{M}(d)$ and $\mathcal{M}(d')$ are distributed identically except for the first coordinate. Hence we transfer it to a one-dimension problem. Let μ_0 denote the probability density function of $\mathcal{N}(0, \sigma^2)$ and let μ_1 denote probability density function of $\mathcal{N}(1, \sigma^2)$. Thus,

$$\begin{aligned} &\mathcal{M}(d) \sim \mu_0, \\ &\mathcal{M}(d') \sim \mu \coloneqq \begin{cases} \mu_0, & \text{w.p. } 1-q \\ \mu_1, & \text{w.p. } q \end{cases}, \end{aligned}$$

where $q = 1/m$. To avoid the difficulty of analysis this complex mixture distribution, we decompose $\mathcal{M}$ as a composition of two algorithm $\mathcal{M}_0 \circ \texttt{subsample}$ which is defined as: (1) `subsample`: subsample without replacement 1 datapoint of the dataset, and (2) a randomized algorithm taking the subsampled dataset as the input. Next, we use the amplification properties for RDP via subsampling to obtain $\alpha_{\mathcal{M}}(\lambda)$.

First, we analysis the RDP for $\mathcal{M}_0$ as:

$$\begin{aligned} \varepsilon_0(\alpha) \coloneqq& D_\alpha(\mathcal{M}_0(d) \| \mathcal{M}_0(d')) \\ =& D_\alpha(\mu_0 \| \mu_1) \\ =& \frac{\alpha}{2\sigma^2}, \end{aligned} \tag{B.1}$$

where the last equation follows from Lemma 3.

By the amplification properties for RDP via subsampling (Theorem 9 in [Wang et al., 2018]), we can obtain $\mathcal{M} = \mathcal{M}_0 \circ \texttt{subsample}$ is $(\alpha, \varepsilon(\alpha))$-RDP,

$$\varepsilon(\alpha) \leq \frac{1}{\alpha - 1} \log \left(1 + \frac{\alpha(\alpha-1)}{2m^2} \min \left\{ 4 \left(e^{\varepsilon_0(2)} - 1 \right), e^{\varepsilon_0(2)} \min \left\{ 2, \left(e^{\varepsilon_0(\infty)} - 1 \right)^2 \right\} \right\} \right), \tag{B.2}$$

where $\varepsilon_0(\alpha)$ is defined in (B.1), and we ignored the higher-order terms since $m \gg 1$. According to the definition of RDP, we have

$$D_\alpha(\mathcal{M}(d) \| \mathcal{M}(d')) \leq \varepsilon(\alpha). \tag{B.3}$$

Since the Gaussian mechanism does not have a bound $\varepsilon_0(\infty)$, term $\min \left\{ 4 \left(e^{\varepsilon_0(2)} - 1 \right), e^{\varepsilon_0(2)} \min \left\{ 2, \left(e^{\varepsilon_0(\infty)} - 1 \right.\right.\right.$ in the bound (B.2) can be simplified as $\min \left\{ 4 \left(e^{\varepsilon_0(2)} - 1 \right), 2e^{\varepsilon_0(2)} \right\}$, where $\varepsilon_0(2) = 1/\sigma^2$ according to (B.1).

By properties of Rényi divergence, we have

$$\begin{aligned} \alpha_{\mathcal{M}}(\lambda) =& \lambda D_{\lambda+1}(\mathcal{M}(d) \| \mathcal{M}(d')) \\ \leq& \lambda \varepsilon(\lambda + 1) \\ \leq& \lambda \frac{1}{\lambda} \log \left(1 + \frac{\alpha(\alpha-1)}{2m^2} \min \left\{ 4 \left(e^{\varepsilon_0(2)} - 1 \right), 2e^{\varepsilon_0(2)} \right\} \right) \\ =& \log \left(1 + \frac{\lambda(\lambda+1)}{2m^2} \min \left\{ 4 \left(e^{1/\sigma^2} - 1 \right), 2e^{1/\sigma^2} \right\} \right) \\ \leq& \frac{\lambda(\lambda+1)}{2m^2} \min \left\{ 4 \left(e^{1/\sigma^2} - 1 \right), 2e^{1/\sigma^2} \right\}, \end{aligned}$$

where the second inequality follows from (B.3), the third inequality follow from (B.2). This completes the proof.

□

B.2 Proof of Theorem 1

We first introduce a useful theorem for the calculation of

Theorem 4 (Theorem 2 in [Abadi et al., 2016]). *Let $\alpha_{\mathcal{M}}(\lambda)$ be the moments accountant of a randomized mechanism $\mathcal{M}$.*

1. ***[Composability]*** *Suppose that a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where $\mathcal{M}_i$: $\prod_{j=1}^{i-1} \mathcal{Y}_j \times \mathcal{D} \to \mathcal{Y}_i$. Then, for any λ*

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{M}_i}(\lambda).$$

2. ***[Tail bound]*** *For any $\varepsilon > 0$, the mechanism $\mathcal{M}$ is (ε, δ)-differentially private for*

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon).$$

Theorem 4 enables us to compute and bound the moments accountant, $\alpha_{\mathcal{M}}(\lambda)$, at each iteration and sum them to bound the moments of the whole algorithm. This allows us to convert the moments bound to the (ϵ, δ)-differential privacy guarantee.

Given Lemma 1 and Theorem 4, the proof that Algorithm 1 is (ε, δ)-differential private can be obtained directly, because Lemma 1 bounds the moments of each iteration, and we can calculate the moments accountant of our whole algorithm by applying Theorem 4. The proof of Theorem 1 is as follows.

Proof. We first analysis the term $\left\{4\left(e^{1/\sigma^2}-1\right), 2e^{1/\sigma^2}\right\}$ in Lemma 1. If $\sigma^2 \geq 1/\ln 2$, we have

$$\begin{aligned}
&\min\left\{4\left(e^{\varepsilon_0(2)}-1\right), 2e^{\varepsilon_0(2)}\right\} \\
=&\min\left\{4\left(e^{1/\sigma^2}-1\right), 2e^{1/\sigma^2}\right\} \\
=&4\left(e^{1/\sigma^2}-1\right) \\
\leq&\frac{8}{\sigma^2},
\end{aligned}$$

and

$$\alpha_{\mathcal{M}}(\lambda) \leq \frac{4\lambda(\lambda+1)}{m^2\sigma^2}.$$

If $\sigma^2 < 1/\ln 2$, we have

$$\begin{aligned}
&\min\left\{4\left(e^{\varepsilon_0(2)}-1\right), 2e^{\varepsilon_0(2)}\right\} \\
=&\min\left\{4\left(e^{1/\sigma^2}-1\right), 2e^{1/\sigma^2}\right\} \\
=&2e^{1/\sigma^2},
\end{aligned}$$

and

$$\alpha_{\mathcal{M}}(\lambda) \leq \frac{\lambda(\lambda+1)e^{1/\sigma^2}}{m^2}.$$

By Theorem 4 and Lemma 1, the log moment of Algorithm 1 can be bounded as $\alpha(\lambda) \leq 4N\lambda^2/(m^2\sigma^2)$. (assuming we set σ explicitly to satisfy $\sigma \geq \sqrt{1/\ln 2} \approx 1.201$). In order to use Theorem 4 to guarantee the (ε, δ)-differential privacy of Algorithm 1, we need λ satisfy

$$\lambda \leq \sigma^2 \log\left(\frac{m}{\sigma}\right)$$

and σ to satisfy

$$\delta = \min_{\lambda} \exp\left(\frac{N\lambda^2}{m^2\sigma^2} - \lambda\varepsilon\right) \leq \exp\left(-\frac{m^2\sigma^2\varepsilon^2}{4N}\right).$$

Thus, when $\varepsilon = c_1 N/m^2$, all these conditions are satisfied by setting

$$\sigma = \frac{c_2\sqrt{N\log(1/\delta)}}{m\varepsilon},$$

for some explicit constants c_1 and c_2. □

C Proofs in Utility Analysis

In this section we provide detailed proofs of utility analysis. First, we derive properties of each iteration of our algorithm. We assume that all transitions in the sampled trajectory are used in this subsection (as in the GPOPE algorithm).

We first provide the proof of lemma 2.

C.1 Proof of Lemma 2

Proof. The updates have the following iteration

$$\begin{aligned}
&\begin{bmatrix}\theta_{i+1}\\ w_{i+1}\end{bmatrix}\\
&= \begin{bmatrix}\theta_i\\ w_i\end{bmatrix} - \beta_i B_i(\theta_i, w_i),\\
&= \begin{bmatrix}\theta_i\\ w_i\end{bmatrix} - \beta_i B(\theta_i, w_i) + \beta_i\Delta_i,\\
&= \begin{bmatrix}\theta_i\\ w_i\end{bmatrix} - \beta_i\left(\begin{bmatrix}0 & -A^\intercal\\ A & C\end{bmatrix}\begin{bmatrix}\theta_i\\ w_i\end{bmatrix} - \begin{bmatrix}0\\ b\end{bmatrix}\right) + \beta_i\Delta_i.
\end{aligned}$$

Subtracting optimal solution (θ^*, w^*) (defined in (6.1)) from both sides and using the first order optimally condition, we obtain

$$\begin{aligned}
&\begin{bmatrix}\theta_{i+1} - \theta^*\\ w_{i+1} - w^*\end{bmatrix}\\
&= \begin{bmatrix}\theta_t - \theta^*\\ w_t - w^*\end{bmatrix} - \beta_i\begin{bmatrix}0 & -A^\intercal\\ A & C\end{bmatrix}\begin{bmatrix}\theta_t - \theta^*\\ w_t - w^*\end{bmatrix} + \beta_i\Delta_i.
\end{aligned}$$

The analysis of the convergence rate examines the difference between the current parameters and the optimal solution. Note the residual vector ξ_i in (6.2), obeys the following iteration:

$$\xi_{i+1} = (I - \beta_i Q)\xi_i + \beta_i\Delta_i, \tag{C.1}$$

where Q is also defined in (6.2). Taking the Euclidean norm of each side of Eq. (C.1), we obtain

$$\begin{aligned}&\|\xi_{i+1}\|_2^2\\ =&\|(I-\beta_i Q)\xi_i\|_2^2+\beta_i^2\|\Delta_i\|_2^2+2\langle(I-\beta_i Q)\xi_i,\beta_i\Delta_i\rangle,\end{aligned}\tag{C.2}$$

which follows from the rule that $\|a+b\|_2^2=\|a\|_2^2+\|b\|_2^2+2\langle a,b\rangle$. Taking the expectation of both sides of Eq. (C.2), we obtain

$$\begin{aligned}&\mathbf{E}[\|\xi_{i+1}\|_2^2]\\ \overset{(a)}{=}&\mathbf{E}[\|(I-\beta_i Q)\xi_i\|_2^2]+\beta_i^2\mathbf{E}[\|\Delta_i\|_2^2]\\ \overset{(b)}{\leq}&\mathbf{E}[\|I-\beta_i Q\|_S^2\|\xi_i\|_2^2]+\beta_i^2\mathbf{E}[\|\Delta_i\|_2^2]\\ =&\|I-\beta_i Q\|_S^2\mathbf{E}[\|\xi_i\|_2^2]+\beta_i^2\mathbf{E}[\|\Delta_i\|_2^2]\end{aligned}\tag{C.3}$$

where $\|I-\beta_i Q\|_S$ denotes the spectral norm of $(I-\beta_i Q)$, i.e., the square root of the maximum eigenvalue of $(I-\beta_i Q)$, and where **(a)** holds because $\mathbf{E}[\Delta_i]=0$, **(b)** holds by a property of spectral norm [Meyer, 2000].

In order to obtain $\|I-\beta_i Q\|_S$, we calculate the maximum eigenvalue of Q (we use $\lambda_{\max}(\cdot)$ to denote maximum eigenvalue), and the minimum eigenvalue of Q (we use $\lambda_{\min}(\cdot)$ to denote minimum eigenvalue). Using the eigen-analysis of Q in the previous work, in [Du et al., 2017], Appendix A.3, we have

$$\begin{aligned}\lambda_{\max}(Q)\leq&9\kappa(C)\lambda_{\max}(A^\intercal C^{-1}A),\\ \lambda_{\max}(Q)\geq&\frac{8}{9}\lambda_{\min}(A^\intercal C^{-1}A)>0,\end{aligned}$$

where we use $\kappa(\cdot)$ to denote $\lambda_{\max}(\cdot)/\lambda_{\min}(\cdot)$.

Choosing $\beta_i\leq 1/\lambda_{\max}(Q)$, then we have that

$$\|I-\beta_\theta Q\|_S^2=\left(1-\beta_i\lambda_{\min}(Q)\right)^2.\tag{C.4}$$

Substituting (C.4) into (C.3),

$$\begin{aligned}&\mathbf{E}[\|\xi_{i+1}\|_2^2]\\ \leq&\left(1-\beta_i\lambda_{\min}(Q)\right)^2\mathbf{E}[\|\xi_i\|_2^2]+\beta_i^2\mathbf{E}[\|\Delta_i\|_2^2]\\ \leq&\left(1-\beta_i\lambda_{\min}(Q)\right)^2\mathbf{E}[\|\xi_i\|_2^2]+\beta_i^2(G^2+cN/m^2),\end{aligned}$$

where the second inequality follows from the assumption of variance bound in Assumption 1. □

We now prove the utility theorems using lemma 2.

C.2 Proof of Theorem 2

Proof. Let $\beta_i=\beta$. Then (6.3) leads to

$$\begin{aligned}&\mathbf{E}[\|\xi_{i+1}\|_2^2]-\frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}\\ \leq&\left(1-\beta\lambda_{\min}(Q)\right)^2\mathbf{E}[\|\xi_i\|_2^2]+\beta^2(G^2+cN/m^2)-\frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}\\ =&\left(1-\beta\lambda_{\min}(Q)\right)^2\left(\mathbf{E}[\|\xi_i\|_2^2]-\frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}\right).\end{aligned}$$

Thus, recursively we have

$$\begin{aligned}
&\mathbf{E}[\|\xi_{N+1}\|_2^2] - \frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)} \\
\leq &(1-\beta\lambda_{\min}(Q))^{2N}\left(\mathbf{E}[\|\xi_1\|_2^2] - \frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}\right)\mathbf{E}[\|\xi_{N+1}\|_2^2] \\
\leq &(1-\beta\lambda_{\min}(Q))^{2N}\cdot\left(\mathbf{E}[\|\xi_1\|_2^2] - \frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}\right) + \frac{\beta(G^2+cN/m^2)}{2\lambda_{\min}(Q)-\beta\lambda_{\min}^2(Q)}.
\end{aligned}$$

where β can be set as $\beta = \eta/N^k$, for $\forall\ k \in (0,1)$, since

$$\lim_{N\to+\infty}\left(1-1/N^k\right)^{2N} = 1$$

for $k \in [1,+\infty)$. □

C.3 Proof of Theorem 3

Proof. Under $\beta_i = \frac{\eta}{\lambda_{\min}(Q)i}$, (6.3) leads to

$$\begin{aligned}
&\mathbf{E}[\|\xi_{i+1}\|_2^2] \\
\leq &\left(1-\frac{\eta}{i}\right)^2\mathbf{E}[\|\xi_i\|_2^2] + \frac{\eta^2(G^2+cN/m^2)}{\lambda_{\min}^2(Q)i^2}
\end{aligned} \tag{C.5}$$

Let $H(\eta) = \max\left\{\|\xi_1\|_2^2, \frac{\eta^2(G^2+cN/m^2)}{(\eta-1)\lambda_{\min}^2(Q)}\right\}$, so that $\mathbf{E}[\|\xi_i\|_2^2] \leq H(\eta)/i$ by induction. First, note that $\mathbf{E}[\|\xi_1\|_2^2] \leq H(\eta)$. So, if we assume that the convergence rate holds with i, we only need to show that it holds with $i+1$. By (C.5), we have

$$\begin{aligned}
&\mathbf{E}[\|\xi_{i+1}\|_2^2] \\
\leq &\left(1-\frac{\eta}{i}\right)\frac{H(\eta)}{i} + \frac{(\eta-1)H(\eta)}{i^2} \\
\leq &\frac{(i-1)H(\eta)}{t^2} \leq \frac{H(\eta)}{i+1}.
\end{aligned}$$

Thus, we obtain the rate of convergence with diminishing stepsize as

$$\begin{aligned}
&\mathbf{E}[\|\xi_{N+1}\|_2^2] \\
\leq &\frac{\max\left\{\|\xi_1\|_2^2, \frac{\eta^2(G^2+cN/m^2)}{(\eta-1)\lambda_{\min}^2(Q)}\right\}}{N} \\
\leq &\frac{1}{N}\max\left\{\|\xi_1\|_2^2, \frac{\eta^2G^2}{(\eta-1)\lambda_{\min}^2(Q)}\right\} + \frac{\eta^2c}{(\eta-1)\lambda_{\min}^2(Q)}.
\end{aligned}$$

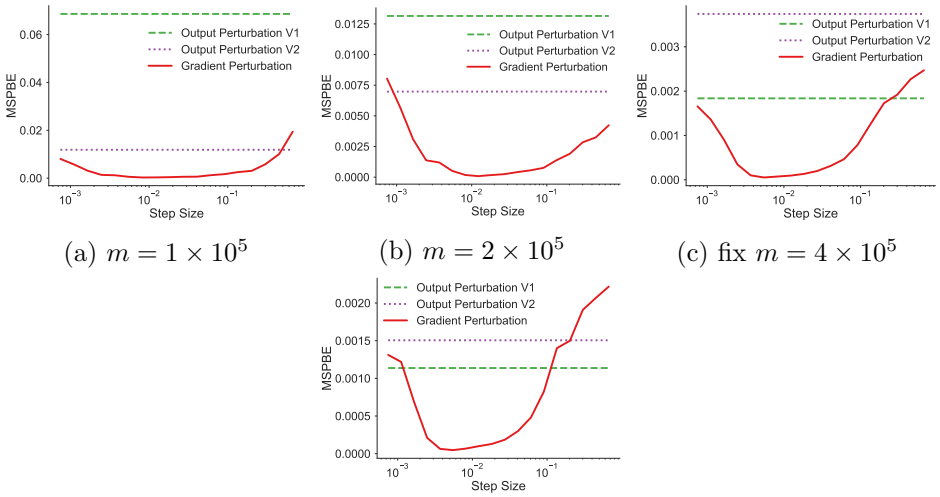
□

D Extra Figures

In this section, we provide extra figures from out experiments.

Figure 5 shows the sensitivity when the step size varies for the on-policy chain domain.

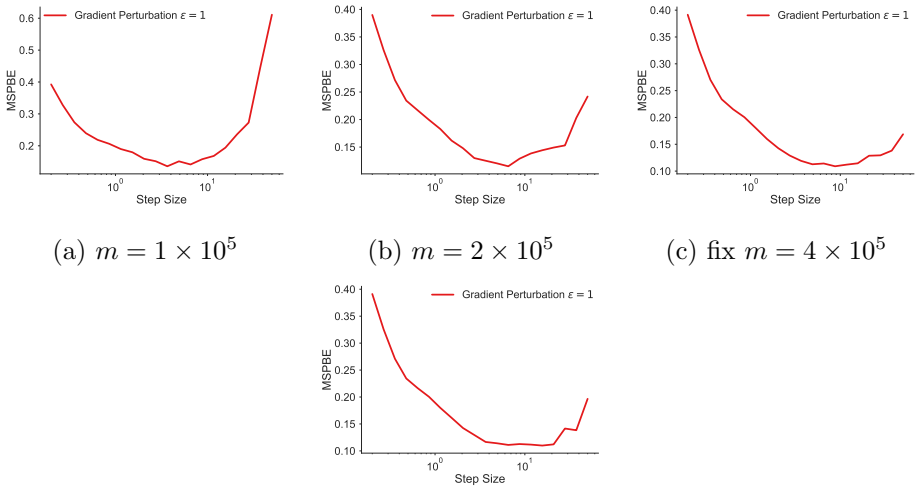
Figure 6 shows the results of additional testing the sensitivity of our algorithm to the step size parameter.



(a) $m = 1 \times 10^5$ (b) $m = 2 \times 10^5$ (c) fix $m = 4 \times 10^5$

(d) $m = 5 \times 10^5$

Figure 5: Sensitivity of step sizes in on-policy chain



(a) $m = 1 \times 10^5$ (b) $m = 2 \times 10^5$ (c) fix $m = 4 \times 10^5$

(d) $m = 5 \times 10^5$

Figure 6: Sensitivity of step sizes in on-policy mountain car