Replicable Reinforcement Learning

Eric Eaton* Marcel Hussing * Michael Kearns † Jessica Sorrell ‡

June 1, 2023

Abstract

The *replicability crisis* in the social, behavioral, and data sciences has led to the formulation of algorithm frameworks for replicability — i.e., a requirement that an algorithm produce identical outputs (with high probability) when run on two different samples from the same underlying distribution. While still in its infancy, provably replicable algorithms have been developed for many fundamental tasks in machine learning and statistics, including statistical query learning, the heavy hitters problem, and distribution testing. In this work we initiate the study of *replicable reinforcement learning*, providing a provably replicable algorithm for parallel value iteration, and a provably replicable version of R-max in the episodic setting. These are the first formal replicability results for control problems, which present different challenges for replication than batch learning settings.

1 Introduction

The growing prominence of machine learning (ML) and its widespread adoption across industries underscore the need for replicable research [Wagstaff, 2012, Pineau et al., 2021]. Many scientific fields have suffered from this same inability to reproduce the results of published studies [Begley and Ellis, 2012]. Replicability in ML requires not only the ability to reproduce published results [Wagstaff, 2012], as may be partially addressed by sharing code and data [Stodden et al., 2014], but also consistency in the results obtained from successive deployments of an ML algorithm in the same environment. However, the inherent variability and randomness present in ML pose challenges to achieving replicability, as these factors may cause significant variations in results.

Building upon foundations of algorithmic stability [Bousquet and Elisseeff, 2002], recent work in learning theory has established rigorous definitions for the study of supervised learning [Impagliazzo et al., 2022] and bandit algorithms [Esfandiari et al., 2023a] that are provably *replicable*, meaning that algorithms produce identical outputs (with high probability) when executed on distinct data samples from the same underlying distribution. However, these results have not been extended to the study of control problems such as reinforcement learning (RL), that have long been known to suffer from stability issues [White and Eldeib, 1994, Mannor et al., 2004, Islam et al., 2017, Henderson et al., 2018]. These stability issues have already sparked research into robustness for control problems including RL [Khalil et al., 1996, Nilim and Ghaoui, 2005, Iyengar, 2005]. Non-deterministic environments and evaluation benchmarks, the randomness of the exploration process, and the sequential interaction of an RL agent with the environment all complicate the ability to make RL replicable. Our work is orthogonal to that of the robustness literature and our goal is not to reduce the effect of these inherent characteristics, such as by decreasing the amount of exploration that an agent performs, but to develop replicable RL algorithms that support these characteristics.

*Department of Computer and Information Sciences, University of Pennsylvania. The research presented in this paper was partially supported by the DARPA SAIL-ON program under contract HR001120C0040, the DARPA ShELL program under agreement HR00112190133, and the Army Research Office under MURI grant W911NF20-1-0080.

†Department of Computer and Information Sciences, University of Pennsylvania.

‡Department of Computer and Information Sciences, University of Pennsylvania. Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness

Toward this goal, we initiate the study of replicable RL and develop the first set of RL algorithms that are provably replicable. We contend that the fundamental theoretical study of replicability in RL might advance our understanding of the aspects of RL algorithms that make replicability hard. In this work, we put on a similar lens as Impagliazzo et al. [2022] and consider replicability as an algorithmic property that can be achieved simultaneously with exploration and exploitation. First, we show that it is possible to obtain a near-optimal, replicable policy given sufficiently many samples from every state in the environment. This notion is then naturally extended to replicable exploration.

Our contributions can be summarized as follows. We provide two novel and efficient algorithms to

- show that stochastic, sample-based value iteration can be done replicably and
- replicably explore the space of an MDP while also finding an optimal policy.

We experimentally validate that our algorithms require much fewer samples than theory suggests.

2 Preliminaries

2.1 Reinforcement Learning

We consider the problem of solving a discounted Markov decision process (MDP) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, P, \gamma, \mu\}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r, transition kernel P, discount factor γ, and initial state distribution μ. We assume that the size of the state space $|\mathcal{S}|$ and number of possible actions $|\mathcal{A}|$ are finite and not too large. Further, we assume that the rewards for every state-action pair are deterministic, bounded, and known. Relaxing this assumption might not necessarily seem straightforward in our goal of replicable RL, as the stochastic reward would need to be made replicable. However, the case can be handled by our algorithms with minor modifications and only constant factor overhead. The goal is to find a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the cumulative discounted reward $J_h = \sum_{k=h}^{\infty} \gamma^{k-h} r_k(s, a)$. We use the typical definitions of the value and Q-value functions for the expected cumulative discounted return from a state or state-action pair, respectively:

$$V_\pi(s) = \mathop{\mathbb{E}}_{\pi,P}[J_h | s_h = s] \qquad\qquad Q_\pi(s, a) = \mathop{\mathbb{E}}_{\pi,P}[J_h | s_h = s, a_h = a] \ . \tag{1}$$

To show the various difficulties that come from trying to achieve replicability in RL, we consider two different settings to examine various components of the problem.

Parallel Sampling Setting First, we ask whether it is even possible to obtain a replicable policy from empirical samples without considering the challenges of exploration. For this we can consider the setting of generative models $G_\mathcal{M}$, or more precisely, the parallel sampling setting. In the parallel sampling model, first introduced by Kearns and Singh [1998a], one has access to a parallel generative sampling subroutine $\mathbf{PS}(G_\mathcal{M})$. A single call to $\mathbf{PS}(G_\mathcal{M})$ will return, for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, a randomly sampled next state $s' \in \mathcal{S}$ drawn from $P(s'|s, a)$. The key advantage is that this model separates learning from the quality of the exploration procedure.

Definition 2.1 (Generative Model)**.** *Let $\mathcal{M}$ denote an arbitrary MDP, then a generative model $G_\mathcal{M}((s, a))$ is a randomized algorithm that, given a state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, outputs a deterministic reward $r(s, a)$ and a next state s' sampled from $P(\cdot|s, a)$.*

Definition 2.2 (Parallel Sampling)**.** *Let $\mathcal{M}$ denote an arbitrary MDP, then a call to the parallel sampling subroutine $\mathbf{PS}(G_\mathcal{M})$ returns exactly one sample $s'_i \sim G_\mathcal{M}((s_i, a_i))$ for every state-action pair (s_i, a_i) in $\mathcal{S} \times \mathcal{A}$ of $\mathcal{M}$ using a generative model.*

Episodic Setting The second setting we consider is one in which an algorithm does have to explore the MDP before it can obtain an optimal policy. More precisely, we consider the episodic setting where, in every episode $t \in \{1, 2, ..., T\}$, the agent starts in a position $s_0 \sim \mu$ and interacts with the environment for a fixed amount of time H. At any step $h \in [1, H]$, the agent is in some state s_h, selects an action a_h, and a new state s_{h+1} is generated using $G_{\mathcal{M}}((s_h, a_h))$. Gathering a trajectory $\tau = (s_0, a_0, r_0, .., s_H, a_H, r_H)$ under policy π can be thought of as a draw from a distribution $\tau \sim P^{\pi}_{\mathcal{M}}(\tau)$. Here, we will omit the sub-and superscripts when clear from context.

2.2 Replicability

We build on the recent framework by Impagliazzo et al. [2022], which considers replicability as a property of randomized algorithms that take as input a dataset sampled i.i.d. from an arbitrary distribution. They consider an algorithm to be replicable if, on two runs in which its internal randomness is fixed and its input data is resampled, it outputs the same result with high probability:

Definition 2.3 (Replicability)**.** *Fix a domain $\mathcal{X}$ and target replicability parameter $\rho \in (0, 1)$. A randomized algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{Y}$ is* ρ-replicable *if for all distributions D over $\mathcal{X}$, randomizing over the internal randomness of $\mathcal{A}$ and choice of samples S_1, S_2, each of size n drawn i.i.d. from D, we have: $\boldsymbol{Pr}_{S_1,S_2,r}[\mathcal{A}(S_1; r) \neq \mathcal{A}(S_2; r)] \leq \rho$.*

Several key tools that were introduced by Impagliazzo et al. [2022] will prove useful or yield inspiration for the algorithms developed in this work. One of the key observations is that many of the computations in RL can be phrased as statistical queries, defined as follows:

Definition 2.4 (Statistical Query, [Kearns, 1998])**.** *Fix a distribution D over $\mathcal{X}$ and an accuracy parameter $\tau \in (0, 1)$. A statistical query is a function $\phi : \mathcal{X} \to [0, 1]$, and a mechanism M answers ϕ with tolerance τ on distribution D if $a \leftarrow M$ satisfies $a \in [\mathbb{E}_{x\sim D}[\phi(x)] \pm \tau]$.*

We will make direct use of the replicable algorithm for answering statistical queries by Impagliazzo et al. [2022] which will be useful to obtain replicable estimates of various measurements such as transition probabilities. We will refer to the replicable statistical query procedure as rSTAT. We note that Impagliazzo et al. [2022] also proves a lower-bound on the sample complexity required for replicable statistical queries, showing that the results below are essentially tight.

Theorem 2.1 (Replicable Statistical Queries, Impagliazzo et al. [2022])**.** *There is a ρ-replicable algorithm* rSTAT *such that for any distribution D over $\mathcal{X}$, replicability parameter $\rho \in (0, 1)$, accuracy parameter $\tau \in (0, 1)$, failure parameter $\delta \in O(\rho)$, and query $\phi : \mathcal{X} \to [0, 1]$, letting S be an i.i.d. sample of $n \in O\left(\frac{\log(1/\delta)}{(\rho-2\delta)^2\varepsilon^2}\right)$ elements drawn i.i.d. from D, we have that $a \leftarrow \mathsf{rSTAT}_{\tau,\rho}(S, \phi)$ satisfies $a \in [\mathbb{E}_{x\sim D}[\phi(x)] \pm \tau]$ except with probability at most δ over the samples S.*

3 Replicable Reinforcement Learning

To define replicability for the RL setting, we can adapt Definition 2.3 more or less exactly. The question that arises is which of the many RL objects should be made replicable? We separate the difficulty of replicability into three levels: replicability of the MDP, the value function, and the policy. Since these objects carry different amounts of information [Farahmand, 2011], the following relationships can be established.

If we are able to replicably (and accurately) estimate an MDP, we can always replicably compute an (optimal) value function using standard techniques on our estimates, and from replicable values functions we can obtain the corresponding policies. Note that the inverse is not true as we lose information when going from MDP to value function and then policy. As a result, we expect that replicable estimation of MDPs is the hardest setting in stochasic RL, followed by replicable value function and then policy estimation.

For replicability of control problems, a sensible measure to ask for is the production of identical policies, which are the ultimate object of primary interest. We would at least like to ensure that with high probability, we can obtain identical optimal policies across two runs of our RL procedures:

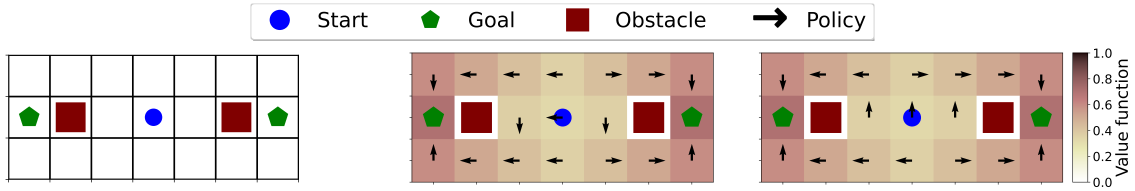


Figure 1: The GridWorld for our experiments (left) and two different policies that were generated by the Phased-Value Iteration Algorithm on this gridworld (center and right). Following the first policy (center) more likely reaches the left goal while following the right policy more likely reaches the right goal. All states except the goals have 0 reward. The actions are up, down, left and right; there is a 30% chance that after choosing an action the agent moves left or right of the target direction.

Definition 3.1 (Replicable policy estimation)**.** *Let $\mathcal{A}$ be a policy estimation algorithm that outputs a policy $\widehat{\pi}^* : \mathcal{S} \mapsto \mathcal{A}$ given a set of trajectories S sampled from an MDP. Algorithm $\mathcal{A}$ is ρ-replicable if, given independently sampled trajectory sets S_1 and S_2, and yielding policies $\widehat{\pi}^*_1$ and $\widehat{\pi}^*_2$, it holds that*

$$\begin{aligned}&\boldsymbol{Pr}_{S_1,S_2,r}[\widehat{\pi}^{*(1)}(a|s) \neq \widehat{\pi}^{*(2)}(a|s)] \leq \rho \\ &s.t.\ \widehat{\pi}^{*(1)}(a|s) \leftarrow \mathcal{A}(S_1; r) \quad \wedge \quad \widehat{\pi}^{*(2)}(a|s) \leftarrow \mathcal{A}(S_2; r)\ ,\end{aligned}$$

where r represents the internal randomness of $\mathcal{A}$. Trajectory sets S_1 and S_2 may potentially be gathered from the environment during the execution of an RL algorithm.

While this definition is the least we would like to achieve, the results we present in this paper provide stronger guarantees. Our Replicable Phased Value Iteration builds on [Kearns and Singh, 1998a] and ensures replicability of value functions, while our Replicable Episodic R-max follows [Kearns and Singh, 1998b, Brafman and Tennenholtz, 2003] and provides replicability of full MDPs. Equivalent formal definitions for replicable value and MDP estimation are given in Appendix A.

Current algorithms for sample-based RL problems will struggle to satisfy the Definition 3.1 of replicability and output different policies even in simple environments (see Figure 1). In some cases, this may not be problematic since the resulting policies will still be ε-optimal, but in practice it is often hard to tell when that is the case. Fixing replicability will support the identification of problematic solutions and encourage procedures that yield more stable solutions in the long run. Varying policies can, for example, arise from sample uncertainty, insufficient state-space coverage or differing exploration. In order to achieve replicability, all of the aforementioned challenges need to be addressed which makes for an intricate but interesting problem. With this in mind, the next section will introduce a first set of formally replicable algorithms that separate out some of these challenges.

4 Algorithms

4.1 Replicable Phased Value Iteration

The first question we would like to answer positively is whether it is even possible to achieve replicability when samples are drawn randomly from some distribution. For this, we use the parallel sampling model described in section 2. This model is well-suited for the task as it allows us to analyze sample-based value iteration independent of the exploration policy that collects the samples.

We provide a replicable version of indirect Phased Value Iteration (PVI) from Kearns and Singh [1998a]. In brief, the algorithm iterates T times and at every iteration makes m calls to $\mathbf{PS}(G_{\mathcal{M}})$, computes an approximate value estimate for every state and does one round of value updates. Kearns and Singh [1998a] provide the following Lemma 4.1 to show the optimality of the original procedure.

Lemma 4.1 (Phased Value Iteration Convergence, [Kearns and Singh, 1998a])**.** *Suppose the expectations* $|\mathbb{E}_{s'\sim\widehat{P}}[\widehat{V}_t(s')] - \mathbb{E}_{s'\sim P}[\widehat{V}_t(s')]| \leq \tau$ *are sufficiently accurate. For any MDP* $\mathcal{M}$*, Phased Value Iteration converges to a policy* $\widehat{\pi}^*$ *whose return is within* ε *of the optimal policy* π^**.*

Our algorithm operates similarly but we would like to achieve replicability on top of optimality. We use a randomized rounding procedure for statistical query estimation (rSTAT) provided by Impagliazzo et al. [2022] to compute the value estimates at every iteration. For this, we assume that the value function is normalized to the interval $[0, 1]$. A detailed description of our algorithm is provided in Algorithm 1. The Replicable Phased Value Iteration (rPVI) algorithm we provide satisfies Definition 3.1 and produces ε-optimal policies. It goes even one step further and produces not only replicable policies but replicable value functions. This is formalized in the following Theorem 4.1.

Theorem 4.1. *Let* $\varepsilon \in (0, 1)$ *be the accuracy and* $\rho \in (0, 1)$ *be the replicability parameter. Let* $\delta \in (0, 1)$ *be the sample failure probability. Set the number of calls to* $\mathbf{PS}(G_{\mathcal{M}})$ *at every iteration to*

$$m = O\left(\frac{\log^2(1/\varepsilon)|\mathcal{S}|^2|\mathcal{A}|^2}{\varepsilon^2(\rho - 2\delta)^2}\log\left(\frac{|\mathcal{S}||\mathcal{A}|}{\delta} + \log\log(1/\varepsilon)\right)\right) .$$

where O *supresses the dependence on* γ*. In two runs* (1) *and* (2) *with shared internal randomness, Algorithm 1 produces identical policies, s.t.* $\boldsymbol{Pr}[\widehat{\pi}^{*(1)} \neq \widehat{\pi}^{*(2)}] \in O(\rho)$*. In every run, the produced policies* $\widehat{\pi}^*$ *achieve return at most* ε *less than the optimal policy* π^* *with all but probability* $O(\delta)$*.*

Proof Sketch. We give a sketch for the proof of the theorem here and refer the reader to a full proof in Appendix B.2. Assume that we can get replicable and accurate estimates of the value function expectations from our rSTAT procedure. One can show by induction that the algorithm consistently produces the same value functions in every iteration. Lemma 4.2 guarantees the convergence to an optimal policy. Finally, we can use union and Chernoff bounds to pick a sufficiently large sample for our rSTAT queries to be replicable and accurate and satisfy our assumption.

An interesting observation is that rPVI discretizes the space of values as a function of the ε-parameter and γ (see Appendix B.2). As a result, replicability becomes harder for larger values of γ as discretization intervals become smaller and we require more samples to obtain a equally sized ρ. This seems intuitive as we need to account for more potential future states that might impact our estimates.

The number of samples to compute a replicable value function is at most $O(|\mathcal{S}|^2|\mathcal{A}|^2/\rho)$ times larger than computing a non-replicable one [Kearns and Singh, 1998a]. Still, a key observations of the original PVI

Algorithm 1 Replicable Phased Value Iteration (rPVI)
Parameters: accuracy ε, failure probability δ, replicability failure probability ρ
Input: Generative Model $G_{\mathcal{M}}$
Output: ε-optimal policy $\widehat{\pi}^*$

```
Initialize $\widehat{Q}_0(s,a)$ to 0 for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
For all $s \in \mathcal{S}$, let $\phi_Q(s) := \max_a Q(s,a)$
for $t = 0, \cdots, T-1$ do
    $S \leftarrow (\mathbf{PS}(G_{\mathcal{M}}))^m$        ▷ do $m$ calls to $\mathbf{PS}(G_{\mathcal{M}})$ and store next-states in a map from
                                                  state-action pairs $(s,a)$ to next states $S[(s,a)]$.
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
        $\mathbb{E}_{s'\sim\widehat{P}(\cdot|s,a)}[\widehat{V}_t(s')] \leftarrow \mathsf{rSTAT}(S[(s,a)], \phi_{\widehat{Q}_t}(s'))$
        $\widehat{Q}_{t+1}(s,a) \leftarrow r(s,a) + \gamma\,\mathbb{E}_{s'\sim\widehat{P}(\cdot|s,a)}[\widehat{V}_t(s')]$
    end for
end for
return $\widehat{\pi}^* = \arg\max_a \widehat{Q}_T(s,a)$
```

Algorithm 2 Replicable Episodic R-max (RepRMAX)
Parameters: Accuracy ε, accuracy failure probability δ, replicability failure probability ρ, horizon H
Input: MDP $\mathcal{M}$, maximum reward $R_{\max}$
Output: ε-optimal policy $\pi_{\hat{\mathcal{M}}_K}$

```
Initialize $\pi_{\hat{\mathcal{M}}_K}$ to a random policy, counters for state-action-visitation $n(s,a)$ to 0
Initialize $K$, the set collecting known state-action pairs, to the empty set $\emptyset$
Initialize $S$, the set collecting trajectories to be used for estimating transition probabilities, to $\emptyset$
Initialize $\widehat{\mathcal{M}}_K$ as
    $\widehat{P}_K(s'|s,a) \coloneqq \mathbb{1}[s'=s]$ for all $(s,a,s')$
    $\widehat{r}_K(s,a) \coloneqq R_{\max}$ for all $(s,a)$
$i = 1$
while $\pi_{\hat{\mathcal{M}}_K}$ is not $\varepsilon$-optimal do
    Collect a sample of trajectories $S_i \leftarrow P(\tau)^m$ and add $S_i$ to $S$
    $K_i \leftarrow \mathsf{RepUpdateK}(S_i, K, \{n(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}})$, identify new known states
    For all $(s,a) \in K_i$, let $S[(s,a)]$ be the multiset of $s'$ visited from $(s,a)$ for all $\tau \in S$
    For all $s' \in \mathcal{S}$, let $\phi_{s'}(s) \coloneqq \mathbb{1}[s = s']$
    Update $\widehat{\mathcal{M}}_K$ for all $(s,a) \in K_i$:
        $\widehat{P}_K(s'|s,a) \coloneqq \mathsf{rSTAT}(S[(s,a)], \phi_{s'})$
        $\widehat{r}_K(s,a) \coloneqq r(s,a)$
    $K = K \cup K_i$
    Compute $\pi_{\hat{\mathcal{M}}_K}$ from $\widehat{\mathcal{M}}_K$
end while
return $\pi_{\hat{\mathcal{M}}_K}$
```

result was that it is sufficient for every state-action pair to have a sample size logarithmic in $|\mathcal{S}||\mathcal{A}|$ making the procedure cheaper than estimating the full transition dynamics of an MDP. In our approach, the cost of replicability is the loss of this property. However, we note that rPVI does not yield replicable transition probability estimation. Using the idea of rSTAT queries to obtain transitions estimates turns out to be significantly more expensive than the replicable value estimation done by Algorithm 1 (see Appendix B.2.1). Our results retain the notion that direct value estimation is much cheaper than estimating the full transition kernel even in the presence of replicability.

4.2 Replicable RL with Exploration

Next, we consider the setting of episodic exploration. We show that, despite the stochastic nature of exploration, it is possible to guarantee replicability while still outputting an ε-optimal policy.

We take the R-max algorithm of Brafman and Tennenholtz [2003] as the starting point for our replicable algorithm RepRMAX (Algorithm 2). It proceeds in rounds where the agent interacts with the environment for multiple episodes. The collection of trajectories encountered during exploration is used to incrementally build a model $\widehat{\mathcal{M}}$ of the underlying MDP $\mathcal{M}$. The algorithm implicitly partitions the set of state-action pairs $\mathcal{S} \times \mathcal{A}$ into two groups: known and unknown. All $(s,a) \in \mathcal{S} \times \mathcal{A}$ are initialized to be unknown. While a state is unknown, the model $\widehat{\mathcal{M}}$ maintains that (s,a) is a self-loop with probability 1, and that (s,a) has maximum reward, thereby promoting exploration of unknown states. After a state-action pair (s,a) has been visited sufficiently many times, it is added to the collection of known states K and its transition probabilities $\widehat{P}$ and reward $\widehat{r}$ are updated with an empirical approximation of $\widehat{P}_K(s' \mid s,a)$ for all $s' \in \mathcal{S}$ and the observed reward r, respectively. After every update, the policy $\pi_{\widehat{\mathcal{M}}_K}$ is computed as the optimal policy of the current model estimate.

While convergence of Algorithm 2 to an ε-optimal policy follows from familiar arguments [Brafman and Tennenholtz, 2003], proving replicability will require a great deal of additional care. To ensure that two runs of RepRMAX (with shared internal randomness) converge to the same policy with high probability, we will show something even stronger: we prove that two such runs will with high probability perform the same sequence of updates to their respective models $\widehat{\mathcal{M}}_K$ and policies $\pi_{\hat{\mathcal{M}}_K}$.

To enforce this property, we introduce a sub-routine in Algorithm 3 which replicably identifies state-action pairs that should be added to the collection of known states. Guaranteeing that at each iteration the set of known states K will be the same for two independent runs of the algorithm helps ensure that the models of the MDP $\widehat{\mathcal{M}}_K$, and consequently the policies $\pi_{\hat{\mathcal{M}}_K}$ learned at each iteration, will also be identical. To ensure replicability, we will want to avoid using a fixed threshold for the number of times a state-action pair (s, a) must be visited before it is considered "known". Under small deviations in realized transitions, a fixed threshold might lead to some (s, a) becoming known in one run of the algorithm and not another. Instead, we use a randomized threshold. In a call to Algorithm 3, the sample drawn at that round is used to estimate the expected number of visits to (s, a) in a single trajectory, for every (s, a). This estimate is added to the count $n(s, a)$ and a new threshold k' is sampled uniformly from $[k, k + w]$. If $n(s, a) \geq k'$, it is added to the set of known states K. We show in Theorem 4.2 that so long as the sample size m and the window w are taken large enough, the update to the set of known states at each round will be replicable.

Algorithm 3 RepUpdateK

Parameters: Accuracy failure probability δ, replicability failure probability ρ

Input: Sample of trajectories S_i, set of known states K, set of state-visit counts $\{n(s, a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$

Output: List of new known state-action pairs K_i

```
$K_i = \{(s, a) : (s, a)$ appears in $S_i\}$
$k' \leftarrow \mathcal{U}[k, k + w]$
for $(s, a) : (s, a) \notin K$ do
    $\widehat{c}_{s,a} = \frac{1}{|S_i|} \sum_{\tau\in S_i} \sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s, a)]$
    $n(s, a) = n(s, a) + \widehat{c}_{s,a}$
    if $n(s, a) < k'$ then
        $K_i = K_i \setminus \{(s, a)\}$
    end if
end for
return $K_i$
```

Now that we have understood the intricacies on an intuitive level, we will prove convergence (Lemma 4.2) and replicability (Theorem 4.2) of Algorithm 2.

Lemma 4.2 (Convergence)**.** *Consider $\mathcal{A}$ to be Algorithm 4. Let $\varepsilon \in (0, 1)$ be the accuracy parameter, $\rho \in (0, 1)$ the replicability parameter, and $\delta \in (0, 1)$, be the sample failure probability, with $\delta < \rho/2$. Suppose $m \in \tilde{O}\left(\frac{H^2|\mathcal{S}|^4|\mathcal{A}|^4 \log^2(1/\delta)\log(1/\rho)}{\rho^2\varepsilon^2}\right)$ is the number of trajectories per iteration, $k \in \tilde{O}\left(\frac{|\mathcal{S}|^6|\mathcal{A}|^2}{m(\varepsilon\rho)^2(1-\gamma)^4}\right)$ is the lowest expected visit count of a state-action pair before it is known. Let $w \in O(k)$ define the window $[k, k + w]$ for sampling the randomized threshold k'. With all but probability δ, $\mathcal{A}$ yields an ε-optimal policy in $T \in \tilde{O}(kH^2|\mathcal{S}||\mathcal{A}|\log(1/\delta))$ iterations.*

The proof that Algorithm 2 converges to an ε-optimal policy makes use of lemmas from Brafman and Tennenholtz [2003] and Kearns and Singh [1998b]. We will use a lemma showing that at each iteration, $\pi_{\hat{\mathcal{M}}_K}$ is already ε-optimal or there is a high probability that $n(s, a)$ increases for some $(s, a) \notin K$. We will also make use of the simulation lemma, which shows that if a model $\widehat{\mathcal{M}}_K$ is a good enough approximation of a model $\mathcal{M}$, then an optimal policy for $\widehat{\mathcal{M}}_K$ is an approximately optimal policy for $\mathcal{M}$. We refer the reader to those works for proof.

Lemma 4.3 (Kearns and Singh [1998b])**.** *Let* Explore(τ) *denote the event that $(s_h, a_h) = (s, a)$ for some $(s, a) \notin K$ and some $h \in [1, H]$. Then for any episode in which $\pi_{\hat{\mathcal{M}}_K}$ is not ε-optimal, it holds that*

$$\boldsymbol{Pr}_{\tau\sim P(\tau)}[\mathsf{Explore}(\tau)] \geq \varepsilon - \left(\tfrac{1}{1-\gamma}\right) \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \|P_K(s, a) - \widehat{P}_K(s, a)\|_1 \ .$$

Lemma 4.4 (Kearns and Singh [1998b]). *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two MDPs, differing only in their transition probabilities $P_1(\cdot|s,a)$ and $P_2(\cdot|s,a)$. Then for any policy π,*

$$|J_{\mathcal{M}_1}(\pi) - J_{\mathcal{M}_2}(\pi)| \leq \tfrac{R_{\max}}{2(1-\gamma)^2} \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \|P_1(s,a) - P_2(s,a)\|_1$$

With these lemmas in hand, we now proceed with the proof of Lemma 4.2.

Proof of Lemma 4.2. We use Lemma 4.3 to ensure that progress is made with probability at least $\varepsilon/2$ per episode, whenever $\pi_{\hat{\mathcal{M}}_K}$ is suboptimal. To ensure $|P_K(s'|s,a) - P_K(s'|s,a)| < \frac{\varepsilon(1-\gamma)^2}{|\mathcal{S}|}$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ and $s' \in \mathcal{S}$ with high probability, we must set parameters appropriately when estimating these quantities with replicable statistical queries. Taking $\rho_{SQ} \in O(\frac{\rho}{|\mathcal{S}|^2|\mathcal{A}|})$, $\tau_{SQ} \in O(\frac{\varepsilon(1-\gamma)^2}{|\mathcal{S}|})$, and $\delta_{SQ} \in O(\frac{\delta}{|\mathcal{S}|^2|\mathcal{A}|})$ to be the replicability, accuracy, and failure parameters respectively for the replicable statistical queries, a sample of size $O(\frac{|\mathcal{S}|^2\log(1/\delta_{SQ})}{(\varepsilon(\rho_{SQ}-2\delta_{SQ}))^2(1-\gamma)^4})$ is required by Theorem 2.1. Taking $k \in O(\frac{|\mathcal{S}|^2\log(1/\delta_{SQ})}{m(\varepsilon(\rho_{SQ}-2\delta_{SQ}))^2(1-\gamma)^4})$ and requiring that a state-action pair (s,a) be visited $O(km)$ times before being added to K suffices to guarantee all replicable statistical queries made by Algorithm 2 are $\frac{\varepsilon(1-\gamma)^2}{|\mathcal{S}|}$ accurate. It follows that at each iteration,

$$\mathbf{Pr}_{\tau\sim P(\tau)}[\mathsf{Explore}(\tau)] \in O(\varepsilon).$$

We sample m i.i.d. trajectories at each iteration and so, in expectation, at least $O(\varepsilon m)$ visits to unknown (s,a) occur in a round. Let $\pi_{\hat{\mathcal{M}}_{K,i}}$ denote the policy at the start of iteration i and observe that the sequence of random variables

$$X_i := \sum_{j=1}^{i}\left(\sum_{\tau\in S_j}\sum_{h=1}^{H}\mathbb{1}[(s_h,a_h)\notin K] - \mathop{\mathbb{E}}_{S}\left[\sum_{\tau\in S}\sum_{h=1}^{H}\mathbb{1}[(s_h,a_h)\notin K]\right]\right)$$

is a martingale with difference bounds $[-mH, mH]$. Azuma's inequality then gives us that

$$\begin{aligned}\mathbf{Pr}_S[X_T \leq -mkH^2|\mathcal{S}||\mathcal{A}|\log(1/\delta)] &\leq \exp(-\tfrac{2(mkH^2|\mathcal{S}||\mathcal{A}|\log(1/\delta))^2}{Tm^2H^2})\\ &\leq \exp(-O(k|\mathcal{S}||\mathcal{A}|\varepsilon\log(1/\delta)))\\ &= O(\delta).\end{aligned}$$

Therefore, except with probability $O(\delta)$, we can lower-bound the number of visits to unknown (s,a) over T iterations as follows.

$$\begin{aligned}\sum_{j=1}^{T}\sum_{\tau\in S_j}\sum_{h=1}^{H}\mathbb{1}[(s_h,a_h)\notin K] &\geq \sum_{j=1}^{T}\mathop{\mathbb{E}}_{S}\left[\sum_{\tau\in S}\sum_{h=1}^{H}\mathbb{1}[(s_h,a_h)\notin K]\right] - mkH^2|\mathcal{S}||\mathcal{A}|\log(1/\delta)\\ &\geq \varepsilon mT - mkH^2|\mathcal{S}||\mathcal{A}|\log(1/\delta)\\ &\in \Omega(mkH^2|\mathcal{S}||\mathcal{A}|).\end{aligned}$$

If all of these visits usefully contributed to the counts of unknown (s,a), we could immediately conclude that Algorithm 2 converges in T iterations, because each (s,a) only needs to be visited $O(mk)$ times to be added to K and there are $|\mathcal{S}||\mathcal{A}|$ many (s,a) to add. It is possible, however, that not every visit to an (s,a) that is unknown at the start of the iteration is useful in terms of making progress. It could be the case that only the first visit to some (s,a) in an iteration was required for (s,a) to be added to K, and so any subsequent visits are "wasted" in terms of making progress. We therefore consider two cases for each iteration: either some (s,a) is added to K or every visit to an unknown (s,a) is useful. When some (s,a) is added to K, in the worst case $mH-1$ of the total visits to unknown (s,a) can be wasted by repeated visits to (s,a) at that iteration, and so $mH|\mathcal{S}||\mathcal{A}|$ is an upper-bound on the number of unproductive visits to unknown (s, a). Of the remaining visits, at most $O(mk|\mathcal{S}||\mathcal{A}|)$ can contribute to making progress over the course of the algorithm before some (s, a) must become known. So after T iterations, we have

$$|K| \in \Omega(mkH^2|\mathcal{S}||\mathcal{A}|) - mH|\mathcal{S}||\mathcal{A}| - mk|\mathcal{S}||\mathcal{A}| \in \Omega(|\mathcal{S}||\mathcal{A}|)$$

and so all $|\mathcal{S}||\mathcal{A}|$ must be added to K after T iterations. Every $(s, a) \in K$ satisfies

$$\|P(\cdot|s, a) - \widehat{P}(\cdot|s, a)\|_1 \leq \varepsilon(1 - \gamma)^2$$

except with probability $O(\delta)$, and so $\pi_{\hat{\mathcal{M}}_K}$ is ε-optimal by Lemma 4.4. □

We now proceed to prove the main result of this section, showing that Algorithm 2 replicably converges to an ε-optimal policy in a number of iterations polynomial in all relevant parameters.

Theorem 4.2. *Let parameters be set as in Lemma 4.2. Then with all but probability δ, $\mathcal{A}$ converges to an ε-optimal policy in T iterations and samples mT trajectories, each of length H, drawing a total of*

$$O\left(\tfrac{|\mathcal{S}|^7|\mathcal{A}|^3H^3\log(1/\delta)}{\rho^2\varepsilon^3(1-\gamma)^4}\right)$$

samples. Further, let S_1 and S_2 be two trajectory sets, independently sampled over two runs of $\mathcal{A}$ with shared internal randomness, and let $\pi^{(1)}_{\hat{\mathcal{M}}_K}(a|s) \leftarrow \mathcal{A}(S_1; r)$ and $\pi^{(2)}_{\hat{\mathcal{M}}_K}(a|s) \leftarrow \mathcal{A}(S_2; r)$. Then

$$\boldsymbol{Pr}_{S_1,S_2,r}\left[\pi^{(1)}_{\hat{\mathcal{M}}_K}(a|s) \neq \pi^{(2)}_{\hat{\mathcal{M}}_K}(a|s)\right] \in O(\rho).$$

Proof. Lemma 4.2 gives us that, for our settings of k and T, Algorithm 2 converges to an ε-optimal policy in T iterations, except with probability δ. The sample complexity follows immediately from the bound on T and the setting of m, so it remains to analyze replicability. Our analysis will make use of some additional shorthand. We use $\rho_K \in O(\rho/(T|\mathcal{S}||\mathcal{A}|))$ to denote the replicability parameter for the decision to add a single (s, a) to K, in a single call to Algorithm 3. We similarly use $\rho_{SQ} \in O(\rho/(|\mathcal{S}|^2|\mathcal{A}|))$, $\tau_{SQ} \in O(\varepsilon(1-\gamma)^2/|\mathcal{S}|)$, and $\delta_{SQ} \in O(\delta/(|\mathcal{S}|^2|\mathcal{A}|))$ to denote the replicability, accuracy, and failure parameters for the rSTAT queries made during the updates to $P(s'|s, a)$. We will use $t \in O(w\rho_K)$ to denote a high probability bound on the difference between the empirical estimates for the expected visits to a given (s, a) in a trajectory across two runs of Algorithm 3, i.e. $|\widehat{c}^{(1)}_{s,a} - \widehat{c}^{(2)}_{s,a}| \in O(t)$. We are now ready to prove the following stronger claim:

Claim 4.1. *If two runs of Algorithm 2 begin iteration i with*

$$\widehat{\mathcal{M}}^{(1)}_K = \widehat{\mathcal{M}}^{(2)}_K,\ \pi^{(1)}_{\hat{\mathcal{M}}_K} = \pi^{(2)}_{\hat{\mathcal{M}}_K},\ \text{and } |n(s, a)^{(1)} - n(s, a)^{(2)}| \in O(it)\ \forall (s, a),$$

then at the end of iteration i, it holds that

$$\widehat{\mathcal{M}}^{(1)}_K = \widehat{\mathcal{M}}^{(2)}_K,\ \pi^{(1)}_{\hat{\mathcal{M}}_K} = \pi^{(2)}_{\hat{\mathcal{M}}_K},\ \text{and } |n(s, a)^{(1)} - n(s, a)^{(2)}| \in O(it + t)\ \forall (s, a),$$

except with probability $O(\rho_K|\mathcal{S}||\mathcal{A}| + \rho_{SQ}|K_1||\mathcal{S}|)$.

We take the initialization of Algorithm 2 as the base case for our inductive proof. Before the first iteration, $\pi_{\hat{\mathcal{M}}_K}$ is initialized randomly and shared internal randomness yields $\pi^{(1)}_{\hat{\mathcal{M}}_K} = \pi^{(2)}_{\hat{\mathcal{M}}_K}$. We deterministically initialize $\widehat{\mathcal{M}}_K$ and all $n(s, a)$, and so $\widehat{\mathcal{M}}^{(1)}_K = \widehat{\mathcal{M}}^{(2)}_K$ and $n(s, a)^{(1)} = n(s, a)^{(2)}$.

Next, we prove the inductive step. Observe that $\widehat{\mathcal{M}}^{(1)}_K = \widehat{\mathcal{M}}^{(2)}_K$ at the end of the iteration unless at least one of the following two events occurs:

1. $K^{(1)}_i \neq K^{(2)}_i$ - the set of new known (s, a) pairs differs across the two runs.

2. The updates to $\widehat{P}_K(s'|s, a)$ and $r(s, a)$ differ for at least one (s, a).

The first event occurs exactly when k' falls in between $n(s,a)^{(1)} + \widehat{c}_{s,a}^{(1)}$ and $n(s,a)^{(2)} + \widehat{c}_{s,a}^{(2)}$, and so we wish to bound $|n(s,a)^{(1)} + \widehat{c}_{s,a}^{(1)} - n(s,a)^{(2)} - \widehat{c}_{s,a}^{(1)}|$ with high probability. Our inductive hypothesis gives us that $|n(s,a)^{(1)} - n(s,a)^{(2)}| \in O(it)$, so it suffices to show that $\widehat{c}_{s,a}^{(1)} - \widehat{c}_{s,a}^{(2)} \in O(t)$ except with probability $O(\rho_K)$. To obtain high probability bounds on $|\widehat{c}_{s,a}^{(1)} - \widehat{c}_{s,a}^{(2)}|$, we will rely on our assumption that at the start of the iteration, $\pi_{\hat{\mathcal{M}}_K}^{(1)} = \pi_{\hat{\mathcal{M}}_K}^{(2)}$. It follows that, for every state-action pair (s,a), the expected number of visits to (s,a) in a single episode is the same for both iterations. That is, for every (s,a), letting

$$c_{s,a} := \mathop{\mathbb{E}}_{\tau \sim P(\tau)} \left[\sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s,a)] \right]$$

we have $c_{s,a}^{(1)} = c_{s,a}^{(2)}$.

For a particular (s,a), Chernoff bounds applied to the average observed counts $\widehat{c}_{s,a}^{(1)}$ and $\widehat{c}_{s,a}^{(2)}$ show that they must both be close to their (shared) expectation with high probability. We draw a sample of $m \in O(\frac{H^2 \log(1/\rho_K)}{t^2})$ trajectories, ensuring that, except with probability $4\exp\left(\frac{-2t^2m^2}{H^2m}\right) \in O(\rho_K)$,

$$\begin{aligned} |\widehat{c}_{s,a}^{(1)} - \widehat{c}_{s,a}^{(2)}| &= \frac{1}{m} \left| \sum_{\tau \in S_1^{(1)}} \sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s,a)] - \sum_{\tau \in S_1^{(2)}} \sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s,a)] \right| \\ &\le \left| \tfrac{1}{m} \sum_{\tau \in S_1^{(1)}} \sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s,a)] - c_{s,a}^{(1)} \right| + \left| \tfrac{1}{m} \sum_{\tau \in S_1^{(2)}} \sum_{h=1}^{H} \mathbb{1}[(s_h, a_h) = (s,a)] - c_{s,a}^{(1)} \right| \in O(t). \end{aligned}$$

Thus, for any (s,a), we bound the probability that $(s,a) \in K_i^{(1)} \triangle K_i^{(2)}$ by the probability that $|\widehat{c}_{s,a}^{(1)} - \widehat{c}_{s,a}^{(2)}| \notin O(t)$, or k' falls between $n(s,a)^{(1)} + \widehat{c}_{s,a}^{(1)}$ and $n(s,a)^{(2)} + \widehat{c}_{s,a}^{(2)}$. Then, for all (s,a),

$$\mathbf{Pr}_{k', S_1, S_2}[(s,a) \in K_i^{(1)} \triangle K_i^{(2)}] \in O(\rho_K + t/w).$$

We took $w \in O(t/\rho_K)$, so by union bounding over all of $\mathcal{S} \times \mathcal{A}$ the probability of the first event is at most $O(|\mathcal{S}||\mathcal{A}|\rho_K)$.

To bound the probability of the second event conditioned on the first event not occurring, it suffices to bound the probability that the updates to $\widehat{P}_K(s'|s,a)$ for $(s,a) \in K_i$ differ across both runs, as rewards are assumed to be deterministic. By the conditioning, we have $K_1^{(1)} = K_1^{(2)}$, and so it suffices to show that each call to rSTAT returns the same value for both runs. Taking ρ_{SQ}, τ_{SQ}, and δ_{SQ} as the replicability, tolerance, and failure parameters respectively in Theorem 2.1 gives that a sample of size

$$s \in O\left(\frac{|\mathcal{S}|^2 \log(1/\delta_{SQ})}{(\varepsilon(\rho_{SQ} - 2\delta_{SQ}))^2 (1-\gamma)^4} \right)$$

is sufficient. Each (s,a) is added to K_i only if it was visited at least km times. We have taken

$$k \in O\left(\frac{|\mathcal{S}|^2 \log(1/\delta_{SQ})}{m(\varepsilon(\rho_{SQ} - 2\delta_{SQ}))^2 (1-\gamma)^4} \right),$$

therefore $S[(s,a)]$ comprises at least s i.i.d. samples from $P(\cdot \mid s,a)$, as desired. Union bounding over the $|K_i||\mathcal{S}|$ queries in the ith iteration gives a bound of $|K_i||\mathcal{S}|\rho_{SQ}$ on the probability of the second event, conditioned on the first event not happening.

We now assemble our inductive argument into a proof of the theorem. At the start of iteration i, the inductive hypothesis holds except with probability

$$\sum_{j=1}^{i-1} \rho_K |\mathcal{S}||\mathcal{A}| + \rho_{SQ}|K_j||\mathcal{S}|.$$

Noting that $\sum_{j=1}^{T} |K_j| \leq |\mathcal{S}||\mathcal{A}|$, and recalling that we have taken replicability parameters $\rho_K \in O(\rho/(T|\mathcal{S}||\mathcal{A}|))$ and $\rho_{SQ} \in O(\rho/(|\mathcal{S}|^2|\mathcal{A}|))$, ensures we achieve a replicability parameter ρ after the T iterations of Algorithm 2. □

4.3 Limitations

As mentioned previously, our bounds lose some of the properties that standard RL results provide, such as the ability to estimate value functions with only a logarithmic dependence on relevant parameters. We expect that some of the sample complexity overhead from achieving replicability is inevitable, as seen in the statistical query lower-bound of Impagliazzo et al. [2022]. Nonetheless, we hope that future work can improve on the sample-complexities of our algorithms.

Our work is in part motivated by the recent replicability concerns in deep RL [Islam et al., 2017, Henderson et al., 2018]. However, establishing formal guarantees in these highly complicated settings is often not easy. As such, our algorithms suffer the weakness that many theoretical results in RL have to deal with, namely their lack of immediate applicability to real-world problems. Yet, our empirical evaluation in section 5 will show that there is hope for practical application.

5 Experiments

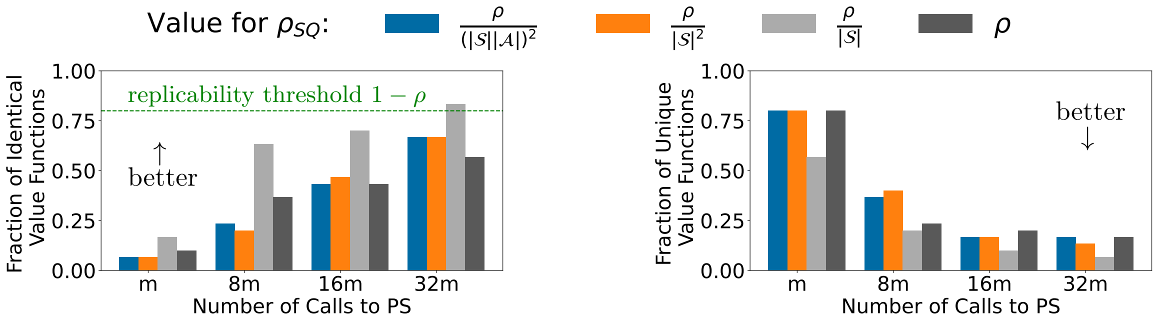
While our theoretical bounds have sample complexity overhead from the introduction of replicability, we would like to analyze the actual requirements in practice. We introduce a simple MDP in Figure 1 that contains several ways of reaching the two goals. We analyze the impact of the number of calls to $\mathbf{PS}(G_{\mathcal{M}})$ on replicability for rPVI. In theory, our dependence on the number of calls is not logarithmic with respect to $|\mathcal{S}||\mathcal{A}|$ but we would like to see if can draw a sample that is considerably smaller, maybe even on the order of the logarithmic requirement. We choose accuracy $\varepsilon = 0.02$, failure rate $\delta = 0.001$ and replicability $\rho = 0.2$. The number of calls that would be required by standard PVI is at most $m \approx 13000$ (ignoring γ factors). We take several multiples of m and measure the fraction of identical and unique value functions, treating the rSTAT ρ_{SQ} as a hyperparameter.

The results are presented Figure 2, revealing that the number of samples needed to replicably produce the same value function can be several orders of magnitude lower than suggested by our bounds. We also find that a significantly larger ρ_{SQ} for every statistical query call is feasible, which also allows for fewer samples. The algorithm quickly produces a small set of value functions that may not be identical but, with a little more data, minor differences are removed. Note that using a replicable procedure naturally incurs overhead, which is expected. However, the overhead is significantly better than the theoretically required sample-size with squared $|\mathcal{S}||\mathcal{A}|$ dependence. Replicability can be achieved with fewer samples, which should allow us to scale to more complex problems in the future.

6 Related Work

Our work builds upon the foundational ideas by Impagliazzo et al. [2022], who introduce formal notions of replicability that are strongly related to robustness, privacy, and generalization [Bun et al., 2023, Kalavasis et al., 2023]. Building on these formal definitions of replicability, researchers have provided algorithms for replicable bandits [Esfandiari et al., 2023a] and replicable clustering [Esfandiari et al., 2023b]. Ahn et al. [2022] introduce algorithms for convex optimization using a slightly different notion of replicability. Our paper presents the first results for formally replicable algorithms in a control setting.

From an RL perspective, our work is strongly related to understanding exploration in MDPs [Kearns and Singh, 1998b, Brafman and Tennenholtz, 2003, Kakade, 2003]. In the finite-horizon episodic setting, researchers made progress on upper bounds for exploration Auer and Ortner [2006], Auer et al. [2008], Jaksch et al. [2010] that ultimately led to the development of a near-complete understanding of the problem [Azar et al., 2017, Zanette and Brunskill, 2019, Simchowitz and Jamieson, 2019]. Lower bounds are provided in other



The largest percentage of identical value functions across 30 runs. The quantity increases with more data and drastically increases with correctly picked ρ_{SQ}.

The percentage of unique value functions across 30 runs. Varying ρ_{SQ} has negligible impact, while more samples reduces it quickly.

Figure 2: The rPVI algorithm evaluated on varying numbers of calls to $\mathbf{PS}(G_{\mathcal{M}})$, with several values for the internal rSTAT parameter ρ_{SQ}. Results are provided across 30 runs with different random sampling seeds. The number of calls is set to constant factor multiples of $m = 13000$. The dotted green line denotes the replicability threshold of $1 - \rho$. The results show that, in practice, the number of samples needed for replicability can be orders of magnitude lower than our bounds suggest.

works [Dann and Brunskill, 2015, Osband and Roy, 2016]. Further, Jin et al. [2020], Kaufmann et al. [2021] provide results on a reward-free framework that allows for the optimization of any reward function. While a good amount of progress has been made on understanding the base problem, the notion of replicability is not considered in any of them.

Given the connections of replicability and robustness, our work is related but orthogonal to that of the study of worst-case optimal policies and value functions. These worst-case results are often obtained via the study of robust Markov decision processes, first introduced by Nilim and Ghaoui [2005], Iyengar [2005]. One line of work here has focused on relaxation of assumptions and combatting conservativeness in robust MDPs [Wiesemann et al., 2013, Mannor et al., 2016, Petrik and Russel, 2019, Panaganti and Kalathil, 2022]. Others have focused on various new formulations such as distributional robustness [Xu and Mannor, 2010, Yu and Xu, 2016]. However, all of the above work focuses on understanding worst-cases and finding policies that do not have to be replicable.

Finally, our work is related to efforts in practical RL to ensure replicability, such as benchmark design [Guss et al., 2021, Mendez et al., 2022] and robust implementation [Nagarajan et al., 2018, Seno and Imai, 2022] and evaluation [Lynnerup et al., 2020, Jordan et al., 2020, Agarwal et al., 2021].

7 Conclusion & Future Work

We introduced the notion of formal replicability to the field of RL and established various novel algorithms for replicable RL. While these first results might have sub-optimal sample complexities, they highlight the crucial fact that replicability in RL is hard and requires study of the various aspects that impact it. We hope that future work can alleviate some of these efficiency challenges. A general open question is if replicable RL might simply be harder by nature than standard RL? This question needs to be posed on various levels because, as we argue in Section 3, finding a replicable policy might be easier than requiring the value function to be replicable. Finally, we believe the development of replicable algorithms for other settings such as the non-episodic setting as well as practical application are of great importance.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320. Curran Associates, Inc., 2021.

Kwangjun Ahn, Prateek Jain, Ziwei Ji, Satyen Kale, Praneeth Netrapalli, and Gil I. Shamir. Reproducibility in optimization: Theoretical framework and limits. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Peter Auer and Ronald Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 263–272. PMLR, 06–11 Aug 2017.

C. Glenn Begley and Lee M Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, March 2012.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2 (Mar):499–526, 2002.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3:213–231, mar 2003.

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 520–527. ACM, 2023.

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2818–2826, Cambridge, MA, USA, 2015. MIT Press.

Hossein Esfandiari, Alkis Kalavasis, Amin Karbasi, Andreas Krause, Vahab Mirrokni, and Grigoris Velegkas. Replicable bandits. In *The Eleventh International Conference on Learning Representations*, 2023a.

Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velegkas, and Felix Zhou. Replicable clustering, 2023b.

Amir-massoud Farahmand. Action-gap phenomenon in reinforcement learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

William Hebgen Guss, Stephanie Milani, Nicholay Topin, Brandon Houghton, Sharada Mohanty, Andrew Melnik, Augustin Harter, Benoit Buschmaas, Bjarne Jaster, Christoph Berganski, Dennis Heitkamp, Marko Henning, Helge Ritter, Chengjie Wu, Xiaotian Hao, Yiming Lu, Hangyu Mao, Yihuan Mao, Chao Wang, Michal Opanowicz, Anssi Kanervisto, Yanick Schraner, Christian Scheller, Xiren Zhou, Lu Liu, Daichi

Nishio, Toi Tsuneda, Karolis Ramanauskas, and Gabija Juceviciute. Towards robust and domain agnostic reinforcement learning competitions: Minerl 2020. In Hugo Jair Escalante and Katja Hofmann, editors, *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 233–252. PMLR, 06–12 Dec 2021.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 818–831, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. doi: 10.1145/3519935.3519973.

Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. In *Reproducibility in Machine Learning Workshop (ICML)*, 2017.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4870–4879. PMLR, 13–18 Jul 2020.

Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip Thomas. Evaluating the performance of reinforcement learning algorithms. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4962–4973. PMLR, 13–18 Jul 2020.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning.* Phd thesis, 2003.

Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms, 2023.

Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 865–891. PMLR, 16–19 Mar 2021.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, Nov 1998.

Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in Neural Information Processing Systems*, 11, 1998a.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 1998b.

IS Khalil, JC Doyle, and K Glover. *Robust and optimal control.* Prentice hall, 1996.

Nicolai A. Lynnerup, Laura Nolling, Rasmus Hasle, and John Hallam. A survey on reproducibility by evaluating deep reinforcement learning algorithms on real-world robots. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 466–489. PMLR, 30 Oct–01 Nov 2020.

Shie Mannor, Duncan Simester, Peng Sun, and John N. Tsitsiklis. Bias and variance in value function estimation. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 72, New York, NY, USA, 2004. Association for Computing Machinery.

Shie Mannor, Ofir Mebel, and Huan Xu. Robust mdps with k-rectangular uncertainty. *Mathematics of Operations Research*, 41(4):1484–1509, 2016.

Jorge A. Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. Composuite: A compositional reinforcement learning benchmark. In *1st Conference on Lifelong Learning Agents*, 2022.

Prabhat Nagarajan, Garrett Warnell, and Peter Stone. Deterministic implementations for reproducibility in deep reinforcement learning. In *2nd Reproducibility in Machine Learning Workshop at ICML 2018*, Stockholm, Sweden, July 2018.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning, 2016.

Kishan Panaganti and Dileep Kalathil. Sample complexity of robust reinforcement learning with a generative model. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 9582–9602. PMLR, 28–30 Mar 2022.

Marek Petrik and Reazul Hasan Russel. Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Lariviere, Alina Beygelzimer, Florence d'Alche Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program). *Journal of Machine Learning Research*, 22(164): 1–20, 2021.

Takuma Seno and Michita Imai. D3rlpy: An offline deep reinforcement learning library. 23(1), 2022.

Max Simchowitz and Kevin G Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

V. Stodden, F. Leisch, and R.D. Peng. *Implementing Reproducible Research*. Chapman & Hall/CRC The R Series. Taylor & Francis, 2014. ISBN 9781466561595.

Kiri L. Wagstaff. Machine learning that matters. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, page 1851–1856, Madison, WI, USA, 2012. Omnipress.

Chelsea C. White and Hany K. Eldeib. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, 1994.

Wolfram Wiesemann, Daniel Kuhn, and Breç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Huan Xu and Shie Mannor. Distributionally robust markov decision processes. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Pengqian Yu and Huan Xu. Distributionally robust counterpart in markov decision processes. *IEEE Transactions on Automatic Control*, 61(9):2538–2543, 2016.

Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7304–7312. PMLR, 09–15 Jun 2019.

A Further Definitions

Definition A.1 (Replicable value function estimation)**.** *Let $\mathcal{A}$ be a policy estimation algorithm that outputs an estimated value function $\widehat{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, from which a policy may be computed, and where $\widehat{Q}$ is computed from a set of trajectories S sampled from an MDP. Algorithm $\mathcal{A}$ is ρ-replicable for value function estimation if, given independently sampled trajectory sets S_1 and S_2, and letting $\widehat{Q}^{*(1)}(s,a), \leftarrow \mathcal{A}(S_1; r)$ and $\widehat{Q}^{*(2)}(s,a) \leftarrow \mathcal{A}(S_2; r)$,*

$$\boldsymbol{Pr}_{S_1,S_2,r}[\widehat{Q}^{*(1)}(s,a) \neq \widehat{Q}^{*(2)}(s,a)] \leq \rho,$$

where r represents the internal randomness of $\mathcal{A}$. Trajectory sets S_1 and S_2 may potentially be gathered from the environment during the execution of an RL algorithm.

Definition A.2 (Replicable MDP estimation)**.** *Let $\mathcal{A}$ be a policy estimation algorithm that outputs a model of an MDP $\widehat{\mathcal{M}}$, from which a policy may be computed, and where $\widehat{\mathcal{M}}$ is computed from a set of trajectories S, sampled from an MDP. Algorithm $\mathcal{A}$ is ρ-replicable for MDP estimation if, given independently sampled trajectory sets S_1 and S_2, and letting $\widehat{\mathcal{M}}^{*(1)} \leftarrow \mathcal{A}(S_1; r)$ and $\widehat{\mathcal{M}}^{*(2)} \leftarrow \mathcal{A}(S_2; r)$, it holds that*

$$\boldsymbol{Pr}_{S_1,S_2,r}[\widehat{\mathcal{M}}^{*(1)} \neq \widehat{\mathcal{M}}^{*(2)}] \leq \rho,$$

where r represents the internal randomness of $\mathcal{A}$. Trajectory sets S_1 and S_2 may potentially be gathered from the environment during the execution of an RL algorithm.

B Proofs

B.1 rPVI Convergence for Lemma 4.1

Proof. We want to prove that after T iterations of Phased Q-learning, it holds that

$$\|\widehat{Q}_T(s,a) - Q^*(s,a)\|_\infty \leq \varepsilon.$$

We can decompose this into two steps by bounding the error introduced from sampling and the error introduced via only running for t iterations using the triangle inequality.

$$\|\widehat{Q}_T(s,a) - Q^*(s,a)\|_\infty \leq \|\widehat{Q}_T(s,a) - Q_T(s,a)\|_\infty + \|Q_T(s,a) - Q^*(s,a)\|_\infty$$

Note that as long as we choose the number of samples to be sufficiently large, our statistical queries will give us accuracy guarantees because for every call to $\mathbf{PS}(G_{\mathcal{M}})$ we get a sample for every state-action pair. These samples are i.i.d. and across state-action pairs they are independent. So, suppose that the following expectations can be estimated accurately

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}, 0 \leq t \leq T, \; \mathbf{Pr}\left(\left| \mathop{\mathbb{E}}_{s' \sim \widehat{P}} \left[\hat{V}_T(s') \right] - \mathop{\mathbb{E}}_{s' \sim P} \left[\hat{V}_t(s') \right] \right| \geq \alpha \right) \leq 2e^{-2m\alpha^2}.$$

Now, to bound the first term, we can derive a recurrence relation as follows.

$$\begin{aligned}
\|\widehat{Q}_{t+1}(s,a) - Q_{t+1}(s,a)\|_\infty &= \max_{(s,a)} |\widehat{Q}_{t+1}(s,a) - Q_{t+1}(s,a)| \\
&= \max_{(s,a)} \left| r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim\widehat{P}}[\hat{V}_t(s')] - r(s,a) - \gamma \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \right| \\
&= \max_{(s,a)} \left| \gamma \mathop{\mathbb{E}}_{s'\sim\widehat{P}}[\hat{V}_t(s')] - \gamma \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \right| \\
&= \gamma \left| \mathop{\mathbb{E}}_{s'\sim\widehat{P}}[\hat{V}_t(s')] - \mathop{\mathbb{E}}_{s'\sim P}[\hat{V}_t(s')] + \mathop{\mathbb{E}}_{s'\sim P}[\hat{V}_t(s')] - \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \right| \\
&\leq \gamma \left| \mathop{\mathbb{E}}_{s'\sim\widehat{P}}[\hat{V}_t(s')] - \mathop{\mathbb{E}}_{s'\sim P}[\hat{V}_t(s')] \right| + \left| \mathop{\mathbb{E}}_{s'\sim P}[\hat{V}_t(s')] - \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \right| \\
&\leq \gamma\tau + \gamma \left| \mathop{\mathbb{E}}_{s'\sim P}[\hat{V}_t(s')] - \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \right| \\
&\leq \gamma\tau + \gamma \max_{s} \left| \hat{V}_t(s) - V_t(s) \right| \\
&\leq \gamma\alpha + \gamma \max_{(s,a)} \left| \hat{Q}_t(s,a) - Q_t(s,a) \right| \\
&\leq \gamma\alpha + \gamma \|\hat{Q}_t(s,a) - Q_t(s,a)\|_\infty
\end{aligned}$$

At $t = 0$, it holds that $\hat{Q}_0 = Q_0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$. As a result, the previous result forms a geometric series and for any t

$$\|\hat{Q}_t(s,a) - Q_t(s,a)\|_\infty \leq \tau \frac{\gamma}{1-\gamma}.$$

We upper bound the second term in the triangle inequality as

$$\begin{aligned}
\|Q_t(s,a) - Q^*(s,a)\|_\infty &= \max_{(s,a)} |Q_t(s,a) - Q^*(s,a)| \\
&= \max_{(s,a)} |\mathcal{T}^t Q_0(s,a) - \mathcal{T}^t Q^*(s,a)| \\
&\leq \gamma^t \max_{(s,a)} |Q_0(s,a) - Q^*(s,a)| \\
&= \gamma^t \max_{(s,a)} |Q^*(s,a)| \\
&\leq \frac{\gamma^t}{1-\gamma}
\end{aligned}$$

As a result, we obtain that

$$\begin{aligned}
\|\hat{Q}_t(s,a) - Q^*(s,a)\|_\infty &\leq \tau \frac{\gamma}{1-\gamma} + \frac{\gamma^t}{1-\gamma} \\
&= \tau \frac{\gamma}{1-\gamma} + \frac{1-(1-\gamma^t)}{1-\gamma} \\
&\leq \tau \frac{\gamma}{1-\gamma} + \frac{e^{-(1-\gamma)t}}{1-\gamma}
\end{aligned}$$

Now, all we need to do is choose τ and t accordingly. If we choose $T = t \geq \log\left(\frac{2}{(1-\gamma)^2\varepsilon}\right)/(1-\gamma)$ and we pick $\tau = (1-\gamma)\frac{\varepsilon}{2}$ we obtain

$$\|\hat{Q}_t(s,a) - Q^*(s,a)\|_\infty \leq \gamma\frac{\varepsilon}{2} + \frac{\varepsilon}{2}(1-\gamma) = \frac{\varepsilon}{2}.$$

□

B.2 Proof of Theorem 4.1

Proof. We must show that the algorithm is reproducible and that the accuracy constraints are not violated. Suppose that m is sufficiently large to guarantee replicable as well as sufficiently accurate estimates. We show by induction that this yields replicability across two runs. Then we use a standard contraction argument to ensure policy convergence.

First, fix some MDP $\mathcal{M}$ and consider two independent runs of the Replicable Phased Value Iteration algorithm with shared internal randomness r. Let $S^{(i)}$ denote the set of trajectories drawn and $V^{(i)}$ the value function in the ith run. Suppose that m is sufficiently large such that our statistical query estimate yields reproducible values estimates such that for all $s \in \mathcal{S}$, $t \in T$, it holds that $\mathbb{E}[\widehat{V}_t^{(1)}(s')] = \mathbb{E}[\widehat{V}_t^{(2)}(s')]$. We show via induction on t that the Q-function is exactly the same across both runs at every step of Replicable Phased Value Iteration. Let $\widehat{Q}_t^{(1)}$ and $\widehat{Q}_t^{(2)}$ be the two Q-functions of the first and second run at iteration t respectively.

Base Case: In the base case at $t = 0$, by choice of our intialization for the Q-functions, it holds that $\widehat{Q}_0^{(1)} = \widehat{Q}_0^{(2)} = \vec{0}$ which is always replicable.

Inductive step: Suppose that $\widehat{Q}_t^{(1)} = \widehat{Q}_t^{(2)}$. After one more iteration of value updates,

$$\begin{aligned}\widehat{Q}_{t+1}^{(1)}(s,a) \leftarrow r(s,a) + \mathbb{E}[\widehat{V}_t^{(1)}(s')] \quad &\wedge \quad \widehat{Q}_{t+1}^{(2)}(s,a) \leftarrow r(s,a) + \mathbb{E}[\widehat{V}_t^{(1)}(s')] \\ \implies \widehat{Q}_{t+1}^{(1)} = \widehat{Q}_{t+1}^{(2)} \ ,&\end{aligned}$$

where we used the fact that rewards are deterministic and $\mathbb{E}[\widehat{V}_t^{(1)}(s')] = \mathbb{E}[\widehat{V}_t^{(2)}(s')]$ is computed to be exactly the same by assumption.

Finally, since $\widehat{Q}_t^{(1)} = \widehat{Q}_t^{(2)}$ it also holds for all states $s \in \mathcal{S}$ that $\max_a \widehat{Q}_t^{(1)}(s,a) = \max_a \widehat{Q}_t^{(2)}(s,a)$. The procedure maintains the exact same Q-function across two runs which yield the same policy.

To show convergence to an ε-optimal policy, we can use a standard contraction argument provided in lemma 4.1. If our value estimates are not too far off from their expectation which can be ensured via sufficiently large sample size for the statistical query procedure.

It remains to show that our sample size is sufficiently large to ensure both replicability as well as accuracy. For this we are interested in the following two quantities $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, t \in [0,T]$,

$$\mathbf{Pr}\left[\mathop{\mathbb{E}}_{s\sim\widehat{P}}\left[\widehat{V}_t(s)\right] - \mathop{\mathbb{E}}_{s\sim P}\left[\widehat{V}_t(s)\right] > \tau\right] \leq \delta_{SQ} \qquad \mathbf{Pr}\left[\left[\mathop{\mathbb{E}}_{s\sim\widehat{P}}[\widehat{V}_t^{(1)}(s)\right] \neq \mathop{\mathbb{E}}_{s\sim\widehat{P}}\left[\widehat{V}_t^{(2)}(s)\right]\right] \leq \rho_{SQ} \ .$$

To ensure the first probability holds, we require that our statistical queries return sufficiently accurate estimates. For this we take a closer look at how the replicable statistical queries give us this guarantee. In the replicable statistical query procedure, the error is split into a sample approximation error and the error from discretization

$$\tau = \frac{\tau(\rho_{SQ} - 2\delta_{SQ})}{\rho_{SQ} + 1 - 2\delta_{SQ}} + \frac{\tau}{\rho_{SQ} + 1 - 2\delta_{SQ}} = \tau' + \frac{\alpha}{2} \ .$$

By union bound and Chernoff inequality we have that

$$\mathbf{Pr}\left[\bigcup_{(s,a),t}\left(\mathop{\mathbb{E}}_{s\sim\widehat{P}}\left[\widehat{V}_t(s)\right]-\mathop{\mathbb{E}}_{s\sim P}\left[\widehat{V}_t(s)\right]>\tau'\right)\right]\leq|\mathcal{S}||\mathcal{A}|Te^{-2m\tau'^2}\leq\delta$$

$$\implies m\geq\frac{1}{2\tau'^2}\log\left(\frac{2|\mathcal{S}||\mathcal{A}|T}{\delta}\right)\ .$$

As long as we pick m at least this large, our value estimates will be accurate. Finally, we are interested in the probability that in two separate runs, rSTAT fails to output the same estimate for one expected value computation. Conditioning on accurate estimation in each run, the probability that two estimates fall into different regions in the rSTAT procedure is given by $2\tau'/\alpha$. Again via union bound, we have that

$$\begin{aligned}\mathbf{Pr}\left[\bigcup_{(s,a),t}\left(\left[\mathop{\mathbb{E}}_{s\sim\widehat{P}}[\widehat{V}_t^{(1)}(s)\right]\neq\mathop{\mathbb{E}}_{s\sim\widehat{P}}\left[\widehat{V}_t^{(2)}(s)\right]\right)\right]&\leq|S||\mathcal{A}|T(2\tau'/\alpha)\\&=|S||A|T\rho_{SQ}-2\delta\ .\end{aligned}$$

As long as we pick $\rho_{SQ}=\rho/|S||A|T$, we are guaranteed with probabability ρ that all estimates will be reproducible. Plugging this back into our sample complexity, we obtain

$$\begin{aligned}\frac{(\rho_{SQ}+1-2\delta_{SQ})^2}{2\tau^2(\rho_{SQ}-2\delta_{SQ})^2}\log\left(\frac{2|\mathcal{S}||\mathcal{A}|T}{\delta}\right)&\leq\frac{4}{2\tau^2(\rho_{SQ}-2\delta_{SQ})^2}\log\left(\frac{2|\mathcal{S}||\mathcal{A}|T}{\delta}\right)\\&=\frac{2(|S||A|T)^2}{\tau^2(\rho-2\delta)^2}\log\left(\frac{2|\mathcal{S}||\mathcal{A}|T}{\delta}\right)\leq m\ .\end{aligned}$$

Setting τ and T according to the convergence criteria in Appendix B.1 concludes the proof. □

B.2.1 Replicable Approximate MDPs

Note that the transition model built in standard PVI is very sparse and so are the transitions that are implicitly used in every statisical query of our algorithm. The number of samples that are used to estimate transition probabilities of a single state are of size $\tilde{O}(\log(|\mathcal{S}||\mathcal{A}|))$ while the vector that represents the full probability vector is of size $|\mathcal{S}|$. This open up the question whether we would be able to reproducibly approximate the full model of the MDP rather than just obtaining estimates of values. We show that is in fact possible to obtain an exactly reproducible MDP in algorithm 4.

Algorithm 4 Replicable ApproximateMDP
Parameters: accuracy ϵ, failure probability δ, replicability failure probability ρ
Input: Generative Model $G_{\mathcal{M}}$
Output:

```
For all $s \in \mathcal{S}$, let $\phi_s(s') \coloneqq \mathbb{1}[s = s']$
for $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ do
    $S \leftarrow (G_{\mathcal{M}}(s, a))^m$                ▷ do $m$ calls to $G_{\mathcal{M}}$ and store next states in a set $S$.
    $\widehat{P}(s'|s, a) = \text{rSTAT}(S[s, a], \phi_s(s'))$
    $\widehat{r}(s, a) = r(s, a)$
end for
return $\widehat{\mathcal{M}}$ built from $\widehat{P}(\cdot|s, a)$ and $\widehat{r}$
```

While our rPVI algorithm achieves cubed dependence on $|S|$, trying to obtain replicable transition dynamics is significantly harder using the rSTAT approach as we show in the following Theorem B.1.

Theorem B.1. *Let $\mathcal{M}$ be a fixed MDP and assume access to a generative model $G_{\mathcal{M}}$. Let $\epsilon \in [0,1]$ be the accuracy parameter, $\rho \in [0,1]$ be the reproducibility parameter. Suppose*

$$m = O\left(\frac{|S|^6|\mathcal{A}|^3}{\varepsilon^2\rho^2}\log\left(\frac{|S||\mathcal{A}|}{\delta}\right)\right).$$

is the number of calls to $G_{\mathcal{M}}$ for every (s,a,s') tuple, it holds for all (s,a,s') across two runs that

$$\boldsymbol{Pr}[|P(s'|s,a) - \widehat{P}(s'|s,a)| \geq \varepsilon] \in O(\delta) \quad \wedge \quad Pr[\widehat{P}^{(1)}(s'|s,a) \neq \widehat{P}^{(2)}(s'|s,a)] \in O(\rho) \tag{2}$$

where $\widehat{P}^{(i)}$ is our approximation of the transitions P in the ith run.

*Proof Sketch.*The analysis that falls out of using statistical queries for the model approximation requires us to distribute the probability or replicability failure across all possible state-action-state tuples. The proof then is similar to that of rPVI. We use Chernoff bounds to get a sample-complexity for failre and reproducbility but this time we need to union bound over all of $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Since the union bound dependency from the rSTAT procedure enters our sample size quadratically, we end up picking $\rho_{SQ} = \rho/(|S|^2|A|)$ and $\delta_{SQ} = \delta/(|S|^2|A|)$. Then, we have consider sampling data for every (s,a,s') tuple which leads to the bound in Theorem B.1. This highlights the difficulty of the statistical query approach for full model-based reinforcement learning. It is, however, not unlikely that more refined tools that utilize vector concentrations could lead to improved sample complexities for replicably approximate MDPs.

C Computational Requirements

Our code is written in Python and mostly uses functions from the numpy library for parallelization. Our algorithms can easily run on house-hold grade computers using central processing units (CPUs) with 2-4 cores. Yet, depending on the speed of the CPUs and the chosen sample-size one run may take up to 4 hours. Most of this runtime comes from numpy's sampling procedures. For our experiments, we had access to 3 Lambda server machines with AMD EPYC™ CPUs and 128-thread support.