Differentially Private Deep Model-Based Reinforcement Learning

Alexandre Rio
Huawei Noah's Ark Lab
Paris, France
alexandre.rio2@huawei.com

Merwan Barlier
Huawei Noah's Ark Lab
Paris, France
merwan.barlier@huawei.com

Igor Colin
Télécom Paris
Paris, France
igor.colin@telecom-paris.fr

Albert Thomas
Huawei Noah's Ark Lab
Paris, France
albert.thomas@huawei.com

Abstract

We address deep offline reinforcement learning with privacy guarantees, where the goal is to train a policy that is differentially private with respect to individual trajectories in the dataset. To achieve this, we introduce DP-MORL, an MBRL algorithm with differential privacy guarantees. A private model of the environment is first learned from offline data using DP-FEDAVG, a training method for neural networks that provides differential privacy guarantees at the trajectory level. Then, we use model-based policy optimization to derive a policy from the (penalized) private model, without any further interaction with the system or access to the dataset. We empirically show that DP-MORL enables the training of private RL agents from offline data in continuous control tasks and we furthermore outline the price of 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 training machine learning models, as these models can memorize part of the training data. For instance, Rigaki and 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

17th European Workshop on Reinforcement Learning (EWRL 2024).

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.

Fortunately, a large body of work has focused on protecting against such privacy leakages. 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 ML domains, such as neural network training [Abadi et al., 2016] and multi-armed bandits [Tossou and 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 they mostly extend definitions from bandits and proposed methods do not scale to the state and action spaces typically encountered in deep RL.

More akin to supervised learning, the particular setting of offline RL arguably provides a more natural approach to 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 and 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 and Wang [2023a] have proposed the first private algorithms for offline RL, building on value iteration methods, their scope is limited to tabular and linear Markov decision processes (MDPs), and experiments are only conducted on simplistic, 2-state environments. Such methods cannot scale to complex continuous control tasks that often require deep neural function approximations, leaving a huge gap between the current private RL literature and real-world applications. In this work, we are the first to tackle deep RL tasks with continuous state and action spaces 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 tabular and linear Markov decision processes (MDPs), with experiments reduced to simple, very low-dimensional tasks , this work is the first attempt to tackle practical deep RL problems with continuous state and action spaces. To this end, we use a deep model-based approach, exploiting a model of the environment to generalize the information contained in the offline data to unexplored regions of the state-action space. Training the model with a user-centered, differentially private optimizer and mitigating the increased model uncertainty during model-based policy optimization, the resulting algorithm, named DP-MORL, can to train trajectory-level DP policies with competitive privacy-performance trade-offs. Experiments on standard continuous control benchmarks show the potential of our approach.

2 Related Work

2.1 Model-Based Offline Reinforcement Learning

Unlike classical reinforcement learning [Sutton and 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 and 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 mroe recently COUNT-MORL [Kim and 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 **?**, which extensively study and challenge key design choices in offline MBRL algorithms.

2.2 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 and Rothblum, 2016, Bun and Steinke, 2016, Mironov, 2017]. 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 and Garcia, 2020]). *Membership inference attacks* [Shokri et al., 2017], where the adversary has access to a black-box model trainer and tries to predict whether a particular data point has been used to train the private model, are the most popular category of attacks. Other kinds of attacks include *model inversion attacks* (*e.g.*, [Krishna et al., 2020]) and *reconstruction attacks* (*e.g.*, [He et al., 2019]).

Recent works on RL-specific attacks have demonstrated that reinforcement learning (RL) is no more immune to privacy threats. [Gomrokchi et al., 2023] exploit the temporal correlation of RL samples to perform powerful membership inference attacks, while [Pan et al., 2019] use classifiers and genetic algorithms to infer the transition dynamics of an environment. Additionally, [Prakash et al., 2022] show that Inverse RL can be used to recover the reward function from a trained policy. With RL being increasingly used to provide personalized services [**?**], 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 and 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 and Zhou [2021], Zhou [2022], Ngo et al. [2022], Qiao and Wang [2023b]) addressed privacy in online RL, using adapted definitions like Joint Differential Privacy (JDP) and Local Differential Privacy (LDP), with both model-free and model-based approaches. 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. Few works have proposed private RL methods for more general problems, however with significant limitations or in different contexts. **?** 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, **?** 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 and 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 differentially private methods that can handle deep RL environments with continuous state and action spaces, 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 a MDP, that is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $P : \mathcal{S} \times \mathcal{A} \longrightarrow \Delta(\mathcal{S})$ is the transition dynamics (where $\Delta(\mathcal{X})$ denotes the space of probability distributions over $\mathcal{X}$), $r : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$ is the reward function, $\gamma \in [0, 1)$ is a discount factor and $\rho_0 \in \Delta(\mathcal{S})$ is the initial state distribution. The dynamics satisfy the Markov property, *i.e.*, the next state $s' \in \mathcal{S}$ only depends on current state and action. The goal of RL is to learn a policy $\pi : \mathcal{S} \longrightarrow \Delta(\mathcal{A})$ that maximizes the expected discounted return $\eta_{\mathcal{M}}(\pi) := \mathop{\mathbb{E}}_{\tau \sim \pi, \mathcal{M}}[R(\tau)]$, 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 $\tau_k = \left(s_t^{(k)}, a_t^{(k)}, r_t^{(k)}\right)_{t \geq 0}$ has been collected following an unknown behavioral policy π^B. For instance, 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 the true MDP $\mathcal{M}$.

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 $\hat{\mathcal{M}} = \left(\mathcal{S}, \mathcal{A}, \hat{P}, \hat{r}, \gamma, \rho_0\right)$. 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:

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

A small ϵ ensures that h's output remains close between two neighboring datasets, the corresponding probability ratio being bounded by $\exp(\epsilon)$ when $\delta = 0$. When δ is non-zero, this bound may only hold with probability greater than $1 - \delta$: we talk about *approximate* DP, as opposed to *pure* DP.

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. Intuitively, the magnitude σ of the noise should be large enough to cover the largest amount by which the function f can change between two neighboring datasets. This quantity $\Delta_\ell(f) := \max_{d(D,D')=1} \|f(D) - f(D')\|_\ell$, where $\|\cdot\|_\ell$ is the L_ℓ-norm, is called the sensitivity of the function. Differential privacy can thus be seen as a worst-case guarantee. However, too much noise may dramatically hurt the utility of the mechanism. This is why σ typically scales with $\Delta_\ell(f)/\epsilon$, where ϵ controls the strength of the privacy guarantees and hence the trade-off between privacy and utility. 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 at the core of DP-SGD [Abadi et al., 2016], a learning algorithm that applies a series of modifications to the classic SGD algorithm to optimize functions while

guaranteeing (approximate) differential privacy. In particular, it has been successfully used to train neural networks privately [Ponomareva et al., 2023]. The main idea behind DP-SGD is to add a well-scaled Gaussian noise to the gradients, which first requires bounding the norm of each per-sample gradient with a constant C (as gradients can have unbounded sensitivity). To quantify the total privacy budget ϵ_{tot} spent through repeated applications of the Gaussian mechanism during the training of a neural network, the authors have developed the *moments accounting* method, which allows to derive a $\left(\mathcal{O}\left(q\epsilon\sqrt{T}\right), \delta\right)$-DP guarantee for DP-SGD, where q is the sampling ratio (the expected size of the batch drawn by Poisson sampling at each iteration, relative to the size of the dataset), T is the total number of iterations (*i.e.*, number of gradient descent steps), and ϵ is the privacy parameter of each call to the Gaussian mechanism. DP-SGD thus makes heavy use of privacy amplification by sub-sampling [Balle et al., 2018], and its privacy guarantees have been subsequently refined with more advanced privacy analysis tools like Rényi DP [Mironov, 2017]. Moreover, in terms of performance, several works have studied error bounds for DP-SGD under various assumptions on the loss [Bassily et al., 2014, Kang et al., 2023].

4 Differentially Private Model-Based Offline Reinforcement Learning

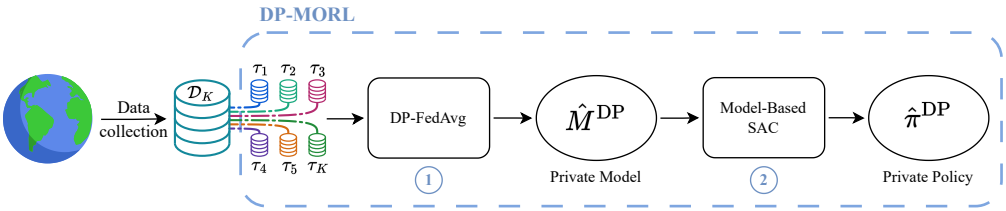

Figure 1: DP-MORL with its two main components: (1) private model training; (2) MBPO.

We now describe our model-based approach for learning differentially private RL agents from offline data, which we call DP-MORL. After defining trajectory-level differential privacy (TDP) in the setting of 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 from the learned 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 DP-MORL.

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 and 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

4.2.1 Dynamics Model

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 $\theta = (\phi, \psi)$:

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

Note that we learn to predict the consecutive state difference $\Delta_t^{t+1}(s) = s_{t+1} - s_t$ instead of next state s_{t+1}, in order to guarantee continuity, which is standard in model-based RL. Moreover, instead of training a single dynamics model, we train an ensemble of N models $\left\{\hat{M}_{\theta_i}\right\}_{i=1}^{N}$, to allow for uncertainty estimation (see Section 4.3).

4.2.2 Trajectory-level DP Training

Such a model is typically trained on the offline dataset $\mathcal{D}_K$ using first-order optimizers like SGD or Adam. The core aspect of our approach, as illustrated in Figure 1, is to learn a private model. To achieve this, we replace the standard optimizer with a DP optimizer. A natural choice is to use DP-SGD. However, DP-SGD is not the most suitable to handle trajectory-level privacy, especially as it clips per-sample gradients. Instead, we use the DP training method developed in McMahan et al. [2018], namely DP-FEDAVG. While this algorithm based on FEDAVG [McMahan et al., 2017] is originally intended to achieve client-level privacy in federated settings, we note that it can be used in any context where the training data can be partitioned. This is the case in offline RL, where the data can be segmented by trajectory, making DP-FEDAVG an interesting approach to training a model with trajectory-level privacy.

The core idea behind DP-FEDAVG is to draw, at each iteration t, a random subset $\mathcal{U}_t$ of the K trajectories. Each trajectory is drawn with probability q, so that the expected number of trajectories selected at each step is qK. Then, we compute one gradient $\Delta_{t,k}$ per trajectory (*i.e.*, each gradient is calculated from a single trajectory's data), that we clip with a constant C using per-layer clipping. This ensures that no trajectory will carry more weight than another in the optimization of the model, hence preventing privacy leakage from a specific trajectory. The next step is to compute an unbiased estimator of the subset gradient average, which is $\Delta_t^{\text{avg}} = (qK)^{-1} \sum_{k \in \mathcal{U}_t} \Delta_{t,k}^{\text{clipped}}$, whose sensitivity is bounded by C/qK. We can then apply the Gaussian mechanism with magnitude $\sigma = zC/qK$, where z controls the strength of the privacy guarantee and therefore the final privacy budget ϵ, and update the model with noisy gradient:

$$\theta_{t+1} \longleftarrow \theta_t + \Delta_t^{\text{avg}} + \mathcal{N}\left(0, \sigma^2\right) \quad .$$

Algorithm 1 provides a condensed pseudo-code for the above training procedure. Further details about the implementation and the training of the model are provided in appendix.

Algorithm 1 Model Training with DP-FEDAVG

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 model $\theta_{\text{start}} \leftarrow \theta_t$
    $\theta \leftarrow \text{CLIPPEDGD}\left(\tau_k, \theta_{\text{start}}; C, \text{local epochs } E, \text{batch size } B\right)$
    $\Delta_{t,k}^{\text{clipped}} \leftarrow \theta - \theta_{\text{start}}$
  end for
  $\Delta_t^{\text{avg}} = \frac{\sum_{k \in \mathcal{U}_t} \Delta_{t,k}^{\text{clipped}}}{qK}$
  $\theta_{t+1} \leftarrow \theta_t + \Delta_t^{\text{avg}} + \mathcal{N}\left(0, \left(\frac{zC}{qK}\right)^2\right)$
end for

4.2.3 Privacy Guarantees for the Model

In this section, we derive formal privacy guarantees for the model trained with Algorithm 1. We must consider that we are not training a single model but rather an ensemble of N models to estimate uncertainty, all consuming the same dataset $\mathcal{D}_K$. In a naive implementation of Algorithm 1, the gradients for each of the N models would be processed independently, with separate clipping and noise addition before aggregation. This means that the noise hides the contribution of a given trajectory only to an individual model, each training step thus corresponding to N separate queries to the private dataset. By the sequential composition property of differential privacy, the total privacy

budget would scale linearly with N, which could drastically limit ensemble size and harm the trade-off between performance and privacy.

To circumvent this issue, we start by training the ensemble as a single big model, as in Yu et al. [2020] and Kidambi et al. [2020]: the current training batch b is fed to all models $\theta = \{\theta_i\}_{i=1}^N$ at once, a single aggregated loss $\mathcal{L}(\theta; b)$ is computed, and the corresponding gradient $g = \nabla_\theta \mathcal{L}(\theta; b)$, which is the concatenation of the gradients of the individual models (*i.e.*, $g = (g_1, ..., g_N)$), is back-propagated through all models in one pass. To ensure a privacy loss independent of the number of models, we distribute the global clipping norm C across all models, on the same principle as per-layer clipping used in McMahan et al. [2017]. That is, for each model $i \in [\![1, N]\!]$, we set a clipping norm C_i, such that $C = \sqrt{\sum_{i=1}^N C_i}$, and scale i's gradient g_i accordingly:

$$g_i^{\text{clipped}} \leftarrow \frac{g_i}{\max\left(1, \frac{\|g_i\|_2}{C_i}\right)} \ .$$

Therefore, $\|g\|_2 \leq C$. In practice, we use a fixed clipping norm $C_i = \frac{C}{\sqrt{N}}$. This technique, which we refer to as *per-model* clipping, guarantees that the contribution of each trajectory in the model update at step t is limited by C, regardless of the number of models. Then, after averaging the gradient across all sampled trajectories in $\mathcal{U}_t$ (line 8 of Algorithm 1), we add a single noise vector to the clipped ensemble gradient (line 9) whose magnitude is calibrated with C, ensuring that we consume a fixed privacy budget independent of N at training round t [1].

From a privacy perspective, we can treat the process of training the ensemble as equivalent to training a single model, and Algorithm 1 behaves similarly to FEDAVG. Theorem 1 from McMahan et al. [2018] shows that the moments accounting method from Abadi et al. [2016] correctly computes the privacy loss of FEDAVG at user-level for the noise multiplier $z = \sigma/\mathbb{C}$ with $\mathbb{C} = C/qK$. Given $\delta \in (0, 1)$, $q \in (0, 1)$ and $T \in \mathbb{N}$, we can therefore use the moments accountant to compute the total privacy budget ϵ spent by Algorithm 1, and obtain (ϵ, δ)-TDP guarantees for our dynamics model, as outlined in Theorem 4.2.

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

4.3 Policy Optimization with the Private Model

Now that we learned a private model from offline data, we can use it as a simulator of the environment to learn a private policy with a model-based policy optimization approach.

4.3.1 Pessimistic Private MDP

A major challenge faced in offline RL is *distribution shift*: since the offline data only covers a portion of the state-action space, the model can be arbitrarily bad in unexplored regions. The policy optimization algorithm can then exploit these inaccuracies, which can impact dramatically the performance of the policy in the real environment. To mitigate these issues, offline model-based methods like MOPO [Yu et al., 2020], MOReL [Kidambi et al., 2020] and Count-MORL [Kim and Oh, 2023] incorporate model uncertainty within the model MDP, penalizing the regions of the space where the model is not confident in its predictions.

In this work, we choose the approach used in MOPO, easier to implement and analyze from a privacy perspective [2]. For a given state-action pair (s, a), MOPO uses the ensemble maximum variance norm $u(s, a) = \max_{i=1,...,N} \|\Sigma_{\psi_i}(s, a)\|_F$, where $\|\cdot\|_F$ is the Frobenius norm, to penalize the reward $\hat{r}(s, a)$ predicted by the model, with a hyperparameter λ controlling the strength of the penalty:

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

[1] Regarding the privacy-performance trade-off, this approach simply shifts the ensemble size problem from scaling the privacy budget to impairing performance: a larger N implies either a smaller C_i or a larger C, both potentially degrading model estimation. In practice, it still yields better trade-offs.

[2] We could use any other penalization strategy (*e.g.*, MOReL) as long as we can compute the corresponding privacy guarantees.

λ is an hyperparameter adjusting the strength of the reward penalty. Policy optimization therefore occurs in the pessimistic private MDP $\tilde{\mathcal{M}} = \left(\mathcal{S}, \mathcal{A}, \hat{P}, \tilde{r}, \rho_0, \gamma\right)$, which ensures that the learned policy $\hat{\pi}_{\tilde{\mathcal{M}}}$ should perform at least as well as the policy (or mixture of policies) used to collect the data (Yu et al. [2020], Theorem 4.4).

4.3.2 Private Policy Optimization

Still following MOPO, we use Soft Actor-Critic (SAC, Haarnoja et al. [2018])[3], a classic off-policy algorithm with entropy regularization, to learn the policy from the pessimistic private MDP. 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, 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.

4.3.3 Privacy Guarantees for the Policy

We now show that, given the (ϵ, δ)-TDP model $\hat{M} = (\hat{P}, \hat{r})$ learned as described in Section 4.2, the policy learned with SAC under the corresponding pessimistic model $\hat{M}$ is also (ϵ, δ)-TDP. First, we observe that since the covariance estimators $\{\Sigma_{\psi_i}\}_{i=1}^N$ are learned privately, the uncertainty estimator $u(s,a) = \|\Sigma_{\psi_i}(s,a)\|_F$ is also private thanks to the post-processing property of DP. Therefore, the pessimistic model $\hat{M}$ remains (ϵ, δ)-TDP. Now, we can think of SAC model-based policy optimization 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 4.3.2, 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.

Theorem 4.3. (ϵ, δ)-TDP guarantees for DP-MORL. *Given an (ϵ, δ)-TDP model $\left(\hat{P}, \hat{r}\right)$ learned with Algorithm 1, the policy obtained with private policy optimization within the pessimistic model $\left(\hat{P}, \hat{r} - \lambda u\right)$ is (ϵ, δ)-TDP.*

4.3.4 Performance Guarantees for the Policy

We expect that private model training will negatively impact policy performance, as gradient perturbations are likely to hurt model convergence. In the simpler case when the model is trained with a vanilla DP noisy gradient descent algorithm, and under several assumptions on the model loss function, Theorem 4.4 evaluates the value evaluation error for the learned policy $\hat{\pi}$.

Theorem 4.4. Value evaluation error in private MBRL. *Let the model loss function be L-Lipschitz and Δ-strongly convex, and $\max_s D_{KL}\left(\hat{\pi}(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_B$. If the model is learned with (ϵ, δ)-DP gradient descent, then, with probability at least $1-\alpha$, there exists a constant M such that for sufficiently large N, the value evaluation error is bounded as:*

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

Full proof is provided in the appendix. Since the policy term is assumed constant between the private and the non-private case (see the proof), the model term bound is to be compared to the non-private bound $\frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M' \cdot \frac{L\log^{1/2}(N/\alpha)}{\sqrt{\Delta N}}$, with M' another constant. In particular, the private bound has an explicit dependence on the problem dimension d which is not present in the non-private bound. Moreover, the $\sqrt{\epsilon}$ factor in the denominator shows that the error will degrade with strong privacy guarantees.

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

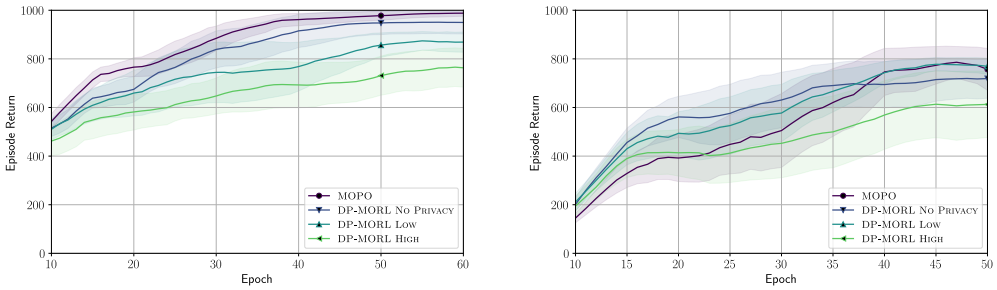

Figure 2: Learning curves on CARTPOLE-BALANCE (*left*) and CARTPOLE-SWINGUP (*right*).

5 Experiments

We assess DP-MORL empirically and evaluate the impact of privacy in two continuous control tasks: CARTPOLE-BALANCE and CARTPOLE-SWINGUP from the DeepMind Control Suite [Tassa et al., 2018]. We also conduct experiments on HALFCHEETAH [Wawrzynski, 2009], which we detail in appendix (Section F). We refer the reader to the appendix for a detailed presentation of the tasks.

Following common practice, we evaluate the policies learned offline by running them in the real environment (*i.e.*, online policy evaluation). We consider MOPO as our non-private baseline, as DP-MORL uses the same reward penalty. Since it is expected that DP training impacts adversely the policy, our goal is to assess how much the performance of the policy drops for different privacy levels. As there is no benchmark for our CARTPOLE datasets, we also report results with the online model-free method DDPG [Lillicrap et al., 2016] to ensure that we actually learn good policies from our offline dataset. For DP-MORL, we consider different configurations outlined in Table 2. The NO PRIVACY variant corresponds to our method without noise, which allows us to isolate the impact of trajectory-level model training on performance. For CARTPOLE, we provide two private configurations ($\epsilon < \infty$): DP-MORL LOW and DP-MORL HIGH, corresponding to different noise multipliers. We detail the choice of privacy parameters in appendix.

5.1 CARTPOLE Experiments

As existing CARTPOLE-SWINGUP offline benchmark from Gülçehre et al. [2020] is extremely small ($K = 40$), and given that DP training of machine learning models typically requires significantly more data compared to non-private training (see, for instance, Ponomareva et al. [2023]), we build our own large offline dataset for CARTPOLE-SWINGUP, with $K = 30,000$ trajectories (*i.e.*, 30 million steps). We proceed the same way for CARTPOLE-BALANCE for which we know no offline benchmark. Data collection, which we detail in appendix, follows the same philosophy as standard offline RL benchmarks like D4RL.

We report experimental results on CARTPOLE-BALANCE and CARTPOLE-SWINGUP for DP-MORL and the aforementioned baselines in Table 1 and Figure 2 (*left*). Both report policy performance in the real MDP, measured at the end of each training epoch of the SAC policy as the mean episodic return across 10 independent episodes. Average and 95% confidence intervals are then computed by re-training the model and the policy from scratch on 10 different random seeds, in order to assess the stability of the whole training process. These results show a well-expected trade-off: performance tends to drop with stronger privacy guarantees (*i.e.*, smaller ϵ's), as the model training gets perturbed with higher levels of noise. But noise is not the sole factor that negatively impacts performance, as suggested by the gap between MOPO and the NO-PRIVACY variant: clipping the gradients, and more importantly partitioning the data by trajectory, which is also crucial to ensure TDP, also contribute to performance drop. In some cases, a small amount of DP noise might even be beneficial, acting as a kind of regularization, as suggested by the results on CARTPOLE-SWINGUP. Nonetheless, the performance cost is rather limited overall, even for ϵ in the 10^1 to 10^2 range, demonstrating that DP-MORL can train competitive private deep RL agents in continuous control tasks.

If such ϵ's only provide weak theoretical DP guarantees, they are competitive in comparison to the current private RL literature which addresses much simpler problems. Our results show that DP-MORL enables the training of RL agents under finite privacy budgets in continuous, higher-

Table 1: Results for CARTPOLE-BALANCE and CARTPOLE-SWINGUP.

	CARTPOLE-BALANCE		CARTPOLE-SWINGUP	
METHOD	ϵ	RETURN	ϵ	RETURN
DDPG	∞	966.1 ± 29.5	∞	827 ± 91.6
MOPO	∞	987.8 ± 13.6	∞	773.1 ± 80.6
DP-MORL NO PRIVACY	∞	950.1 ± 47.1	∞	717.4 ± 83.8
DP-MORL LOW	50.2	868.7 ± 41.7	802.6	773.7 ± 31.6
DP-MORL HIGH	14.4	765.9 ± 80.3	73.6	611.3 ± 139.1

dimensional tasks. Moreover, as pointed out in Ponomareva et al. [2023] and backed by recent work on empirical privacy auditing (*e.g.*, Carlini et al. [2019], Ponomareva et al. [2022]), such ϵ's can already offer enough protection against privacy attacks in practice, especially since DP is a worst-case guarantee, assuming the release of all gradients and strong assumptions about the adversary. In particular, the definition of DP assumes the adversary only has to discriminate between two precise neighboring datasets D and $D' = D \cup \{\tau\}$. 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. It is thus likely that such ϵ's are enough protection in offline RL. However, Section 5.2 shows how better privacy budgets could be achieved, demonstrating even greater potential for DP-MORL.

5.2 The Price of Privacy in Offline RL

In light of these results, it becomes clear what the price of privacy is in offline RL: the need for very large datasets. It is already an accepted fact in the general ML community that, to obtain acceptable privacy-utility trade-offs, DP training requires significantly more data compared to non-private training. In offline RL, however, it has the obvious implication that current benchmarks, whose datasets only contain dozens to thousands of trajectories, are not suitable for studying privacy. Indeed, we could never have reached such privacy-performance trade-offs on CARTPOLE-SWINGUP had we used RL UNPLUGGED benchmark ($K \approx 10^1$), and would have required a significantly larger dataset than D4RL's ($K \approx 10^3$) to obtain similar achievements on HALFCHEETAH. For comparison, 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. Unfortunately, leading experiments with such large datasets in deep offline RL would require huge computational resources. In the appendix (Section H), we instead provide theoretical and practical arguments that demonstrate how increasing the size K of the dataset is likely to lead to better privacy-performance trade-offs. We therefore argue that DP-MORL, already capable of producing good policies with significant noise levels, has the potential to achieve both strong formal privacy guarantees and good performance provided access to large enough datasets.

6 Discussion

In this work, we are the first to address the problem of deep offline RL with privacy guarantees, and propose a model-based approach named DP-MORL. We show empirically that this method is capable of learning policies under a controlled privacy budget in continuous control tasks, with only limited performance cost. While the reported privacy budgets ϵ's are typically considered too large to stand as formal DP guarantees, these results show great potential for achieving meaningful privacy trade-offs in benchmark control tasks, given access to large enough offline datasets. We indeed pointed out that the current offline datasets were inadequate to study private RL, calling for new benchmarks in this increasingly important field. Moreover, it is important to keep in mind that the worst-case nature of differential privacy, which in such iterative training assumes that all intermediate results (*i.e.*, the gradients) are released, can yield too pessimistic privacy budgets in practice. That is why, as Ponomareva et al. [2023] point out, even substantial ϵ values can offer good protection against privacy attacks. 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. All in all, we believe that our work represents a significant step towards the deployment of private RL methods in more complex, high-dimensional control problems.

References

M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016. URL https://doi.org/10.1145/2976749.2978318.

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

B. Balle, G. Barthe, and M. 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.

R. Bassily, A. D. Smith, and A. 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*, pages 464–473. IEEE Computer Society, 2014. URL https://doi.org/10.1109/FOCS.2014.56.

D. Basu, C. Dimitrakakis, and A. 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.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

M. Bun and T. 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*, pages 635–658, 2016. URL https://doi.org/10.1007/978-3-662-53641-4_24.

N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In N. Heninger and P. Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 267–284. USENIX Association, 2019. URL https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

S. R. Chowdhury and X. Zhou. Differentially Private Regret Minimization in Episodic Markov Decision Processes, Dec. 2021. URL http://arxiv.org/abs/2112.10599. arXiv:2112.10599 [cs, math].

K. Chua, R. Calandra, R. McAllister, and S. 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.

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

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

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

J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

S. Fujimoto, D. Meger, and D. 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*, pages 2052–2062. PMLR, 2019. URL http://proceedings.mlr.press/v97/fujimoto19a.html.

E. Garcelon, V. Perchet, C. Pike-Burke, and M. 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.

M. Gomrokchi, S. Amin, H. Aboutalebi, A. Wong, and D. 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.

Ç. Gülçehre, Z. Wang, A. Novikov, T. Paine, S. G. Colmenarejo, K. Zolna, R. Agarwal, J. Merel, D. J. Mankowitz, C. Paduraru, G. Dulac-Arnold, J. Li, M. Norouzi, M. Hoffman, N. Heess, and N. 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.

T. Haarnoja, A. Zhou, P. Abbeel, and S. 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*, pages 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Z. He, T. Zhang, and R. B. Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 148–162, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450376280. doi: 10.1145/3359789.3359824. URL https://doi.org/10.1145/3359789.3359824.

Y. Kang, J. Li, Y. Liu, and W. 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*, pages 119–133. Springer, 2023. URL https://doi.org/10.1007/978-3-031-35995-8_9.

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

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

B. Kim and M. H. 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*, pages 16728–16746. PMLR, 2023. URL https://proceedings.mlr.press/v202/kim23q.html.

B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. K. Yogamani, and P. 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.

K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byl5NREFDr.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

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

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016*, 2016. URL http://arxiv.org/abs/1509.02971.

M. Liu, X. Shen, and W. Pan. Deep reinforcement learning for personalized treatment recommendation. *Statistics in Medicine*, 41, 06 2022.

S. Liu, K. C. See, K. Y. Ngiam, L. A. Celi, X. Sun, and M. 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.

P. Luyo, E. Garcelon, A. Lazaric, and M. Pirotta. Differentially private exploration in reinforcement learning with linear representation, 2021. URL https://arxiv.org/abs/2112.01585.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. 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*, pages 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

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

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

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

D. D. T. Ngo, G. Vietri, and S. Wu. Improved regret for differentially private exploration in linear MDP. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16529–16552. PMLR, 2022. URL https://proceedings.mlr.press/v162/ngo22a.html.

X. Pan, W. Wang, X. Zhang, B. Li, J. Yi, and D. Song. How You Act Tells a Lot: Privacy-Leaking Attack on Deep Reinforcement Learning. *Reinforcement Learning*, 2019.

N. Ponomareva, J. Bastings, and S. Vassilvitskii. Training text-to-text transformers with privacy guarantees. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 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.

N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. Thakurta. How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy, Mar. 2023. URL http://arxiv.org/abs/2303.00654. arXiv:2303.00654 [cs, stat].

K. Prakash, F. Husain, P. Paruchuri, and S. 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.

R. F. Prudencio, M. R. O. A. Máximo, and E. L. 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.

D. Qiao and Y. 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.

D. Qiao and Y. Wang. Near-optimal differentially private reinforcement learning. In F. J. R. Ruiz, J. G. Dy, and J. van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 9914–9940. PMLR, 2023b. URL https://proceedings.mlr.press/v206/qiao23a.html.

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

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. 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)*, pages 3–18. IEEE Computer Society, 2017. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

B. Singh, R. Kumar, and V. P. 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.

R. S. Sutton and A. 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.

Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. P. Lillicrap, and M. A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

E. Todorov, T. Erez, and Y. 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*, pages 5026–5033. IEEE, 2012. URL https://doi.org/10.1109/IROS.2012.6386109.

A. Tossou and C. 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/.

G. Vietri, B. Balle, A. Krishnamurthy, and Z. S. 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*, pages 9754–9764. PMLR, 2020. URL http://proceedings.mlr.press/v119/vietri20a.html.

P. 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*, pages 380–390. Springer, 2009. URL https://doi.org/10.1007/978-3-642-04921-7_39.

T. Xu, Z. Li, and Y. 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.

T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. MOPO: model-based offline policy optimization. In *Proceedings of NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. 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*, pages 167–176. ACM, 2018. URL https://doi.org/10.1145/3178876.3185994.

X. 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 $\delta \in (0, 1)$, sampling ratio q and T training iterations, let $\epsilon := \epsilon^{MA}(\delta, q, T)$ be the privacy budget computed by the moments accounting method from [Abadi et al., 2016]. The dynamics model output by Algorithm 1 is (ϵ, δ)-TDP.*

Proof. Theorem 1 McMahan et al. [2018] shows that the moments accounting method from Abadi et al. [2016] computes correctly the privacy loss of FEDAVG at user-level for the noise multiplier $z = \sigma/\mathbb{C}$ with $\mathbb{C} = C/qK$. We can therefore use the moments accountant to compute, given $\delta \in (0, 1)$, $q \in (0, 1)$ and $T \in \mathbb{N}$, the total privacy budget ϵ_1 spent by Algorithm 1, and obtain (ϵ_1, δ)-TDP guarantees for a single dynamics model.

Since we train an ensemble of N models and release the best $N_e \leq N$ models to optimize the final policy, each model $\hat{M}_{\theta_i}$ consuming the same dataset $\mathcal{D}_K$, we use the sequential composition property of differential privacy to show that the total privacy budget spent by Algorithm 1 is $\epsilon_{N_e} = N_e \times \epsilon_1$.

The dyanmics model output by Algorithm 1 is therefore $(N_e \times \epsilon_1, \delta)$-TDP. □

Theorem 4.3. (ϵ, δ)-TDP guarantees for DP-MORL. *Given an (ϵ, δ)-TDP model $\left(\hat{P}, \hat{r}\right)$ learned with Algorithm 1, the policy obtained with private policy optimization within the pessimistic model $\left(\hat{P}, \hat{r} - \lambda u\right)$ is (ϵ, δ)-TDP.*

Proof. First, we observe that since the covariance estimators $\{\Sigma_{\psi_i}\}_{i=1}^N$ are learned privately, the uncertainty estimator $u(s, a) = \|\Sigma_{\psi_i}(s, a)\|_F$ is also private thanks to the post-processing property of DP. Therefore, the pessimistic model $\hat{M}$ remains (ϵ, δ)-TDP.

Now, we can think of SAC model-based policy optimization 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 4.3.2, 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. □

Theorem 4.4. Value evaluation error in private MBRL. *Let the model loss function be L-Lipschitz and Δ-strongly convex, and $\max_s D_{KL}\left(\hat{\pi}(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_B$. If the model is learned with (ϵ, δ)-DP gradient descent, then, with probability at least $1 - \alpha$, there exists a constant M such that for sufficiently large N, the value evaluation error is bounded as:*

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

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:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathbb{E}_{(s,a)\sim\rho^{\pi^B}_{P}}\left[D_{\text{KL}}\left(P(s,a), \hat{P}(s,a)\right)\right] \ . \tag{1}$$

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 $1 - \alpha$:

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

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

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathcal{O}\left(\frac{L^2\log(N/\alpha)}{\Delta N}\right) \ . \tag{3}$$

On the other hand, we have from the Simulation Lemma [Kearns and 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 learn transition model $\hat{P}$ with:

$$\mathbb{E}_{(s,a)\sim\rho^{\hat{\pi}^D}_{P}}\left[D_{\text{KL}}\left(P(s,a), \hat{P}(s,a)\right)\right] \leq \varepsilon_M \ , \tag{4}$$

which by 1 is equivalent to:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) \leq \varepsilon_M \ , \tag{5}$$

if the divergence between $\hat{\pi}$ and the behavioral policy is bounded:

$$\max_s D_{\text{KL}}\left(\hat{\pi}(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_B \ , \tag{6}$$

then the value evaluation error of $\hat{\pi}$ is bounded as:

$$|\hat{V}^{\hat{\pi}} - V^{\hat{\pi}}| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_M} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_B} \ . \tag{7}$$

Since $f(x) = \mathcal{O}(g(x))$ implies $\sqrt{f(x)} = \mathcal{O}(\sqrt{g(x)})$ [4], we note that we can replace $\sqrt{\varepsilon_M}$ in the model term of the right-hand side of 7 by the (square root of) the bounds from 2 and 3 in the private case and in the non-private case, respectively.

We must also analyze the policy term of the right-hand side of 7, which scales with the divergence between π and π^B. On the one hand, privacy aims at preventing overfitting particular data points, and is therefore likely to drive $\hat{\pi}$ away from the offline data, hence increasing the divergence between $\hat{\pi}$ and π^B. On the other hand, the reward penalization used in model-based methods like MOPO implicitly constrains $\hat{\pi}$ to stay close to the data collection policy π^B, as it penalizes the reward on regions of the state-action space where the model uncertainty is high, corresponding to the regions not covered by the offline dataset. This is likely to cancel the diverging effect of privacy. Therefore, we consider that $\max_s D_{\text{KL}}\left(\pi(\cdot|s), \pi^B(\cdot|s)\right)$ will not change significantly due to privacy, and assume that ε_B stays constant between the private and the non-private case.

□

[4] 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 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. 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. Both are 1,000-step episodic tasks.

C CARTPOLE Data Collection

To collect our offline dataset for CARTPOLE, 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 BALANCE and 150 independent runs of $200,000$ steps each for SWINGUP, collecting all training episodes. As DDPG solves the task in roughly $30,000$ and $100,000$ steps, respectively, this ensures that the dataset is a good mix between random, medium and expert episodes.

D Baselines

Table 2: DP-MORL configurations.

VARIANT	FEDAVG	CLIP	NOISE	DP
NO CLIP	✓	✗	✗	$\epsilon = \infty$
NO PRIVACY	✓	✓	✗	$\epsilon = \infty$
LOW, HIGH	✓	✓	✓	$\epsilon < \infty$

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

E Experiment Details

E.1 Datasets

Table 3 provides additional details on the offline datasets used in experiments.

Table 3: Dataset details

	CARTPOLE	HALFCHEETAH
ORIGIN	CUSTOM	D4RL
OBSERVATION SPACE $\mathcal{S}$	$\mathbb{R}^5$	$\mathbb{R}^{17}$
ACTION SPACE $\mathcal{A}$	$[-1, 1]$	$[0, 1]^6$
NB. OF EPISODES K	30,000	2,003

E.2 Implementation Details

For CARTPOLE, the model is approximated with a neural network with two hidden layers of 128 neurons each and SWISH activation functions with decaying weights. For HALFCHEETAH, the model is approximated with a neural network with four hidden layers of 200 neurons each and SWISH activation functions, also with 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 4 provides further implementation details.

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 DP-MORL, which mainly uses PyTorch, is also based on these codebases. For the model-free baseline DDPG, we use the implementation from `https://github.com/schatty/DDPG-pytorch`.

Model training with DP-FEDAVG is parallelized over 16 CPUs using JobLib, while SAC training is conducted over a single Nvidia Tesla P100 GPU.

Table 4: Implementation details

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

E.3 Training Details

Table 5: Training and Hyperparameters details

	CARTPOLE	HALFCHEETAH
TEST SET SIZE	$1\% \times K$	$10\% \times K$
EARLY STOPPING	✓ PATIENCE $= 5$	✓ PATIENCE $= 5$
SAMPLING RATIO q	10^{-3}	10^{-2}
MODEL LOCAL EPOCHS E	1	1
MODEL BATCH SIZE B	16	16
MODEL LR η	10^{-3}	10^{-3}
SAC LR	3.10^{-4}	3.10^{-4}
ROLLOUT LENGTH H	20	5
REWARD PENALTY λ	0.5	1.0
AUTO-α	✓	✓
TARGET ENTROPY H	-3	-3

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 compute its prediction error. The split is made by episode (instead of by transitions), so that the test set contains 1% of the episodes for CARTPOLE and 20% for HALFCHEETAH. To tune the clipping norm, we set $z = 0$ and progressively decreased C until it started to adversely affect performance, and found that $C = 10^{-2}$ provided the best results. Moreover, we set the sampling ratio to $q = 10^{-3}$. The model is trained until convergence using *early stopping* with patience $p = 5$ over chunks of 100 steps. 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 5.

E.4 Hyperparameters

The model is trained using DP-FEDAVG with learning rate $\eta = 10^{-3}$, batch size $B = 16$, and number of local epochs $E = 1$. While MOPO trains an ensemble of $N = 7$ and picks the best $N_e = 5$ best models, for CARTPOLE we choose smaller numbers $(N, N_e) = (3, 2)$ to limit the privacy loss incurred by training multiple models, which works just as well in our experiments.

The policy is optimized within the model using Soft Actor-Critic with rollout length $H = 20$ and reward penalty $\lambda = 0.5$. 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 5. We do not report the privacy loss resulting from hyperparameter tuning, although we recognize its importance in real-world applications.

E.5 Privacy Parameters

In Table 1, we provide the privacy budget ϵ corresponding to the two configurations of DP-MORL for both CARTPOLE tasks, computed with the moments accountant method from Abadi et al. [2016]. We use the DP accounting tools from Google's Differential Privacy library, availabe on GitHub. Privacy budget are computed for $\delta = 10^{-5}$, *i.e.* less than K^{-1} as recommended in the literature.

For CARTPOLE-BALANCE, we use $z = 0.25$ and $z = 0.38$ for DP-MORL LOW and DP-MORL HIGH, respectively. For CARTPOLE-SWINGUP, we use $z = 0.15$ and $z = 0.3$ for DP-MORL LOW and DP-MORL HIGH, respectively. The value for DP-MORL 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 DP-MORL 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.

E.6 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.

F Experiments on HALFCHEETAH

We conduct experiments on the MEDIUM-EXPERT dataset ($K = 2,003$) from the classic D4RL benchmark [Fu et al., 2020], where agents are anticipated to achieve peak performance, facilitating the analysis of privacy cost. Experimental results are reported in Figure 3 and Table 6 (in appendix), using $C = 15.0$ and $q = 10^{-2}$.

If DP-MORL 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. However, we argue that these results are mainly attributable to the limited dataset size and that more episodes would translate into competitive privacy-performance trade-offs, as we develop in Section 5.2.

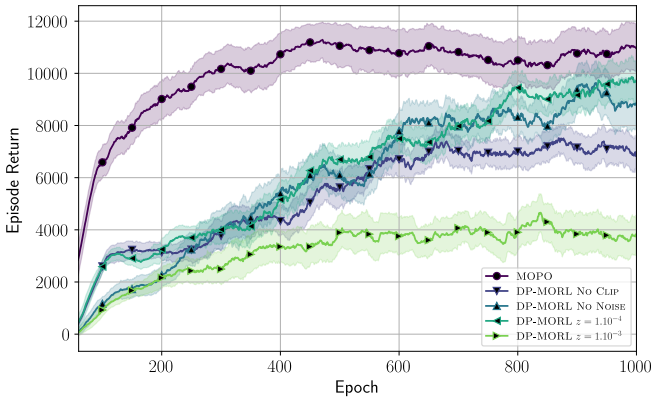

Figure 3: Learning curves for the SAC policy on HALFCHEETAH (*right*). Policy performance (episodic return) is evaluated in the true MDP at the end of each training epoch, over 10 evaluation episodes with different random seeds.

Table 6: 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
DP-MORL NO CLIP	0.0	7062 ± 2230
DP-MORL NO NOISE	0.0	8792 ± 2053
DP-MORL	$z = 1.10^{-4}$	9729 ± 2018
	$z = 1.10^{-3}$	3697 ± 1465

G Algorithms

Algorithm 2 is the fully detailed pseudo-code for DP-MORL. Algorithm 3 details the clipping method we use in DP-FEDAVG, adapted to private 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 DP-FEDAVG

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 model $\theta_{\text{start}} \leftarrow \theta_t$
    $\theta \leftarrow \theta_{\text{start}}$
    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**
        $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta; b)$
        $\theta \leftarrow \theta_{\text{start}} + \text{PERMODELCLIP}(\theta - \theta_{\text{start}}; C)$
      end for
    end for
    } CLIPPEDGD $(\tau_k, \theta_{\text{start}}; C, E, B)$
    $\Delta_{t,k}^{\text{clipped}} \leftarrow \theta - \theta_{\text{start}}$
  end for
  $\Delta_t^{\text{avg}} = \frac{\sum_{k \in \mathcal{U}_t} \Delta_{t,k}^{\text{clipped}}}{qK}$
  $\theta_{t+1} \leftarrow \theta_t + \Delta_t^{\text{avg}} + \mathcal{N}\left(0, \left(\frac{zC}{qK}\right)^2\right)$
end for

Algorithm 3 Per-model Clipping (PERMODELCLIP)

Input: ensemble size N, unclipped gradient $\{g_i\}_{i \in [\![1,N]\!]}$, clipping norm C
Output: clipped gradient Δ^{clipped}
$C_i = \frac{C}{\sqrt{N}}$ s.t. $C = \sqrt{\sum_{i=1}^N C_i^2}$.

$$g_i^{\text{clipped}} \leftarrow \frac{g_i}{\max\left(1, \frac{\|g_i\|_2}{C_i}\right)}, \quad i \in [\![1, N]\!] \ .$$

Algorithm 4 Private Model-Based Optimization with SAC

Input: private model $\hat{M} = (\hat{P}, \hat{r})$, empty replay buffer $\mathcal{B}$
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

H 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 5.2).

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, DP-MORL, 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 DP-FEDAVG) 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 in Section 5.2 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.

Figure 4 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.

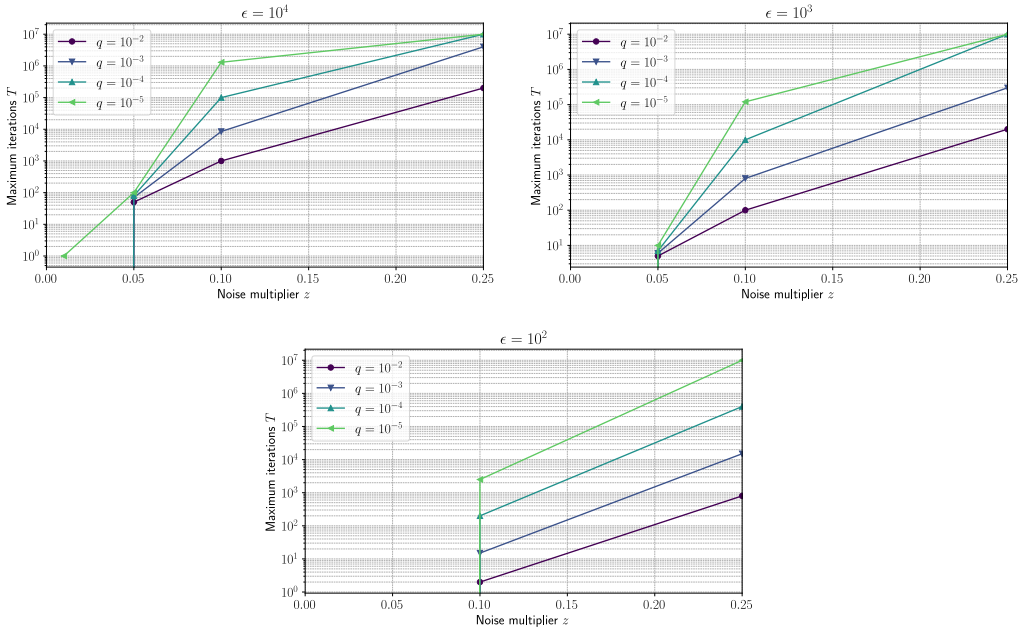

Figure 4: Maximum number of iterations T so that the privacy loss does not exceed ϵ, as function of the noise multiplier z.

I 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.