The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

PAC Reinforcement Learning with an Imperfect Model

Nan Jiang
Microsoft Research
New York, NY 10011
nanjiang@umich.edu

Abstract

Reinforcement learning (RL) methods have proved to be successful in many simulated environments. The common approaches, however, are often too sample intensive to be applied directly in the real world. A promising approach to addressing this issue is to train an RL agent in a simulator and transfer the solution to the real environment. When a high-fidelity simulator is available we would expect significant reduction in the amount of real trajectories needed for learning. In this work we aim at better understanding the theoretical nature of this approach. We start with a perhaps surprising result that, even if the approximate model (e.g., a simulator) only differs from the real environment in a single state-action pair (but which one is unknown), such a model could be information-theoretically useless and the sample complexity (in terms of real trajectories) still scales with the total number of states in the worst case. We investigate the hard instances and come up with natural conditions that avoid the pathological situations. We then propose two conceptually simple algorithms that enjoy polynomial sample complexity guarantees with *no dependence* on the size of the state-action space, and prove some foundational results to provide insights into this important problem.

1 Introduction

Recently, Reinforcement learning (RL) methods have achieved impressive successes in many challenging domains (Mnih et al. 2015; Heess et al. 2015; Silver et al. 2016; Levine et al. 2016; Mnih et al. 2016). Many of these successes occur in simulated environments (e.g., video / board games, simulated robotics domains), and the state-of-the-art approaches require a large number of training samples, rendering them inapplicable in non-simulator problems where data acquisition may be costly.

A promising approach to addressing this issue is to train an RL agent in a simulator and transfer the solution to the real environment, which is particularly relevant but not limited to robotics domains (Koos, Mouret, and Doncieux 2010; Cutler and How 2015; Hanna and Stone 2017). The approach faces a significant challenge that the policy trained in a simulator may have degenerate performance in the real environment due to the imperfectness of the simulator (Kober, Bagnell, and Peters 2013).

There are many aspects from which one could address this challenge. For example, the simulator and the real environment may not share the same observation spaces and we may need to learn a transfer function that corrects the mismatch, or the simulator may be significantly different from the real environment that we should only transfer useful features instead of actual policies (Rusu et al. 2016), etc. While there has been active empirical research in this area, little in theory is known in terms of when transfer is possible and what guarantees we can have.

In this paper we focus on a particular angle of this problem, and provide some foundational results under stylized assumptions to help understand the theoretical nature of this approach. We start with a simple question: given an approximate model (e.g., a simulator) that only differs from the real environment in 1 state-action pair (but which one is unknown), can we always learn a near-optimal policy by collecting significantly fewer real trajectories compared to RL from scratch, i.e., without the model? Perhaps surprisingly, the answer is no due to a lower bound. We understand and draw insights from the hard instances, and come up with natural conditions that exclude the pathological scenarios (Sec.4). Under these conditions, we describe and analyze two algorithms whose sample complexity guarantees only depend on the number of incorrect state-action pairs and have no dependence on the size of the state and action spaces (Sec.5 and 6).

2 Preliminaries

We consider episodic RL problems where the real environment is specified by a finite-horizon MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and for simplicity we assume that $\mathcal{S}$ and $\mathcal{A}$ are finite but can be arbitrarily large. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function ($\Delta(\mathcal{S})$ is the probability simplex over $\mathcal{S}$, i.e., the set of all probability distributions). $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function; we assume rewards are non-negative. H is the horizon, and $\mu \in \Delta(\mathcal{S})$ is the initial distribution.

In general, optimal policies in the finite-horizon setting are non-stationary, i.e., they are time dependent. To keep the notations simple, w.l.o.g. we assume that each state only appears in a particular time step (or *level*) $1 \le h \le H$, and the state space can be partitioned into disjoint sets $\mathcal{S} = \bigcup_{h=1}^{H} \mathcal{S}_h$, where μ is supported on $\mathcal{S}_1$ and states

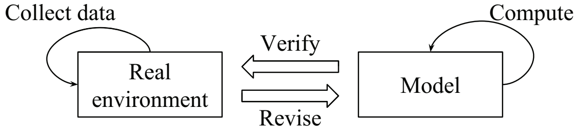


Figure 1: Protocol of how the learner interacts with the real environment and the approximate model.

in $\mathcal{S}_h$ only transition to those in $\mathcal{S}_{h+1}$. Assume that the total reward has bounded magnitude for any sequence of state-actions, i.e., $\sum_{h=1}^{H} R(s_h, a_h) \in [0,1]$ holds for all $s_1 \in \mathcal{S}_1, a_1 \in \mathcal{A}, \ldots, s_H \in \mathcal{S}_H, a_H \in \mathcal{A}$.[1]

Given a policy $\pi : \mathcal{S} \to \mathcal{A}$, a random trajectory is generated by $s_1 \sim \mu$, and for $h = 1, \ldots, H$, $a_h \sim \pi(s_h)$, $r_h = R(s_h, a_h)$, $s_{h+1} \sim P(s_h, a_h)$. The ultimate measure of π's performance is $v_M^\pi := \mathbb{E}[\sum_{h=1}^{H} r_h \,|\, \pi]$. Also define the value function of π as $V_M^\pi(s) := \mathbb{E}[\sum_{h'=h}^{H} r_{h'} \,|\, \pi, s_h = s]$ where h is such that $s \in \mathcal{S}_h$. Note that all such value functions have bounded range $[0,1]$.

Let $\pi_M^\star$ be the optimal policy in M, which maximizes V_M^π for all $s \in \mathcal{S}$. We use $V_M^\star$ as a shorthand for $V_M^{\pi_M^\star}$, which satisfies the Bellman optimality equation $V_M^\star = \mathcal{T} V_M^\star$, where $\mathcal{T} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ is the Bellman update operator[2] $(\mathcal{T}f)(s) := \max_{a \in \mathcal{A}} \left[R(s,a) + \mathbb{E}_{s' \sim P(s,a)}[f(s')] \right]$.

3 Problem formulation

Our learning algorithm is given an approximate model $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{P}, R, H, \mu)$ as input. For simplicity we assume that $\widehat{M}$ and M only differ in dynamics, and our analyses extend straightforwardly to approximate reward functions.

Abstraction of computation

Since this work focuses on the sample efficiency regarding the trajectories in the real environment, we abstract away the *computation* in the approximate model by assuming an oracle that can take any M' as input and return $\pi_{M'}^\star$ and $v_{M'}^\star$. Later in Algorithm 2 we will also need the oracle to return $V_{M'}^\star$, but that is often a side product of computing $\pi_{M'}^\star$.[3]

Protocol

We consider a learner that interacts with the real environment and the approximate model in an alternating manner (see Figure 1). The learner repeats the following steps until it finds a satisfying policy: (1) carrying out computation in the model, (2) collecting data in the real environment, and (3) revising the model. In Sec.7 we show that the interactivity in the protocol is crucial — under a non-interactive protocol no algorithm can achieve polynomial sample complexity.

Under this protocol, an algorithm needs to specify what computation to carry out in the approximate model, what actions to take and how much data to collect in the real environment, and how to revise the model based on the real data. For now assume that the learner can revise the model arbitrarily, i.e., it can change any entry of the transition function of $\widehat{M}$. Of course, this assumption can be unrealistic when the simulator is sophisticated and can only be accessed in a black-box manner; we relax this assumption in Sec.6.

Incorrect state-action pairs

When the approximate model is very close to the real environment, intuitively we would expect significantly reduced sample complexity (in terms of real trajectories) compared to RL from scratch. To formalize this intuition, we define a soft notion of incorrect state-action pairs in Definition 1, and use the number of incorrect state-action pairs $|\mathcal{X}_{\xi\text{-inc}}|$ to characterize the imperfectness of the model.

Definition 1 (ξ-correctness)**.** *Given M and $\widehat{M}$, we say that (s,a) is ξ-incorrect if*[4]

$$d_{M,\widehat{M}}(s,a) := \|P(s,a) - \widehat{P}(s,a)\|_{TV} > \xi.$$

Let $\mathcal{X}_{\xi\text{-inc}}$ be the set of state-action pairs that are ξ-incorrect.

Fact 1. *$\mathcal{X}_{\xi\text{-inc}} \subseteq \mathcal{X}_{\xi'\text{-inc}}$ if $\xi \geq \xi'$.*

Remark 1. When $\mathcal{S}$ is large, it may be difficult to have a model $\widehat{M}$ that matches M on the transition probability to each next-state in most state-action pairs. In Sec.8 we give alternative definitions of $\mathcal{X}_{\xi\text{-inc}}$ that are more lenient and show that our analyses automatically extend.

Goal: no dependence on $|\mathcal{S}|$ or $|\mathcal{A}|$

We are interested in the scenario where $|\mathcal{S}|$ and $|\mathcal{A}|$ may be very large but $|\mathcal{X}_{\xi\text{-inc}}|$ is small (for some reasonable choice of ξ). Our goal is to develop algorithms that can learn a near-optimal policy in M using polynomially many real trajectories, where the polynomial can depend on $|\mathcal{X}_{\xi\text{-inc}}|$ (and other parameters such as H) but *not* on $|\mathcal{S}|$ or $|\mathcal{A}|$.

4 Sufficient conditions for avoiding dependence on $|\mathcal{S}|$ and $|\mathcal{A}|$

Unfortunately, the goal stated above is impossible without further assumptions. In particular, there can be situations where $|\mathcal{X}_{0\text{-inc}}| = 1$ but the sample complexity is polynomial in $|\mathcal{S}|$. This result is formalized in Proposition 1. The proof builds on the lower bound from (Auer, Cesa-Bianchi, and Fischer 2002) and is deferred to the appendix.[5]

Proposition 1. *Without further assumptions, no algorithm can return an ϵ-optimal policy with a probability higher than $2/3$ and a sample complexity of poly$(|\mathcal{X}_{0\text{-inc}}|, H, 1/\epsilon)$ (recall from Fact 1 that $|\mathcal{X}_{\xi\text{-inc}}|$ is the largest when $\xi = 0$).*

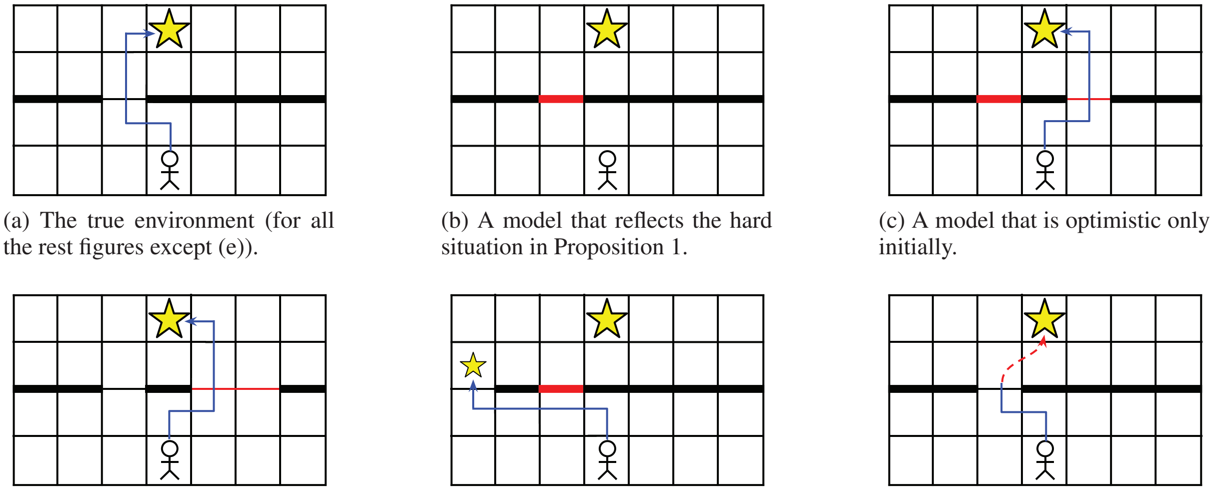
[1] The assumption makes no reference to the transition dynamics, which leads to boundedness of total reward in the approximate model and its revisions to be introduced later.

[2] Let $f(s_{H+1}) \equiv 0$ so that the same equation applies to $s \in \mathcal{S}_H$.

[3] To approximate the oracle in problems with large state spaces, we can use Sparse Sampling (Kearns, Mansour, and Ng 2002) or any Monte-Carlo tree search methods that do not depend on the state branching factor (Bjarnason, Fern, and Tadepalli 2009; Grill, Valko, and Munos 2016). Practically speaking, deep RL methods, which are empirically state-of-the-art, are also reasonable approximations (Mnih et al. 2015).

[4] For distributions over finite spaces, $\|\cdot\|_{TV} = \frac{1}{2}\|\cdot\|_1$.

[5] All appendices can be found in
https://sites.google.com/a/umich.edu/nanjiang/aaai18-pac.pdf.



(a) The true environment (for all the rest figures except (e)).

(b) A model that reflects the hard situation in Proposition 1.

(c) A model that is optimistic only initially.

(d) A desired approximate model.

(e) A situation where Assumption 1 fails but Eq.(1) promises nontrivial value.

(f) A situation where Eq.(1) delivers nontrivial guarantee and Eq.(5) is vacuous.

Figure 2: An environment **(a)** and a series of models. See text in Sec.4 for the description of the domain and the models in **(b)** to **(d)**. The red lines and bars indicate the mistakes of the model, and the blue path shows an optimal policy of the model.
(e): Here we modify the real environment by removing the leftmost obstacle and putting a smaller reward behind it. Since an episode terminates at any star, the optimal policy is still to go through the middle gate, which is erroneously blocked in the model. In this case, Eq.(1) will guarantee that we can get the smaller reward, which is suboptimal but nontrivial.
(f): This model erroneously believes that the agent will be teleported to the reward upon passing through the gate (red dashed line). In this case, Definition 6 terminates a model episode when the agent goes through the gate, essentially blocking it. As a result, the value guaranteed by Theorem 2 (Eq.(5)) becomes vacuous in this case, while Theorem 1 still competes against $v^\star_M$.

Understand the hard instances

We explain the hardness result in Figure 2 and draw insights from it. Here the real environment is depicted in 2(a): the agent can move in 4 directions in this grid world, and the thick bars represent obstacles. An episode ends either when the agent runs into an obstacle or when it gets to a star (reward). The optimal policy is to go through the only gate to get the star, but the agent has no knowledge of where the gate is. Without additional information, the agent needs to try each obstacle one by one and incurs $\Omega(|\mathcal{S}|)$ sample complexity (note that we can easily scale up the problem).

The hard instance in Proposition 1 is similar to the one depicted in Figure 2(b): the model claims that there is no gate, hence no policy can get to the reward. While such a model is obviously useless, it indeed satisfies $|\mathcal{X}_{\text{0-inc}}| = 1$. Therefore, a small $|\mathcal{X}_{\text{0-inc}}|$ does not necessarily imply a useful model, and we need to impose additional conditions to exclude such degenerate cases.

Exclude the degenerate cases: a first attempt

One thing that we might notice in Figure 2(b) is that the optimal value in the model is very low ($v^\star_{\widehat{M}} = 0$). Intuitively, a model should claim that there exists a policy that achieves a high value to be any useful. Based on this observation, we might come up with the condition that $v^\star_{\widehat{M}} \geq v^\star_M$, which excludes the model in 2(b).

However, it is easy to construct another degenerate case without violating the condition; see 2(c). The model satisfies the condition by claiming the existence of a gate in the wrong location, which yields $|\mathcal{X}_{\text{0-inc}}| = 2$. Note, however, that the learner could generate such a model by choosing a location randomly, which does not require any external information hence the model is still useless.

A sufficient condition

An alternative explanation of the failure of 2(c) is that, if we follow the optimal policy suggested by the model in the real environment, we will realize that the red gate is actually an obstacle. Once this mistake is fixed, however, we literally get back to the model in 2(b).

Based on the above intuitions, we come up with the a sufficient condition that excludes all degenerate cases that obscure the desired sample complexity. Roughly speaking, we require the optimal value in the approximate model to always stay high whenever we replace the dynamics of any subset of state-action pairs in $\mathcal{X}_{\xi\text{-inc}}$ with the true dynamics (see Figure 2(d) for example). This idea is formalized in the following definitions and Assumption 1.

Definition 2 (Partially repaired model)**.** *Given M and $\widehat{M}$ which only differ in the transition dynamics P and $\widehat{P}$, and $\mathcal{X} \subseteq \mathcal{S} \times \mathcal{A}$, define $\widehat{M}_{\mathcal{X}}$ as the MDP $(\mathcal{S}, \mathcal{A}, \widehat{P}_{\mathcal{X}}, R, H, \mu)$ where*

$$\widehat{P}_{\mathcal{X}}(s,a) := \begin{cases} P(s,a), & \textit{if } (s,a) \in \mathcal{X}, \\ \widehat{P}(s,a), & \textit{otherwise.} \end{cases}$$

Definition 3. $\mathcal{M} := \{\widehat{M}_{\mathcal{X}} : \mathcal{X} \subseteq \mathcal{X}_{\xi\text{-}inc}\}$.

Assumption 1 (Always optimistic). $\forall M' \in \mathcal{M}$, $v^\star_{M'} \geq v^\star_M$.

While Assumption 1 is sufficient, it is also too strict since it fails if any $v^\star_{M'}$ is slightly below $v^\star_M$. In the next section we will not make any explicit assumptions but rather use an agnostic version of Assumption 1: instead of requiring the algorithm to compete against $v^\star_M$ (i.e., return a policy with at least $v^\star_M - \epsilon$ value), we only require the algorithm to compete against

$$\inf_{M' \in \mathcal{M}} v^\star_{M'}. \tag{1}$$

If Assumption 1 holds, we compete against $v^\star_M$ as usual; when Assumption 1 fails, our optimality guarantee degrades gracefully with the violation. Figure 2(e) illustrates a situation where Eq.(1) delivers nontrivial guarantee while Assumption 1 fails (see caption for details).

5 Repair the model

In this section, we describe a conceptually simple algorithm whose sample complexity has no dependence on $|\mathcal{S}|$ or $|\mathcal{A}|$. The pseudocode is given in Algorithm 1. We first walk through the pseudocode and give some intuitions, and then state and prove the sample complexity guarantee.

The outer loop of the algorithm computes $\pi_t = \pi^\star_{M_t}$, the optimal policy of the current model M_t (initialized as $\widehat{M}$), and use Monte-Carlo policy evaluation to estimate its return. If the policy's performance is satisfying, we simply output it. Otherwise, the inner loop uses the same policy to collect trajectories, and add every next-state to the dataset associated with the preceding state-action pair.

The inner loop stops whenever the size of a dataset $D_{s,a}$ for some (s, a) increases to a pre-determined threshold, n_{est}, which will be set later in the analysis. This triggers a model revision: we replace $\widehat{P}(s, a)$ with the empirical frequency of states observed in $D_{s,a}$, and produce model M_{t+1} for the next iteration of outer loop.

A straightforward analysis of the above procedure, however, would incur polynomial dependence on $|\mathcal{S}|$: we recover the multinomial distributions $\{P(s, a) : (s, a) \in \mathcal{S} \times \mathcal{A}\}$ from sample data, and such distributions are supported on $\mathcal{S}$. In general, if we want to guarantee low estimation error (measured by e.g., total variation), we would incur dependence on size of the support.

To overcome this difficulty, note that there is no need to recover the detailed transition distribution over states. Instead, we simply need to guarantee that when we use the estimated probabilities in the Bellman update operators, the value function(s) of interest are updated correctly. That is, when we have enough samples for some $D_{s,a}$, we would like to guarantee that

$$d^f_{M,D}(s, a) := \left|\mathbb{E}_{s' \sim P(s,a)}[f(s')] - \mathbb{E}_{s' \sim D_{s,a}}[f(s')]\right| \tag{2}$$

is small for some careful choice(s) of f. In standard RL literature, f is often chosen to be $V^\star_M$, the optimal value function which we compete against (see e.g., (Kearns, Mansour, and Ng 2002)). For such a fixed f, we can use Hoeffding's inequality to guarantee concentration, and the necessary size of $D_{s,a}$ has no dependence on $|\mathcal{S}|$.

Algorithm 1 MODEL_REPAIR($\widehat{M}$)

```
$M_0 \leftarrow \widehat{M}$. $D_{s,a} \leftarrow \{\}$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$.
for $t = 0, 1, \ldots$ do
    $\pi_t \leftarrow \pi^\star_{M_t}$.
    Collect $n_{\text{eval}}$ trajectories using $\pi_t$, and let $\hat{v}^{\pi_t}_M$ be the Monte-Carlo estimate of value.
    if $\hat{v}^{\pi_t}_M \geq v^{\pi_t}_{M_t} - 7\epsilon/10$ then return $\pi_t$.
    repeat
        Collect trajectory $s_1, a_1, \ldots, s_H, a_H$ using $\pi_t$.
        $\forall\, h$, add $s_{h+1}$ to $D_{s_h,a_h}$ if $|D_{s_h,a_h}| < n_{\text{est}}$.
    until some $|D_{s,a}|$ increases to $n_{\text{est}}$ for the 1st time
    Construct $M_{t+1}$ from $\widehat{M}$ by plugging in $D_{s,a}$ for each $(s, a)$ in $\mathcal{X}'_{t+1} := \{|D_{s,a}| = n_{\text{est}}\}$.
end for
```

In our case, however, we compete against multiple value functions (Eq.(1)), and need to guarantee that all of them are updated correctly. In particular, when we have enough samples in $D_{s,a}$, we want Eq.(2) to be small for any f that is the optimal value function of some partially repaired model (Definition 2). Interestingly, there are at most $2^{|\mathcal{X}_{\xi\text{-inc}}|}$ such models (Fact 2), and by union bound we pay logarithmic dependence on the number of functions, which is $O(|\mathcal{X}_{\xi\text{-inc}}|)$. Below we state the formal guarantees for the algorithm and prove it using the intuitions described above.

Theorem 1. *Given any $\delta \in (0, 1), \epsilon \in (0, 1)$, run Algorithm 1 with parameters $\xi = \frac{\epsilon}{10H^2}$, $n_{eval} = \tilde{O}(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$, $n_{est} = \tilde{O}(\frac{H^4}{\epsilon^2}(|\mathcal{X}_{\xi\text{-}inc}| + \log\frac{1}{\delta}))$. With probability at least $1 - \delta$ the algorithm will return a policy π_T such that $v^{\pi_T}_M \geq \inf_{M' \in \mathcal{M}} v^\star_{M'} - \epsilon$ after acquiring $\tilde{O}\left(|\mathcal{X}_{\xi\text{-}inc}|(|\mathcal{X}_{\xi\text{-}inc}| + \log(1/\delta))\frac{H^4}{\epsilon^3}\right)$ sample trajectories.*[6]

To prove Theorem 1, we introduce some further definitions and helping lemmas. We first define the value function class of interest, $\mathcal{F}$, and establish some basic properties.

Definition 4. *Given $M, \widehat{M}, \xi$, let $\mathcal{F} := \{V^\star_{M'} : M' \in \mathcal{M}\}$ (recall the definition of $\mathcal{M}$ from Definition 3).*

Fact 2. $|\mathcal{F}| \leq |\mathcal{M}| \leq 2^{|\mathcal{X}_{\xi\text{-}inc}|}$.

Definition 5. *Given value function f and M_1 and M_2 that differ only in dynamics P_1 and P_2, define*

$$d^f_{M_1,M_2}(s, a) := \left|\mathbb{E}_{s' \sim P_1(s,a)}[f(s')] - \mathbb{E}_{s' \sim P_2(s,a)}[f(s')]\right|,$$

and $d^{\mathcal{F}}_{M_1,M_2} := \sup_{f \in \mathcal{F}} d^f_{M_1,M_2}(s, a)$.

Fact 3. *If $f \in [0, 1]$ $\forall f \in \mathcal{F}$, then for any M_1, M_2, s, a,*

$$d^f_{M_1,M_2}(s, a) \leq d^{\mathcal{F}}_{M_1,M_2}(s, a) \leq d_{M_1,M_2}(s, a) \leq 1$$

Lemma 1. *In Algorithm 1, for any fixed t and any $\delta \in (0, 1)$, w.p. at least $1 - \delta$, $|\hat{v}^{\pi_t}_M - v^{\pi_t}_M| \leq \sqrt{\frac{1}{2n_{eval}}\log\frac{2}{\delta}}$.*

[6] We use $\tilde{O}$ to suppress logarithmic dependence on $|\mathcal{X}_{\xi\text{-inc}}|$, H, $1/\epsilon$, and $\log(1/\delta)$. However, no $\log|\mathcal{S}|$ or $\log|\mathcal{A}|$ is incurred here.

Lemma 2. *In Algorithm 1, for any fixed $(s,a) \in \mathcal{S} \times \mathcal{A}$ and any $\delta \in (0,1)$, when $|D_{s,a}| = n_{est}$, w.p. at least $1-\delta$, $d_{M,D}^{\mathcal{F}}(s,a) \le \sqrt{\frac{1}{2n_{est}} \log \frac{2|\mathcal{F}|}{\delta}}$, where $d_{M,D}^{\mathcal{F}}(s,a) := \sup_{f \in \mathcal{F}} d_{M,D}^{f}(s,a)$.*

The proofs of the above two lemmas are elementary and deferred to the appendix. Our argument that $D_{s,a}$ only needs to update certain value functions correctly is supported by Lemma 3. Its proof is deferred to the appendix.

Lemma 3. *Given any M_1 and M_2 where $V_{M_1}^{\star} \in \mathcal{F}$, we have $\|V_{M_1}^{\star} - V_{M_2}^{\star}\|_\infty \le H \|d_{M_1,M_2}^{\mathcal{F}}\|_\infty$.*

The last lemma in this section is Lemma 4, which is a fine-grained version of the *simulation lemma* (Kearns and Singh 2002), a result commonly found in the analyses of PAC exploration algorithms. The proof is deferred to the appendix.

Lemma 4. *Suppose M_1 and M_2 only differ in dynamics. If there exists f' such that $\|f' - V_{M_2}^{\pi}\|_\infty \le \xi'$, we have*

$$|v_{M_1}^{\pi} - v_{M_2}^{\pi}| \le \langle \eta_{M_1}^{\pi}, d_{M_1,M_2}^{f'} \rangle + 2H\xi',$$

where $\eta_{M_1}^{\pi}$ is the state-action occupancy of π in M_1, defined as $\eta_{M_1}^{\pi}(s,a) := \sum_{h=1}^{H} \mathbb{P}[s_h = s, a_h = a \,|\, \pi, P_1]$.

With all the preparation, we are ready to prove Theorem 1.

Proof of Theorem 1. Throughout the analysis we assume that two type of events always hold, which are later guaranteed by concentration inequalities and union bound: (A) whenever a (s,a) pair satisfies $|D_{s,a}| = n_{\text{est}}$ we have $d_{M,D}^{\mathcal{F}}(s,a) \le \xi$; (B) $|v_M^{\pi_t} - \hat{v}_M^{\pi_t}| \le \epsilon/10$ (Line 5).

Given these high probability events, we first show the correctness of the algorithm. That is, when it terminates at $t = T$, the returned policy π_T satisfies the theorem statement. Define $\mathcal{X}_t'$ as on Line 10 (i.e., the set of (s,a) with sufficient samples at the beginning of round t) and $\mathcal{X}_t := \mathcal{X}_{\xi\text{-inc}} \bigcap \mathcal{X}_t'$. Since $\widehat{M}_{\mathcal{X}_t} \in \mathcal{M}$ and $V_{\widehat{M}_{\mathcal{X}_t}}^{\star} \in \mathcal{F}$, claim that

$$\forall t, \; \|d_{M_t, \widehat{M}_{\mathcal{X}_t}}^{\mathcal{F}}\|_\infty \le 2\xi. \tag{3}$$

This is because, for $(s,a) \in \mathcal{X}_t$, $\widehat{P}_{\mathcal{X}_t}(s,a) = P(s,a)$, and $P_t(s,a)$ uses the empirical estimate which is guaranteed to be ξ-close to $P(s,a)$ with respect to $\mathcal{F}$; for $(s,a) \notin \mathcal{X}_t'$, $P_t(s,a) = \widehat{P}_{\mathcal{X}_t}(s,a) = \widehat{P}(s,a)$; for the remaining case, both $\widehat{P}_{\mathcal{X}_t}(s,a)$ and $P_t(s,a)$ are ξ-close to $P(s,a)$ with respect to $\mathcal{F}$, so they differ by at most 2ξ.

Now we invoke Lemma 3 on $\widehat{M}_{\mathcal{X}_T}$ and M_T, and obtain $|v_{\widehat{M}_{\mathcal{X}_T}}^{\star} - v_{M_T}^{\pi_T}| \le 2H\xi$. Hence,

$$\begin{aligned} v_M^{\pi_T} &\ge \hat{v}_M^{\pi_T} - \tfrac{\epsilon}{10} \ge v_{M_T}^{\pi_T} - \tfrac{4\epsilon}{5} \\ &\ge v_{\widehat{M}_{\mathcal{X}_T}}^{\star} - 2H\xi - \tfrac{4\epsilon}{5} \ge \inf_{M' \in \mathcal{M}} v_{M'}^{\star} - \epsilon. \end{aligned}$$

Next we show that when $t < T$, π_t always puts significant occupancy on unlearned state-action pairs in $\mathcal{X}_{\xi\text{-inc}}$; we will refer to the visits to such state-action pairs as *effective visits* in the remainder of the proof. $\forall\, 0 \le t < T$, $v_{M_t}^{\pi_t} - v_M^{\pi_t} \ge \hat{v}_M^{\pi_t} + 7\epsilon/10 - \hat{v}_M^{\pi_t} - \epsilon/10 = 3\epsilon/5$. Let p be the occupancy of $\pi_{M_t}^{\star}$ on the subset of $\mathcal{X}_{\epsilon\text{-inc}}$ that are unlearned, i.e., p is the expected number of effective visits. We would like to upper bound $v_{M_t}^{\pi_t} - v_M^{\pi_t}$ via Lemma 4 by letting $M_1 := M$, $M_2 := M_t$. To do that we first have to bound ξ' in Lemma 4 by approximating $V_{M_t}^{\pi_t}$ with $V_{\widehat{M}_{\mathcal{X}_t}}^{\star}$ (the latter is in $\mathcal{F}$): recall that $\pi_t = \pi_{M_t}^{\star}$, and Lemma 3 implies that $\|V_{M_t}^{\pi_t} - V_{\widehat{M}_{\mathcal{X}_t}}^{\star}\|_\infty \le 2H\xi$. Now we can invoke Lemma 4 on M and M_t with $f' = V_{\widehat{M}_{\mathcal{X}_t}}^{\star}$ and $\xi' = 2H\xi$, and

$$\begin{aligned} v_{M_t}^{\pi_t} - v_M^{\pi_t} &\le p + (H-p)\xi + 4H^2\xi \\ &\le p + 5H^2\xi = p + \epsilon/2. \end{aligned} \tag{4}$$

From this we conclude that $p + \epsilon/2 \ge 3\epsilon/5$, so $p \ge \epsilon/10$. In other words, in expectation we have $\epsilon/10$ effective visits per trajectory. If we ignore the randomness in effective visits, $10 n_{\text{est}} |\mathcal{X}_{\xi\text{-inc}}|/\epsilon$ sample trajectories would guarantee successful termination of the algorithm, which matches the theorem statement. The remainder of the proof applies concentration inequalities to deal with the randomness and is deferred to the appendix. □

Before concluding the section, we comment on the guarantee in Theorem 1. Perhaps the most outstanding term is H^4, which seems unreasonably high. The main difficulty here is that M_t is random, and it is hard to capture it in a model class with reasonable size specified a priori. What we do is to approximate its optimal value function using $V_{\widehat{M}_{\mathcal{X}_t}}^{\star}$ before invoking Lemma 4, which blows up one-step transition error twice (see the $H^2\xi$ term in Eq.(4)).[7] In the next section we consider a similar but slightly different setting, where M_t can be exactly captured in a deterministic model class and the sample complexity is quadratic in H.

6 What if we cannot change the model?

As discussed in Sec.3, one of the unrealistic assumptions so far is that we can manipulate the model and make arbitrary changes to the transition function, which is seldom possible for a sophisticated simulator. To remove the assumption, we need to incorporate the knowledge learned from real trajectories without changing the model itself. But how?

We borrow intuitions from Figure 2 again. An alternative way of explaining 2(b) to 2(e) is that, we should actually compete against policies that only visit (s,a) pairs where the model dynamics are correct. Such an objective is also consistent with the optimality criterion of Eq.(1) for the models in 2(b) to 2(e).

To make this objective more robust, we may want to allow the agent to visit incorrect (s,a) pairs with small probabilities. Instead of imposing constraints on the state-action occupancy of a policy, a more lenient solution is to penalize a policy for visiting incorrect (s,a) pairs. A natural penalty would be to fix the future value of incorrect state-action pairs as the minimum value 0: the future value predicted by the

[7] The recent work of (Azar, Osband, and Munos 2017) faces a similar difficulty and they avoid heavy dependence on H by bounding a particular residual (see their Sec 5.1). However, their technique incurs dependence on $|\mathcal{S}|$ and cannot be applied here.

Algorithm 2 MODEL_PENALIZE($\widehat{M}$)

```
1: $M_0 \leftarrow \widehat{M}$. $\mathcal{X}_0 \leftarrow \{\}$. $D_{s,a} \leftarrow \{\}$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$.
2: for $t = 0, 1, \ldots$ do
3:     $\pi_t \leftarrow \pi^\star_{M_t}$, $f_t \leftarrow V^\star_{M_t}$.
4:     Collect $n_{\text{eval}}$ trajectories using $\pi_t$, and let $\hat{v}^{\pi_t}_M$ be the Monte-Carlo estimate of value.
5:     if $\hat{v}^{\pi_t}_M \geq v^{\pi_t}_{M_t} - 4\epsilon/5$ then return $\pi_t$.
6:     while $\forall (s,a) \notin \mathcal{X}_t$, ISFINE($s$, $a$, $f_t$) do
7:         Collect a trajectory $s_1, a_1, \ldots, s_H, a_H$ using $\pi_t$.
8:         $\forall\, h$, add $s_{h+1}$ to $D_{s_h,a_h}$ if $|D_{s_h,a_h}| < n_{\text{est}}$.
9:     end while
10:    $\mathcal{X}_{\text{tmp}} \leftarrow \{\}$.
11:    for $h = H-1, \ldots, 2, 1$ do
12:        $M_{\text{tmp}} \leftarrow \widehat{M}_{\setminus \mathcal{X}_{\text{tmp}}}$, $f_{\text{tmp}} \leftarrow V^\star_{M_{\text{tmp}}}$.
13:        $\mathcal{X}_{\text{tmp}} \leftarrow \mathcal{X}_{\text{tmp}} \bigcup$
               $\{(s,a) \in \mathcal{S}_h \times \mathcal{A} : \neg\text{ISFINE}(s, a, f_{\text{tmp}})\}$.
14:    end for
15:    $\mathcal{X}_{t+1} \leftarrow \mathcal{X}_{\text{tmp}}$, $M_{t+1} \leftarrow \widehat{M}_{\setminus \mathcal{X}_{t+1}}$.
16: end for

17: function ISFINE($s$, $a$, $f$)
18:    if $|D_{s,a}| < n_{\text{est}}$ then return true.
19:    if $d^f_{\widehat{M},D}(s,a) \leq 1.5\xi$ then return true.
        // recall definition of $d^f_{\widehat{M},D}$ from Eq.(2)
20:    return false.
21: end function
```

model is not trustworthy due to incorrect dynamics and we replace it with a pessimistic guess.

Finally, due to our assumption of non-negative rewards, the penalty can be simply implemented by terminating a model episode upon running into an incorrect (s,a) pair. We will still treat the penalized model as a new MDP to facilitate analysis, with the understanding that such changes can be implemented in a black-box manner. The penalized model is formally defined below.

Definition 6 (Partially penalized model)**.** *Given MDP $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{P}, R, H, \mu)$ and $\mathcal{X} \subseteq \mathcal{S} \times \mathcal{A}$, define $\widehat{M}_{\setminus \mathcal{X}}$ as the MDP $(\mathcal{S}, \mathcal{A}, \widehat{P}_{\setminus \mathcal{X}}, R, H, \mu)$ where*

$$\widehat{P}_{\setminus \mathcal{X}}(s,a) := \begin{cases} \texttt{termination}, & \textit{if } (s,a) \in \mathcal{X}, \\ \widehat{P}(s,a), & \textit{otherwise.} \end{cases}$$

We define $\widetilde{\mathcal{M}}$ and $\widetilde{\mathcal{F}}$, the analogies of $\mathcal{M}$ and $\mathcal{F}$.

Definition 7. *Given M, $\widehat{M}$, and ξ, define $\widetilde{\mathcal{M}} := \{\widehat{M}_{\setminus \mathcal{X}} : \mathcal{X} \subseteq \mathcal{X}_{\xi\text{-inc}}\}$, $\widetilde{\mathcal{F}} := \{V^\star_{M'} : M' \in \widetilde{\mathcal{M}}\}$.*

Similarly to Eq.(1), in this section we compete against:

$$\inf_{M' \in \widetilde{\mathcal{M}}} v^\star_{M'}. \tag{5}$$

It is worth noting that the value in Eq.(5) is always less than or equal to that in Eq.(1). Intuitively, whenever an incorrect state-action pair is discovered, we allow the agent to avoid it as opposed to fixing the incorrect dynamics since we refrain ourselves from modifying the model. This comes with the cost that we give up the opportunity of reusing these state-action pairs in future policies. This intuition is formalized in Fact 4, and we give a concrete example in Figure 2(f) where Eq.(1) and (5) have a nontrivial gap.

Fact 4. *For any $\mathcal{X} \subseteq \mathcal{S} \times \mathcal{A}$ and $\pi : \mathcal{S} \to \mathcal{A}$, $v^\pi_{\widehat{M}_{\setminus \mathcal{X}}} \leq v^\pi_{\widehat{M}_{\mathcal{X}}}$.*

In the remainder of this section, we introduce Algorithm 2 and state the sample complexity result. Overall Algorithm 2 is similar to Algorithm 1, but with a few differences:

1. Unlike Algorithm 1 where we blindly replace $\widehat{P}(s,a)$ with $D_{s,a}$ (which is valid as $D_{s,a}$ is always unbiased), here we penalize (s,a) selectively as penalizing a correct (s,a) may affect the value we could obtain.
2. While we would like to compare $\widehat{P}(s,a)$ and $D_{s,a}$ against all functions in $\widetilde{\mathcal{F}}$, this is impossible as we do not know $\widetilde{\mathcal{F}}$ in advance (because $\mathcal{X}_{\xi\text{-inc}}$ is unknown). Fortunately, it is sufficient to compare them against the current value function f_t, which gives the criterion on Line 6.
3. The for-loop computes a set of incorrect (s,a) pairs from the bottom up. This is necessary because penalizing an (s,a) pair at a lower level (i.e., a later time step) may trigger a change in value function, which affects whether some other (s,a) at a higher level should be penalized or not. The bottom-up procedure guarantees that `isfine` is never invoked on an outdated f.
4. Thanks to the binary nature of the penalty, any M_t that we could run into is a member of $\widetilde{\mathcal{M}}$ (with high probability), hence M_t and π_t are much more deterministic objects than in Algorithm 1. As a result, we avoid the difficulty in Theorem 1 and can show a quadratic dependence on H.

Theorem 2. *Given any $\delta \in (0,1), \epsilon \in (0,1)$, we run Algorithm 2 with parameters $\xi = \frac{\epsilon}{5H}$, $n_{eval} = \tilde{O}(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$, $n_{est} = \tilde{O}(\frac{H^2}{\epsilon^2}(|\mathcal{X}_{\xi\text{-inc}}| + \log \frac{1}{\delta}))$. W.p. at least $1 - \delta$ the algorithm will return a policy π_T such that $v^{\pi_T}_M \geq \inf_{M' \in \widetilde{\mathcal{M}}} v^\star_{M'} - \epsilon$ after acquiring $\tilde{O}(|\mathcal{X}_{\xi\text{-inc}}|(|\mathcal{X}_{\xi\text{-inc}}| + \log(1/\delta))\frac{H^2}{\epsilon^3})$ sample trajectories.*

The proof of Theorem 2 is similar to that of Theorem 1 and is deferred to the appendix.

7 Non-interactive algorithms are inefficient

In the previous sections we give two algorithms that enjoy polynomial sample complexity guarantee without any dependence on $|\mathcal{S}|$ or $|\mathcal{A}|$. Both algorithms fit into the abstract protocol we introduce at the beginning, that is, they alternate between computation in the approximate model and data collection in the real environment.

In this section we show that such interactivity is crucial for our purpose. In particular, we prove a hardness result that, if all the data are collected before we perform any computation in $\widehat{M}$, no algorithm can achieve the desired polynomial sample complexity even if the conditions required by Algorithms 1 and 2 are satisfied. We briefly sketch the proof idea and the full proof is deferred to the appendix.

Theorem 3. *If the data collection strategy is independent of $\widehat{M}$, no algorithm can learn an ϵ-optimal policy with probability $2/3$ using $poly(|\mathcal{X}_0|, H, 1/\epsilon)$ sample trajectories, even if $\inf_{M' \in \mathcal{M}} v^\star_{M'} = \inf_{M' \in \widetilde{\mathcal{M}}} v^\star_{M'} = v^\star_M$.*

Proof sketch. Assume towards contradiction that such an algorithm exists. We can solve the hard instance of best arm identification with $poly(\log|\mathcal{A}|, 1/\epsilon)$ samples, which is against the known lower bound. Concretely, we design $|\mathcal{A}|^2$ models, where each model claims that a pair of arms are more rewarding than others. Applying the hypothetical algorithm to each model allows us to make reliable pair-wise comparison between the arms using only $O(\log|\mathcal{A}|)$ independent datasets, each of size $O(1/\epsilon^2)$. □

8 Relax the definition of ξ-correctness

In this section we relax the definition of ξ-correctness as promised in Remark 1. In Definition 1, whether an (s, a) pair is correct is determined by how large $d_{M,\widehat{M}}(s, a)$ is. The key observation here is that in the proof of Theorem 1, we only use the fact $d_{M,\widehat{M}}(s, a) \le \xi$ for $(s, a) \notin \mathcal{X}_{\xi\text{-inc}}$ in Eq.(3) through $d^{\mathcal{F}}_{M,\widehat{M}}$ (recall Fact 3). Therefore, we can simply re-define $\mathcal{X}_{\xi\text{-inc}}$ based on $d^{\mathcal{F}}_{M,\widehat{M}}$ instead of $d_{M,\widehat{M}}$, and all the analyses and guarantees extend straightforwardly. Due to Fact 3, the new definition will result in a smaller number of incorrect state-action pairs, and hence yield improved sample complexity in Theorem 1. The same thing happens to Theorem 2, where we can re-define $\mathcal{X}_{\xi\text{-inc}}$ based on $d^{\widetilde{\mathcal{F}}}_{M,\widehat{M}}$.

The complication here, however, is that the definition of $d^{\mathcal{F}}_{M,\widehat{M}}$ ($d^{\widetilde{\mathcal{F}}}_{M,\widehat{M}}$) depends on $\mathcal{F}$ ($\widetilde{\mathcal{F}}$), which further depends on $\mathcal{X}_{\xi\text{-inc}}$ (recall Definitions 3, 4, and 7), and we are now modifying the definition of $\mathcal{X}_{\xi\text{-inc}}$ to make it depend on $\mathcal{F}$ ($\widetilde{\mathcal{F}}$). The resolution to the recursive dependence is to define things in a bottom-up order; see Algorithm 3. This procedure is very similar to the for-loop in Algorithm 2, where the same issue has already been encountered. The formal statement of the tightened results is given below. In the appendix we also describe a scenario where $|\overline{\mathcal{X}}_{\xi\text{-inc}}|$ and $|\widetilde{\mathcal{X}}_{\xi\text{-inc}}|$ are substantially smaller than $|\mathcal{X}_{\xi\text{-inc}}|$.

Proposition 2. *Let $\overline{\mathcal{X}}_{\xi\text{-inc}}$ and $\widetilde{\mathcal{X}}_{\xi\text{-inc}}$ be defined via Algorithm 3. We have: (1) $\overline{\mathcal{X}}_{\xi\text{-inc}} \subseteq \mathcal{X}_{\xi\text{-inc}}$, $\widetilde{\mathcal{X}}_{\xi\text{-inc}} \subseteq \mathcal{X}_{\xi\text{-inc}}$.*
(2) Theorems 1 and 2 still hold if we replace $|\mathcal{X}_{\xi\text{-inc}}|$ in the theorem statements by $|\overline{\mathcal{X}}_{\xi\text{-inc}}|$ and $|\widetilde{\mathcal{X}}_{\xi\text{-inc}}|$ respectively.

Algorithm 3 CONSTRUCTBADSET(M, $\widehat{M}$, ξ)

```
$\overline{\mathcal{X}}_{\xi\text{-inc}} \leftarrow \{\}$.
for $h = H-1, \ldots, 2, 1$ do
    $\mathcal{M} \leftarrow \{\widehat{M}_{\mathcal{X}} : \mathcal{X} \subseteq \overline{\mathcal{X}}_{\xi\text{-inc}}\}$. // use $\widehat{M}_{\backslash\mathcal{X}}$ for $\widetilde{\mathcal{X}}_{\xi\text{-inc}}$
    $\mathcal{F} \leftarrow \{V^\star_{M'} : M' \in \mathcal{M}\}$.
    $\overline{\mathcal{X}}_{\xi\text{-inc}} \leftarrow \overline{\mathcal{X}}_{\xi\text{-inc}} \bigcup$
            $\{(s, a) \in \mathcal{S}_h \times \mathcal{A} : d^{\mathcal{F}}_{M,\widehat{M}}(s, a) > \xi\}$.
end for
```

9 Conclusions and discussions

In this paper, we investigate the theoretical properties of reinforcement learning with an approximate model as side information. We believe that there are 3 high-level insights that can be drawn from the paper:

1. We need the model to *always* stay optimistic (Assumption 1), otherwise there are degenerate cases where the model is useless even if it is correct in all but a constant number of state-action pairs (Proposition 1).
2. It is important that the learner interacts with the environment and the model in an alternating manner (Theorem 3).
3. Under 1+2, we can achieve polynomial sample complexity in the number of incorrect state-action pairs and incur no dependence on $|\mathcal{S}|$ and $|\mathcal{A}|$ (Theorems 1 and 2).

We conclude the paper by discussing related work, limitations of our assumptions and results, and open questions.

- The most related work is (Cutler, Walsh, and How 2015), who consider multiple simulators with varying levels of fidelity. They implicitly assume that a simulator's quality is homogeneous across the state-action space, evidenced by their fidelity defined as the worst-case error over states and actions. Consequently, they incur dependence on $|\mathcal{S}|$ and $|\mathcal{A}|$. In another related work, (Ha and Yamane 2015) propose an algorithm for linear control problems that is similar in spirit to our Algorithm 1. While no sample complexity guarantee is given, their algorithm is empirically validated and produces promising results.
- A major limitation of our work is that we assume $|\mathcal{S}|$ (and $|\mathcal{A}|$) is large but $|\mathcal{X}_{\xi\text{-inc}}|$ is small, which can be unrealistic. While it is possible to extend the analyses to accommodate continuous $\mathcal{X}_{\xi\text{-inc}}$ by a covering argument (Kakade, Kearns, and Langford 2003; Pazis and Parr 2013), such arguments incur dependence on the covering numbers, which are typically exponential in the dimension. Taking our analyses forward to a practical scenario may require satisfying theoretical solutions to exploration in large state spaces, an important research direction that is relatively understudied by itself despite a few very recent advances (Krishnamurthy, Agarwal, and Langford 2016; Jiang et al. 2017).
- The conditions we have identified (e.g., Assumption 1 and the agnostic version) are sufficient. Are they necessary and are there weaker conditions?
- There is a gap between the interactivity of our algorithms (polynomially many alternations) and the non-interactive lower bound (no alternation at all; see Theorem 3). While we might expect a stronger lower bound to exclude even a small (e.g., constant) number of alternations, the current formulation cannot prevent the agent from loading the entire $\widehat{M}$ into its memory to consult the model at any future time. Obtaining the stronger lower bound (if it exists) would require a careful characterization of what kind of computation is allowed within each round.

- The sample complexity guarantees obtained in Theorems 1 and 2 may not be optimal. In particular, it might be possible to remove one $1/\epsilon$ term by carefully distinguishing important states from the unimportant ones (Dann and Brunskill 2015). It might also be possible to reduce the dependence on $|\mathcal{X}_{\xi\text{-inc}}|$ by determinizing the order in which we fix / penalize incorrect (s, a) pairs, e.g., by delaying model revision and updating multiple state-action pairs at once. Tightening the upper bounds and finding matching lower bounds are interesting directions for future work.
- Our algorithms explore with the optimal policy of the model. Are there more sophisticated strategies that improve sample complexity? Since we want to avoid dependence on $|\mathcal{A}|$, standard operations such as taking actions uniformly are prohibited. Any intelligent exploration in this case might have to be heavily informed by the model.

Acknowledgements

The author thanks the anonymous reviewers for the insightful comments. The research question was inspired by a conversation with David Meger at McGill University.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Azar, M. G.; Osband, I.; and Munos, R. 2017. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, 263–272.

Bjarnason, R.; Fern, A.; and Tadepalli, P. 2009. Lower bounding Klondike solitaire with Monte-Carlo planning. In *Proceedings of International Conference on Automated Planning and Scheduling*, 26–33.

Cutler, M., and How, J. P. 2015. Efficient reinforcement learning for robots using informative simulated priors. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2605–2612. IEEE.

Cutler, M.; Walsh, T. J.; and How, J. P. 2015. Real-world reinforcement learning via multifidelity simulators. *IEEE Transactions on Robotics* 31(3):655–671.

Dann, C., and Brunskill, E. 2015. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, 2818–2826.

Grill, J.-B.; Valko, M.; and Munos, R. 2016. Blazing the trails before beating the path: Sample-efficient monte-carlo planning. In *Advances in Neural Information Processing Systems*, 4680–4688.

Ha, S., and Yamane, K. 2015. Reducing hardware experiments for model learning and policy optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2620–2626.

Hanna, J. P., and Stone, P. 2017. Grounded action transformation for robot learning in simulation. In *AAAI*, 3834–3840.

Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T.; Erez, T.; and Tassa, Y. 2015. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, 2944–2952.

Jiang, N.; Krishnamurthy, A.; Agarwal, A.; Langford, J.; and Schapire, R. E. 2017. Contextual decision processes with low Bellman rank are PAC-learnable. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 1704–1713.

Kakade, S.; Kearns, M. J.; and Langford, J. 2003. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 306–312.

Kearns, M., and Singh, S. 2002. Near-optimal reinforcement learning in polynomial time. *Machine Learning* 49(2-3):209–232.

Kearns, M.; Mansour, Y.; and Ng, A. Y. 2002. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning* 49(2-3):193–208.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 0278364913495721.

Koos, S.; Mouret, J.-B.; and Doncieux, S. 2010. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 119–126. ACM.

Krishnamurthy, A.; Agarwal, A.; and Langford, J. 2016. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, 1840–1848.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17(39):1–40.

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

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 1928–1937.

Pazis, J., and Parr, R. 2013. PAC optimal exploration in continuous space Markov Decision Processes. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Rusu, A. A.; Vecerik, M.; Rothörl, T.; Heess, N.; Pascanu, R.; and Hadsell, R. 2016. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489.