DIFFERENTIALLY PRIVATE DEEP MODEL-BASED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We address private deep offline reinforcement learning (RL), where the goal is to train a policy on standard control tasks that is differentially private (DP) with respect to individual trajectories in the dataset. To achieve this, we introduce PRIMORL, a model-based RL algorithm with formal differential privacy guarantees. PRIMORL first learns an ensemble of trajectory-level DP models of the environment from offline data. It then optimizes a policy on the penalized private model, without any further interaction with the system or access to the dataset. In addition to offering strong theoretical foundations, we demonstrate empirically that PRIMORL enables the training of private RL agents on offline continuous control tasks with deep function approximations, whereas current methods are limited to simpler tabular and linear Markov Decision Processes (MDPs). We furthermore outline the trade-offs involved in achieving privacy in this setting.

1 INTRODUCTION

Despite Reinforcement Learning's (RL) notable advancements in various tasks, there have been many obstacles to its adoption for the control of real systems in the industry. In particular, online interaction with the system may be impractical or hazardous in real-world scenarios. Offline RL (Levine et al., 2020) refers to the set of methods enabling the training of control agents from static datasets. While this paradigm shows promise for real-world applications, its deployment is not without concerns. Many studies have warned of the risk of privacy leakage when deploying machine learning models, as these models can memorize part of the training data. For instance, Rigaki & Garcia (2020) review the proliferation of sophisticated privacy attacks. Of the various attack types, membership inference attacks (Shokri et al., 2017) stand out as the most prevalent. In these attacks, the adversary, with access to a black-box model trainer, attempts to predict whether a specific data point was part of the model's training data. Unfortunately, reinforcement learning is no exception to these threats. In a recent contribution, Gomrokchi et al. (2023) exploit the temporal correlation of RL samples to perform powerful membership inference attacks using convolutional neural classifiers. More precisely, they demonstrate that given access to the output policy, an adversary can learn to infer the presence of a specific trajectory — which is the result of a sequence of interactions between a user and the system — in the training dataset with great accuracy. The threat of powerful membership inference attacks is particularly concerning in reinforcement learning, where a trajectory can unveil sensitive user information. For instance, when using RL to train autonomous vehicles (Kiran et al., 2022), we need to collect a large number of trips that may disclose locations and driving habits. Similarly, a browsing journey collected to train a personalized recommendation engine may contain sensitive information about the user's behavior (Zheng et al., 2018). In healthcare, RL's potential for personalized treatment recommendation (Liu et al., 2022) underscores the need to safeguard patients' treatment and health history.

A large body of work has focused on protecting against privacy leakages in machine learning. Differential Privacy (DP), which allows learning models without exposing sensitive information about any particular user in the training dataset, has emerged as the gold standard. While successfully applied in various domains, such as neural network training (Abadi et al., 2016) and multi-armed bandits (Tossou & Dimitrakakis, 2016), extending differential privacy to reinforcement learning poses challenges. In particular, the many ways of collecting data and the correlated nature of training samples resulting from online interactions make it difficult to come up with a universal and meaningful DP definition in this setting. Several attempts, such as joint differential privacy (Vietri et al., 2020),

have been made, but extending definitions from bandits, they do not scale to large state and action spaces. Indeed, their scope is limited to tabular and linear Markov decision processes (MDPs) with finite horizon, which make them not suitable to the tasks typically encountered in deep RL.

In addition to its practical significance, the offline RL setting arguably offers a more natural framework for privacy. In contrast to online RL, which inherently blends input and output data throughout the process, an offline RL method can be seen as a black-box randomized algorithm h taking in as input a fixed dataset $\mathcal{D}$, partitioned in trajectories, and outputting a policy $\hat{\pi}$. An adversary having access to $\hat{\pi}$ may successfully learn to infer the membership of a specific trajectory in $\mathcal{D}$, which can, as emphasized before, reveal sensitive user information. Hence, similarly to Qiao & Wang (2023a), we use the following informal DP definition for offline RL, which we refer to as *trajectory-level differential privacy* (TDP): adding or removing a single trajectory from the input dataset of an offline RL algorithm must not impact significantly the distribution of the output policy. If Qiao & Wang (2023a) have proposed the first private algorithms for offline RL, building on value iteration methods, their approach is also restricted to finite-horizon tabular and linear MDPs, limiting its applicability. It is not suited for standard control tasks such as those from Gym (Brockman et al., 2016) and the DeepMind Control Suite (Tassa et al., 2018), which often require deep neural function approximations. This leaves a huge gap between the current private RL literature and real-world applications. In this work, we are, according to our knwoledge, the first to tackle deep RL tasks in the infinite-horizon discounted setting under differential privacy guarantees, paving the way for enhanced applications of private RL in more complex scenarios.

Contributions. While previous work in the differentially private RL literature is essentially restricted to finite-horizon tabular and linear Markov decision processes (MDPs), with experiments reduced to simple numerical simulations, this work is the first attempt to tackle deep RL problems in the infinite-horizon discounted setting. To this end, we use a model-based approach, named PRIMORL, which exploits a model of the environment to generalize the information contained in the offline data to unexplored regions of the state-action space. We introduce a method for training an ensemble of models with differential privacy guarantees at the trajectory-level, and mitigate the increased model uncertainty during model-based policy optimization. In addition to offering strong theoretical foundations and formal privacy guarantees, we show empirically that PRIMORL can train private policies with competitive privacy-performance trade-offs on standard continuous control benchmarks, demonstrating the potential of our approach.

2 RELATED WORK

Offline RL (Levine et al., 2020; Prudencio et al., 2022) focuses on training agents without further interactions with the system, making it essential in scenarios where data collection is impractical (Singh et al., 2022; Liu et al., 2020; Kiran et al., 2022). Model-based RL (Moerland et al., 2023) can further reduce costs or safety risks by using a learned environment model to simulate beyond the collected data and improve sample efficiency (Chua et al., 2018). Argenson & Dulac-Arnold (2021) demonstrate that model-based offline planning, where the model is trained on a static dataset, performs well in robotic tasks. However, offline RL faces challenges like *distribution shift* (Fujimoto et al., 2019), where the limited coverage of the dataset can lead to inaccuracies in unexplored state-action regions, affecting performance. Methods like MOPO (Yu et al., 2020), MOREL (Kidambi et al., 2020), and COUNT-MORL (Kim & Oh, 2023) address this by penalizing rewards based on model uncertainty, achieving strong results on offline benchmarks. Still, key design choices in offline MBRL require further exploration, as highlighted by Lu et al. (2022).

On the other hand, Differential Privacy (DP), established by Dwork (2006), has become the standard for privacy protection. Recent research has focused on improving the privacy-utility trade-off, with relaxations of DP and advanced composition tools enabling tighter privacy analyses (Dwork et al., 2010; Dwork & Rothblum, 2016; Bun & Steinke, 2016; Mironov, 2017a). Notably, DP-SGD (Abadi et al., 2016) has facilitated the development of private deep learning algorithms, despite ongoing practical challenges (Ponomareva et al., 2023). Concurrently, sophisticated attack strategies have underscored the necessity for robust DP algorithms (Rigaki & Garcia, 2020). Recent studies have shown that reinforcement learning (RL) is also vulnerable to privacy threats (Pan et al., 2019; Prakash et al., 2022; Gomrokchi et al., 2023). As RL is increasingly applied in personalized services (den Hengst et al., 2020), the need for privacy-preserving training techniques is critical. Although DP

has been successfully extended to multi-armed bandits (Tossou & Dimitrakakis, 2016; Basu et al., 2019), existing RL algorithms (*e.g.*, Vietri et al. (2020), Zhou (2022), Qiao & Wang (2023b)) with formal DP guarantees mainly apply to episodic tabular or linear MDPs and lack empirical validation beyond basic simulations. Moreover, private offline RL remains underexplored. Only Qiao & Wang (2023a) have proposed DP offline algorithms, which, while theoretically strong, are also restricted to finite-horizon tabular and linear MDPs. Consequently, no existing work has introduced DP methods that can handle deep RL environments in the infinite-horizon discounted setting, a critical step toward deploying private RL algorithms in real-world applications. With this work, we aim to fill this gap by proposing a differentially private, deep model-based RL method for the offline setting.

3 Preliminaries

3.1 Offline Model-Based Reinforcement Learning

We consider an infinite-horizon discounted MDP, that is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ where $\mathcal{S}$ and $\mathcal{A}$ are respectively the state and action spaces, $P : \mathcal{S} \times \mathcal{A} \longrightarrow \Delta(\mathcal{S})$ the transition dynamics (where $\Delta(\mathcal{X})$ denotes the space of probability distributions over $\mathcal{X}$), $r : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$ the reward function, $\gamma \in [0, 1)$ a discount factor and $\rho_0 \in \Delta(\mathcal{S})$ the initial state distribution. The dynamics satisfy the Markov property, *i.e.*, the next state s' only depends on current state and action. The goal is to learn a policy $\pi : \mathcal{S} \longrightarrow \Delta(\mathcal{A})$ maximizing the expected discounted return $\eta_{\mathcal{M}}(\pi) := \mathbb{E}_{\tau \sim \pi, \mathcal{M}}[R(\tau)]$, where $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$. The expectation is taken w.r.t. the trajectories $\tau = ((s_t, a_t, r_t))_{t \geq 0}$ generated by π in the MDP $\mathcal{M}$, *i.e.*, $s_0 \sim \rho_0$, $s_{t+1} \sim P(\cdot|s_t, a_t)$ and $a_t \sim \pi(\cdot|s_t)$.

In offline RL, we assume access to a dataset of K trajectories $\mathcal{D}_K = (\tau_k)_{k=1}^K$, where each $\tau_k = (s_t^{(k)}, a_t^{(k)}, r_t^{(k)})_{t \geq 0}$ has been collected with an unknown behavioral policy π^B. τ_k can be seen as the result of the interaction of a user u_k with the environment. The objective is then to learn a policy $\hat{\pi}$ from $\mathcal{D}_K$ (without any further interaction with the environment) which performs as best as possible in $\mathcal{M}$. To achieve this goal, we consider a model-based approach. In this context, we learn estimates of both the transition dynamics and the reward function, denoted $\hat{P}$ and $\hat{r}$ respectively, from the offline dataset $\mathcal{D}_K$. This results in an estimate of the MDP $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \hat{r}, \gamma, \rho_0)$. We can then use the model $\hat{\mathcal{M}}$ as a simulator of the environment to learn a policy $\hat{\pi}_{\hat{\mathcal{M}}}$, without further access to the dataset or interactions with the real environment modeled by $\mathcal{M}$. Note that if the policy $\hat{\pi}_{\hat{\mathcal{M}}}$ is trained to maximize the expected discounted return in the MDP model $\hat{\mathcal{M}}$, *i.e.*, $\hat{\pi}_{\hat{\mathcal{M}}} \in \operatorname{argmax} \eta_{\hat{\mathcal{M}}}(\pi)$, we eventually want to evaluate the policy in the true environment $\mathcal{M}$, that is using $\eta_{\mathcal{M}}$.

3.2 Differential Privacy

When learning patterns from a dataset, differential privacy (Dwork, 2006) protects against the leakage of sensitive information in the data by ensuring that the output of the algorithm does not change significantly when adding or removing a data point, as formally stated in Definition 3.1.

Definition 3.1. (ϵ, δ)*-differential privacy.* Given $\epsilon > 0$, $\delta \in [0, 1)$, a *mechanism* h (*i.e.*, a randomized function of the data) is (ϵ, δ)-DP if for any pair of datasets D, D' that differ in a most one element (referred to as *neighboring datasets*, and denoted $d(D, D') = 1$), and any subset $\mathcal{E}$ in h's range:

$$\mathbb{P}(h(D) \in \mathcal{E}) \leq e^{\epsilon} \cdot \mathbb{P}(h(D') \in \mathcal{E}) + \delta \ .$$

In particular, ϵ controls the strength of the privacy guarantees with larger ϵ, which increases as ϵ grows. To achieve (ϵ, δ)-DP, the standard approach is to add a zero-mean random noise to the output of the (non-private) function f, whose magnitude σ scales with $\Delta_\ell(f)/\epsilon$, where $\Delta_\ell(f) := \max_{d(D,D')=1} \|f(D) - f(D')\|_\ell$ is the sensitivity of f. One of the most used DP mechanisms is the *Gaussian mechanism*, which provably guarantees (ϵ, δ)-DP for $\epsilon, \delta \in (0, 1)$ by adding random noise from a Gaussian distribution with magnitude $\sigma = \epsilon^{-1}\sqrt{2\log(1.25/\delta)} \cdot \Delta_2(f)$. From such simple mechanisms, we can derive complex DP algorithms using the *sequential* and *parallel composition* properties of DP, as well as its *immunity to post-processing* (*i.e.*, if h is (ϵ, δ)-DP and g is data-independent, then $g \circ h$ remains (ϵ, δ)-DP).

The Gaussian mechanism is central to DP-SGD (Abadi et al., 2016), a learning algorithm that modifies classic SGD to ensure (approximate) differential privacy. By adding Gaussian noise to the

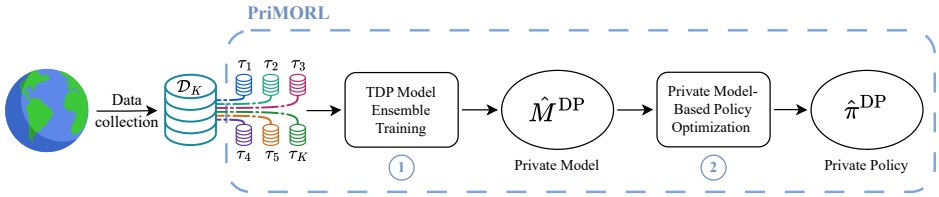


Figure 1: PRIMORL with its two main components: (1) private model training; (2) MBPO.

gradients and bounding their norm by a constant C, DP-SGD enables private neural network training (Ponomareva et al., 2023). To track the total privacy budget ϵ_{tot} spent by DP-SGD, Abadi et al. (2016) developed the *moments accounting* method that provides a $\left(\mathcal{O}(q\epsilon\sqrt{T}), \delta\right)$-DP guarantee, where q is the sampling ratio, T is the number of iterations, and ϵ is the privacy parameter. DP-SGD relies strongly on privacy amplification by sub-sampling (Balle et al., 2018). Studies have also analyzed error bounds for DP-SGD under various loss assumptions (Bassily et al., 2014; Kang et al., 2023).

4 Differentially Private Model-Based Offline Reinforcement Learning

We now describe our model-based approach for learning differentially private RL agents from offline data, which we call PRIMORL (for Private Model-Based Offline RL). After defining trajectory-level differential privacy (TDP) in offline RL (Section 4.1), we address the learning of a private model from offline data (Section 4.2). Finally, we demonstrate how we optimize a policy under the private model (Section 4.3). Exploiting the *post-processing* property of DP, we show that ensuring model privacy alone is enough to achieve a private policy. Figure 1 provides a high-level description of PRIMORL.

4.1 Trajectory-level Privacy in Offline Reinforcement Learning

We introduce the following formal definition for trajectory-level differential privacy (TDP) in offline RL. It can be seen as a reformulation of the definition used in Qiao & Wang (2023a), which is the first work to tackle differential privacy in this setting.

Definition 4.1. (ϵ, δ)-*TDP*. Let h be an offline RL algorithm, that takes as input an offline dataset and outputs a policy. Given $\epsilon > 0$ and $\delta \in (0, 1)$, h is (ϵ, δ)-TDP if for any trajectory-neighboring datasets $\mathcal{D}_K$, $\mathcal{D}_{K\setminus\{k\}}$, and any subset of policies Π:

$$\mathbb{P}\left(h(\mathcal{D}_K) \in \Pi\right) \leq e^{\epsilon} \cdot \mathbb{P}\left(h(\mathcal{D}_{K\setminus\{k\}}) \in \Pi\right) + \delta \ .$$

4.2 Model Learning with Differential Privacy

Following previous work (Yu et al., 2020; Kidambi et al., 2020), we jointly model the transition dynamics $\hat{P}$ and reward $\hat{r}$ with a Gaussian distribution $\hat{M}$ conditioned on the current state and action. Its mean and covariance are parameterized with neural networks $\theta = (\phi, \psi)$:

$$\hat{M}_\theta\left(\Delta_t^{t+1}(s), r_t | s_t, a_t\right) = \mathcal{N}\left(\mu_\phi(s_t, a_t), \Sigma_\psi(s_t, a_t)\right) \ .$$

To carry out uncertainty estimation (see Section 4.3), we train an ensemble of N models $\hat{M}_{\theta_i}$, $i \in [\![1, N]\!]$, all sharing the same architecture. The core aspect of PRIMORL, as illustrated in Figure 1, is therefore to learn a trajectory-level DP dynamics model ensemble.

A straightforward approach would be to train each model independently using DP-SGD, but this is inefficient. DP-SGD introduces excessive noise by perturbing gradients at the transition level, harming model performance. Moreover, since all models use the same dataset $\mathcal{D}_K$, the privacy budget

scales with N, increasing privacy leakage. A key contribution of our work, developed in Section 4.2.1, is thus to introduce a training method that 1) ensures privacy guarantees at the trajectory level and 2) efficiently manages the privacy budget across an ensemble of models.

4.2.1 TRAJECTORY-LEVEL DP TRAINING FOR MODEL ENSEMBLES

We identified that the idea behind the DP training method developed in McMahan et al. (2017), DP-FEDAVG, although originally designed to achieve client-level privacy in federated settings, could be effectively adapted for trajectory-level private training in offline RL. Specifically, our training data can be partitioned into trajectories in a manner analogous to how data is partitioned across clients in federated learning. This insight allowed us to leverage this approach to address the unique privacy challenges of our task, adapting it to model ensembles.

We present the resulting training procedure in Algorithm 1. The core idea behind TDP MODEL ENSEMBLE TRAINING is to draw, at each iteration t, a random subset $\mathcal{U}_t$ of the K trajectories (line 2). Each trajectory is drawn with probability q, so that the expected number of trajectories selected at each step is qK. For each trajectory $\tau_k \in \mathcal{U}_t$, the clipped gradients $\{\Delta_{i,k}^{\text{clipped}}(t)\}$ are then computed from τ_k's data only (line 3 to 7). Processing and clipping gradients per trajectory is essential to provide trajectory-level privacy, as this ensures that no trajectory will carry more weight than another in the optimization of the model. We later introduce ensemble-adapted clipping strategies to control the privacy budget over model ensembles, ensuring that the sensitivity of the ensemble gradient $\Delta_k^{\text{clipped}}(t) = \left(\Delta_{i,k}^{\text{clipped}}(t)\right)_{i=1}^{N}$ is bounded by C. We then compute an unbiased estimator of the subset gradient average whose sensitivity is bounded by C/qK (line 8). We can then apply the Gaussian mechanism with magnitude $\sigma = zC/qK$, where z controls the strength of the privacy guarantee ϵ, and update the ensemble model $\theta(t) = (\theta_i(t))_{i=1}^{N}$ with noisy gradient (line 9):

$$\theta(t+1) \longleftarrow \theta(t) + \Delta^{\text{avg}}(t) + \mathcal{N}\left(0_{Nd}, \sigma^2 I_{Nd}\right) \quad .$$

Algorithm 1 TDP MODEL ENSEMBLE TRAINING

1: **for** each iteration $t \in [\![0, T-1]\!]$ **do**
2: $\quad \mathcal{U}_t \leftarrow$ (sample with replacement trajectories from $\mathcal{D}_K$ with prob. q)
3: $\quad$ **for** each trajectory $\tau_k \in \mathcal{U}_t$ **do**
4: $\quad\quad$ Clone current models $\{\theta_i^{\text{start}}\}_{i=1}^{N} \leftarrow \{\theta_i(t)\}_{i=1}^{N}$
5: $\quad\quad \{\theta_{i,k}\}_{i=1}^{N} \leftarrow$ ENSCLIPGD $\left(\tau_k, \{\theta_i^{\text{start}}\}_{i=1}^{N}; C, \text{local epochs } E, \text{batch size } B\right)$
6: $\quad\quad \Delta_{i,k}^{\text{clipped}}(t) \leftarrow \theta_{i,k} - \theta_i^{\text{start}}, \;\; i = 1, ..., N$
7: $\quad$ **end for**
8: $\quad \Delta_i^{\text{avg}}(t) = \frac{\sum_{k \in \mathcal{U}_t} \Delta_{i,k}^{\text{clipped}}(t)}{qK}, \;\; i = 1, ..., N$
9: $\quad \theta(t+1) \leftarrow \theta(t) + \Delta^{\text{avg}}(t) + \mathcal{N}\left(0_{Nd}, \left(\frac{zC}{qK}\right)^2 I_{Nd}\right)$
10: **end for**

4.2.2 PRIVACY GUARANTEES FOR THE MODEL

We can now derive formal privacy guarantees for a model trained using Algorithm 1. A key challenge in our setting arises from training an ensemble of N models for uncertainty estimation, all using the same dataset $\mathcal{D}_K$. Treating each model independently, with separate clipping and noise addition, would be inefficient and significantly increase the privacy budget by composition. This could be mitigated by limiting the ensemble size, but at the cost of performance, as shown in Lu et al. (2022).

To address this challenge, we process all the gradients of the model ensemble simultaneously and distribute the global clipping norm C across all models, on the same principle as the per-layer clipping used in McMahan et al. (2017). Denoting $\Delta_{i,\ell}$ the gradient of layer ℓ for model i, we propose and experiment with two ensemble clipping strategies: **Flat Ensemble Clipping**, which clips the whole model gradient $\Delta_i = (\Delta_{i,\ell})_{\ell=1}^{L}$ with $C_i = C/\sqrt{N}$; and **Per Layer Ensemble Clipping**, which clips per-layer gradients $\Delta_{i,\ell}$ with $C_{i,\ell} = C/\sqrt{N \times L}$, so that $C = \sqrt{\sum_{i=1}^{N} C_i^2} = \sqrt{\sum_{i=1}^{N} \sum_{\ell=1}^{L} C_{i,\ell}^2}$.

For both strategies, we verify that that $\Delta_k^{\text{clipped}} = \left(\Delta_{i,k}^{\text{clipped}}\right)_{i=1}^{K}$ has sensitivity bounded by C (see Theorem 4.2's proof in appendix), and that the contribution of a given trajectory to the *model ensemble* is appropriately limited. Ensemble clipping eliminates the linear dependence of the privacy budget on the number of models. However, it does not entirely remove the negative impact of increasing N. Indeed, for a given noise level, a larger N requires a smaller clipping threshold C_i or $C_{i,\ell}$, which can degrade model convergence by losing too much information from the original gradient. Nevertheless, the clipping threshold scales with the square root of N, mitigating the impact to some extent.

We now formally derive the privacy guarantees for an ensemble of models trained with Algorithm 1. Mapping users in federating learning to trajectories in offline RL, we can directly adapt Theorem 1 from McMahan et al. (2018) to state that, with the sensitivity of clipped gradients $\Delta_{i,k}^{\text{clipped}}$ effectively bounded by C, the moments accounting method from Abadi et al. (2016) computes correctly the privacy loss of Algorithm 1 at trajectory-level for the noise multiplier $z = \sigma/\mathbb{C}$ with $\mathbb{C} = C/qK$. We can therefore use the moments accountant to compute, given $\delta \in (0,1)$, $z > 0$, $q \in (0,1)$ and $T \in \mathbb{N}$, the total privacy budget ϵ spent by Algorithm 1, and obtain (ϵ, δ)-TDP guarantees for our dynamics model, as stated in Theorem 4.2 (full proof in appendix).

Theorem 4.2. (ϵ, δ)-TDP guarantees for dynamics model. *Given $\delta \in (0,1)$, noise multiplier z, sampling ratio q and number of training iterations T, let $\epsilon := \epsilon^{MA}(z, q, T, \delta)$ be the privacy budget computed by the moments accounting method from (Abadi et al. (2016), more details in Section H.6). The dynamics model output by Algorithm 1 is (ϵ, δ)-TDP.*

4.3 Policy Optimization under a Private Model

Now that we learned a private model $\hat{M}$ from offline data, we use it as a simulator of the environment to learn a private policy $\hat{\pi}$ with a model-based policy optimization approach. The use of a private model and the privacy constraints on the end policy introduce additional challenges compared to the non-private case, as demonstrated in Section 4.3.1. We study solutions to mitigate the detrimental effects of private training on policy performance in Section 4.3.2, before deriving formal privacy guarantees for a policy learned under a private model in Section 4.3.3.

4.3.1 Impact of Privacy on Policy Optimization

It is first essential to examine the complexities of policy optimization in model-based offline RL and assess whether they are amplified in the private setting. A major challenge in model-based offline RL is to handle the discrepancy between the true and the learned dynamics when optimizing the policy. Indeed, model inaccuracies cause errors in policy evaluation that may be exploited, resulting in poor performance in the real environment. According to the Simulation Lemma (Kearns & Singh, 2002; Xu et al., 2020), the value evaluation error of a policy π in model-based RL can be decomposed into a *model error* term and a *policy distribution shift* term. Formally, denoting $\rho_P^{\pi^B}$ the state-action discounted occupancy measure of the data-collection policy π^B under the true MDP, if the model error is bounded as $\mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}}\left[D_{KL}\left(P(\cdot|s,a)\|\hat{P}(\cdot|s,a)\right)\right] \leq \varepsilon_m$ and the distribution shift is bounded as $\max_s D_{KL}\left(\pi(\cdot|s)\|\pi^B(\cdot|s)\right)\right) \leq \varepsilon_\pi$, then the value evaluation error of π is bounded as:

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\epsilon_m} + \frac{2\sqrt{2}}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ , \tag{1}$$

where $\hat{V}^\pi$ and V^π denote the value of π under the learned and the true dynamics, respectively. Controlling this quantity for an arbitrary π is crucial in our setting, as it ensures that the learned MDP is a reasonable simulator of the true environment. Moreover, (1) directly implies a bound on the sub-optimality gap, since $|V^\star - V^{\hat{\pi}}| \leq 2\sup_\pi |\hat{V}^\pi - V^\pi|$. Under some assumptions regarding the model loss function, Proposition 4.3 states the model error term in terms of the size N of the dataset.

Proposition 4.3. Value evaluation error in non-private offline MBRL. *Let the model loss function be L-Lipschitz and Δ-strongly convex, and assumptions from the simulation lemma hold. There is a stochastic convex optimization algorithm for learning the model and a constant M such that, with probability at least $1-\alpha$, and for sufficiently large N, the value evaluation error of π is bounded as:*

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M \cdot \frac{L\log^{1/2}(N/\alpha)}{\sqrt{\Delta N}} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ .$$

When we learn the model with differential privacy, we disrupt model convergence because of gradient clipping and noise. This likely results in a less accurate dynamics model (although it may help prevent overfitting in some cases) and increased value evaluation error. Intuitively, DP training impacts model error in (1) as a direct result of gradient perturbations: Bassily et al. (2014), in particular, shows that noisy gradient descent (GD) has increased excess risk compared to non-private GD. In the simpler case where the model is trained with a vanilla DP noisy GD algorithm, Proposition 4.4 states the value evaluation error under the private model.

Proposition 4.4. Value evaluation error in private offline MBRL. *Let assumptions from Proposition 4.3 hold. If the model is learned with (ϵ, δ)-DP gradient descent, then, with probability at least $1-\alpha$, there is a constant M' such that for large enough N, the value evaluation error of π:*

$$|\hat{V}^{\pi}_{DP} - V^{\pi}| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M' \cdot \frac{Ld^{1/4}\log(N/\delta) \cdot poly\log(1/\alpha)}{\sqrt{\Delta N \epsilon \alpha}} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_{\pi}} \ ,$$

where $\hat{V}^{\pi}_{\text{DP}}$ is the value of π under the privately learned dynamics. Comparing the value evaluation errors in Propositions 4.3 and 4.4 (both proven in appendix), we observe how DP training may degrade performance in MBRL. The private bound has an explicit dependence on the problem dimension d which is not present in the non-private bound, and the $\sqrt{\epsilon}$ factor in the denominator shows that the error will degrade with strong privacy guarantees. On the other hand, the distribution shift term does not depend on the learned dynamics and is therefore not affected by private training.

4.3.2 Mitigating Private Model Uncertainty

In Section 4.3.1, we showed that private model training impacts the reliability of our model for evaluating policies due to an increased dynamics error, which can lead to misjudging the quality of a policy in the true environment. In the non-private case, this is typically handled by penalizing the reward with a measure of the uncertainty of the model, denoted $u : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$. Therefore, if the model is believed to be unreliable at a given state-action pair (s, a) (*i.e.*, large $u(s, a)$), the possibly over-estimated reward will be corrected as:

$$\tilde{r}(s, a) = \hat{r}(s, a) - \lambda \cdot u(s, a) \ , \tag{2}$$

where λ is an hyperparameter. The policy is then optimized under the resulting pessimistic MDP $\tilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \tilde{r}, \gamma, \rho_0)$. MOPO (Yu et al., 2020), MOREL (Kidambi et al., 2020) and more recently COUNT-MORL (Kim & Oh, 2023) achieve impressive results on traditional offline RL benchmarks with this approach, using different heuristics to estimate model uncertainty.

As suggested by the simulation lemma, the valuation error can depend on both model error and distribution shift. However, we showed that private training only affects model error. Interestingly, Lu et al. (2022), which study design choices in offline model-based RL and the properties of different uncertainty estimators, find that the uncertainty measures proposed in the literature have a good correlation to model error, more so than with distribution shift. Therefore, we believe that existing measures are appropriate for mitigating the worse reliability of the model under private training. In particular, we consider the maximum aleatoric uncertainty $u_{\text{MA}}(s, a) = \max_{i \in [\![1,N]\!]} \|\Sigma_{\psi_i}(s, a)\|_F$ (Yu et al., 2020) and the maximum pairwise difference $u_{\text{MPD}}(s, a) = \max_{i,j \in [\![1,N]\!]} \|\mu_{\phi_i}(s, a) - \mu_{\phi_j}(s, a)\|_2$ (Kidambi et al., 2020). We compare both estimators (see Figure 3 in the appendix) and find that neither is consistently superior. However, we observe that the choice of estimator can affect performance on a specific task. In addition, it seems reasonable to moderately increase the reward penalty λ compared to the non-private case to take into account the greater uncertainty.

4.3.3 Private Policy Optimization

Given a choice of uncertainty estimator $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$, we now consider optimizing the policy within the pessimistic private MDP $\tilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \tilde{r}_u, \gamma, \rho_0)$, with $\tilde{r}_u = \hat{r}(s, a) - \lambda \cdot u(s, a)$. We use Soft Actor-Critic (SAC, Haarnoja et al. (2018))[1], a classic off-policy algorithm with entropy regularization, to learn the policy from $\tilde{\mathcal{M}}$, in line with existing approaches in the offline MBRL literature. Offline model-based methods typically mix real offline data from $\mathcal{D}_K$ with model data during policy learning (in MOPO, for instance, each batch contains 5% of real data). Here, however,

[1]This could be any model-based policy optimization or planning algorithm that does not use offline data.

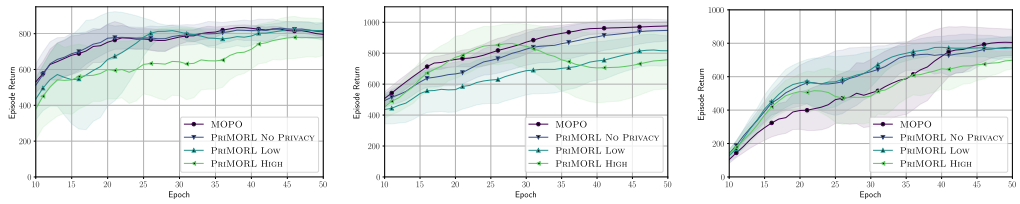


Figure 2: Learning curves on PENDULUM (*left*), BALANCE (*middle*) and SWINGUP (*right*).

we learn the policy exclusively from model data to avoid incurring privacy loss beyond what is needed to train the model, and thus control the privacy guarantees. Algorithm 4 in appendix provides a pseudo-code for SAC policy optimization in the pessimistic private MDP. Using the post-processing property of DP, we can now state in Theorem 4.5 that, given the (ϵ, δ)-TDP model $\hat{M} = (\hat{P}, \hat{r})$ learned as described in Section 4.2, the policy learned with Algorithm 4 under $\tilde{M}$ is also (ϵ, δ)-TDP. The full proof of this theorem is provided in appendix.

Theorem 4.5. (ϵ, δ)-TDP guarantees for PRIMORL. *Given an (ϵ, δ)-TDP model $\left(\hat{P}, \hat{r}\right)$ learned with Algorithm 1, the policy obtained with private policy optimization (Algorithm 4) within the pessimistic model $\left(\hat{P}, \hat{r} - \lambda u\right)$ is (ϵ, δ)-TDP.*

5 EXPERIMENTS

We empirically assess PRIMORL in three continuous control tasks: CARTPOLE-BALANCE and CARTPOLE-SWINGUP from the DeepMind Control Suite (Tassa et al., 2018) as well as PENDULUM from OpenAI's Gym (Brockman et al., 2016). We also conduct experiments on HALFCHEETAH (Wawrzynski, 2009), which we present in appendix (Section J). For simplicity, we refer to CARTPOLE-BALANCE and CARTPOLE-SWINGUP as BALANCE and SWINGUP.

5.1 EXPERIMENTAL SETTING

Following common practice, we evaluate the offline policies by running them in the real environment We aim to assess the policy's performance degradation when varying the privacy level, as DP training may negatively affect it. We consider MOPO as our non-private baseline. For PRIMORL, we consider different configurations outlined in Table 2. The NO PRIVACY variant, without noise ($z = 0$), isolates the impact of trajectory-level model training on performance. The two private variants ($\epsilon < \infty$) PRIMORL LOW and PRIMORL HIGH correspond to different noise multipliers. We detail the choice of privacy parameters in appendix (Section H). As the existing SWINGUP offline benchmark from Gülçehre et al. (2020) is very small ($K = 40$), and DP training of ML models typically requires significantly more data compared to non-private training (see, for instance, Ponomareva et al. (2023), and our discussion in appendix, Section L), we build our own dataset with 30k trajectories (*i.e.*, 30M steps). We follow the same approach for BALANCE and PENDULUM for which we are not aware of any existing offline benchmark. Data collection, which we detail in appendix (Section D), follows the philosophy of standard benchmarks like D4RL (Fu et al., 2020).

5.2 MAIN RESULTS

We present results on BALANCE, SWINGUP and PENDULUM for PRIMORL and baselines in Table 1 and Figure 2. Both report policy performance in the real MDP as the mean episodic return over 10 episodes per SAC training epoch. Average performance and 95% confidence intervals are computed by re-training the model and the policy from scratch on at least 5 random seeds to assess the stability of the full training process. Based on preliminary results, we use *flat clipping* for both CARTPOLE tasks and *per-layer clipping* for PENDULUM. During policy optimization, SWINGUP uses u_{MA} to estimate uncertainty while others use u_{MPD}. These results show a well-expected trade-off: performance tends to degrade with stronger privacy guarantees (*i.e.*, smaller ϵ's), as the model training gets perturbed with higher levels of noise. Moreover, private model training makes the policy performance less stable over several runs, which is also expected since differential privacy adds another source of randomness

Table 1: Results for PENDULUM, BALANCE and SWINGUP.

	PENDULUM		CARTPOLE-BALANCE		CARTPOLE-SWINGUP	
METHOD	ϵ	RETURN	ϵ	RETURN	ϵ	RETURN
MOPO	∞	795.9 ± 6.5	∞	976.3 ± 26.8	∞	804.9 ± 89.6
PRIMORL NO PRIV.	∞	810.4 ± 27.5 **(101.8%)**	∞	947.5 ± 68.3 **(97.1%)**	∞	774.1 ± 81.7 **(96.17%)**
PRIMORL LOW	22.3	817.4 ± 21.7 **(102.7%)**	85.0	815.8 ± 97.2 **(83.6%)**	94.2	772.4 ± 73.9 **(95.96%)**
PRIMORL HIGH	5.1	778.9 ± 53.5 **(97.9%)**	8.2	758.2 ± 187.2 **(77.7%)**	17.0	698.3 ± 57.5 **(86.75%)**

during training. We notice that noise is not the sole factor that negatively impacts performance, as suggested by the gap between MOPO and PRIMORL NO PRIVACY: gradient clipping and trajectory-level training also contribute to performance degradation. In some cases, a small amount of DP noise might actually be beneficial, acting as a kind of regularization, as in SWINGUP and PENDULUM. Moreover, experiments on HALFCHEETAH (Section J) show that PRIMORL performs worse in higher-dimensional tasks. This could be expected based on the theoretical analysis led in Section 4.3.1, as DP training adds a dependence on the dimension d of the task in the valuation gap.

Despite this trade-off, private agents trained with PRIMORL remain competitive with MOPO for ϵ in the 10^1 to 10^2 range. For PENDULUM, we plot policy performance against ϵ (Figure 4 in appendix) and observe even no performance degradation until ϵ reaches the 1 to 10 range. Although algorithms from Qiao & Wang (2023a) are not suited for direct comparison on the same tasks, we argue that our empirical results are significantly stronger. Indeed, converting their ρ-zero-concentrated DP guarantees into standard (ϵ, δ)-DP guarantees for clarity and fair comparison, we observe that PRIMORL achieves comparable privacy-performance trade-offs, but on much more complex environments (more details in appendix, Section F). While the privacy budgets ϵ from Table 1 do not correspond to strong theoretical privacy guarantees, we must consider the worst-case nature of the differential privacy definition, along with its very strong assumptions on the adversary side. Therefore, backed by recent work on empirical privacy auditing (*e.g.*, Carlini et al. (2019); Ponomareva et al. (2022)), we argue that such ϵ's can provide adequate privacy protection in practical offline RL applications. According to Ponomareva et al. (2023), $\epsilon \lessapprox 10$ is actually a realistic and widely used goal in private deep learning applications. We discuss this matter more in depth in appendix (Section G).

We also point out that achieving a strong privacy-utility trade-off in offline RL requires access to datasets with a very large number of trajectories and that current benchmarks, with datasets of only dozens to thousands of trajectories, are insufficient for studying privacy effectively. In contrast, other fields often use datasets containing millions of users (between 10^6 to 10^9 users in McMahan et al. (2018)) to ensure robust privacy guarantees, which would be very costly to study in offline RL. In appendix (Section L), we provide evidence that increasing dataset size improves the privacy-performance trade-off, showing even greater potential for PRIMORL.

6 DISCUSSION

While existing DP RL methods are limited to simple finite-horizon MDPs, we are the first to address deep offline RL with privacy guarantees in the infinite-horizon discounted setting, and propose a model-based approach named PRIMORL. We empirically show that PRIMORL is capable of learning trajectory-level private, neural-based policies in standard control tasks with only limited performance cost, achieving unprecedented results. Although the reported privacy budgets are typically considered too large to stand as formal DP guarantees, we argue based on recent studies on practical DP that they can offer satisfying privacy protection in practice, especially considering the worst-case nature of DP which can yield too pessimistic privacy budgets. Empirical evaluation of the robustness of our algorithm against privacy attacks, for which a rigorous benchmark has to be developed, will thus be an important research direction for future work. We further point out that our approach has the potential for achieving greater privacy-utility trade-offs given access to large enough offline datasets, hence calling for new benchmarks in the increasingly important field of private offline RL. All in all, we believe that our work represents a significant step towards the much-needed deployment of private RL methods in more complex, high-dimensional control problems.

REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016. URL https://doi.org/10.1145/2976749.2978318.

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *9th International Conference on Learning Representations, ICLR*, 2021. URL https://openreview.net/forum?id=OMNB1G5xzd4.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Proceedings of NeurIPS*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3b5020bb891119b9f5130f1fea9bd773-Abstract.html.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pp. 464–473. IEEE Computer Society, 2014. URL https://doi.org/10.1109/FOCS.2014.56.

Debabrota Basu, Christos Dimitrakakis, and Aristide C. Y. Tossou. Differential privacy for multi-armed bandits: What is it and what is its cost? *CoRR*, abs/1905.12298, 2019. URL http://arxiv.org/abs/1905.12298.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pp. 635–658, 2016. URL https://doi.org/10.1007/978-3-662-53641-4_24.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In Nadia Heninger and Patrick Traynor (eds.), *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pp. 267–284. USENIX Association, 2019. URL https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

Sayak Ray Chowdhury and Xingyu Zhou. Differentially Private Regret Minimization in Episodic Markov Decision Processes, December 2021. URL http://arxiv.org/abs/2112.10599. arXiv:2112.10599 [cs, math].

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Proceedings of NeurIPS*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Chris Cundy, Rishi Desai, and Stefano Ermon. Privacy-constrained policies via mutual information regularized policy gradients. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2809–2817. PMLR, 2024. URL https://proceedings.mlr.press/v238/j-cundy24a.html.

Floris den Hengst, Eoin Grua, Ali el Hassouni, and Mark Hoogendoorn. Reinforcement learning for personalization: A systematic literature review. *Data Science*, 3:1–41, 04 2020. doi: 10.3233/DS-200028.

Cynthia Dwork. Differential Privacy. In *Proceedings of ICALP*, 2006. URL https://www.microsoft.com/en-us/research/publication/differential-privacy/.

Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016. URL http://arxiv.org/abs/1603.01887.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 51–60. IEEE Computer Society, 2010. URL https://doi.org/10.1109/FOCS.2010.12.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL http://proceedings.mlr.press/v97/fujimoto19a.html.

Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differential privacy for regret minimization in reinforcement learning. In *Proceedings of NeurIPS*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/580760fb5def6e2ca8eaf601236d5b08-Abstract.html.

Maziar Gomrokchi, Susan Amin, Hossein Aboutalebi, Alexander Wong, and Doina Precup. Membership inference attacks against temporally correlated data in deep reinforcement learning. *IEEE Access*, 11:42796–42808, 2023. URL https://doi.org/10.1109/ACCESS.2023.3270860.

Çaglar Gülçehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel J. Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A collection of benchmarks for offline reinforcement learning. In *Proceedings of NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/51200d29d1fc15f5a71c1dab4bb54f7c-Abstract.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Yilin Kang, Jian Li, Yong Liu, and Weiping Wang. Data heterogeneity differential privacy: From theory to algorithm. In *Computational Science - ICCS 2023 - 23rd International Conference, Prague, Czech Republic, July 3-5, 2023, Proceedings, Part I*, volume 14073 of *Lecture Notes in Computer Science*, pp. 119–133. Springer, 2023. URL https://doi.org/10.1007/978-3-031-35995-8_9.

Michael J. Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Mach. Learn.*, 49(2-3):209–232, 2002. URL https://doi.org/10.1023/A:1017984413808.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Proceedings of NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f7efa4f864ae9b88d43527f4b14f750f-Abstract.html.

Byeongchan Kim and Min Hwan Oh. Model-based offline reinforcement learning with count-based conservatism. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16728–16746. PMLR, 2023. URL https://proceedings.mlr.press/v202/kim23q.html.

B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Kumar Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.*, 23(6):4909–4926, 2022. URL https://doi.org/10.1109/TITS.2021.3054625.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Chonghua Liao, Jiafan He, and Quanquan Gu. Locally differentially private reinforcement learning for linear mixture markov decision processes. *CoRR*, abs/2110.10133, 2021. URL https://arxiv.org/abs/2110.10133.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016*, 2016. URL http://arxiv.org/abs/1509.02971.

Mingyang Liu, Xiaotong Shen, and Wei Pan. Deep reinforcement learning for personalized treatment recommendation. *Statistics in Medicine*, 41, 06 2022.

Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: Comprehensive review. *J Med Internet Res*, 22(7):e18477, Jul 2020. URL https://www.jmir.org/2020/7/e18477.

Cong Lu, Philip J. Ball, Jack Parker-Holder, Michael A. Osborne, and Stephen J. Roberts. Revisiting design choices in offline model based reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022. URL https://openreview.net/forum?id=zz9hXVhf40.

Paul Luyo, Evrard Garcelon, Alessandro Lazaric, and Matteo Pirotta. Differentially private exploration in reinforcement learning with linear representation, 2021. URL https://arxiv.org/abs/2112.01585.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *6th International Conference on Learning Representations, ICLR*, 2018. URL https://openreview.net/forum?id=BJ0hF1Z0b.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, aug 2017a. URL https://doi.org/10.1109%2Fcsf.2017.11.

Ilya Mironov. Renyi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017b. URL http://arxiv.org/abs/1702.07476. arXiv:1702.07476 [cs].

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118, 2023. URL https://doi.org/10.1561/2200000086.

Dung Daniel T. Ngo, Giuseppe Vietri, and Steven Wu. Improved regret for differentially private exploration in linear MDP. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16529–16552. PMLR, 2022. URL https://proceedings.mlr.press/v162/ngo22a.html.

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. *Reinforcement Learning*, 2019.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. Training text-to-text transformers with privacy guarantees. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2182–2193. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.171. URL https://doi.org/10.18653/v1/2022.findings-acl.171.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H. Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy, March 2023. URL http://arxiv.org/abs/2303.00654. arXiv:2303.00654 [cs, stat].

Kritika Prakash, Fiza Husain, Praveen Paruchuri, and Sujit Gujar. How Private Is Your RL Policy? An Inverse RL Based Analysis Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):8009–8016, June 2022. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v36i7.20772. URL https://ojs.aaai.org/index.php/AAAI/article/view/20772.

Rafael Figueiredo Prudencio, Marcos R. O. A. Máximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *CoRR*, abs/2203.01387, 2022. URL https://doi.org/10.48550/arXiv.2203.01387.

Dan Qiao and Yu-Xiang Wang. Offline reinforcement learning with differential privacy. In *Proceedings of NeurIPS*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/c1aaf7c3f306fe94f77236dc0756d771-Abstract-Conference.html.

Dan Qiao and Yu-Xiang Wang. Near-optimal differentially private reinforcement learning. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9914–9940. PMLR, 2023b. URL https://proceedings.mlr.press/v206/qiao23a.html.

Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *CoRR*, abs/2007.07646, 2020. URL https://arxiv.org/abs/2007.07646.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT 2009*, 2009. URL http://www.cs.mcgill.ca/%7Ecolt2009/papers/018.pdf#page=1.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE Computer Society, 2017. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artif. Intell. Rev.*, 55(2):945–990, 2022. URL https://doi.org/10.1007/s10462-021-09997-9.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL https://www.worldcat.org/oclc/37293240.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pp. 5026–5033. IEEE, 2012. URL https://doi.org/10.1109/IROS.2012.6386109.

Aristide Tossou and Christos Dimitrakakis. Algorithms for Differentially Private Multi-Armed Bandits. In *Proceedings of AAAI*, 2016. URL https://aaai.org/papers/212-algorithms-for-differentially-private-multi-armed-bandits/.

Giuseppe Vietri, Borja Balle, Akshay Krishnamurthy, and Zhiwei Steven Wu. Private reinforcement learning with PAC and regret guarantees. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9754–9764. PMLR, 2020. URL http://proceedings.mlr.press/v119/vietri20a.html.

Baoxiang Wang and Nidhi Hegde. Privacy-preserving q-learning with functional noise in continuous spaces. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/6646b06b90bd13dabc11ddba01270d23-Paper.pdf.

Pawel Wawrzynski. A cat-like robot real-time learning to run. In *Adaptive and Natural Computing Algorithms, 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers*, volume 5495 of *Lecture Notes in Computer Science*, pp. 380–390. Springer, 2009. URL https://doi.org/10.1007/978-3-642-04921-7_39.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b5c01503041b70d41d80e3dbe31bbd8c-Abstract.html.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In *Proceedings of NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pp. 167–176. ACM, 2018. URL https://doi.org/10.1145/3178876.3185994.

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. URL https://dl.acm.org/doi/10.1145/3508028.

A PROOFS

Theorem 4.2. (ϵ, δ)-TDP guarantees for dynamics model. *Given $\delta \in (0,1)$, noise multiplier z, sampling ratio q and number of training iterations T, let $\epsilon := \epsilon^{MA}(z, q, T, \delta)$ be the privacy budget computed by the moments accounting method from (Abadi et al. (2016), more details in Section H.6). The dynamics model output by Algorithm 1 is (ϵ, δ)-TDP.*

Proof. Theorem 1 from McMahan et al. (2018) shows that the moments accounting method from Abadi et al. (2016) computes correctly the privacy loss of DP-FEDAVG at user-level for the noise multiplier $z = \sigma/\mathbb{C}$ with $\mathbb{C} = C/qK$ if, for each user u_k, the clipped gradient $\Delta_k^{\text{clipped}}$ computed from u_k's data has sensitivity bounded by C (referred to as **Condition 1**). With TDP MODEL ENSEMBLE TRAINING, we train the model ensemble as a single big model: at each training iteration, the same input batch is processed forward by all models in a single pass, a single loss is computed for the ensemble, and the parameters are then updated in a single backward pass. The ensemble of models can therefore be seen as a concatenation of all individual models, equivalent to a larger model $\theta = (\theta_i)_{i=1}^N$. We can therefore extend this theorem by mapping users in federating learning to trajectories in offline RL, as long as **Condition 1** holds for every trajectory τ_k.

Since we use **ensemble clipping**, we verify that, for trajectory τ_k, the ensemble gradient $\Delta_k^{\text{CLIPPED}} = \left(\Delta_{i,k}^{\text{CLIPPED}}\right)$ has sensitivity bounded by C. With **flat ensemble clipping**, the gradient of each model $i \in [\![1, N]\!]$ is clipped by a factor $C_i = \frac{C}{\sqrt{N}}$ (see Algorithm 3). By construction, $\Delta_{i,k}^{\text{CLIPPED}}$ has sensitivity bounded by C_i, *i.e.*, $\max_{d(D,D')=1} \|\Delta_{i,k}^{\text{CLIPPED}}(D) - \Delta_{i,k}^{\text{CLIPPED}}(D')\|_2 \leq C_i$. Therefore, for two neighboring datasets D and D':

$$
\begin{aligned}
\|\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\|_2 &= \| \left(\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\right)_{i=1}^N \|_2 \\
&= \sqrt{\sum_{i=1}^N \|\Delta_{i,k}^{\text{CLIPPED}}(D) - \Delta_{i,k}^{\text{CLIPPED}}(D')\|_2^2} \\
&\leq \sqrt{\sum_{i=1}^N C_i^2} \\
&= \sqrt{\sum_{i=1}^N \frac{C^2}{N}} \\
&= C \ .
\end{aligned}
$$

This implies $\max_{d(D,D')=1} \|\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\|_2 \leq C$: $\Delta_k^{\text{CLIPPED}}$ has sensitivity bounded by C. We can derive the same proof for **per-layer ensemble clipping**. Therefore, Theorem 1 from McMahan et al. (2018) holds for TDP MODEL ENSEMBLE TRAINING.

We can therefore use the moments accountant ϵ^{MA} to compute, given $z > 0$, $\delta \in (0,1)$, $q \in (0,1)$ and $T \in \mathbb{N}$, the total privacy budget ϵ spent by Algorithm 1, *i.e.*, $\epsilon = \epsilon^{\text{MA}}(z, q, T, \delta)$.

The dyanmics model output by Algorithm 1 is therefore (ϵ, δ)-TDP. □

Theorem 4.5. (ϵ, δ)-TDP guarantees for PRIMORL. *Given an (ϵ, δ)-TDP model $\left(\hat{P}, \hat{r}\right)$ learned with Algorithm 1, the policy obtained with private policy optimization (Algorithm 4) within the pessimistic model $\left(\hat{P}, \hat{r} - \lambda u\right)$ is (ϵ, δ)-TDP.*

Proof. First, we establish that the pessimistic MDP $\tilde{M}$ is private for $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$. By Theorem 4.2, both the mean estimators $\{\mu_{\phi_i}\}_{i=1}^N$ the covariance estimators $\{\Sigma_{\psi_i}\}_{i=1}^N$ are private. Therefore, both uncertainty estimators $u_{\text{MA}}(s,a) = \|\Sigma_{\psi_i}(s,a)\|_F$ and $u_{\text{MPD}}(s,a) = \max_{i,j} \|f_{\phi_i} - f_{\phi_j}\|_2$, as data-independent transformations of the above quantities, are also private thanks to the post-processing property of DP. Therefore, the pessimistic model $\tilde{M}$ remains (ϵ, δ)-TDP.

Now, we can think of SAC model-based policy optimization (Algorithm 4) as an abstract, randomized function h_Π, that takes as input $\hat{M}$ and outputs as policy $\hat{\pi}$. Furthermore, let h_M denote the mechanism that takes as input the private offline dataset $\mathcal{D}_K$ and outputs the private pessimistic model $\hat{M}$, and which is (ϵ, δ)-TDP following 4.2. We observe that $h = h_\Pi \circ h_M$, where h is the global offline RL algorithm which is the object of Definition 4.1. Since SAC only uses data from the model, as stated in Section 4.3.3, h_Π is independent of the private offline data $\mathcal{D}_K$. In other words, h_Π is a data-independent transformation of the private mechanism h_M. Thanks again to the post-processing property of differential privacy, h is also (ϵ, δ)-TDP. □

We now prove the following two propositions:

Proposition 4.3. Value evaluation error in non-private offline MBRL. *Let the model loss function be L-Lipschitz and Δ-strongly convex, and assumptions from the simulation lemma hold. There is a stochastic convex optimization algorithm for learning the model and a constant M such that, with probability at least $1-\alpha$, and for sufficiently large N, the value evaluation error of π is bounded as:*

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M \cdot \frac{L \log^{1/2}(N/\alpha)}{\sqrt{\Delta N}} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ .$$

Proposition 4.4. Value evaluation error in private offline MBRL. *Let assumptions from Proposition 4.3 hold. If the model is learned with (ϵ, δ)-DP gradient descent, then, with probability at least $1-\alpha$, there is a constant M' such that for large enough N, the value evaluation error of π:*

$$|\hat{V}^\pi_{DP} - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M' \cdot \frac{L d^{1/4} \log(N/\delta) \cdot \textit{poly} \log(1/\alpha)}{\sqrt{\Delta N \epsilon \alpha}} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ ,$$

Proof. Let $\mathcal{F}$ denote the function class of the model. The model is estimated by maximizing the likelihood of the data $\mathcal{D}_K = (s_i, a_i, s'_i)_{i=1}^N$, which is collected by an unknown behavioral policy π^B. This is equivalent to minimizing the negative log-likelihood. The population risk of the estimated model $\hat{P}$ obtained with DP-SGD, is therefore:

$$\mathcal{L}(\hat{P}) = \mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}, s'\sim P(\cdot|s,a)} \left[-\log \hat{P}(s'|s,a) \right] \ ,$$

where $\rho_P^{\pi^B}$ is the (normalized) state-action occupancy measure under policy π^B and dynamics P.

Let us further assume that the true model P belongs to the function class $\mathcal{F}$, and that $P \in \text{argmin}_{P'\in\mathcal{F}} \mathcal{L}(P')$. We can therefore write the excess population risk of the model estimator $\hat{P}$ as:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}, s'\sim P(\cdot|s,a)} \left[\frac{\log P(s'|s,a)}{\log \hat{P}(s'|s,a)} \right] \ .$$

But, denoting $D_{\text{KL}}(A, B)$ the Kullback-Leibler divergence between distributions A, B:

$$D_{\text{KL}}\left(P(s,a), \hat{P}(s,a)\right) = \mathbb{E}_{s'\sim P(\cdot|s,a)} \left[\frac{\log P(s'|s,a)}{\log \hat{P}(s'|s,a)} \right] \ .$$

We can therefore rewrite the above excess population risk as:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}} \left[D_{\text{KL}}\left(P(s,a), \hat{P}(s,a)\right) \right] \ . \tag{3}$$

If the objective function $\mathcal{L}$ is L-Lipschitz and Δ-strongly convex, Bassily et al. (2014) shows (Theorem F.2) that a noisy gradient descent algorithm with (ϵ, δ)-DP guarantees satisfies, with probability at least $1-\alpha$:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathcal{O}\left(\frac{L^2 \sqrt{d} \log^2(N/\delta) \cdot \text{poly} \log(1/\alpha)}{\Delta N \epsilon \alpha} \right) \ . \tag{4}$$

In the non-private case, Shalev-Shwartz et al. (2009) provides the following bound under the same assumptions:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathcal{O}\left(\frac{L^2 \log(N/\alpha)}{\Delta N} \right) \ . \tag{5}$$

On the other hand, we have from the Simulation Lemma (Kearns & Singh, 2002; Xu et al., 2020) that for a MDP $\mathcal{M}$ with reward upper bounded by $r_{\max} = 1$ and dynamics P, a behavioral policy π^B and a learned transition model $\hat{P}$ with:

$$\mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}}\left[D_{\mathrm{KL}}\left(P(s,a), \hat{P}(s,a)\right)\right] \leq \varepsilon_M \ , \tag{6}$$

which by 3 is equivalent to:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) \leq \varepsilon_M \ , \tag{7}$$

Let π be an arbitrary policy. If the divergence between π and the behavioral policy is bounded:

$$\max_s D_{\mathrm{KL}}\left(\pi(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_\pi \ , \tag{8}$$

then the value evaluation error of π is bounded as:

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_M} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ . \tag{9}$$

Since $f(x) = \mathcal{O}(g(x))$ implies $\sqrt{f(x)} = \mathcal{O}(\sqrt{g(x)})$ [2], we note that we can replace $\sqrt{\varepsilon_M}$ in the model term of the right-hand side of 9 by the (square root of) the bounds from 4 and 5 in the private case and in the non-private case, respectively.

This result holds for any policy π verifying 8. In particular, if:

$$\max_s D_{\mathrm{KL}}\left(\hat{\pi}(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_{\hat{\pi}} \ , \tag{10}$$

then:

$$|\hat{V}^{\hat{\pi}} - V^{\hat{\pi}}| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_M} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_{\hat{\pi}}} \ . \tag{11}$$

□

[2] Indeed, for $f(x)$ positive, for any $x \geq x_0$, $|f(x)| = f(x) \leq M' \times g(x)$, then, for any $x \geq x_0$, $\sqrt{f(x)} = |\sqrt{f(x)}| \leq \sqrt{M'} \times \sqrt{g(x)} = M \times \sqrt{g(x)}$

B Related Work (Extended)

B.1 Model-Based Offline Reinforcement Learning

Unlike classical RL (Sutton & Barto, 1998) which is online in nature, offline RL (Levine et al., 2020; Prudencio et al., 2022) aims at learning and controlling autonomous agents without further interactions with the system. This approach is preferred or even unavoidable in situations where data collection is impractical (see for instance Singh et al. (2022); Liu et al. (2020); Kiran et al. (2022)). Model-based RL (Moerland et al., 2023) can also help when data collection is expensive or unsafe as a good model of the environment can generalize beyond in-distribution trajectories and allow simulations. Moreover, model-based RL has been shown to be generally more sample efficient than model-free RL (Chua et al., 2018). Argenson & Dulac-Arnold (2021) also show that model-based offline planning, where the model is learned offline on a static dataset and subsequently used for control without further accessing the system, is a viable approach to control agents on robotic-like tasks with good performance. Unfortunately, the offline setting comes with its own major challenges. In particular, when the data is entirely collected beforehand, we are confronted to the problem of *distribution shift* (Fujimoto et al., 2019): as the logging policy used to collect the training dataset only covers a limited (and potentially small) region of the state-action space, the model can only be trusted in this region, and may be highly inaccurate in other parts of the space. This can lead to a severe decrease in the performance of classic RL methods, particularly in the model-based setting where the acting agent may exploit these inaccuracies in the model , causing large gap between performances in the true and the learned environment. MOPO (Yu et al., 2020) and MOREL (Kidambi et al., 2020), and more recently COUNT-MORL (Kim & Oh, 2023) have effectively tackled this issue by penalizing the reward proportionally to the model's uncertainty, achieving impressive results on popular offline benchmarks. Nonetheless, there remain many areas for improvement, as highlighted by Lu et al. (2022), which extensively study and challenge key design choices in offline MBRL algorithms.

B.2 Privacy in Reinforcement Learning

Differential Privacy (DP), first formalized in Dwork (2006), has become the gold standard in terms of privacy protection. Over the recent years, the design of algorithms with better privacy-utility trade-offs has been a major line of research. In particular, relaxations of differential privacy and more advanced composition tools have allowed tighter analysis of privacy bounds (Dwork et al., 2010; Dwork & Rothblum, 2016; Bun & Steinke, 2016; Mironov, 2017a). Leveraging these advances, the introduction of DP-SGD (Abadi et al., 2016) has allowed to design private deep learning algorithms, paving the way towards a wider adoption of DP in real-world settings, although the practicalities of differential privacy remain challenging (Ponomareva et al., 2023). In parallel to the theoretical analysis of privacy, many works have focused on designing more and more sophisticated attacks, justifying further the need to design DP algorithms ((Rigaki & Garcia, 2020)).

Recent works on RL-specific attacks (Pan et al., 2019; Prakash et al., 2022; Gomrokchi et al., 2023) have demonstrated that reinforcement learning (RL) is no more immune to privacy threats. With RL being increasingly used to provide personalized services (den Hengst et al., 2020), which may expose sensitive user data, developing privacy-preserving techniques for training policies has become crucial. Shortly after DP was successfully extended to multi-armed bandits (Tossou & Dimitrakakis, 2016; Basu et al., 2019), a substantial body of work (*e.g.*, Vietri et al. (2020); Garcelon et al. (2021); Liao et al. (2021); Luyo et al. (2021); Chowdhury & Zhou (2021); Zhou (2022); Ngo et al. (2022); Qiao & Wang (2023b)) addressed privacy in online RL, extending definitions from bandits. However, relying on count-based and UCB-like methods, current RL algorithms with formal DP guarantees are essentially limited to episodic tabular or linear MDPs, and have not been assessed empirically beyond simple numerical simulations. However, current RL algorithms with formal DP guarantees are essentially limited to episodic tabular or linear MDPs, and have not been assessed empirically beyond simple numerical simulations. Few works have proposed private RL methods for more general problems, however with significant limitations or in different contexts. Wang & Hegde (2019) tackle continuous state spaces by adding functional noise to Q-Learning, but the approach is restricted to unidimensional states and focuses on protecting reward information. Recently, Cundy et al. (2024) addressed high-dimensional control and robotic tasks; however, they consider a specific notion of privacy that protects sensitive state variables based on a mutual information framework.

Despite the relevance of the setting for real-world RL deployments, private offline RL has received comparatively less attention. To date, only Qiao & Wang (2023a) have proposed DP offline algorithms, building on non-private value iteration methods. While their approach lays the groundwork for private offline RL and offers strong theoretical guarantees, it remains limited to episodic tabular and linear MDPs. Consequently, no existing work has introduced DP methods that can handle deep RL environments in the infinite-horizon discounted setting, a critical step toward deploying private RL algorithms in real-world applications. With this work, we aim to fill this gap by proposing a differentially private, deep model-based RL method for the offline setting.

C PRESENTATION OF THE TASKS

CARTPOLE requires to swing up then balance an unactuated pole by applying forces on a cart at its base, while CARTPOLE-BALANCE only requires keeping balance. The duration of both tasks is 1,000 steps. PENDULUM involves controlling an inverted pendulum by applying torque to keep it upright and balanced over 200 steps. For this task, we normalize the episodic return to obtain a normalized score between 0 and 1000, using the following formula: $s_{\text{normalized}} = \frac{s-(-1500)}{0-(-1500))}$. HALFCHEETAH is another, higher-dimensional, continuous control task from OpenAI's Gym (Brockman et al., 2016) based on the physics engine MuJoCo (Todorov et al., 2012) where we move forward a 2D cat-like robot by applying torques on its joints. The duration of an episode is 1,000 steps.

D DATA COLLECTION

To collect our offline dataset for CARTPOLE and PENDULUM, we used DDPG (Lillicrap et al., 2016), a model-free RL algorithm for continuous action spaces. We ran 600 independent runs of $50,000$ steps each for CARTPOLE-BALANCE, 150 independent runs of $200,000$ steps each for CARTPOLE-SWINGUP, and 6 independent runs of $1M$ steps each for PENDULUM. We collect all training episodes to ensure a correct mix between random, medium and expert episodes (similar to *replay* datasets in Fu et al. (2020)).

E BASELINES

Table 2: PRIMORL configurations.

VARIANT	TRAJECTORY-LEVEL ENS. TRAINING	CLIP	NOISE	DP
NO CLIP	✓	✗	✗	$\epsilon = \infty$
NO PRIVACY	✓	✓	✗	$\epsilon = \infty$
LOW, HIGH	✓	✓	✓	$\epsilon < \infty$

The first two baselines, PRIMORL NO CLIP and PRIMORL NO PRIVACY are not private ($\epsilon < \infty$) but allow us to isolate the impact of trajectory-level model ensemble training (without clipping and noise addition) and clipping on policy performance. We do not report results for PRIMORL NO CLIP for CARTPOLE and PENDULUM as we found that the model optimized with TDP MODEL ENSEMBLE TRAINING diverges without clipping.

F COMPARISON TO EXISTING METHODS

The closest and only comparable work in offline DPRL is Qiao & Wang (2023a). Although their scope is limited to tabular and linear MDPs and their algorithms are not suited for direct comparison on the same benchmarks, we present below a side-by-side comparison of our respective results.

First, we compare the complexity of the benchmark tasks considered here and the evaluation environment used in Qiao & Wang (2023a). Qiao & Wang (2023a) evaluate their algorithms on an episodic synthetic linear MDP with 2 states and 100 actions, and horizon $H = 20$. On the other hand, we consider standard control tasks with multi-dimensional continuous state and action spaces. Moreover, our tasks have long horizons and high frequency, which makes them typically represented in the infinite-horizon discounted setting.

We then compare the privacy-performance trade-offs achieved by Qiao & Wang (2023a) and PRIMORL. In Qiao & Wang (2023a), they do not mention explicitly the privacy budgets ϵ, but instead mention the zero-concentrated differential privacy (z-CDP) parameter ρ. For clarity and fair comparison, we convert the z-CDP guarantee into a DP guarantee. For this, we use Proposition 1.3 from Bun & Steinke (2016): if a mechanism is ρ-zero-concentrated DP, then for any $\delta > 0$ it is (ϵ, δ)-DP, with $\epsilon = \rho + 2\sqrt{\rho + 2\log(1/\delta)}$. As they evaluate their algorithms for a dataset size up to 1000, we consider two values of $\delta \in \{1/100, 1/1000\}$. Table F show the results for the various

parameters ρ mentioned in Figure 1 from Qiao & Wang (2023a). We observe Qiao & Wang (2023a) also considers the low privacy regime with $\rho = 25$ yielding ϵ close to 50, which is comparable to our low privacy variant. They indeed consider ϵ close to 1 with $\rho = 0.1$, but the cost is a 2 to 3 times worse utility. Other configurations proposed are closed in privacy budgets to what we consider in our paper. Overall, our work achieves comparable privacy-utility trade-offs than Qiao & Wang (2023a), but on significantly more complex tasks.

Table 3: Results from Qiao & Wang (2023a), converted from z-CDP

z-CDP guarantee ρ	DP ϵ for $\delta = 10^{-1}$	DP ϵ for $\delta = 10^{-3}$
25	40.2	51.3
5	11.8	16.8
1	4.0	6.26
0.1	1.1	1.8

G Discussion on the ϵ parameter

As the privacy budgets ϵ's presented in our experimental results do not provide strong theoretical DP guarantees, we would like to further discuss the implications of such privacy budgets in practice.

First, we point out that such ϵ values are comparable to existing work. In particular, as pointed out in Section F, Qiao & Wang (2023a) achieves similar privacy-performance trade-offs and also consider the "low privacy regime" with ϵ's approaching 50 for their best-performing variant. We argue that studying different privacy regimes allows us to clearly highlight the trade-offs between privacy and performance.

Moreover, in light of recent literature on achieving differential privacy in practical deep learning (Carlini et al., 2019; Ponomareva et al., 2022; 2023), we argue that these ϵ values may offer an adequate level of privacy in real-world applications. Ponomareva et al. (2023) states $\epsilon \lessapprox 10$ as a realistic and widely used goal in DP deep learning and a "sweet spot" where it is possible to preserve acceptable utility for complex ML models. Moreover, these studies point out the overly restrictive assumptions on the adversary side, which may yield unnecessarily pessimistic privacy bounds. In offline RL especially, the definition of DP assumes the adversary only has to discriminate between two precise neighboring datasets D and $D' = D \cup \{\tau\}$ as well as the release of all gradients and strong assumptions about the adversary, whereas in practice the adversary faces the much harder task of reconstructing a high-dimensional trajectory based on the output policy and limited side information only.

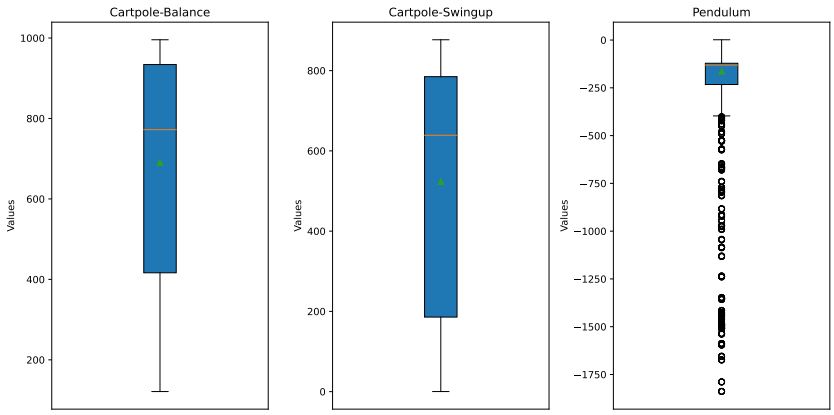
H Experiment Details

H.1 Datasets

Table 4 provides additional details on the offline datasets used in experiments. Figure H.1 shows episode return statistics for each dataset.

Table 4: Dataset details

	CartPole	Pendulum	HalfCheetah
Origin	Custom	Custom	D4RL
Observation space $\mathcal{S}$	$\mathbb{R}^5$	$\mathbb{R}^3$	$\mathbb{R}^{17}$
Action space $\mathcal{A}$	$[-1, 1]$	$[-2, 2]$	$[0, 1]^6$
Nb. of episodes K	30,000	30,000	2,003



H.2 IMPLEMENTATION DETAILS

For all tasks, the model is approximated with a deep neural network with SWISH activation functions and decaying weights. Models take as input a concatenation of the current state s and the taken action a and predict the difference between the next state s' and the current state s along with the reward r. Table 5 provides further implementation details.

The code repository for PRIMORL is provided as part of the supplementary material and will be made public upon acceptance. For MOPO, we use the official implementation from `https://github.com/tianheyu927/mopo`, as well as the PyTorch re-implementation from `https://github.com/junming-yang/mopo`. Our implementation of PRIMORL, which mainly uses PyTorch, is also based on these codebases. To collect the datasets, we use DDPG implementation from `https://github.com/schatty/DDPG-pytorch`.

Model training with TDP MODEL ENSEMBLE TRAINING is parallelized over 16 CPUs using JobLib, while SAC training is conducted over a single Nvidia Tesla P100 GPU.

Table 5: Implementation details

	CARTPOLE	PENDULUM	HALFCHEETAH
MODEL INPUT DIMENSION	6	4	23
MODEL OUTPUT DIMENSION	6	4	18
MODEL HIDDEN LAYERS	2	2	4
NEURONS PER LAYER	128	64	200
WEIGHT DECAY	✓	✓	✓
ACTIVATION FUNCTIONS	SWISH	SWISH	SWISH
ENSEMBLE SIZE N	5	3	7

H.3 TRAINING DETAILS

Before model training, we split the offline dataset into two parts: a train set used to train the model, and a test set used to track model performance. We consider the test set public so that this operation does not involve additional privacy leakage. The split is made by episode (instead of by transitions), so that the test set contains 1% of the episodes for CARTPOLE and PENDULUM and 20% for HALFCHEETAH. To tune the clipping norm, we set $z = 0$ and progressively decreased C until it started to adversely affect performance provided the best results. Moreover, we set the sampling ratio so that a few dozen episodes are randomly selected at each step, which proved to work best in our experiments, which correspond to $q = 10^{-3}$ for CARTPOLE and PENDULUM. The model is trained until convergence using *early stopping*. Test set prediction error is used to track model improvement. For SAC training, the real-to-model ratio r_{real} is zero, meaning that SAC is trained using only simulated data from the model, and does not access any data from the offline dataset. Training details are provided in Table 6.

Table 6: Training and Hyperparameters details

	CARTPOLE	PENDULUM	HALFCHEETAH
TEST SET SIZE	$1\% \times K$	$1\% \times K$	$10\% \times K$
EARLY STOPPING	✓ PATIENCE $= 10$	✓ PATIENCE $= 10$	✓ PATIENCE $= 5$
SAMPLING RATIO q	10^{-3}	10^{-3}	10^{-2}
MODEL LOCAL EPOCHS E	1	1	1
MODEL BATCH SIZE B	16	16	16
MODEL LR η	10^{-3}	10^{-3}	10^{-3}
CLIPPING STRATEGY	FLAT	PER-LAYER	PER-LAYER
SAC LR	3.10^{-4}	3.10^{-4}	3.10^{-4}
ROLLOUT LENGTH H	20	30	5
REWARD PENALTY λ	2.0	2.0	1.0
AUTO-α	✓	✓	✓
TARGET ENTROPY H	-3	-3	-3

H.4 HYPERPARAMETERS

The model is trained using TDP MODEL ENSEMBLE TRAINING with learning rate $\eta = 10^{-3}$, batch size $B = 16$, and number of local epochs $E = 1$.

The policy is optimized within the model using Soft Actor-Critic with rollout, with rollout length and penalty depending on the task. We use a learning rate of 3.10^{-4} for both the actor and the critic. For entropy regularization, we use auto-α with target entropy $H = -3$.

Hyperparameters are summarized in Table 6. We do not report the privacy loss resulting from hyperparameter tuning, although we recognize its importance in real-world applications.

H.5 PRIVACY PARAMETERS

In Table 1, we provide the privacy budgets ϵ computed with the moments accountant method from Abadi et al. (2016). We use the DP accounting tools from Google's Differential Privacy library, available on GitHub. Privacy budget are computed for $\delta = 10^{-5}$, *i.e.* less than K^{-1} as recommended in the literature. It also depends on the noise multiplier z, the number of training round T and the sampling ratio q. Since we use early stopping and the different training runs have different durations, we use the average number of training rounds in the privacy budget computations.

For CARTPOLE-BALANCE, we use $z = 0.25$ and $z = 0.45$ for PRIMORL LOW and PRIMORL HIGH, respectively. For CARTPOLE-SWINGUP, we use $z = 0.25$ and $z = 0.38$ for PRIMORL LOW and PRIMORL HIGH, respectively. The value for PRIMORL HIGH is chosen by incrementally increasing z until policy performance drops below acceptable levels. The corresponding ϵ is therefore roughly the best privacy budget we can obtain while keeping acceptable policy performance. The value for PRIMORL LOW is chosen arbitrarily to provide a weaker level of privacy that typically yields higher policy performance, illustrating the trade-off between the strength of the privacy guarantee and the performance.

H.6 COMPUTING ϵ: THE MOMENTS ACCOUNTANT

Theorem 1 from McMahan et al. (2018) allows us to compute the privacy guarantees $\left(\epsilon^{\text{MA}}(z, q, T, \delta), \delta\right)$ of Algorithm 1 using the Moments Accountant from Abadi et al. (2016). To compute $\epsilon^{\text{MA}}(z, q, T, \delta)$ in our experiments, we use the DP accounting tools from Google's Differential Privacy library, which provides an improved version of the moments accountant based on Rényi Differential Privacy (RDP) Mironov (2017b). Since the computations of the RDP accountant are quite involved while the underlying principles are the same, we rather present the original moments accounting method based on Section 3.2 from Abadi et al. (2016).

By taking into account the DP noise distribution, the moments accountant allows to get tighter bound on the total privacy leakage compared to the standard strong composition theorem. Using an

(ϵ, δ)-DP mechanism at each gradient step, Algorithm 1 with T training steps and a sampling ratio q is $\left(\mathcal{O}(q\epsilon\sqrt{T}), \delta\right)$-DP by the moments accountant. For comparison, the strong composition theorem would yield $\left(\mathcal{O}(q\epsilon\sqrt{T\log(1/\delta)}), Tq\delta\right)$.

The moments accountant works by computing the log moments of the privacy loss random variable. We denote $\mathcal{M}_{\sigma^2}$ the Gaussian mechanism at each training step t, which is characterized by the magnitude $\sigma^2 := \sigma^2(z, q, T, \delta)$ of the Gaussian noise. The privacy loss for $\mathcal{M}_{\sigma^2}$ at output o is defined as follows:

$$c(o; \sigma^2, D, D') = \log \frac{\mathbb{P}(\mathcal{M}_{\sigma^2}(D) = o)}{\mathbb{P}(\mathcal{M}_{\sigma^2}(D') = o)} \quad ,$$

where D, D' are neighboring datasets. It quantifies the privacy leakage for the specific output o taking into account the randomness of the algorithm. The λ-th moment $\alpha(\lambda; D, D')$ is defined as the logarithm of the moment generating function:

$$\alpha_{\mathcal{M}_{\sigma^2}}(\lambda) = \max_{D,D'} \log \mathbb{E}_{o\sim\mathcal{M}_{\sigma^2}(D)} \left[\exp(\lambda c(o; \mathcal{M}_{\sigma^2}, D, D'))\right] \quad .$$

To bound $\alpha_{\mathcal{M}_{\sigma^2}}(\lambda)$ for a Gaussian mechanism of scale σ^2, Abadi et al. (2016) show that, denoting μ_x the p.d.f. of $\mathcal{N}(x, \sigma^2)$ and $\mu = (1-q)\mu_0 + q\mu_1$, it suffices to estimate $\alpha(\lambda) = \log\max(E_1, E_2)$ with:

$$E_1 = \mathbb{E}_{z\sim\mu_0}\left[(\mu_0(z)/\mu(z))^\lambda\right]$$
$$E_2 = \mathbb{E}_{z\sim\mu}\left[(\mu(z)/\mu_0(z))^\lambda\right] \quad .$$

Implementations of the moments accountant typically use numerical integration to estimate $\alpha(\lambda)$.

To compute $\epsilon^{\text{MA}}(z, q, T, \delta)$, a bound on the total privacy loss of Algorithm 1, it then suffices to compute a bound on $\alpha_{\mathcal{M}_{\sigma^2}}(\lambda)$ at each step and sum over all steps. Since we cannot compute a bound for all λ, we need to specify as input a discrete list $\Lambda = \{\lambda_1, ..., \lambda_S\}$ of moments to bound, and select the λ yielding the best privacy budget. Abadi et al. (2016) find that it usually suffices to compute $\alpha(\lambda)$ for $\lambda \leq 32$ (see Section 4).

H.7 Computational Resources

We perform training on a single machine with 64 CPUs and 6 Tesla P100 GPUs with 16GB RAM each. The full training of a single policy, from model learning to policy optimization, takes several hours.

I Additional Experiments

Figures 3 and 4 provide additional empirical insights about PriMORL. In Figure 3, we can see that neither uncertainty estimators is superior overall, but the choice of estimator may impact privacy performance for a specific tasks. On the other hand, Figure 4 shows the performance of PriMORL on Pendulum as a function of the privacy strength ϵ. We can see that performance does not degrade until ϵ goes in the 1 to 10 range, where it starts to drop significantly.

J Experiments on HalfCheetah

We conduct experiments on the Medium-Expert dataset ($K = 2,003$) from the classic D4RL benchmark (Fu et al., 2020). Experimental results are reported in Figure 5 and Table 7 (in appendix), using $C = 15.0$ and $q = 10^{-2}$.

If PriMORL can train competitive policies with small enough noise levels — a tiny amount of noise like $z = 10^{-4}$ proving even beneficial, possibly acting as a kind of regularization —, we were not able to obtain reasonable ϵ's. Indeed, a noise multiplier as small as $z = 10^{-3}$ is enough to cause a significant decline in performance. HalfCheetah thus appears a significantly harder tasks than Cartpole and Pendulum. It is not surprising as HalfCheetah is higher-dimensional, and the theoretical analysis led in Section4.3.1 showed that the dimension d of the problem could

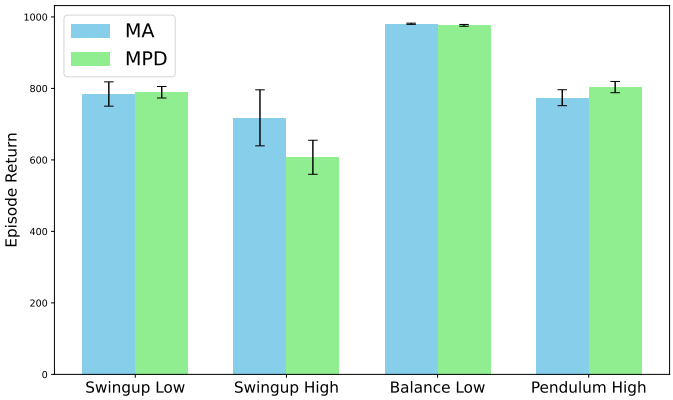


Figure 3: Comparison of policy performance with u_{MA} and u_{MPD} for a fixed model. We measure the average performance of the policy over the last 10 epochs of training. Average and confidence intervals are computed over 5 random seeds.

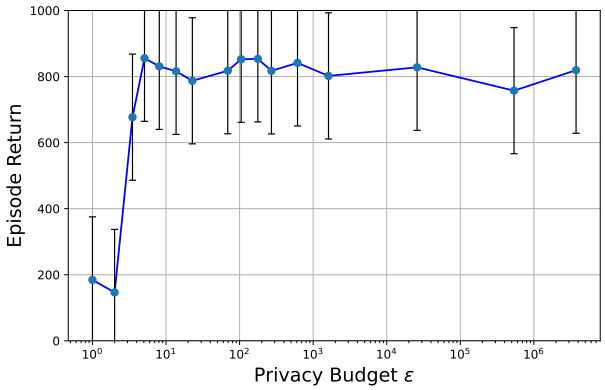


Figure 4: Policy performance on PENDULUM as a function of the privacy budget ϵ. We measure the average performance of the policy over the last 5 epochs of training. Average and confidence intervals are computed over 5 random seeds.

negatively impact the performance of the policy. However, we point out that the size of the dataset for HALFCHEETAH is very limited, and argue that larger datasets with substantially more episodes would translate into competitive privacy-performance trade-offs, as we develop in Section **??**.

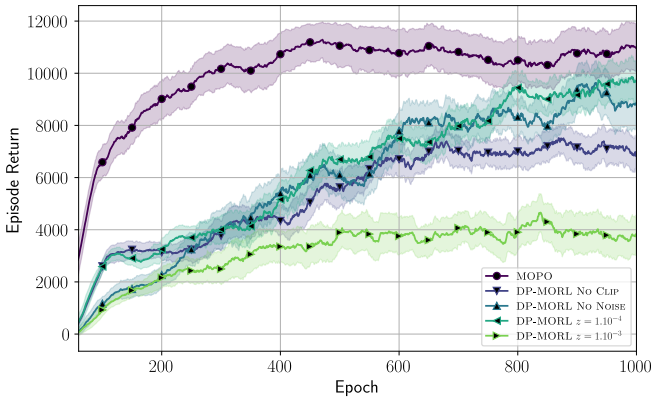


Figure 5: Learning curves for the SAC policy on HALFCHEETAH (*right*). Policy performance (episodic return) is evaluated in the true MDP at the end of each training epoch, over 10 evaluation episodes with different random seeds.

Table 7: Results for HALFCHEETAH MEDIUM-EXPERT. RETURN is the return of the SAC policy evaluated over 10 episodes at the end of each training epoch, averaged over the last 20 epochs.

METHOD	z	RETURN
MOPO	0.0	10931 ± 1326
PRIMORL NO CLIP	0.0	7062 ± 2230
PRIMORL NO NOISE	0.0	8792 ± 2053
PRIMORL	$z = 1.10^{-4}$	9729 ± 2018
	$z = 1.10^{-3}$	3697 ± 1465

K ALGORITHMS

Algorithm 2 is the fully detailed pseudo-code for PRIMORL. Algorithm 3 details the clipping method used in TDP MODEL ENSEMBLE TRAINING. Algorithm 4 is the pseudo-code for SAC policy optimization on the pessimistic private model. This pseudo-code is based on `https://spinningup.openai.com/en/latest/algorithms/sac.html`

Algorithm 2 Model Training with TDP MODEL ENSEMBLE TRAINING

Input: offline dataset $\mathcal{D}_K$, sampling ratio $q \in (0,1)$, noise multiplier $z \geq 0$, clipping norm $C > 0$, local epochs E, batch size B, learning rate η
Output: private model $\hat{M}_\theta$
Initialize model parameters θ_0
for each iteration $t \in [\![0, T-1]\!]$ **do**
  $\mathcal{U}_t \leftarrow$ (sample with replacement trajectories from $\mathcal{D}_K$ with prob. q)
  for each trajectory $\tau_k \in \mathcal{U}_t$ **do**
    Clone current models $\{\theta_i^{\text{start}}\}_{i=1}^N \leftarrow \{\theta_i(t)\}_{i=1}^N$
    $\theta \leftarrow \theta^{\text{start}} := (\theta^{\text{start}})_{i=1}^N$
    for each local epoch $i \in [\![1, E]\!]$ **do**
      $\mathcal{B} \leftarrow$ (τ_k's data split into size B batches)
      for each batch $b \in \mathcal{B}$ **do**
        $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta; b)$
        $\theta \leftarrow \theta^{\text{start}} + \text{ENSEMBLECLIP}(\theta - \theta^{\text{start}}; C)$
      end for
    end for
    $\Delta_{t,k}^{\text{clipped}} \leftarrow \theta - \theta^{\text{start}}$
  end for
  $\Delta_i^{\text{avg}}(t) = \frac{\sum_{k \in \mathcal{U}_t} \Delta_{i,k}^{\text{clipped}}(t)}{qK}$
  $\theta(t+1) \leftarrow \theta(t) + \Delta^{\text{avg}}(t) + \mathcal{N}\left(0_{N_d}, \left(\frac{zC}{qK}\right)^2 I_{N_d}\right)$
end for

$\left.\right\}$ ENSCLIPGD $\left(\tau_k, \{\theta_i^{\text{start}}\}_{i=1}^N; C, E, B\right)$

Algorithm 3 Ensemble Clipping (ENSEMBLECLIP)

Input: ensemble size N, number of model layers L, unclipped gradient $\Delta = \{\Delta_{i,\ell}\}_{i,\ell=1}^{N,L}$, clipping norm C
Output: clipped gradient Δ^{clipped}
$\Delta_i \leftarrow (\Delta_{i,\ell})_{\ell=1}^L, \;\; C_i = \frac{C}{\sqrt{N}}$

$$\Delta_i^{\text{clipped}} \leftarrow \frac{\Delta_i}{\max\left(1, \frac{\|\Delta_i\|_2}{C_i}\right)}, \;\; j = 1, ..., m.$$

Algorithm 4 Private Model-Based Optimization with SAC

Input: private model $\hat{M} = (\hat{P}, \hat{r})$, empty replay buffer $\mathcal{B}$, uncertainty estimator $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$
Output: private policy $\hat{\pi}^{\text{DP}}$
Initialize policy parameters ξ, Q-function parameters ω_1, ω_2 and target parameters $\omega_{\text{targ},1}, \omega_{\text{targ},2}$
for epoch $e \in [\![1, E]\!]$ **do**
  while episode is not terminated **do**
    Observe state s and select action $a \sim \pi_\xi(\cdot|s)$
    Execute a in the pessimistic MDP $\tilde{\mathcal{M}}$ and observe next state $s' \sim \hat{P}(\cdot|s,a)$, reward $r \sim \hat{r}(s,a) - \lambda u(s,a)$ and done signal d
    Store (s, a, r, s', d) in replay buffer $\mathcal{B}$
    if time to update **then**
      Sample a batch of transitions $B = \{(s, a, r, s', d)\}$ from buffer $\mathcal{B}$
      Compute targets for Q-functions:

$$y(r, s', d) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\omega_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\xi(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi_\xi(\cdot|s') \ .$$

      Update Q-functions by one step of gradient descent using:

$$\nabla_{\omega_i} \frac{1}{|B|} \sum_{(s,a,r,s',d)\in B} \left(Q_{\omega_i}(s,a) - y(r,s',d)\right)^2, \quad \text{for } i = 1, 2.$$

      Update policy by one step of gradient ascent using:

$$\nabla_\xi \frac{1}{|B|} \sum_{s\in B} \left(\min_{i=1,2} Q_{\omega_i}(s, \tilde{a}_\zeta(s)) - \alpha \log \pi_\xi\left(\tilde{a}_\zeta(s)|s\right)\right), \quad \tilde{a}_\zeta(s) \sim \pi_\xi(\cdot|s).$$

      Update target networks with:

$$\omega_{\text{targ},i} \leftarrow \rho\omega_{\text{targ},i} + (1-\rho)\omega_i, \quad \text{for } i = 1, 2 \ .$$

    end if
  end while
  Evaluate π_ξ is the true environment $\mathcal{M}$.
end for

L THE PRICE OF PRIVACY IN OFFLINE RL

In this section, we provide theoretical and practical arguments to further justify the need for (much) larger datasets in order to achieve competitive privacy trade-offs in offline RL, as pointed out in (Section 5).

Why does privacy benefit so much from large datasets? From a theoretical perspective, it stems from two facts: 1) ϵ scales with the sampling ratio q (*privacy amplification by subsampling*), and 2) noise magnitude σ is inversely proportional to $\mathbb{E}\left[|\mathcal{U}_t|\right] = qK$. Clearly, the privacy-performance trade-off would benefit from both small q (reducing ϵ) and large qK (reducing noise levels and thus improving performance), which are conflicting objectives for a fixed K. However, if we consider using larger datasets of size $K' \gg K$, it becomes possible to find a K' large enough so that we can use $q' \ll q$ and $q'K' \gg qK$, achieving both much stronger privacy and better performance. We can even argue that for a given privacy budget ϵ (obtained for a given q) and an unlimited capacity to increase K, we could virtually tend to zero noise levels and achieve optimal performance. Therefore, PRIMORL, already capable of producing good policies with significant noise levels and ϵ, has the potential to achieve stronger privacy guarantees provided access to large enough datasets.

An aspect that deserves further development is the iterative aspect of the used training methods and its effect on privacy. Differential privacy being a worst-case definition, it assumes that all intermediate models are released during training. Although the practicality of this hypothesis is debatable, it definitely impacts privacy: privacy loss is incurred at each training iteration (corresponding to a gradient step on the global model in DP-SGD and TDP MODEL ENSEMBLE TRAINING) and privacy budget, therefore, scales with the number of iterations T. Consequently, limiting the number of iterations is even more crucial with DP training than with non-private training. Training a model on the kind of tasks we considered nonetheless requires a lot of iterations to reach convergence (empirically, thousands of iterations for CARTPOLE and tens of thousands of iterations for HALFCHEETAH), and the privacy budget suffers unavoidably.

However, one way to circumvent this is to leverage privacy amplification by subsampling. Indeed, as McMahan et al. (2017) observe, the additional privacy loss incurred by additional training iterations becomes negligible when the sampling ratio q is small enough, which is a direct effect of privacy amplification by subsampling. We discussed above how increasing dataset size K allowed to decrease both sampling ratio q and noise levels. Therefore, by increasing the size of the dataset, we also greatly reduce the impact of the number of training iterations, likely promoting model convergence. This further reinforces the need for large datasets in offline RL in order to study privacy. As an example, McMahan et al. (2018) consider datasets with 10^6 to 10^9 users to train DP recurrent language models, and this is arguably the main reason why they achieve formal strong privacy guarantees. For comparison, the classical RL UNPLUGGED and D4RL benchmarks provide datasets with $K \approx 10^1$ to $K \approx 10^3$ datasets. Achieving the privacy-performance trade-offs demonstrated in Section 5 would not have been possible without the collection of large datasets. Moreover, datasets orders of magnitude larger would be required to attain formal, strong privacy guarantees, such as $\epsilon < 1$. While conducting experiments in deep offline RL with such extensive datasets demands substantial computational resources, we argue that scenarios involving access to datasets with a vast number of trajectories are reflective of real-world situations. For this reason, we consider this case worthy of thorough investigation.

Figure 6 illustrates this point in another way. Given $\epsilon \in \{10^{-4}, 10^{-3}, 10^{-2}\}$, we plot for a range of sampling ratio q the maximum number of iterations T that is allowed so that the total privacy loss does not exceed ϵ, as a function of the noise multiplier z. We can see how decreasing q makes it well easier to train a private model: dividing q by 10, we "gain" roughly 10 times more iterations across all noise levels.

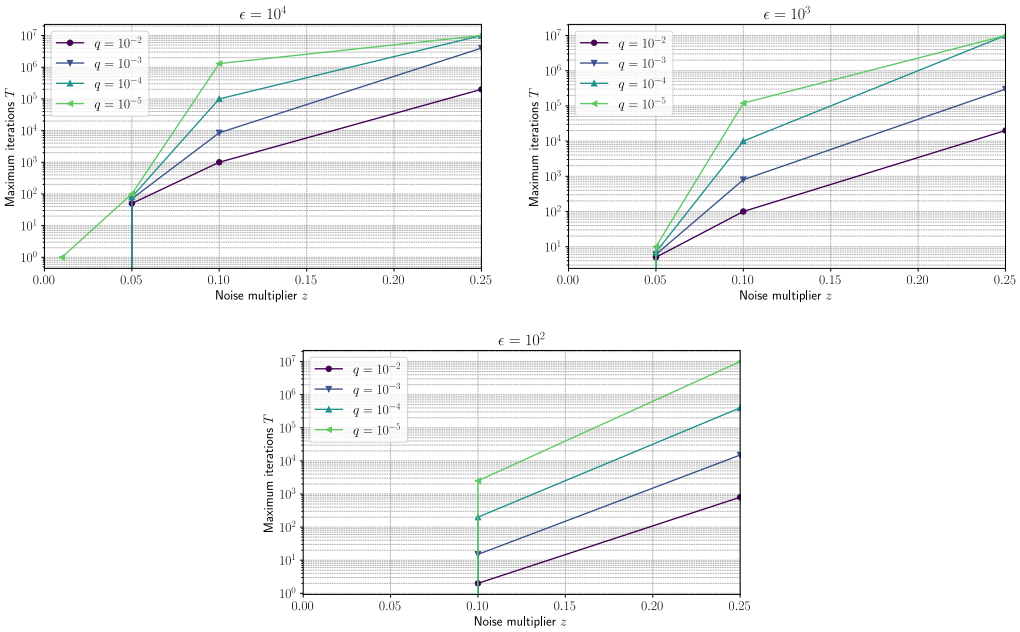


Figure 6: Maximum number of iterations T so that the privacy loss does not exceed ϵ, as function of the noise multiplier z.

M BROADER IMPACTS

As recent advances in the field have moved reinforcement learning closer to widespread real-world application, from healthcare to autonomous driving, and as many works have shown that it is no more immune to privacy attacks than any other area in machine learning, it has become crucial to design algorithmic techniques that protect user privacy. In this paper, we contribute to this endeavor by introducing a new approach to privacy in offline RL, tackling more complex control problems and thus paving the way towards real-world private reinforcement learning. We firmly believe in the importance of pushing the boundaries of this research field and are hopeful that this work will contribute to practical advancements in achieving trustworthy machine learning.