A Further definitions

Definition A.1 (Replicable value function estimation)**.** *Let $\mathcal{A}$ be a policy estimation algorithm that outputs an estimated Q-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 $\widehat{Q}^{*(2)}(s,a) \leftarrow \mathcal{A}(S_2; r)$, it holds for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ that*

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

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. The proof closely follows that of Kearns and Singh [1998a]. We want to prove that after T iterations of Replicable Phased Value Iteration, 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 values $\hat{V}_t(s')$ from the rSTAT procedure can be estimated accurately such that the following probabilities are bounded

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

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

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

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

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

We upper bound the second term in the triangle inequality using the standard Bellman operator defined as

$$
(\mathcal{T}Q)(s,a) = R(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim P}[V_t(s')] \tag{1}
$$

as follows

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

As a result, we obtain that

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

Now, all we need to do is choose α and T accordingly. If we choose $T \geq \log\left(\frac{2}{(1-\gamma)^2\varepsilon}\right)/(1-\gamma)$ and we pick $\alpha = (1-\gamma)\frac{\varepsilon}{2}$ 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 replicable 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 transitions drawn and $V^{(i)}$ the value function in the ith run. Suppose that m is sufficiently large such that our statistical query estimate yields replicable values estimates such that for all $s \in \mathcal{S}$, $t \in T$, it holds that $\widehat{V}_t^{(1)}(s') = \widehat{V}_t^{(2)}(s')$. 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,

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

where we used the fact that rewards are deterministic and $\widehat{V}_t^{(1)}(s') = \widehat{V}_t^{(2)}(s')$ 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 $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, t \in [0,T]$,

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

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

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

where β is the bin size of discretization that is chosen according to the original rSTAT procedure. By union bound and Chernoff inequality we have that

$$\begin{aligned}&\textbf{Pr}\left[\bigcup_{(s,a),t}\left(\widehat{V}_t(s) - \mathop{\mathbb{E}}_{s \sim P}\left[\widehat{V}_t(s)\right] > \alpha'\right)\right] \leq |\mathcal{S}||\mathcal{A}|Te^{-2m\alpha'^2} \leq \delta \\ &\implies m \geq \frac{1}{2\alpha'^2}\log\left(\frac{2|\mathcal{S}||\mathcal{A}|T}{\delta}\right) \ .\end{aligned}$$

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 $2\alpha'/\beta$. Again via union bound, we have that

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

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

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

Setting α 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 Phased Q-learning 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 replicably 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 replicable 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 $\phi_s(s') := \mathbb{1}[s = s']$
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 $S$.
    $\widehat{P}(s'|s, a) = \mathsf{rSTAT}(S[s, a], \phi_s(s'))$
    $\widehat{R}(s, a) = R(s, a)$
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 Observation B.1.

Observation 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 replicability parameter. Suppose*

$$m = O\left(\frac{|S|^5|\mathcal{A}|^3}{\varepsilon^2(\rho - 2\delta)^2} \log\left(\frac{|S||\mathcal{A}|}{\delta}\right)\right).$$

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

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

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 failure 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) tuple which leads to the bound in Observation 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.

B.3 Policy convergence for Lemma 4.2

The proof that Algorithm 2 converges to an ε-optimal policy makes use of lemmas from Kearns and Singh [1998b] and Brafman and Tennenholtz [2003]. 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 B.1 (Kearns and Singh [1998b])**.** *Let* $\mathsf{Explore}(\tau)$ *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*

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

Lemma B.2 (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,

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

Lemma 4.2 (Convergence)**.** *Consider* $\mathscr{A}$ *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* $\delta < \rho/4$. *Let* $T \in \Theta(\frac{H|\mathcal{S}||\mathcal{A}|}{\varepsilon} + \frac{H^2 \log(1/\delta)}{\varepsilon^2})$ *be a bound on the number of iterations of Algorithm 2. Suppose* $1 - \gamma > \frac{\sqrt{\varepsilon}}{H|\mathcal{A}|}$ *and let* $m \in \tilde{O}\left(\frac{|\mathcal{S}|^2|\mathcal{A}|^2 T^4 \log(1/\rho)}{\rho^2}\right)$ *be the number of trajectories per iteration. Let* $k = H$ *be 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'. *Then with all but probability* δ, *after* T *iterations,* $\mathscr{A}$ *yields an* ε*-optimal policy.*

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

Proof of Lemma 4.2. We use Lemma B.1 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 $\rho_{SQ} \in O(\frac{\rho}{|\mathcal{S}|^2|\mathcal{A}|})$, $\alpha_{SQ} \in O(\frac{\varepsilon(1-\gamma)^2}{|\mathcal{S}|})$, 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 $[-mH, mH]$. We have taken $T \in \Theta(\frac{H|\mathcal{S}||\mathcal{A}|}{\varepsilon} + \frac{H^2 \log(1/\delta)}{\varepsilon^2})$ and so Azuma's inequality then gives us that

$$\begin{aligned} \mathbf{Pr}_S[X_T \leq -\tfrac{mH^2 \log(1/\delta)}{\varepsilon}] &\leq \exp(-O(\tfrac{m^2 H^4 \log^2(1/\delta)}{\varepsilon^2 T m^2 H^2})) \\ &\leq \exp(-O(\tfrac{H^2 \log^2(1/\delta)}{\varepsilon^2 T})) \\ &\in O(\delta). \end{aligned}$$

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

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

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. We have taken $k = H$, so after T iterations, we have

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

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 B.2. $\square$

To contextualize the sample complexity of Algorithm 2, we first recall that the sample complexity of the original R-max algorithm of Brafman and Tennenholtz [2003], suppressing dependence on γ, is roughly $\tilde{O}\left(\frac{|S|^2|A|\log(1/\delta)}{\varepsilon^3}\right)$. In Theorem 4.2, we show that the total sample complexity of Algorithm 2 is $\tilde{O}\left(\frac{|\mathcal{S}|^7|\mathcal{A}|^7 H^6}{\rho^2\varepsilon^5} + \frac{|\mathcal{S}|^2|\mathcal{A}|^2 H^{10}\log^5(1/\delta)}{\varepsilon^{10}}\right)$, so the sample overhead for replicability that we obtain is $\tilde{O}\left(\frac{|\mathcal{S}|^5|\mathcal{A}|^6 H^6}{\rho^2\varepsilon^2} + \frac{|\mathcal{A}|H^{10}}{\varepsilon^7}\right)$.

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.