Privacy Preserving Reinforcement Learning for Population Processes

Samuel Yang-Zhao [1] **Kee Siong Ng** [1]

Abstract

We consider the problem of building Differential Privacy (DP) into Reinforcement Learning (RL) algorithms that operate over population processes, a practical setting that includes, for example, the control of epidemics in large populations of dynamically interacting individuals. In this setting, the RL algorithm interacts with the population over T time steps by receiving population-level statistics as state and performing actions which can affect the entire population at each time step. An individual's data is collected across T interactions and their privacy must be protected at all times. To achieve this, we consider an approach that uses DP mechanisms to privatise the state and reward signal at each time step before the RL algorithm receives them as input. This approach to privacy separates the privatisation scheme from the RL algorithm, which allows any downstream RL algorithm to be used with differential privacy guarantees. Our main theoretical result shows that the value-function approximation error when applying standard RL algorithms directly to the privatised states shrinks quickly as the population size and privacy budget increase. This highlights that protecting privacy when operating over population-level data can have minimal impact on an RL agent's performance. Our theoretical findings are validated by experiments performed on a simulated epidemic control problem over large population sizes.

1. Introduction

The increasing successes and adoption of Reinforcement Learning (RL) algorithms in many practical applications such as digital marketing, finance and public health (Mao et al., 2020; Wang & Yu, 2021; Charpentier et al., 2020a) have led to new, challenging privacy considerations for the research community. This is a particularly important issue in domains like healthcare where highly sensitive personal information is routinely collected and the use of such data in training RL algorithms must be handled carefully. Unfortunately, (Pan et al., 2019) has shown that tabula rasa RL algorithms are susceptible to leaking information about their training data/environment through released policy or value functions. Privacy Preserving Reinforcement Learning is an active research area looking to address this concern. A large amount of work has sought to address privacy via the now widely accepted concept of Differential Privacy (DP) (Dwork et al., 2006), which confers formal 'plausible deniability' guarantees for users whose data are used in training RL algorithms. Unfortunately it is already clear in the personalized setting, where an RL algorithm interacts with a single individual for each episode, that no differentially private reinforcement learning algorithm can have good utility (Shariff & Sheffet, 2018).

In light of this negative result, we contribute to the study of differentially private reinforcement learning (DP-RL) by instead considering the problem setting where an RL agent interacts with a population of users for a fixed number of interactions. We dub this class of environments as *population processes* and at each time step the RL algorithm receives population-level statistics as state and performs actions which can affect the entire population. Population processes directly model settings such as the control of epidemics in large populations of interacting individuals via government interventions (e.g. see (Kompella et al., 2020)). Our goal is to ensure that an individual's data contributions over all interactions are differentially private. We argue that prior approaches to DP-RL, perhaps surprisingly, do not cater to this natural setting and are unable to exploit the additional structure provided.

The questions we ask in this paper are the following: **(1)** What is the right notion of privacy in population process environments and how do we protect this notion of privacy? **(2)** Are good privacy-utility trade-offs possible for RL algorithms in this problem setting? We answer these questions by making the following contributions: **(1)** We formalize the study of differentially private reinforcement learning when the RL agent interacts with a population of individuals, providing the natural definition of privacy in this setting. We clarify the precise semantics of providing differential privacy guarantees in our problem setting using the Pufferfish privacy framework (Kifer & Machanavajjhala, 2014). **(2)** We provide a desirable black-box privacy solution that allows any downstream RL algorithm (whether online/offline or value-based/policy-based) to be used in a privacy preserv-

ing manner. The trade-off for such generality however is a more difficult control problem, as the underlying environment becomes partially observable. Standard methods for dealing with partial observability typically require expensive state estimation techniques or sampling based approximations (Monahan, 1982; Shani et al., 2013; Kurniawati, 2022). Instead of using these methods, we analyze the performance of standard RL algorithms as the population size and privacy budget increases. Our main theoretical contribution is the following bound on the approximation error:

Theorem 1. *Let M be the MDP environment and $\tilde{M}$ denote the privatised MDP under an (ϵ', δ)-differentially private mechanism. Let Q^* be the optimal value function in M and $\tilde{Q}^*$ be the optimal value function in $\tilde{M}$. Then,*

$$\left\|Q^* - \tilde{Q}^*\right\|_\infty \leq \mathcal{O}\left(N^K \exp\left(-\frac{\sqrt{N}\epsilon'}{K}\right) + \frac{1}{\sqrt{N}}\right),$$

where $\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$, K is the state dimension, and N is the population size.

Theorem 1 highlights that for fixed values of K the approximation error can be made arbitrarily small and reduces quickly as the population size and privacy budget increase. This demonstrates that protecting privacy when operating over population-level data can have minimal impact on an RL agent's performance. We validate this theoretical finding by performing experiments on an epidemic control problem simulated over large graphs.

Related Work. The earliest works to consider differential privacy in a reinforcement learning context were focused on the bandit or contextual bandit settings (Guha Thakurta & Smith, 2013; Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016; 2017; Shariff & Sheffet, 2018; Sajed & Sheffet, 2019; Zheng et al., 2020; Dubey & Pentland, 2020; Ren et al., 2020; Chowdhury & Zhou, 2022b; Azize & Basu, 2022). Differentially private reinforcement learning beyond the bandit setting has been primarily considered in a personalized context, where the agent interacts with a population of users in trajectories or episodes, with each trajectory representing multiple interactions with a single user. In (Balle et al., 2016), the authors study policy evaluation under the setting where the RL algorithm receives a set of trajectories, and a neighbouring dataset is one in which a single trajectory differs. In a regret minimization context, there is a large body of work on designing RL algorithms to satisfy either joint differential privacy (Vietri et al., 2020; Luyo et al., 2021; Chowdhury & Zhou, 2022a; Ngo et al., 2022; Zhou, 2022; Qiao & Wang, 2023) or local differential privacy (Garcelon et al., 2020; Luyo et al., 2021; Chowdhury & Zhou, 2022a; Liao et al., 2023). In all of these works, the RL algorithm is framed as interacting with a single user and a trajectory represents a single user's data. A neighbouring dataset is then defined with respect to a neighbouring trajectory. Furthermore, (Shariff & Sheffet, 2018) prove that sublinear regret is not possible under differential privacy in contextual bandits, a result which translates to the reinforcement learning setting. Such DP-RL approaches also cannot be easily adapted for privacy protection in population processes, since trajectories reflect interactions with an entire population and an individual's data can be present across all trajectories and time steps. Protecting the privacy with respect to a single trajectory is therefore insufficient. In (Wang & Hegde, 2019), the author's analyse the performance of deep Q-learning (Mnih et al., 2015) under differential privacy guarantees specified with respect to neighbouring reward functions. This notion of privacy makes natural sense when the reward function is viewed as an individual's private preferences but is also inapplicable to our setting as it does not consider the privacy of the state.

In relation to our practical experiments, the control of epidemics is a topical subject given the prevalence of COVID-19 in recent years and many different practical approaches have been developed (Arango & Pelov, 2020; Charpentier et al., 2020b; Colas et al., 2020; Berestizshevsky et al., 2021; Kompella et al., 2020). These previous approaches have typically focused on the modelling and performance aspects of the epidemic control problem and the preservation of individual privacy has not been considered.

2. Preliminaries

Reinforcement Learning and Markov Decision Processes. We consider a time-homogeneous Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$, reward function $r : \mathcal{S} \times \mathcal{A} \to [0, r_{\max}]$, discount factor $\gamma \in [0, 1)$. The notation $\mathcal{D}(\mathcal{S})$ defines the set of distributions over $\mathcal{S}$. The rewards are assumed to be bounded between 0 and $r_{\max} \in \mathbb{R}$. A stationary policy is a function $\pi : \mathcal{S} \to \mathcal{D}(\mathcal{A})$ specifying a distribution over actions based on a given state, i.e. $a_t \sim \pi(\cdot|s_t)$. A stationary deterministic policy assigns probability 1 to a single action in a given state. With a slight abuse of notation, we will define a stationary deterministic policy to have the signature $\pi : \mathcal{S} \to \mathcal{A}$. Let Π denote the set of all stationary policies. The action-value function (Q-value) of a policy π is the expected cumulative discounted reward $Q^\pi(s, a) = r(s, a) + \mathbb{E}_{P,\pi}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t)\right]$, where the expectation is taken with respect to the transition function P and policy π at each time step. The value function is defined as $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s, a)]$. The Q-value satisfies the Bellman equation given by $Q^\pi(s, a) = r(s, a) + \gamma\mathbb{E}_P\left[V^\pi(s')\right]$.

When considering the optimal policy, define $V^*(s) = \sup_{\pi\in\Pi} V^\pi(s)$ and $Q^*(s, a) = \sup_{\pi\in\Pi} Q^\pi(s, a)$. The

optimal action-value function satisfies the bellman optimality equation given by $Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_P \left[\max_{a'} Q^*(s',a')\right]$. There exists an optimal stationary deterministic policy $\pi^* \in \Pi$ such that $V^*(s) = V^{\pi^*}(s)$. The greedy policy $\pi^*(s) = \arg\max_a Q^*(s,a)$ is in fact an optimal policy.

We primarily consider the case when $\mathcal{S}$ is finite. In this case, it is helpful to view our functions as vectors and matrices. We use P to refer to a matrix of size $(|\mathcal{S}| \cdot |\mathcal{A}|) \times |\mathcal{S}|$ where $P_{sa}^{s'}$ is equal to $P(s'|s,a)$ and P_{sa} is a length $|\mathcal{S}|$ vector denoting $P(\cdot|s,a)$. Similarly, we can view V^π as a vector of length $|\mathcal{S}|$ and Q^π and r as vectors of length $|\mathcal{S}| \cdot |\mathcal{A}|$. The Bellman equation can now be expressed as

$$Q^\pi = r + \gamma P V^\pi.$$

Differential Privacy. Differential privacy is now a commonly accepted definition of privacy with good guarantees even under adversarial settings (Dwork et al., 2006). The differential privacy definition allows semantically different notions of privacy by defining the notion of a neighbouring dataset appropriately. The following definition is a common choice[1]:

Definition 1 (Differential Privacy)**.** A mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{U}$ is (ϵ, δ)-differentially private if for any $D \in \mathcal{X}^n$ and for any $\Omega \subseteq \mathcal{U}$

$$P(\mathcal{M}(D) \in \Omega) \leq e^\epsilon P(\mathcal{M}(D') \in \Omega) + \delta,$$

for all D' in the Hamming-1 neighbourhood of D. That is, D' may differ in at most one entry from D: there exists at most one $i \in [n]$ such that $D_i \neq D'_i$.

A standard approach to privatising a query over an input dataset is to design a mechanism $\mathcal{M}$ that samples noise from a carefully scaled distribution and add it to the true output of the query. To scale the noise level appropriately, the sensitivity of a query is an important parameter.

Definition 2 (ℓ_1 sensitivity)**.** Let f be a function $f : \mathcal{X} \to \mathcal{U}$. Let $d : \mathcal{X} \times \mathcal{X} \to \{0,1\}$ be a function indicating whether two inputs are neighbours. The sensitivity of f is defined as

$$\Delta_f = \sup_{x,x' \in \mathcal{X} : d(x,x')=1} \|f(x) - f(x')\|_1 .$$

Pufferfish Privacy. Pufferfish Privacy was introduced in (Kifer & Machanavajjhala, 2014) and proposes a generalization of differential privacy from a Bayesian perspective. In pufferfish, privacy requirements are instantiated through three components: $\mathbb{S}$, the set of secrets representing functions of the data that may wish to be hidden, $\mathbb{Q} \subseteq \mathbb{S} \times \mathbb{S}$, a set of secret pairs that need to be indistinguishable to an adversary, and Θ, a class of distributions that can plausibly generate the data. It is commonplace to also think of Θ as the beliefs that an adversary may hold over how the data was generated. Pufferfish privacy is defined as follows:

Definition 3 (Pufferfish Privacy)**.** Let $(\mathbb{S}, \mathbb{Q}, \Theta)$ denote the set of secrets, secret pairs, and data generating distributions and let D be a random variable representing the dataset. A privacy mechanism $\mathcal{M}$ is said to be (ϵ, δ)-Pufferfish private in framework $(\mathbb{S}, \mathbb{Q}, \Theta)$ if for all $\theta \in \Theta$, $D \sim \theta$, for all $(s_i, s_j) \in \mathbb{Q}$, and for all $w \in Range(\mathcal{M})$, we have

$$e^{-\epsilon} \leq \frac{P(\mathcal{M}(D) = \omega | s_i, \theta) - \delta}{P(\mathcal{M}(D) = \omega | s_j, \theta)} \leq e^\epsilon, \tag{1}$$

when s_i, s_j are such that $P(s_i|\theta) \neq 0, P(s_j|\theta) \neq 0$.

For $\delta = 0$, applying Bayes Theorem to Equation 1 shows that Pufferfish privacy can be interpreted as bounding the odds ratio of s_i to s_j; an attacker's belief in s_i being true over s_j can only increase by a factor of e^ϵ after seeing the Mechanism's output. The main advantages of Pufferfish privacy are that it provides a formal way to explicitly codify what privacy means and what impact the data generating process has. This has been used to analyze differential privacy in (Kifer & Machanavajjhala, 2014). Whilst we also wish to provide differential privacy guarantees, the data generating process in our problem is different from usual and we will use the Pufferfish framework to precisely detail our privacy guarantees.

3. Problem Setting

Stochastic Population Processes. We consider the problem of controlling a stochastic population process under the constraint of guaranteeing privacy. In this setting, the environment evolves as a stochastic system over a collection of individuals that interact with each other. Consider a population of N^* individuals indexed by the set $N^* = \{1, \ldots, N^*\}$. Individual i's status at time t is given by the random variable $X_{t,i} \in [K]$, which takes one of K values, and $X_t = (X_{t,1}, \ldots, X_{t,N^*})$ denotes the random vector of all individuals' status. A graph at time t is denoted by $G_t = (\mathcal{V}, \mathcal{E}_t)$ where nodes in the graph $\mathcal{V}$ represent the individuals and $\mathcal{E}_t$ represent denotes the interactions between individuals at time t. We assume that the total number of individuals is fixed but the edges evolve over time. To denote sequences, let $Y_{1:t} = (Y_1, \ldots, Y_t)$ and $Y_{<t} = (Y_1, \ldots, Y_{t-1})$. The graph at time t evolves according to a distribution $G_t \sim \rho(\cdot|G_{<t})$. An individual's status depends only upon the previous graph and the previous status of its neighbours, thus satisfying the Markov property, and can be expressed as follows:

$$P(X_t|G_{<t}, X_{<t}) = P(X_t|G_{t-1}, X_{t-1}).$$

[1] e.g. see (Dimitrakakis et al., 2017)

We will assume that individuals' initial status are drawn independently from a distribution $P(X_0|G_{<0}, X_{<0}) = P(X_0)$ and are not dependent on any interaction graph.

The control of population processes becomes relevant when it is desirable for the population to exist in certain states. We consider the case where actions are available for modifying the edges in the graph G_t at each time step for select individuals. The action space $\mathcal{A}$ is then a set of subsets of the nodes in the graph, i.e. $\mathcal{A} \subseteq 2^{\mathcal{V}}$. Thus when actions are involved, the graph at time t is additionally dependent upon the action chosen and is distributed according to $\rho(G_t|G_{<t}, a_t)$.

Example: Epidemic Control. Throughout this paper we consider the Epidemic Control problem as a concrete instantiation of our problem setting. One particular example is the Susceptible-Exposed-Infected-Recovered-Susceptible (SEIRS) process on contact networks (Pastor-Satorras et al., 2015; Nowzari et al., 2016; Newman, 2018). An SEIRS process on contact networks is parametrized by a graph $G = (\mathcal{V}, \mathcal{E})$ and four transition rates $\beta, \sigma, \gamma, \rho > 0$. At any point in time, each individual is in one of four states: Susceptible (S), Exposed (E), Infected (I), or Recovered (R). If individual i is Susceptible at time t and has interacted with $d_{t,i}$ individuals who are Infected, then individual i becomes Exposed with probability $1 - (1 - \beta)^{d_{t,i}}$. Once Exposed, an individual becomes Infected after Geometric(σ) amount of time. Similarly, an Infected individual becomes Recovered with Geometric(γ) time and a Recovered individual becomes Susceptible again with Geometric(ρ) time. At each time t, the state is a histogram representing the proportion of individuals that are of each status. Instead of allowing all interactions, the agent's actions allow for a subset of nodes to be quarantined for one time step. This has the effect of modifying the graph's edges such that quarantined individuals have no edges for a single time step. A typical reward function provides a cost proportional to the number of individuals quarantined and the number of infected individuals.

Data Generation. We now describe how the data is generated and modelled as an MDP. A population process evolving over N^* individuals is the environment $\mathscr{E}$, the RL agent is denoted by $\mathscr{A}$ and there exists a trusted data curator $\mathscr{D}$ that collects the data. At each time step, N individuals are randomly sampled (not necessarily uniformly) and agree to provide their data to the data curator. In the context of Epidemic Control, this equates to N random individuals agreeing to provide their infection status. We consider the case where the interactions between individuals is unknown, i.e. $\mathscr{D}$ has no access to the graph sequence $G_{1:T}$, and that $\mathscr{D}$ is willing to answer queries on the data at each time step. We consider the case of histogram queries. The agent $\mathscr{A}$ picks actions at each time step depending upon the received state and also computes its reward $r_t = r(s_t, a_{t-1})$ as a function of the current state and previous reward. In summary, the three interact in the following manner.

For time $t = 1, \ldots, T$:

1. $\mathscr{E}$ generates $X_t \sim P(\cdot|G_{t-1}, X_{t-1})$.
2. $\mathscr{D}$ produces the dataset $D_t = (x_{t,i})_{i\in[N]}$, where (without loss of generality) we assume the individuals are labelled 1 to N.
3. $\mathscr{D}$ answers histogram query $q(D_t) = \frac{1}{N}\sum_{i\in[N]}(\mathbb{I}(x_{t,i} = \alpha))_{\alpha\in[K]}$.
4. $\mathscr{A}$ receives state $s_t = q(D_t)$, computes reward $r_t = r(s_t, a_{t-1})$ and forms the transition sample $(s_{t-1}, a_{t-1}, s_t, r_t)$ to learn from.
5. $\mathscr{A}$ picks the next action $a_t \sim \pi_t(\cdot|s_t)$.
6. The graph $G_t \sim \rho(\cdot|G_{t-1}, a_t)$ is chosen depending on the last graph and action selected.

To complete the formulation of the above interaction as an MDP we need to specify the state space and transition function. The state space $\mathcal{S}$ consists of any $z \in [0, 1]^K$ that satisfies $\sum_{i\in[K]} z_i = 1$ and $z_i = c_i/N$ for some $c_i \in [N]$. Also, the underlying transitions on individuals' states will induce a transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ over the state space that implicitly captures the impact of the agent's actions on the underlying graph.

As an individual's data is exposed via the state histogram $s_t = q(D_t)$ at each time step, protecting differential privacy amounts to ensuring that it is indistinguishable whether individual i's data was used or individual j's data was used in the computation of the s_t. The state query has sensitivity $\Delta_q = 2/N$ as using individual j's data instead of individual i's data can change the counts in at most two bins of the histogram. Additionally a participating individual provides their data over T interactions, so we need to ensure that all their data over T steps is differentially private. We provide such guarantees via composition.

4. Differentially Private Reinforcement Learning

Our algorithm for differentially private reinforcement learning is presented in Algorithm 1. The algorithm takes as input an MDP environment M, an RL algorithm `RL`, a privacy mechanism $\mathcal{M}$, and parameters (ϵ, δ) and T that specify the level of privacy that should hold over T interactions. We specify an RL algorithm as any method that takes a transition sample (s, a, r', s') and a policy π_{old} and outputs a new policy $\pi_{\text{new}} = \texttt{RL}((s, a, r', s'), \pi_{\text{old}})$.

Our approach is to privatise the state at every time step, which privatises each transition sample via post-processing.

Algorithm 1 Differentially Private Reinforcement Learning

Input: Environment $M = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$, RL algorithm $\texttt{RL} : \mathcal{S} \times \mathcal{A} \times \mathcal{R} \times \mathcal{S} \times \Pi \to \Pi$, Privacy Mechanism $\mathcal{M} : \mathcal{S} \times \mathbb{R} \to \mathcal{S}$, initial state $s_0 \in \mathcal{S}$.
Parameters Privacy parameters (ϵ, δ), number of interactions T.
Randomly initialize policy π_0.
$\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$.
$\tilde{s}_0 = \mathcal{M}_{\epsilon'}(s_0)$.
for $t = 0, 1, 2, \ldots, T-1$ **do**
  $\tilde{a}_t \sim \pi_t(\cdot|\tilde{s}_t)$.
  Receive $s_{t+1} \sim P(\cdot|s_t, \tilde{a}_t)$.
  $\tilde{s}_{t+1} = \mathcal{M}_{\epsilon'}(s_{t+1})$.
  $\tilde{r}_t = r(\tilde{s}_{t+1}, \tilde{a}_t)$.
  $\pi_{t+1} \leftarrow \texttt{RL}((\tilde{s}_t, \tilde{a}_t, \tilde{r}_t, \tilde{s}_{t+1}), \pi_t)$.
end for

Consequently, the RL algorithm is only trained on privatised transition samples and is also guaranteed private via post-processing. This provides a desirable approach to privacy as it allows any RL algorithm to be used. To ensure that privacy holds over T interactions, we employ the adaptive composition theorem (Dwork et al., 2010) to scale the privacy parameters at each time step appropriately as our setup can be viewed as a subcase of the k-fold adaptive experiment used to prove the adaptive composition theorem. We provide a full analysis of the semantics of our privacy protection in light of how the data is generated using the Pufferfish framework before analyzing the utility and providing approximation error bounds on the optimal value function.

4.1. Privacy Analysis

Our privacy guarantee for Algorithm 1 is provided by the adaptive composition theorem (Dwork et al., 2010). Here we provide a Pufferfish analysis and show an equivalence between the guarantees provided by adaptive composition and Pufferfish privacy in our problem setting. The semantics and assumptions behind differential privacy and k-fold adaptations are not always obvious in complex dynamic scenarios, especially when there can be correlation in the data (Kifer & Machanavajjhala, 2014). Our Pufferfish analysis addresses this issue for DPRL for population processes by (1) clarifying the exact secrets being protected, and (2) providing a precise description of the attackers from whom the secrets are protected, in the form of possible data-generating processes and the associated background knowledge.

Let $I = (i_t)_{t\in[T]}$ and $J = (j_t)_{t\in[T]}$ denote two sequences of individuals. Let $Data_t$ be a random variable denoting the dataset produced at time t. We define the following predicates:

$$\sigma_{t,i} = \{x_{t,i} \in Data_t\}$$
$$\sigma_{t,i}(x_t) = \{x_{t,i} \in Data_t \wedge x_{t,i} = x_t\}$$
$$\sigma_I^J(x) = \bigwedge_{t=1}^{T} (\neg\sigma_{t,j_t} \wedge \sigma_{t,i_t}(x_t))$$

Thus for the data sequence random variable $Data_{1:T}$, $\sigma_I^J(x)$ being true means that the data for the sequence of individuals in I was present whilst the individuals in J were not. Let $\mathcal{B} \coloneqq \{(I, J) : I, J \in [N^*]^T, \forall t \in [T], i_t \neq j_t\}$. The secrets and secret pairs are then defined as:

$$\mathbb{S} \coloneqq \{\sigma_I^J(x) : (I, J) \in \mathcal{B}, x \in [K]^T\}$$
$$\mathbb{Q} \coloneqq \{(\sigma_I^J(x), \sigma_J^I(x') : (I, J) \in \mathcal{B}, x, x' \in [K]^T\}.$$

The secret pairs $\mathbb{Q}$ state that we want to ensure that the participation of individuals in I is indistinguishable from the participation of individuals in J for all non-overlapping sequences I and J, and is a direct translation of the protections given by adaptive composition. The additional element to specify in the Pufferfish framework is the data generating processes Θ, representing the possible ways an attacker believes the data could've been generated. We assume that the attacker's belief on the probabilistic model have the following structure:

1) Conditionally independent participation given past data:
$$\nu_{t,i}(G_{<t}, X_{<t}) \coloneqq P(X_{t,i} \in Data_t | G_{<t}, X_{<t}).$$
2) Conditionally independent values given past data:
$$f_{t,i}(x|G_{<t}, X_{<t}) = P(X_{t,i} = x|G_{<t}, X_{<t}).$$
3) Distribution over possible graph sequences $\rho(G_{1:T})$.

1) and 2) codify the probabilistic model in how the adversary in the k-fold adaptive composition experiment chooses the datasets and how a population process evolves. 3) models the attackers prior belief over the likely sequence of interactions between individuals in the population and indicates the strength of the attackers prior knowledge. For example, a uniform distribution for ρ indicates a weak attacker.

We take distributions of the form $\theta = \{\nu_{t,i}, f_{t,i} : t \in [T], i \in [N^*]\} \cup \{\rho\}$ to be the possible data generating distributions. Under this model, our following result states the equivalence between (ϵ, δ)-Pufferfish privacy and (ϵ, δ)-differential privacy under k-fold adaptive composition. Note also that the number of interactions is $k = T$.

Theorem 2. *A family of mechanisms $\mathcal{F}$ satisfies (ϵ, δ)-differential privacy under T-fold adaptive composition iff every sequence of mechanisms $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_T)$, with $\mathcal{M}_i \in \mathcal{F}$, satisfies (ϵ, δ)-Pufferfish privacy with parameters $(\mathbb{S}, \mathbb{Q}, \Theta)$.*

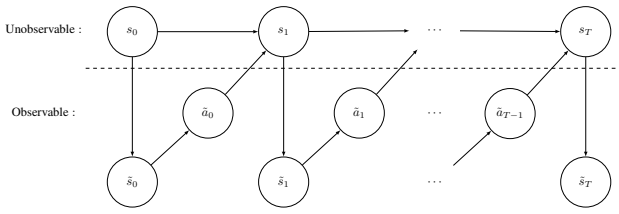

Figure 1. A graphical model of the underlying state and action sequence under our differentially private reinforcement learning approach. The true states are unobservable.

Theorem 2 proves the exact equivalence between the guarantees provided by adaptive composition and Pufferfish privacy under $(\mathbb{S}, \mathbb{Q}, \Theta)$. The full proof is given in Appendix B. This explicitly highlights that the secrets (privacy guarantees) under adaptive composition are protected in the presence of correlations from past information.

4.2. Utility Analysis

We now analyze the utility of our DPRL approach, presenting our main theoretical result that bounds the approximation error of the optimal value function under privacy from the true optimal value function. The analysis we provide is asymptotic in nature; showing asymptotic results however are an important first step in establishing whether good solutions are possible in a problem setting.

Whilst our approach to privacy in Algorithm 1 makes it easy to guarantee the differential privacy of any downstream RL algorithm, the control problem is made more difficult as the true underlying process is unobserved. Figure 1 visualizes the graphical model under our approach and highlights that the state, privatised states and actions evolve according to a partially-observable markov decision process (POMDP). One subtle difference between a POMDP and our privatised system however is that the observed reward is not a direct function of the underlying state as that would constitute a privacy leak. The typical methods for solving POMDP problems resort to computationally expensive state-estimation techniques or sampling based approximations (Monahan, 1982; Shani et al., 2013; Kurniawati, 2022). Instead, we analyse the approximation error when standard MDP RL algorithms are applied directly on the observed privatised states without resorting to state-estimation. The analysis is done under several assumptions.

Assumption 1 (Ergodicity)**.** The underlying MDP environment is ergodic (Puterman, 2014). This means that the transition matrix under any stationary policy consists of a single recurrent class. An ergodic MDP ensures that a stationary distribution over states is well defined under any stationary policy.

Assumption 1 is a necessary tool when analyzing asymptotic performance of RL algorithms and has been commonly used to ensure that every state-action pair is visited infinitely often (see e.g. (Singh et al., 1994)).

Assumption 2 (Lipschitz dynamics)**.** For all $s, s' \in \mathcal{S},\ a \in \mathcal{A}$, there exists $L > 0$ such that

$$\|P(\cdot|s,a) - P(\cdot|s',a)\|_1 \leq L \|s - s'\|_1 .$$

Since the privatised environment evolves as a POMDP, the distribution for the privatised state $\tilde{s}_t$ will in general depend on the entire history of observed states, actions and rewards. However, when an MDP RL algorithm is directly applied on top of privatised states, the transitions between privatised states are assumed to be Markovian. The induced transition model will however depend on the asymptotic state distribution under a behaviour policy generating interactions. For our analysis, we consider an off-policy setting where a stationary behaviour policy π generates a sequence of privatised states, actions, and rewards. The induced Markovian transition model $\tilde{P}^\pi : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ describes the asymptotic transition probabilities and is given by

$$\tilde{P}^\pi(\tilde{s}'|\tilde{s},\tilde{a}) = \sum_{s\in\mathcal{S}} P^\pi(s|\tilde{s},\tilde{a}) \sum_{s'\in\mathcal{S}} P(s'|s,\tilde{a}) P_{\mathcal{M}}(\tilde{s}'|s'). \tag{2}$$

Here $P_{\mathcal{M}}(\tilde{s}'|s') = \mathbb{P}(\mathcal{M}(s') = \tilde{s}')$ denotes the distribution of the state privatization mechanism and $P(s'|s,a)$ denotes the transition matrix of the underlying MDP. The transition model $\tilde{P}^\pi$ depends upon the behaviour policy π through the distribution $P^\pi(s|\tilde{s},\tilde{a})$. The distribution $P^\pi(s|\tilde{s},\tilde{a})$ is the asymptotic probability of the underlying state being s under π when $\tilde{s}$ is observed and $\tilde{a}$ is performed. Using Bayes theorem, it can be expressed as

$$\begin{aligned} P^\pi(s|\tilde{s},\tilde{a}) &= \frac{P_{\mathcal{M}}(\tilde{s}|s)P^\pi(s|\tilde{a})}{\sum_{s'\in\mathcal{S}} P_{\mathcal{M}}(\tilde{s}|s')P^\pi(s'|\tilde{a})} \\ &= \frac{P_{\mathcal{M}}(\tilde{s}|s)P^\pi(s|\tilde{a})}{\tilde{P}^\pi(\tilde{s}|\tilde{a})}, \end{aligned} \tag{3}$$

where $\tilde{P}^\pi(\tilde{s}|\tilde{a}) = \sum_{s'\in\mathcal{S}} P_{\mathcal{M}}(\tilde{s}|s')P^\pi(s'|\tilde{a})$. To make analysis possible, we will assume that the behaviour policy is stochastic and assigns non-zero probability to every action in all states.

Assumption 3. The behaviour policy π is a stochastic policy where $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, $\pi(a|s) > 0$.

Under Assumptions 1 and 3, the distributions $P^\pi(s|\tilde{a})$ and $\tilde{P}^\pi(\tilde{s}|\tilde{a})$ are well defined.

Under our privatisation scheme, the induced MDP on top of privatised states is given by $\tilde{M} = (\mathcal{S}, \mathcal{A}, \tilde{P}^\pi, r, \gamma)$. Note that the reward function does not change as the received rewards are functions of the privatised states. For any policy $\bar{\pi} \in \Pi$, the Q-value $\tilde{Q}^{\bar{\pi}}$ in $\tilde{M}$ satisfies the Bellman equation $\tilde{Q}^{\bar{\pi}} = r + \gamma \tilde{P}^\pi \tilde{V}^{\bar{\pi}}$ where $\tilde{V}^{\bar{\pi}}(s) = \mathbb{E}_{a\sim\bar{\pi}(\cdot|s)}[\tilde{Q}^{\bar{\pi}}(s,a)]$.

We perform our analysis using the exponential mechanism as the state privatisation mechanism. We instantiate the exponential mechanism with the utility function as the ℓ_1 norm $u(s, s') = -\|s - s'\|_1$ (which has sensitivity $\Delta_u = 2/N$) and privacy parameter $\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$.

Our main result on the approximation error between the optimal Q value in the underlying MDP M and the privatised MDP $\tilde{M}$ now follows.

Theorem 1. Let M be the MDP environment and $\tilde{M}$ denote the privatised MDP under an (ϵ', δ)-differentially private mechanism. Let Q^* be the optimal value function in M and $\tilde{Q}^*$ be the optimal value function in $\tilde{M}$. Then

$$\left\|Q^* - \tilde{Q}^*\right\|_\infty \leq \mathcal{O}\left(N^K \exp\left(-\frac{\sqrt{N}\epsilon'}{K}\right) + \frac{1}{\sqrt{N}}\right),$$

where $\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$.

Proof. (Sketch) The full proof is given in Appendix A but we provide a sketch of the main ideas here.

The 'simulation lemma' (Kearns & Singh, 2002; Agarwal et al., 2022) first reduces the problem of bounding the Q-value error to the L_1 error $\|P_{sa} - \tilde{P}^\pi_{sa}\|_1$ between the transition models. Expanding the definition of $\tilde{P}^\pi_{sa}$ then allows us to split into two terms:

$$\|P_{sa} - \tilde{P}^\pi_{sa}\|_1 \leq$$
$$\sum_{s_1 \in \mathcal{S}} P^\pi(s_1|s,a)\left(\underbrace{\left\|P_{sa} - \bar{P}_{sa}\right\|_1}_{(one)} + \underbrace{\left\|\bar{P}_{sa} - \bar{P}_{s_1 a}\right\|_1}_{(two)}\right),$$

where $\bar{P}(s'|s_1, a) = \sum_{s_2 \in \mathcal{S}} P(s_2|s_1, a) P_{\mathcal{M}}(s'|s_2)$. The first term can be viewed as the error due to privatising the output from the transition model and the second term can be viewed as the error due to privatising the input to the transition model.

The first term is bound by first applying the Bretagnolle-Huber inequality (Bretagnolle & Huber, 1979). The KL divergence between P_{sa} and $\bar{P}_{sa}$ can be bound by noting that $\bar{P}_{sa}$ is a convolution between the privacy mechanism and the true transition model. The sum in the convolution can be reduced to a single element, leading to the following bound:

$$\|P_{sa} - \bar{P}^\pi_{sa}\|_1 \leq 2\sqrt{1 - C(s)^{-1}},$$

where $C(s)$ is the normalizing constant for the exponential mechanism at state s. $\sqrt{1 - C(s)^{-1}}$ can then be shown to shrink to zero at a rate of $\mathcal{O}\left(N^K \exp(-N\epsilon'/K)\right)$.

The second term is bound by exploiting the fact that our utility function is symmetric and the concentration properties of the exponential mechanism. Using the fact that the utility function is symmetric allows us to first bound the second term by $\tilde{C}\sum_{s_1 \in \mathcal{S}} P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa} - \bar{P}_{s_1 a}\right\|_1$, where $\tilde{C}$ is a constant. Define the α-ball around state s as $B_\alpha(s) \coloneqq \{s' \in \mathcal{S} : \|s - s'\|_1 < \alpha\}$. We split the sum over s_1 over $B_\alpha(s)$ and its complement $B^c_\alpha(s)$. We bound terms in $B_\alpha(s)$ using the Lipschitz property and the concentration properties of the exponential mechanism state that the error on $B^c_\alpha(s)$ will shrink quickly. This allows us to bound $\sum_{s_1 \in \mathcal{S}} P^\pi(s_1|s,a)\left\|\bar{P}_{sa} - \bar{P}_{s_1 a}\right\|_1$ by $\mathcal{O}\left(N\exp\left(-\frac{N\alpha\epsilon'}{K}\right) + \alpha\right)$. Choosing $\alpha = N^{-\frac{1}{2}}$ and combining terms gives the final result. □

Theorem 1 highlights that for a fixed K, the approximation error shrinks quickly as the population size increases. The error will still shrink as the the privacy budget increases but the second term of the error bound is not a function of ϵ' and is thus unaffected. Thus, increasing the population size is the most important factor in determining good quality solutions for DPRL. Note however our results depend upon K being fixed and not a function of N. This is because the size of the state space scales exponentially as K grows.

5. Experiments

We present empirical results that corroborate our theoretical findings on the SEIRS Epidemic Control problem detailed in Section 3. We simulate the SEIRS Epidemic Control problem over three large social network graphs from the Stanford Large Graph Network Dataset (Leskovec & Krevl, 2014). The number of nodes in each graph are: 82168 (82K), 196194 (196K), and 1134890 (1.1M). These social networks represent reasonable models of social interactions in a population. The state space for each of these models is $\mathcal{O}(N^K)$, and that is computationally challenging for RL algorithms. On each graph, the agent has access to 5 actions $\{\text{Quarantine}(i) : i \in [0, 0.25, 0.5, 0.75, 1.0\}$, where Quarantine$(i)$ quarantines the top ith percent of nodes ranked by degree centrality by modifying the interaction matrix in the way described in Sect. 3. The reward function is given by

$$r(s_t, a_t) = -(\alpha I(s_t) + (1 - \alpha)C(a_t)),$$

and is taken as a convex combination between two functions $I(s_t)$ and $C(a_t)$. The function $I(s_t)$ returns total proportion of Exposed and Infected individuals at time t and $C(a_t)$ returns proportion of individuals quarantined by action a_t. All additional experiment details can be found in Appendix D.

The RL algorithm we choose is the DQN algorithm (Mnih et al., 2015) initialized with an experience replay buffer (Lin, 1992) and we refer to its differentially private version as DP-DQN. DP-DQN can only store privatised transitions in

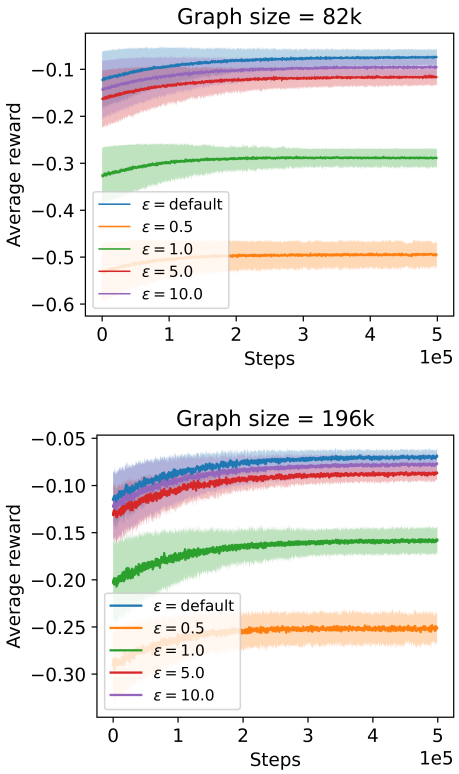

Figure 2. DP-DQN performance as ϵ is varied on graphs with 82k, 196K nodes.

its replay buffer. The DQN algorithm and the environment interact in an online fashion and exploration is performed using epsilon-greedy. The Laplace mechanism with output projected back onto the state space was used as the state privatisation mechanism for computational efficiency but we note that the utility function we use for the exponential mechanism in the proof of Theorem 1 recovers the Laplace mechanism up to normalization factors. Since the projection may not result in a unique element, we choose an element that is a projection and maximises the reward error to make the performance difference as pronounced as possible. A full description of the DP-DQN algorithm and hyperparameters used is presented in Appendix C.

Results. Figures 2 and 3 display how the performance of DP-DQN scales as ϵ increases across each graph over $T = 5\text{e}5$ interactions. Each curve displays the mean and standard deviation over five random seeds. The parameter δ was kept fixed across all runs at $\delta = 1\text{e}{-5}$. The blue line indicates the default performance of DQN without differential privacy across all graphs. For all graphs, the performance of DP-DQN in a low privacy setting (i.e. $\epsilon \geq 5$) clustered closely to the optimal performance. Notably, DP-DQN's performance under low privacy on the 1.1M graph is essentially indistinguishable to the default performance. Across all graphs, the performance of DP-DQN degrades in a high privacy setting (i.e. $\epsilon < 1$). The effect is especially

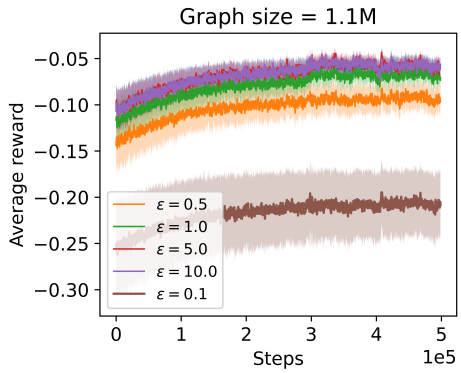

Figure 3. DP-DQN performance as ϵ is varied on graph with 1.1M nodes.

pronounced for the small 81k graph. The performance of DP-DQN under $\epsilon = 0.5$ in the 1.1M graph is quite close to the default performance. This falls in line with intuition as the noise added under a given privacy parameter has absolute scale. As the population size increases, the relative error due to privacy is much smaller. Assuming the environment transition function and reward function are Lipschitz, the small relative error due to privacy would not impact performance in a large population as much as it would in a smaller population. On the 1.1M graph we also plot the performance of DP-DQN under $\epsilon = 0.1$. Again, this is made possible due to the increased population size.

6. Conclusion and Limitations

One limitation of our results is that they are asymptotic in nature. This analysis provides guarantees on the error between the solution under privacy and the true solution without privacy, but does not provide any guidance on whether such a solution can be learned. Nevertheless, our empirical results provide some confidence the solutions found by learning algorithms will scale the way our results predict. Also, understanding the asymptotic properties of a problem setting is an important first step. Investigating whether sublinear-regret algorithms can be constructed in our problem setting is important future work.

Whilst our approach to achieving differential privacy is desirable as it is agnostic to the RL algorithm itself, it leaves open the possibility that the same privacy guarantees could be achieved by directly modifying an RL algorithm. Such an approach may be able to better deal with higher dimensional state spaces, which can adversely impact the approximation error, and is one of the limitations of our approach. Our analysis is also performed under some regularity assumptions; removing these assumptions would provide more generally applicable theoretical guarantees and is an interesting topic for future research.

Impact Statement

As reinforcement learning algorithms see wider adoption and begin interacting with humans and their data, issues around protecting privacy become more pertinent. Our work can be seen as providing a promising start and solid theoretical foundation to providing privacy protections in an important problem class where reinforcement learning may be applied in the future.

References

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. Reinforcement learning: Theory and algorithms. unpublished (publicly available), 2022. URL https://rltheorybook.github.io/rltheorybook_AJKS.pdf.

Arango, M. and Pelov, L. COVID-19 pandemic cyclic lockdown optimization using reinforcement learning. *CoRR*, abs/2009.04647, 2020. URL https://arxiv.org/abs/2009.04647.

Azize, A. and Basu, D. When privacy meets partial information: A refined analysis of differentially private bandits. *Advances in Neural Information Processing Systems*, 35: 32199–32210, 2022.

Balle, B., Gomrokchi, M., and Precup, D. Differentially private policy evaluation, 2016. URL https://arxiv.org/abs/1603.02010.

Berestizshevsky, K., Sadzi, K.-E., Even, G., and Shahar, M. Optimization of resource-constrained policies for covid-19 testing and quarantining. *Journal of Communications and Networks*, 23(5):326–339, 2021. doi: 10.23919/JCN.2021.000029.

Bretagnolle, J. and Huber, C. Estimation des densités: risque minimax. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 47:119–137, 1979.

Charpentier, A., Elie, R., Laurière, M., and Tran, V. C. Covid-19 pandemic control: balancing detection policy and lockdown intervention under icu sustainability. *Mathematical Modelling of Natural Phenomena*, 15:57, 2020a.

Charpentier, A., Elie, R., Laurière, M., and Tran, V. C. Covid-19 pandemic control: balancing detection policy and lockdown intervention under ICU sustainability, 2020b. URL https://arxiv.org/abs/2005.06526.

Cho, E., Myers, S. A., and Leskovec, J. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1082–1090, 2011.

Chowdhury, S. R. and Zhou, X. Differentially private regret minimization in episodic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6375–6383, 2022a.

Chowdhury, S. R. and Zhou, X. Distributed differential privacy in multi-armed bandits. *arXiv preprint arXiv:2206.05772*, 2022b.

Colas, C., Hejblum, B., Rouillon, S., Thiébaut, R., Oudeyer, P.-Y., Moulin-Frier, C., and Prague, M. Epidemioptim: A toolbox for the optimization of control policies in epidemiological models, 2020. URL https://arxiv.org/abs/2010.04452.

Dimitrakakis, C., Nelson, B., Zhang, Z., Mitrokotsa, A., and Rubinstein, B. I. Differential privacy for bayesian inference through posterior sampling. *Journal of machine learning research*, 18(11):1–39, 2017.

Dubey, A. and Pentland, A. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33:6003–6014, 2020.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60. IEEE, 2010.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Garcelon, E., Perchet, V., Pike-Burke, C., and Pirotta, M. Local differential privacy for regret minimization in reinforcement learning, 2020. URL https://arxiv.org/abs/2010.07778.

Guha Thakurta, A. and Smith, A. (nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49:209–232, 2002.

Kifer, D. and Machanavajjhala, A. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):1–36, 2014.

Kompella, V., Capobianco, R., Jong, S., Browne, J., Fox, S., Meyers, L., Wurman, P., and Stone, P. Reinforcement learning for optimization of covid-19 mitigation policies, 2020. URL https://arxiv.org/abs/2010.10560.

Kurniawati, H. Partially observable markov decision processes and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:253–277, 2022.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Liao, C., He, J., and Gu, Q. Locally differentially private reinforcement learning for linear mixture markov decision processes. In *Asian Conference on Machine Learning*, pp. 627–642. PMLR, 2023.

Lin, L.-J. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992.

Luyo, P., Garcelon, E., Lazaric, A., and Pirotta, M. Differentially private exploration in reinforcement learning with linear representation. *arXiv preprint arXiv:2112.01585*, 2021.

Mao, H., Chen, S., Dimmery, D., Singh, S., Blaisdell, D., Tian, Y., Alizadeh, M., and Bakshy, E. Real-world video adaptation with reinforcement learning. *arXiv preprint arXiv:2008.12858*, 2020.

Mishra, N. and Thakurta, A. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 592–601, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Monahan, G. E. State of the art—a survey of partially observable markov decision processes: theory, models, and algorithms. *Management science*, 28(1):1–16, 1982.

Newman, M. *Networks*. Oxford University Press, 2018.

Ngo, D. D. T., Vietri, G., and Wu, S. Improved regret for differentially private exploration in linear mdp. In *International Conference on Machine Learning*, pp. 16529–16552. PMLR, 2022.

Nowzari, C., Preciado, V. M., and Pappas, G. J. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, 36 (1):26–46, 2016.

Pan, X., Wang, W., Zhang, X., Li, B., Yi, J., and Song, D. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 368–376, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3), 2015.

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

Qiao, D. and Wang, Y.-X. Near-optimal differentially private reinforcement learning. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9914–9940. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/qiao23a.html.

Ren, W., Zhou, X., Liu, J., and Shroff, N. B. Multi-armed bandits with local differential privacy. *arXiv preprint arXiv:2007.03121*, 2020.

Sajed, T. and Sheffet, O. An optimal private stochastic-mab algorithm based on optimal private stopping rule. In *International Conference on Machine Learning*, pp. 5579–5588. PMLR, 2019.

Shani, G., Pineau, J., and Kaplow, R. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27:1–51, 2013.

Shariff, R. and Sheffet, O. Differentially private contextual linear bandits, 2018. URL https://arxiv.org/abs/1810.00068.

Singh, S. P., Jaakkola, T., and Jordan, M. I. Learning without state-estimation in partially observable markovian decision processes. In *Machine Learning Proceedings 1994*, pp. 284–292. Elsevier, 1994.

Tossou, A. and Dimitrakakis, C. Algorithms for differentially private multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Tossou, A. and Dimitrakakis, C. Achieving privacy in the adversarial multi-armed bandit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Vietri, G., Balle, B., Krishnamurthy, A., and Wu, Z. S. Private reinforcement learning with pac and regret guarantees, 2020. URL https://arxiv.org/abs/2009.09052.

Wang, B. and Hegde, N. Privacy-preserving q-learning with functional noise in continuous spaces. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, H. and Yu, S. Robo-advising: Enhancing investment with inverse optimization and deep reinforcement learning. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 365–372. IEEE, 2021.

Yang, J. and Leskovec, J. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, pp. 1–8, 2012.

Zheng, K., Cai, T., Huang, W., Li, Z., and Wang, L. Locally differentially private (contextual) bandits learning. *Advances in Neural Information Processing Systems*, 33: 12300–12310, 2020.

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

A. Proof of Theorem 1

We first provide some technical results that we will use.

The utility bound of the exponential mechanism is given as follows.

Theorem 3 ((Dwork et al., 2014)). *Let $\mathcal{M}_{u,\mathcal{R}}(x)$ be the exponential mechanism that, given an input database x, outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp(\frac{\epsilon u(x,r)}{2\Delta_u})$. Let $u^*(x) = \max_{r\in\mathcal{R}} u(x,r)$ and $\mathcal{R}^* = \{r \in \mathcal{R} : u(x,r) = u^*(x)\}$. Then*

$$\mathbb{P}\left[u(x, \mathcal{M}_{u,\mathcal{R}}(x)) \leq u^*(x) - \frac{2\Delta_u}{\epsilon}\left[\ln\frac{|\mathcal{R}|}{|\mathcal{R}^*|} + t\right]\right] \leq e^{-t}.$$

Proposition 1 is a simple consequence of Theorem 3.

Proposition 1. Let $\mathcal{M}_{u,\mathcal{S}}(x)$ be the exponential mechanism with $u(x,s') = -\left\|q(x) - s'\right\|_1$, where $q(x)$ is the histogram query with sensitivity $\Delta_u = 2/N$. Given a database x, let $s = q(x)$ and s' be the output of $\mathcal{M}_{u,\mathcal{S}}(x)$. Then for $\alpha > 0$,

$$\mathbb{P}\left(\left\|s - s'\right\|_1 > \alpha\right) \leq (N+1)\exp\left(-\frac{N\alpha\epsilon}{4K}\right).$$

Proof. Clearly, $u^*(x) = 0$ and is only attained when $s' = s$. Letting $\beta = e^{-t}$, we have by Theorem 3

$$\mathbb{P}\left(\left\|s - s'\right\|_1 \geq \frac{4}{N\epsilon}\ln\frac{|\mathcal{S}|}{\beta}\right) \leq \beta.$$

Noting that $\mathcal{S} \subseteq \{0, \frac{1}{N}, , \ldots, \frac{N-1}{N}, 1\}^K$, we thus have $|\mathcal{S}| \leq (N+1)^K$, yielding

$$\mathbb{P}\left(\left\|s - s'\right\|_1 \geq \frac{4K}{N\epsilon}\ln\frac{N+1}{\beta}\right) \leq \beta.$$

Letting $\alpha = \frac{4K}{N\epsilon}\ln\frac{N+1}{\beta}$ and solving for β gives the result. □

Lemma 1 (Lipschitz preservation under convolution)**.** Suppose f is an L-Lipschitz function and ϕ is a function such that $\int \phi(x)dx = 1$ and $\phi(x) \geq 0$ for all x. Then $g = f * \phi$ is also an L-Lipschitz function.

Proof.

$$\begin{aligned}\|g(z) - g(z')\| &\leq \left\|\int (f(z+x) - f(z'+x))\phi(x)dx\right\| \\ &\leq \int \|f(z+x) - f(z'+x)\|\,\phi(x)dx \\ &\leq L\|z - z'\|\int \phi(x)dx \\ &\leq L\|z - z'\|\ .\end{aligned}$$

□

Lemma 2 lets us change the mean of the exponential mechanism for the summand.

Lemma 2. Let $\mathcal{M}$ be the exponential mechanism and suppose the utility function $u : \mathcal{S}\times\mathcal{S} \to \mathbb{R}$ is symmetric, i.e. $u(s,s') = u(s',s)$. Let $P_{\mathcal{M}}(s|s')$ be the probability that $\mathcal{M}$ outputs s when given input s'. Let $C(s) = \sum_r \exp\left(\frac{\epsilon u(s,r)}{2\Delta_u}\right)$ and $C_{\max} = \max_{s,s'}\frac{C(s)}{C(s')}$. Then given any function $f : \mathcal{S} \to \mathbb{R}$, for all $s \in \mathcal{S}$,

$$\sum_{s'} P_{\mathcal{M}}(s|s')f(s') \leq C_{\max}\sum_{s'} P_{\mathcal{M}}(s'|s)f(s').$$

Proof. Let $\lambda_\epsilon = \exp(\epsilon/2\Delta_u)$.

$$\begin{aligned}&\sum_{s'} P_{\mathcal{M}}(s|s')f(s') = \sum_{s'}\frac{\lambda_\epsilon^{u(s',s)}}{C(s')}f(s') \\ &= \sum_{s'}\frac{C(s)}{C(s')}\frac{\lambda_\epsilon^{u(s',s)}}{C(s)}f(s') = \sum_{s'}\frac{C(s)}{C(s')}\frac{\lambda_\epsilon^{u(s,s')}}{C(s)}f(s') \\ &\leq C_{\max}\sum_{s'} P_{\mathcal{M}}(s'|s)f(s').\end{aligned}$$

□

The Simulation Lemma (Kearns & Singh, 2002; Agarwal et al., 2022) lets us bound the value function error in terms of the error in the transition functions.

Lemma 3 (Simulation Lemma (Kearns & Singh, 2002))**.** Let $M = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$ and $\tilde{M} = (\mathcal{S}, \mathcal{A}, r, \tilde{P}, \gamma)$ be two MDPs that differ only in the transition model. Given a policy π, let Q^π be the value function under π in M and $\tilde{Q}^\pi$ be the value function under π in $\tilde{M}$. Then for all π

$$\left\|Q^\pi - \tilde{Q}^\pi\right\|_\infty \leq \frac{\gamma}{1-\gamma}\left\|(P - \tilde{P})V^\pi\right\|_\infty.$$

A.1. Main Result

Theorem 1. Let M be the MDP environment and $\tilde{M}$ denote the privatised MDP under an (ϵ', δ)-differentially private mechanism. Let Q^* be the optimal value function in M and $\tilde{Q}^*$ be the optimal value function in $\tilde{M}$. Then,

$$\left\|Q^* - \tilde{Q}^*\right\|_\infty \leq \mathcal{O}\left(N^K\exp\left(-\frac{\sqrt{N}\epsilon'}{K}\right) + \frac{1}{\sqrt{N}}\right),$$

where $\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$, K is the state dimension, and N is the population size.

Proof. We have

$$\begin{aligned}\left|Q^*(s,a)-\tilde{Q}^*(s,a)\right| &= \left|\sup_\pi Q^\pi(s,a)-\sup_\pi \tilde{Q}^\pi(s,a)\right| \\ &\leq \sup_\pi \left|Q^\pi(s,a)-\tilde{Q}^\pi(s,a)\right| \\ &\leq \sup_\pi \left\|Q^\pi-\tilde{Q}^\pi\right\|_\infty . \end{aligned} \tag{4}$$

Then (4) can be bound with Lemma 3. For any policy $\bar{\pi}$,

$$\begin{aligned}\left\|Q^{\bar{\pi}}-\tilde{Q}^{\bar{\pi}}\right\|_\infty &\leq \frac{\gamma}{1-\gamma}\left\|(P-\tilde{P}^\pi)V^{\bar{\pi}}\right\|_\infty \\ &\leq \frac{\gamma}{1-\gamma}\max_{s,a}\left\|(P_{sa}-\tilde{P}^\pi_{sa})\right\|_1 \left\|V^{\bar{\pi}}\right\|_\infty \\ &\leq \frac{\gamma r_{\max}}{(1-\gamma)^2}\max_{s,a}\left\|(P_{sa}-\tilde{P}^\pi_{sa})\right\|_1 . \end{aligned} \tag{5}$$

Here $\tilde{P}^\pi$, as defined in (2), denotes the transition model in $\tilde{M}$ under a behaviour policy π. Writing $\bar{P}(s'|s_1,a) = \sum_{s_2\in\mathcal{S}} P(s_2|s_1,a)P_{\mathcal{M}}(s'|s_2)$, we have, for any s,a,

$$\begin{aligned}&\left\|P_{sa}-\tilde{P}^\pi_{sa}\right\|_1 = \left\|P_{sa}-\sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)\bar{P}_{s_1a}\right\|_1 \\ &= \left\|\sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)P_{sa}-\sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)\bar{P}_{s_1a}\right\|_1 \\ &\leq \sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)\left\|P_{sa}-\bar{P}_{s_1a}\right\|_1 \\ &\leq \sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)\left(\underbrace{\left\|P_{sa}-\bar{P}_{sa}\right\|_1}_{(one)}+\underbrace{\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1}_{(two)}\right), \end{aligned}$$

where the last two steps follows from the triangle inequality and splitting $\left\|P_{sa}-\bar{P}_{s_1a}\right\|_1$. The first term can be viewed as the error due to output privatisation and the second term can be viewed as the error due to input privatisation. We will look to bound the two terms separately.

Term (one)
By the Bretagnolle-Huber inequality (Bretagnolle & Huber, 1979), term (one) can be bound as:

$$\left\|P_{sa}-\bar{P}_{sa}\right\|_1 \leq 2\sqrt{1-\exp(-KL(P_{sa}\,||\,\bar{P}_{sa})}.$$

The KL divergence term can be lower bound as follows:

$$\begin{aligned}-KL(P_{sa}\,||\,\bar{P}_{sa}) &= \sum_{s'\in\mathcal{S}}P^{s'}_{sa}\log\frac{\bar{P}^{s'}_{sa}}{P^{s'}_{sa}} \\ &= \sum_{s'\in\mathcal{S}}P^{s'}_{sa}\log\frac{\sum_{s_2\in\mathcal{S}}P_{\mathcal{M}}(s'|s_2)P^{s_2}_{sa}}{P^{s'}_{sa}} \\ &\overset{(a)}{\geq} \sum_{s'\in\mathcal{S}}P^{s'}_{sa}\log\frac{P_{\mathcal{M}}(s'|s')P^{s'}_{sa}}{P^{s'}_{sa}} \\ &= \sum_{s'\in\mathcal{S}}P^{s'}_{sa}\log P_{\mathcal{M}}(s'|s') \\ &\geq \min_{s'\in\mathcal{S}}\log P_{\mathcal{M}}(s'|s'). \end{aligned} \tag{6}$$

Here (a) follows as we shrink the sum to just when $s_2 = s'$. Thus term (one) can be bound as:

$$\begin{aligned}\left\|P_{sa}-\bar{P}_{sa}\right\|_1 &\leq 2\min_{s'\in\mathcal{S}}\sqrt{1-P_{\mathcal{M}}(s'|s')} \\ &= 2\min_{s'\in\mathcal{S}}\sqrt{1-C(s')^{-1}}, \end{aligned}$$

where $C(s') = \sum_{s\in\mathcal{S}}\exp\left(-\frac{N\epsilon'}{4K}\left\|s-s'\right\|_1\right)$. Noting that $C(s') \leq |\mathcal{S}|\exp\left(-\frac{N\epsilon'}{4K}\right)$ gives us:

$$\begin{aligned}\sqrt{1-C(s')^{-1}} &= \sqrt{\frac{C(s')-1}{C(s')}} \\ &\leq \sqrt{C(s')} \\ &\leq C\,|\mathcal{S}|\exp\left(-\frac{N\epsilon'}{K}\right) \\ &\leq \mathcal{O}\left(N^K\exp\left(-\frac{N\epsilon'}{K}\right)\right). \end{aligned}$$

where the first inequality follows as $C(s') \geq 1$ and the last inequality follows as $|\mathcal{S}| \leq \mathcal{O}(N^K)$.

Term (two)
Using Equation 3, we have:

$$\begin{aligned}&\sum_{s_1\in\mathcal{S}}P^\pi(s_1|s,a)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\ &= \sum_{s_1\in\mathcal{S}}\frac{P_{\mathcal{M}}(s|s_1)P^\pi(s_1|a)}{\tilde{P}^\pi(s|a)}\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\ &\leq H\sum_{s_1\in\mathcal{S}}P_{\mathcal{M}}(s|s_1)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\ &\leq HC_{\max}\sum_{s_1\in\mathcal{S}}P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 , \end{aligned}$$

where the last step follows from Lemma 2.

For $\alpha > 0$, define the α-ball around a state s and its complement as:

$$\begin{aligned}B_\alpha(s) &:= \{s'\in\mathcal{S} : \|s-s'\|_1 < \alpha\} \\ B^c_\alpha(s) &:= \{s'\in\mathcal{S} : \|s-s'\|_1 \geq \alpha\}. \end{aligned}$$

Now note that $\bar{P}$ is L-Lipschitz by Lemma 1 and Assumption 2. Splitting the sum over $B_\alpha(s)$ and $B^c_\alpha(s)$ gives us the following bound:

$$\begin{aligned}
&\sum_{s_1\in\mathcal{S}} P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\
&\quad=\sum_{s_1\in B_\alpha(s)} P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 + \\
&\qquad\sum_{s_1\in B^c_\alpha(s)} P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\
&\quad\le L\alpha + 2(N+1)\exp\left(-\frac{N\alpha\epsilon'}{4K}\right) \\
&\quad\le \mathcal{O}\left(N\exp\left(-\frac{N\alpha\epsilon'}{K}\right)+\alpha\right)
\end{aligned}$$

The final inequality follows as the first term is bound by the Lipschitz property. The second term is bound by noting that the L_1 norm between distributions is bound by 2 and applying Proposition 1. Picking $\alpha = N^{-\frac{1}{2}}$ gives us

$$\begin{aligned}
&\sum_{s_1\in\mathcal{S}} P_{\mathcal{M}}(s_1|s)\left\|\bar{P}_{sa}-\bar{P}_{s_1a}\right\|_1 \\
&\qquad\le \mathcal{O}\left(N\exp\left(-\frac{\sqrt{N}\epsilon'}{K}\right)+\frac{1}{\sqrt{N}}\right).
\end{aligned}$$

Now combining the bounds on terms (one) and (two) by picking the higher growth rate terms gives the final result.

□

B. Proof of Theorem 2

Theorem 2. A family of mechanisms $\mathcal{F}$ satisfies (ϵ,δ)-differential privacy under T-fold adaptive composition iff every sequence of mechanisms $\mathcal{M}=(\mathcal{M}_1,\ldots,\mathcal{M}_T)$, with $\mathcal{M}_i\in\mathcal{F}$, satisfies (ϵ,δ)-Pufferfish privacy with parameters $(\mathbb{S},\mathbb{Q},\Theta)$.

Proof. **Showing T-fold adaptive composition implies (ϵ,δ)-Pufferfish privacy.**

Let $y_i=\mathcal{M}_i(D_i$ denote the output from a mechanism $\mathcal{M}_i$ and let $D_{1:T}$ be a sequence of databases where $\sigma^J_{I,x}$ holds and $D'_{1:T}$ be constructed from $D_{1:T}$ by removing individuals in I and replacing them with individuals in J, thus having $\sigma^I_{J,x'}$ hold on $D'_{1:T}$. We will denote $\mathcal{M}(D_{1:T})=y_{1:T}$ as the sequence $\mathcal{M}=(\mathcal{M}_1,\ldots,\mathcal{M}_T)$ applied to a dataset sequence.

Recall that satisfying (ϵ,δ) differential privacy under k-fold adaptive composition provides the following guarantee:

$$\max_{\omega:P(\mathcal{M}(D_{1:T})=\omega)\ge\delta}\ln\frac{P(\mathcal{M}(D_{1:T})=\omega)-\delta}{P(\mathcal{M}(D'_{1:T})=\omega)}\le\epsilon.$$

Let $Data_{1:T}$ be a random variable denoting the sequence of databases. Then

$$\begin{aligned}
&P(\mathcal{M}(Data_{1:T})=y_{1:T}|\sigma^J_{I,x},\theta) \\
&=\int P(Data_{1:T}=D_{1:T}|G_{1:T},y_{1:T},\sigma^J_I(x),\theta) \\
&\cdot P(\mathcal{M}(D_{1:T})=y_{1:t})\rho(G_{1:T})d(G_{1:T},D_{1:T}).
\end{aligned}$$

We now have that $P(\mathcal{M}(D_{1:T}) = y_{1:t}) \le e^\epsilon P(\mathcal{M}(D'_{1:T}) = y_{1:t}) + \delta$ as every element from $\mathcal{M}$ is from the class of mechanisms $\mathcal{F}$ that satisfies (ϵ,δ) differential privacy under k-fold adaptive composition. The distribution over $Data_{1:T}$ can also be factorized as:

$$\begin{aligned}
&P(Data_{1:T}=D_{1:T}|G_{1:T},y_{1:T},\sigma^J_I(x),\theta) \\
&\quad=\prod_{t=1}^{T}P(Data_t=D_t|G_{<t},D_{<t},y_{<t}\sigma^{j_t}_{t,i_t}(x_t),\theta).
\end{aligned}$$

We have that

$$\begin{aligned}
&P(Data_t=D_t|G_{<t},D_{<t},y_{<t},\sigma^{j_t}_{t,i_t}(x_t),\theta) \\
&\quad=P(Data_t=D'_t|G_{<t},D_{<t},y_{<t},\sigma^{i_t}_{t,j_t}(x'_t),\theta),
\end{aligned}$$

for all $\theta\in\Theta$. This is because the participation and values of all other individuals in the datasets D_t and D'_t are conditionally independent of i_t and j_t. Thus we can claim:

$$\begin{aligned}
&P(Data_{1:T}=D_{1:T}|G_{1:T},y_{1:T},\sigma^J_I(x),\theta) \\
&\quad=P(Data_{1:T}=D'_{1:T}|G_{1:T},y_{1:T},\sigma^I_J(x),\theta). \quad (7)
\end{aligned}$$

Substituting Equation 7 in Equation B allows us to claim

$$\begin{aligned}
&P(\mathcal{M}(Data_{1:T}=y_{1:T}|\sigma^J_I(x),\theta) \\
&\qquad=P(\mathcal{M}(Data_{1:T}=y_{1:T}|\sigma^I_J(x),\theta),
\end{aligned}$$

thus satisfying (ϵ,δ)-Pufferfish privacy.

Showing (ϵ,δ)-Pufferfish privacy implies T-fold adaptive composition.

We show for any pair of neighbouring databases there is a $\theta\in\Theta$ that preserves differential privacy. Then, preserving (ϵ,δ)-Pufferfish privacy preserves (ϵ,δ)-differential privacy under k-fold adaptive composition.

Let $I=(i_t)_{t\in[T]}$ and $J=(j_t)_{t\in[T]}$ be two sequences of individuals such that $\forall t\in[T]$, $i_t\neq j_t$. Suppose $D_{1:T}$ and $D'_{1:T}$ are two neighbouring database sequences where $\sigma^J_I(x)$ and $\sigma^I_J(x)$ hold respectively. We choose $\theta\in\Theta$ with the following definitions for all $t\in[T]$:

$$\nu_{t,\ell}(G_{<t},X_{<t})=\begin{cases}1 & \text{if } \ell=k_i, i=2,\ldots,n\\ 0 & \text{if } x_{t,\ell}\notin Data_t\\ p_{t,i} & \text{if } \ell=i_t\\ p_{t,j} & \text{if } \ell=j_t\end{cases} \quad (8)$$

where $p_{t,i}, p_{t,j} \in (0,1)$, and

$$f_{t,\ell}(x_{t,\ell}|G_{<t}, X_{<t}) = \begin{cases} 1 & \text{if } \ell \in \{i, j, k_i, i = 2, \ldots, n\} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Starting from the (ϵ, δ)-Pufferfish privacy guarantee, we have:

$$P(\mathcal{M}_t(Data_t) = \omega|\sigma^j_{i,x}, \theta) \leq e^\epsilon P(\mathcal{M}_t(Data_t) = \omega|\neg\sigma^i_{j,x}, \theta). \tag{10}$$

Under θ, there is only one sequence $D_{1:T}$ where $\sigma^J_I(x)$ holds and one sequence $D'_{1:T}$ where $\sigma^I_J(x')$ holds. Thus Equation 10 is equivalent to the following:

$$P(\mathcal{M}_t(D) = \omega|\sigma^j_{i,x}, \theta) \leq e^\epsilon P(\mathcal{M}_t(D') = \omega|\sigma^i_{j,x}, \theta),$$

which is equivalent to the (ϵ, δ)-differential privacy guarantee under k-fold adaptive composition:

$$P(\mathcal{M}_t(D) = \omega) \leq e^\epsilon P(\mathcal{M}_t(D') = \omega).$$

Choosing $\theta \in \Theta$ in this manner for all neighbouring data sequences and repeating the calculations proves the final result.

□

C. Differentially Private DQN Algorithm (DP-DQN)

Our experimental results use a concrete instantiation of Algorithm 1 with DQN (Mnih et al., 2015) and epsilon-greedy for exploration. We display the full algorithm details in Algorithm 2. The state privatisation mechanism used is the Laplace mechanism projected back to the state space (see Algorithm 3). Lines 10-21 can be considered the RL algorithm expanded. There are two minor differences to how Algorithm 1 is written. Firstly, the replay buffer and target network are written such that they are not a part of the RL algorithm itself. This is done to make clear that the buffer and target network maintain state across all iterations. Finally, DQN returns a value function instead of a policy. This is a minor change however as the policy used is simply the epsilon-greedy policy with respect to the returned value function.

The parameters used in each experiment are listed in Table 1. The learning rate (α) is not listed as we use the default settings of the RMSProp optimizer in PyTorch to optimize the neural network.

Algorithm 2 Differentially Private DQN (DP-DQN)

1: **Input:** Environment $M = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$, initial state $s_0 \in \mathcal{S}$.
2: **Parameters:** privacy parameters (ϵ, δ), time horizon T, batch size B, target update step D, population size N, learning rate α, initial exploration rate $\epsilon_{start} < 1$, decay rate $\kappa < 1$.
3: **Initialize:** network parameters θ randomly, target network parameters $\bar{\theta} = \theta$, $\epsilon_{explore} = \epsilon_{start}$.
4: `Buffer` $\leftarrow \{\}$.
5: $\epsilon' = \frac{\epsilon}{2\sqrt{2T\log(1/\delta)}}$.
6: $\tilde{s}_0 = \texttt{ProjectedLaplace}_{\epsilon'}(s_0)$.
7: **for** $t = 0, 2, \ldots, T$ **do**
8: $p \sim Uniform([0,1])$.
9: $\tilde{a}_t = \arg\max_a Q_\theta(\tilde{s}_t, a)$ **if** $p > \epsilon_{explore}$ **else** $\tilde{a}_t \sim Uniform(\mathcal{A})$.
10: Receive $s_{t+1} \sim P(\cdot|s_t, \tilde{a}_t)$.
11: $\tilde{s}_{t+1} = \texttt{ProjectedLaplace}_{\epsilon'}(s_{t+1})$.
12: $\tilde{r}_t = r(\tilde{s}_t, a_t)$,
13: Append $(\tilde{s}_t, \tilde{a}_t, \tilde{r}_t, \tilde{s}_{t+1})$ to `Buffer`.
14: **if** $t > B$ **then**
15: **for** $i = 1, \ldots, B$ **do**
16: $(s, a, r, s') \sim$ Uniform(`Buffer`).
17: $y_i \leftarrow r + \gamma \max_a Q_{\bar{\theta}}(s', a)$.
18: $\ell_i \leftarrow \frac{1}{2}\left(Q_\theta(s,a) - y_i\right)^2$.
19: **end for**
20: Run one step SGD $\theta \leftarrow \theta + \alpha \frac{1}{B} \nabla_\theta \sum_{i=1}^B \ell_i$.
21: **end if**
22: **if** $t \mod D = 0$ **then**
23: Update target network parameters $\bar{\theta} \leftarrow \theta$.
24: **end if**
25: $\epsilon_{explore} \leftarrow 0.03 + (\epsilon_{start} - 0.03) \cdot e^{-\kappa t}$.
26: **end for**

Parameter	82K	196K	1.1M
δ	1e−3	1e−3	1e−3
γ	0.999	0.999	0.999
T	5e5	5e5	5e5
B	128	128	128
D	800	800	800
ϵ_{start}	0.9999	0.9999	0.9999
κ	1e−5	1e−5	1e−5

Table 1. DP-DQN configuration for each experiment.

Algorithm 3 `ProjectedLaplace` Mechanism

Input: state $s \in \mathcal{S}$
Parameters: Privacy parameter ϵ.
$s' = s + \eta$, where $\eta \in \mathbb{R}^K$ and for $i \in [K]$, $\eta_i \sim \text{Lap}\left(\frac{2}{N\epsilon}\right)$.
$\tilde{s} = \arg\min_{\bar{s} \in \mathcal{S}} \|s' - \bar{s}\|_1$.
Return $\tilde{s}$.

D. Experiment Details

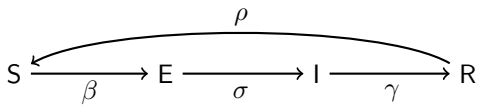

Figure 4. Visualisation of the parameters that govern the transitions between states for individuals in the SEIRS process over contact networks.

All experiments were performed on a shared server with a 32-Core Intel(R) Xeon(R) Gold 5218 CPU, 192 gigabytes of RAM. A single NVIDIA GeForce RTX 3090 GPU was also used.

Table 3 details addition information about each graph dataset that was used. All datasets were retrieved from the Stanford Large Graph Network Dataset (Leskovec & Krevl, 2014).

Table 2 details the parameters used for each experiment. The parameter α denotes the weighting used in the reward function. The remaining parameters correspond to the transition rates in the SEIRS epidemic model (see Section 3). A graphical representation of the parameters governing state transitions is shown in Figure 4.

Experiment	α	β	σ	γ	ρ
82K	0.8	0.2	0.3	0.1	0.01
196K	0.8	0.2	0.3	0.1	0.01
1.1M	0.8	0.2	0.3	0.1	0.01

Table 2. Environment parameters for each experiment.

Dataset	**Name**	**Nodes**	**Edges**	**Description**
`Slashdot`(Leskovec et al., 2009)	82K	82,168	948,464	Slashdot social network from February 2009.
`Gowalla`(Cho et al., 2011)	196K	196,591	950,327	Gowalla location based online social network.
`Youtube`(Yang & Leskovec, 2012)	1.1M	1,134,890	2,987,624	Youtube online social network.

Table 3. Summary of network datasets used in experiments.