Contextual-MDPs for PAC-Reinforcement Learning with Rich Observations

Akshay Krishnamurthy *1, Alekh Agarwal †1, and John Langford ‡1

[1]Microsoft Research, New York, NY 10011

May 25, 2016

Abstract

We propose and study a new tractable model for reinforcement learning with rich observations called Contextual-MDPs, generalizing contextual bandits to sequential decision making. These models require an agent to take actions based on observations (features) with the goal of achieving long-term performance competitive with a large set of policies. To avoid barriers to sample-efficient learning associated with large observation spaces and general POMDPs, Contextual-MDPs can be summarized by a small number of hidden states and long-term rewards are predictable by a reactive function class. In this setting, we design a new reinforcement learning algorithm that engages in global exploration and analyze its sample complexity. We prove that the algorithm learns near optimal behavior after a number of episodes that is polynomial in all relevant parameters, logarithmic in the number of policies, and independent of the size of the observation space. This represents an exponential improvement over all existing alternative approaches and provides theoretical justification for reinforcement learning with function approximation.

1 Introduction

The Atari Reinforcement Learning research program [21] has highlighted a critical deficiency of practical reinforcement learning algorithms in settings with rich observation spaces: they cannot effectively solve problems that require sophisticated exploration. How can we construct Reinforcement Learning (RL) algorithms which effectively plan and plan to explore?

In RL theory, this is a solved problem for Markov Decision Processes (MDPs) [12, 5, 25]. Why do these results not apply?

An easy response is, "because the hard games are not MDPs." This may be true for some of the hard games, but it is misleading—popular algorithms like Q-learning with ϵ-greedy exploration do not even engage in minimal planning and global exploration[1] as is required to solve MDPs efficiently. MDP-optimized global exploration has also been avoided because of a polynomial dependence on the number of unique observations which is intractably large with observations from a visual sensor.

*akshay@cs.umass.edu
†alekha@microsoft.com
‡jcl@microsoft.com

[1]We use "global exploration" to distinguish the sophisticated exploration strategies required to solve an MDP efficiently from exponentially less efficient alternatives such as ϵ-greedy.

In contrast, supervised and contextual bandit learning algorithms have *no* dependence on the number of observations and at most a logarithmic dependence on the size of the underlying policy set. Approaches to RL with a weak dependence on these quantities exist [14] but suffer from an exponential dependence on the time horizon—with K actions and a horizon of H, they require K^H samples. Examples show that this dependence is necessary, although they typically require a large number of states. Can we find an RL algorithm with no dependence on the number of unique observations and a polynomial dependence on the number of actions K, the number of necessary states M, the horizon H, and the policy complexity $\log(|\Pi|)$?

To begin answering this question we consider a simplified setting with episodes of bounded length H and deterministic state transitions. We further assume that we have a function class that contains the optimal observation-action value function $Q^\star$. These simplifications make the problem significantly more tractable without trivializing the core goal of designing a Poly$(K, M, H, \log(|\Pi|)))$ algorithm. To this end, our contributions are:

1. A new class of models (Contextual-MDPs) for the design and analysis of reinforcement learning algorithms. Contextual-MDPs generalize both contextual bandits and MDPs, but, unlike Partially Observable MDPs (POMDPs), the optimal Q function in a Contextual-MDP depends only on the most recent observation rather than the entire trajectory. We further show *exponential lower bounds* on sample complexity if the optimal Q function depends on longer history or is not captured in our policy class.
2. A new reinforcement learning algorithm and a guarantee that it *PAC-learns Contextual-MDPs* (with the above assumptions) using $\mathcal{O}(MK^2H^6\log(|\Pi|))$ samples. This is done by combining ideas from contextual bandits with a novel state equality test and a global exploration technique, yielding the first Poly$(K, M, H, \log(|\Pi|))$ reinforcement learning algorithm with no dependence on the number of unique observations. Like initial contextual bandit approaches [1], the algorithm is computationally inefficient since it requires enumeration of the policy class, an aspect we hope to address in future work.

Our algorithm uses a function class to approximate future rewards, and thus lends theoretical backing for reinforcement learning with function approximation, which is the empirical state-of-the-art.

2 The Model

In this section, we introduce the episodic Contextual-MDP model, starting with basic notation. Let $H \in \mathbb{N}$ denote an episode length, $\mathcal{X} \subseteq \mathbb{R}^d$ an observation space, $\mathcal{A}$ a finite set of actions, and $\mathcal{S}$ a finite set of latent states. Let $K = |\mathcal{A}|$. We partition $\mathcal{S}$ into H disjoint groups $\mathcal{S}_1, \dots, \mathcal{S}_H$, each of size at most M. For a set P, $\Delta(P)$ denotes the set of distributions over P.

2.1 Basic Definitions

An episodic Contextual-MDP is defined by the tuple (Γ_1, Γ, D) where $\Gamma_1 \in \Delta(\mathcal{S}_1)$ denotes a starting state distribution, $\Gamma : (\mathcal{S} \times \mathcal{A}) \to \Delta(\mathcal{S})$ denotes the transition dynamics, and $D_s \in \Delta(\mathcal{X} \times [0,1]^K)$ associates a distribution over observation-reward pairs with each state $s \in \mathcal{S}$. We also use D_s to denote the marginal distribution over observations (usage will be clear from context) and use $D_{s|x}$ for the conditional distribution over reward given the observation x in state s. The marginal and conditional probabilities are referred to as $D_s(x)$ and $D_{s|x}(r)$.

We assume that the process is *layered* (also known as loop-free or acyclic) so that for a state $s_h \in \mathcal{S}_h$ and for any action $a \in \mathcal{A}$, $\Gamma(s_h, a) \in \Delta(\mathcal{S}_{h+1})$. Since the state space is partitioned into disjoint sets, the environment transitions from the state space $\mathcal{S}_1$ up to $\mathcal{S}_H$ via a sequence of actions. Layered structure allows us to avoid indexing policies and Q-functions with time, which enables more concise notation but is mathematically equivalent to an alternative reformulation without layered structure.

Each episode produces a full record of interaction $(s_1, x_1, a_1, r_1, \ldots, s_H, x_H, a_H, r_H)$ where $s_1 \sim \Gamma_1$, $s_h \sim \Gamma(s_{h-1}, a_{h-1})$, $(x_h, r_h) \sim D_{s_h}$ and all actions a_h are chosen by the learning agent. The record of interaction observed by the learner is $(x_1, a_1, r_1(a_1), \ldots, x_H, a_H, r_H(a_H))$ and at time point h, the learner may use all observable information up to and including x_h to select a_h. Notice that all state information and rewards for alternative actions are unobserved by the learning agent.

Over the course of an episode, the reward obtained by the learner is $\sum_{h=1}^{H} r_h(a_h)$, and the goal is to maximize the expected cumulative reward, $R = \mathbb{E}[\sum_{h=1}^{H} r_h(a_h)]$, where the expectation accounts for all randomness in the model and the learner. We assume that almost surely $\sum_{h=1}^{H} r_h(a_h) \in [0, 1]$ for any action sequence.

In this model, the optimal expected reward achievable can be computed recursively as

$$V^\star = \mathbb{E}_{s \sim \Gamma_1}[V^\star(s)] \quad \text{with} \quad V^\star(s) = \mathbb{E}_{x \sim D_s} \max_a \mathbb{E}_{r \sim D_{s|x}} \left[r(a) + \mathbb{E}_{s' \sim \Gamma(s,a)} V^\star(s') \right]. \tag{1}$$

As the base case, we assume that for states $s \in \mathcal{S}_H$, all actions transition to a terminal state s_{H+1} with $V^\star(s_{H+1}) = 0$. For each (s, x) pair such that $D_s(x) > 0$ we also define a $Q^\star$ function as

$$Q_s^\star(x, a) = \mathbb{E}_{r \sim D_{s|x}} \left[r(a) + \mathbb{E}_{s' \sim \Gamma(s,a)} V^\star(s') \right]. \tag{2}$$

This function captures the optimal choice of action given this (state, observation) pair and therefore encodes optimal behavior in the model. With no further assumptions, the above model is a *layered episodic Partially Observable Markov Decision Process* (POMDP). Both learning and planning are notoriously challenging in POMDPs, because the optimal policy may depend on the entire history of interaction and the complexity of learning such a policy grows exponentially with H (see e.g. Kearns et al. [14] as well as Propositions 1 and 2 below). A Contextual-MDP avoids this statistical barrier with two assumptions: (a) we consider only reactive policies, and (b) we assume access to a class of functions that can realize the $Q^\star$ function. Both of these assumptions are implicit in the empirical state of the art reinforcement learning results. We describe both assumptions in detail before formally defining the Contextual-MDP model.

Reactive Policies: One approach taken by some prior theoretical work is to consider *reactive* (or memoryless) policies that use only the current observation to select an action [20, 3]. Memorylessness is slightly generalized in the recent empirical advances in reinforcement learning, which typically employ policies that depend only on the few most recent observations [21].

A reactive policy $\pi : \mathcal{X} \to \mathcal{A}$ is a strategy for navigating the search space by taking actions $\pi(x)$ given observation x. The expected reward for a policy is defined recursively through

$$V(\pi) = \mathbb{E}_{s \sim \Gamma_1}[V(s, \pi)] \quad \text{and} \quad V(s, \pi) = \mathbb{E}_{(x,r) \sim D_s} \left[r(\pi(x)) + \mathbb{E}_{s' \sim \Gamma(s, \pi(x))} V(s', \pi) \right].$$

A natural learning goal is to identify a reactive policy with maximal value $V(\pi)$ from a given collection of policies Π. Unfortunately, even when restricting to reactive policies, learning in POMDPs requires exponentially many samples, as we show in the next lower bound.

Proposition 1. *Fix $H, K \in \mathbb{N}$ with $K \geq 2$ and $\epsilon \in (0, \sqrt{1/8})$. There exists a layered episodic POMDP with time horizon H, K actions, and $2H$ total states; a class Π of reactive policies with $|\Pi| = K^H$; and a universal constant $c > 0$ such that, for any algorithm and any $T \leq cK^H/\epsilon^2$, the probability that the algorithm outputs a policy $\hat{\pi}$ with $V(\hat{\pi}) > \max_{\pi \in \Pi} V(\pi) - \epsilon$ after collecting T trajectories is at most $2/3$.*

This lower bound precludes a Poly$(K, M, H, \log(|\Pi|))$ sample complexity bound for learning reactive policies in general POMDPs as $\log(|\Pi|) = H \log(K)$ in the construction, but the number of samples required is exponential in H. The lower bound instance provides essentially no instantaneous feedback and therefore forces the agent to reason over K^H paths independently.

Predictability of $Q^\star$: The assumption underlying the empirical successes in reinforcement learning is that the $Q^\star$ function can be well-approximated by some large set of functions $\mathcal{F}$. To formalize this assumption, note that for some POMDPs, we may be able to write $Q^\star$ as a function of the observed history of interaction $(x_1, a_1, r_1(a_1), \ldots, x_h)$ at time h. For example, this is always true in deterministic-transition POMDPs, since the sequence of previous actions encodes the state and $Q^\star$ as in Eq. (2) depends only on the state, the current observation, and the proposed action. In the *realizable* setting, we have access to a collection of functions $\mathcal{F}$ mapping the observed history to $[0, 1]$, and we assume that $Q^\star \in \mathcal{F}$.

Unfortunately, even with realizability, learning in POMDPs can require exponentially many samples.

Proposition 2. *Fix $H, K \in \mathbb{N}$ with $K \geq 2$ and $\epsilon \in (0, \sqrt{1/8})$. There exists a layered episodic POMDP with time horizon H, K actions, and $2H$ total states; a class of predictors $\mathcal{F}$ with $|\mathcal{F}| = K^H$ and $Q^\star \in \mathcal{F}$; and a universal constant $c \geq 0$ such that, for any algorithm and any $T \leq cK^H/\epsilon^2$, the probability that the algorithm outputs a policy $\hat{\pi}$ with $V(\hat{\pi}) > V^\star - \epsilon$ after collecting T trajectories is at most $2/3$.*

As with Proposition 1, this lower bound precludes a $\text{Poly}(K, M, H, \log(|\Pi|))$ sample complexity bound for learning POMDPs with realizability. The lower bound shows that even with realizability, the agent may have to reason over K^H paths independently since the functions can depend on the entire history. Proofs of both lower bounds here are deferred to Appendix A.

Both lower bounds use POMDPs with deterministic transitions and an extremely small observation space. Consequently, even learning in deterministic-transition POMDPs requires further assumptions.

2.2 The Contextual-MDP Model

As we have seen, neither restricting to reactive policies, nor imposing realizability enable tractable learning in POMDPs on their own. Combined however, we will see that sample-efficient learning is possible, and the combination of these two assumptions is precisely how we characterize Contextual-MDPs. Specifically, a Contextual-MDP is a POMDP for which $Q^\star$ can be realized by a predictor that uses only the current observation and proposed action.

Definition 1 (*Contextual-MDP*)**.** Let $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \Gamma_1, \Gamma, D)$ be a layered episodic POMDP. Let $Q^\star$ be correspondingly defined as in Equation 2. The POMDP is called a Contextual-MDP if for all $x \in \mathcal{X}, a \in \mathcal{A}$ and any two states s, s' such that $D_s(x), D_{s'}(x) > 0$ we have $Q^\star_s(x, a) = Q^\star_{s'}(x, a)$.

The restriction on the $Q^\star$ function implies both that the optimal policy is reactive and also that the optimal predictor of long-term reward depends only on the current observation. In the following section, we describe how this condition relates to other reinforcement learning models in the literature. However, we first justify the assumptions with a natural example.

Example 1 (*Disjoint observations*)**.** The simplest example of a Contextual-MDP is one where each state s can be identified with a subset $\mathcal{X}_s$ with $D_s(x) > 0$ only for $x \in \mathcal{X}_s$ and where $\mathcal{X}_s \cap \mathcal{X}_{s'} = \emptyset$ when $s \neq s'$. In this case, a realized observation x uniquely identifies the underlying state s so that Contextual-MDP assumption trivially holds. On the other hand, this underlying mapping from s to $\mathcal{X}_s$ is unknown to the learning agent so the problem cannot be easily reduced to a classical tabular MDP with a small number of states. This example might appear limited at the first glance, but it is quite natural in several robotic and navigation tasks. In such scenarios, the visual signals are rich enough to identify the agent's position (and hence state) uniquely, but learning this mapping is challenging.

More generally, Contextual-MDPs provide a framework to reason about reinforcement learning with function approximation. This is highly desirable as such approaches are the empirical state-of-the-art, but the limited supporting theory provides little advice on systematic global exploration.

2.3 Connections to Other Models and Techniques

Our model is closely related to several well-studied models in the literature, namely:

Contextual Bandits: If $H = 1$, then Contextual-MDPs reduce to stochastic contextual bandits [15, 7], a well-studied simplification of the general reinforcement learning problem. The main difference is that the choice of action *does not* influence the future observations, meaning there is only one state, and algorithms do not need to perform long-term planning to obtain low sample complexity.

Markov Decision Processes: If $\mathcal{X} = \mathcal{S}$ and $D_s(x)$ for each state s is concentrated on s, then Contextual-MDPs reduce to MDPs, which can be efficiently solved by tabular approaches [12, 5, 25]. The key differences in our setting are that the observation space $\mathcal{X}$ is extremely large or infinite and the underlying state is unobserved, so tabular methods are not viable and algorithms need to *generalize* across observations.

When the number of states is large, existing methods typically require exponentially many samples such as the $\mathcal{O}(K^H)$ result of Kearns et al. [14]. Others depend poorly on the complexity of the policy set or scale linearly in the size of a covering over the state space [11, 9]. Lastly, policy gradient methods avoid dependence on size of the state space, but do not achieve global optimality [26, 10] in theory and in practice, unlike our algorithm which is guaranteed to find the globally optimal policy.

POMDPs: By definition a Contextual-MDP is a Partially Observable Markov Decision Process (POMDP) where the $Q^\star$ function is consistent across states. This restriction implies that the learning algorithm does not have to reason over belief states as is required in POMDPs.

There are some sample complexity guarantees for learning in arbitrarily complex POMDPs, but the bounds we are aware of are quite weak as they scale linearly with $|\Pi|$ [13, 19].

Predictive State Representations (PSRs): PSRs [18] encode states as a a collection of *tests*, a test being a sequence of (a, x) pairs observed in the history. Representationally, PSRs are even more powerful than POMDPs [24] which make them also more general than Contextual-MDPs. However, we are not aware of finite sample bounds for learning PSRs.

State Abstraction: State abstraction (see [17] for a survey) focuses on understanding what optimality properties are preserved in an MDP after the state space is compressed. While Contextual-MDPs do have a small number of underlying states, they do not necessarily admit non-trivial state abstractions that are easy to discover (i.e. that do not amount to learning the optimal behavior) as the optimal behavior can depend on the observation in an arbitrary manner. Furthermore, most sample complexity results cannot search over large abstraction sets (see e.g. Jiang et al. [8]), limiting their scope.

Function Approximation: Our approach uses function approximation to address the generalization problem implicit in Contextual-MDPs. Function approximation is the empirical state-of-the-art in reinforcement learning [21], but theoretical analysis has been quite limited. Several authors have studied linear function approximation (See [27, 22]) but none of these results give finite sample bounds, as they do not address the exploration question. Baird [4] analyzes more general function approximation for predicting the value function in a Markov Chain, but does not show convergence when the agent is also selecting actions. More closely to our work, Li and Littman [16] do give finite sample bounds for RL with function approximation, but they assume access to a "Knows-what-it-knows" (KWIK) oracle, which cannot exist even for simple problems. We are not aware of finite sample results for approximating $Q^\star$ with a function class, which is precisely what we do here.

3 The Result

We consider the task of Probably Approximately Correct (PAC) learning of Contextual-MDPs. In this task, we are given a class of predictors $\mathcal{F} \subset (\mathcal{X} \times \mathcal{A} \to [0,1])$ of size $|\mathcal{F}| = N$, with the assumption that

Algorithm 1 CTXMDPLEARN $(\mathcal{F}, \epsilon, \delta)$

1: $\mathcal{F} \leftarrow$ DFS-LEARN$(\varnothing, \mathcal{F}, \epsilon, \delta/2)$.
2: Let $\hat{V}^\star$ be a Monte Carlo estimate of $V^f(\varnothing, \pi_f)$ for any $f \in \mathcal{F}$. (Use existing data, see Eq. (3))
3: Set $\epsilon_{\text{demand}} = \epsilon/2, n_1 = \frac{32\log(12MH/\delta)}{\epsilon^2}$ and $n_2 = \frac{8\log(6MH/\delta)}{\epsilon}$.
4: **while** `true` **do**
5: Fix a regressor $f \in \mathcal{F}$.
6: Collect n_1 trajectories according to π_f and estimate $V(\pi_f)$ via Monte-Carlo estimate $\hat{V}(\pi_f)$.
7: If $|\hat{V}(\pi_f) - \hat{V}^\star| \le \epsilon_{\text{demand}}$, return π_f.
8: Otherwise update $\mathcal{F}$ by calling DFS-LEARN $(p, \mathcal{F}, \epsilon, \frac{\delta}{6MH^2 n_2})$ on each of the $H-1$ prefixes p of each of the first n_2 paths collected in step 6.
9: **end while**

Algorithm 2 DFS-LEARN $(p, \mathcal{F}, \epsilon, \delta)$

Set $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$ and $\epsilon_{\text{test}} = 20(H - |p| - 5/4)\sqrt{K}\phi$.
for $a \in \mathcal{A}$, if not CONSENSUS$(p \circ a, \mathcal{F}, \epsilon_{\text{test}}, \phi, \frac{\delta/2}{MKH})$ **do**
$\mathcal{F} \leftarrow$ DFS-LEARN$(p \circ a, \mathcal{F}, \epsilon, \delta)$.
end for
Collect $n_{\text{train}} = \frac{24}{\phi^2}\log\left(\frac{8MHN}{\delta}\right)$ observations (x_i, a_i, r_i) where $(x_i, r_i') \sim D_p$, a_i is chosen uniformly at random, and $r_i = r_i'(a_i)$.
Return $\left\{f \in \mathcal{F} : \tilde{R}(f) \le \min_{f' \in \mathcal{F}} \tilde{R}(f') + 2\phi^2 + \frac{22\log(4MHN/\delta)}{n_{\text{train}}}\right\}$, $\tilde{R}(f)$ defined in Eq. 4.

$Q^\star = f^\star \in \mathcal{F}$. We identify each predictor f with a policy $\pi_f(x) = \text{argmax}_a f(x, a)$. Notice that the optimal policy for the Contextual-MDP is $\pi_{f^\star}$ which satisfies $V(\pi_{f^\star}) = V^\star$.

Given $\mathcal{F}$, we say that an algorithm PAC learns a Contextual-MDP if for any $\epsilon, \delta \in (0, 1)$, the algorithm outputs a policy $\hat{\pi}$ with $V(\hat{\pi}) \ge V^\star - \epsilon$ with probability at least $1 - \delta$. The *sample complexity* is a function $n : (0, 1)^2 \to \mathbb{N}$ such that for any $\epsilon, \delta \in (0, 1)$, the algorithm returns an ϵ-suboptimal policy with probability at least $1 - \delta$ using at most $n(\epsilon, \delta)$ episodes. We refer to a Poly$(M, K, H, \epsilon, \log(N), \log(1/\delta))$ sample complexity bound as polynomial in all relevant parameters. Notably, there should be no dependence on $|\mathcal{X}|$, which may be infinite.

Our algorithm operates on Contextual-MDPs with one additional assumptions.

Assumption 1 (*Deterministic Transitions*)**.** We assume that the transition model is deterministic. This means that the starting distribution Γ_1 is a point-mass on some state s_1 and $\Gamma : (\mathcal{S} \times \mathcal{A}) \to \mathcal{S}$.

Even with deterministic transitions, PAC-learning Contextual-MDPs requires systematic global exploration that is unaddressed in previous work. Recall that the lower bound constructions for Propositions 1 and 2 actually use deterministic transition POMDPs. Therefore, deterministic transitions combined with either the reactive or the realizability assumption still precludes tractable learning.

3.1 The Algorithm

Before turning to the algorithm, it is worth clarifying some additional notation. Since we are focused on the deterministic transition setting, it is natural to think about the Contextual-MDP as an exponentially large search tree with fan-out K and depth H. Each node in the search tree is labeled with an (unobserved) state

Algorithm 3 CONSENSUS$(p, \mathcal{F}, \epsilon_{\text{test}}, \phi, \delta)$

1: Set $n_{\text{test}} = \frac{2}{\phi^2}\log(2N/\delta)$. Collect n_{test} observations $x_i \sim D_p$.
2: Compute for each function, $\hat{V}^f(p, \pi_f) = \frac{1}{n_{\text{test}}}\sum_{i=1}^{n_{\text{test}}} f(x_i, \pi_f(x_i))$.
3: Return $\mathbf{1}\left[|\hat{V}^f(p,\pi_f) - \hat{V}^g(p,\pi_g)| \le \epsilon_{\text{test}}\ \forall f, g \in \mathcal{F}\right]$.

$s \in \mathcal{S}$, and each edge is labeled with an action $a \in \mathcal{A}$, both of which are consistent with the transition model. A path $p \in \mathcal{A}^\star$ corresponds to a sequence of actions from the root of the search tree, and we also use p to denote the state reached after executing the corresponding sequence of actions from the root. We often call such a path a *roll-in*, in line with existing terminology. For a roll-in p, we use $p \circ a$ to denote a path formed by executing all actions in p and then executing action a, and we use $|p|$ to denote the length of the path. Let $\varnothing$ denote the empty path, which corresponds to the root of the search tree.

The pseudocode for the algorithm, which we call CTXMDPLEARN, is displayed in Algorithm 1. The algorithm has two main components: a depth-first-search routine with a learning step (step 6 in Algorithm 2) and an on-demand exploration technique (steps 5-8 in Algorithm 1). The high-level idea of the algorithm is to eliminate regression functions that do not meet Bellman-like consistency properties of the $Q^\star$ function. We now describe both components and their properties in detail.

The DFS routine: When the DFS routine, displayed in Algorithm 2, is run at some path p, we first decide whether to recursively expand the descendants $p \circ a$ by performing a *consensus test*. Given a path p', this test, displayed in Algorithm 3, computes estimates of *value predictions*,

$$V^f(p', \pi_f) = \mathbb{E}_{x \sim D_{p'}} f(x, \pi_f(x)), \tag{3}$$

for all the surviving regressors. These value predictions are easily estimated by collecting many observations after rolling in to p' and using empirical averages (See line 2 in Algorithm 3). If all the functions agree on this value for p' the DFS need not visit this path.

After the recursive calls, the DFS routine performs the *elimination step* (line 6). When this step is invoked at path p, the algorithm collects n_{train} observations (x_i, a_i, r_i) where $(x_i, r'_i) \sim D_p$, a_i is chosen uniformly at random, and $r_i = r'_i(a_i)$ and eliminates regressors that have high empirical risk,

$$\tilde{R}(f) = \frac{1}{n_{\text{train}}}\sum_{i=1}^{n_{\text{train}}}(f(x_i, a_i) - r_i - \hat{V}^f(p \circ a_i, \pi_f))^2. \tag{4}$$

The value prediction estimates $\hat{V}^f(p \circ a_i, \pi_f)$ from the consensus tests are reused here (Line 2 in Algorithm 3).

Intuition for DFS: This regression problem is motivated by the realizability assumption and the definition of $Q^\star$ in Eq. 2, which imply that at path p,

$$f^\star(x, a) = \mathbb{E}_{r \sim D_{p|x}} r(a) + V(p \circ a, \pi_{f^\star}) = \mathbb{E}_{r \sim D_{p|x}} r(a) + \mathbb{E}_{x' \sim D_{p \circ a}} f^\star(x', \pi_{f^\star}(x')). \tag{5}$$

Thus $f^\star$ is consistent between its estimate at the current state s and the future state $s' = \Gamma(s, a)$.

The regression problem (4) is essentially a finite sample version of this identity. However, some care must be taken as the target for the regression function f includes $V^f(p \circ a, \pi_f)$, which is f's value prediction for the future. The fact that the target differs across functions can cause instability in the regression problem, as some targets may have substantially lower variance than $f^\star$'s. To ensure correct behavior, we must obtain high-quality future value prediction estimates, and so, we re-use the Monte-Carlo estimates $\hat{V}^f(p \circ a, \pi_f)$

in Eq. 3 from the consensus tests. With these high-quality estimates, step 6 ensure that the algorithm retains only good regressors, which induce good policies.

Invoking elimination at p produces a reduced function set with many desirable properties, provided consensus is reached at the descendants and the estimates $\hat{V}^f(p \circ a, \pi_f)$ are accurate. When these conditions hold, we prove that (i) $f^\star$ is not eliminated, (ii) consensus is reached at p, and (iii) surviving policies choose good actions at p. Property (ii) controls the sample complexity, since consensus tests at state s return true once elimination has been invoked on s, so DFS avoids exploring the entire search space. Property (iii) leads to the PAC-bound, since if we have run the elimination step on all states visited by a surviving policy, then that policy must be near-optimal.

DFS invokes the elimination step and the consensus test in tandem, which ensures that the two preconditions for elimination are always satisfied. First, since we ran a consensus test on each descendant, we can guarantee accurate estimation of the value predictions $V^f(p \circ a, \pi_f)$. Second, for each descendant, either we invoked the elimination step, inductively guaranteeing consensus, or the consensus test returned true. Thus, we always execute the elimination step correctly.

To bound the sample complexity of the DFS routine, since there are M states per level and the consensus test returns true once elimination has been performed, we know that the DFS does not visit a large fraction of the search tree. Specifically, this means DFS is invoked on at most MH nodes in total, so we run elimination at most MH times, and we perform at most MKH consensus tests. Each of these operations requires polynomially many samples.

The elimination step is inspired by the RegressorElimination algorithm of Agarwal et. al [1] for contextual bandit learning in the realizable setting. Apart from the differences between the regression problem, motivated by the discussion above, the other main difference between the algorithms is the choice of action-selection distribution. RegressorElimination must carefully choose actions to balance exploration and exploitation which leads to an optimal regret bound. In contrast, we are pursuing a PAC-guarantee here, for which it suffices to focus exclusively on exploration.

On-demand Exploration: The other component of the algorithm is an *on-demand exploration* technique, the second half of Algorithm 1 (steps 5-8). At each iteration, we select a policy π_f and estimate its value via Monte Carlo sampling. If the policy has sub-optimal value, we invoke the DFS procedure on many of the paths visited. If the policy has near-optimal value, we have found a good policy, so we are done. This procedure requires an accurate estimate of the optimal value, which we obtain by invoking the DFS routine at the root, since it guarantees that all surviving regressors agree with $f^\star$'s value on the starting state distribution. $f^\star$'s value is precisely the optimal value.

Intuition for On-demand Exploration: Running the elimination step at some path p ensures that all surviving regressors take good actions at p, in the sense that taking one action according to any surviving policy and then behaving optimally thereafter achieves near-optimal reward for path p. Unfortunately, this does not ensure that all surviving policies achieve near-optimal reward, because they may take highly sub-optimal actions after the first one. On the other hand, if a surviving policy π_f visits only states for which the elimination step has been invoked, then it must have near-optimal reward. More precisely, letting L denote the set of states for which the elimination step has been invoked (the "learned" states), we prove that any surviving π_f satisfies,

$$V^\star - V(\pi_f) \leq \epsilon/8 + \mathbb{P}\left[\pi_f \text{ visits } \bar{L}\right]$$

Thus, if π_f is highly sub-optimal, it must visit some unlearned states with substantial probability. By calling DFS-LEARN on the paths visited by π_f, we ensure that the elimination step is run on at least one unlearned states. Since there are only MH distinct states and each non-terminal iteration ensures training on an unlearned state, the algorithm must terminate and output a near-optimal policy.

Computationally, the running time of the algorithm may be $O(N^2)$, since eliminating regression functions according to Eq. 4 may require enumerating over the class and the consensus function further enumerates over pairs of functions in the class. This may be intractably slow for rich function classes, but our focus is on statistical efficiency, so we ignore computational issues here.

3.2 The PAC Guarantee

The main result of this paper certifies that CTXMDPLEARN can PAC-learn deterministic transition Contextual-MDPs with polynomial sample complexity.

Theorem 1 (PAC bound). *For any $(\epsilon, \delta) \in (0, 1)$ and any Contextual-MDP (Definition 1) with deterministic transitions for which $Q^\star \in \mathcal{F}$, with probability at least $1 - \delta$, the policy π returned by* CTXMDPLEARN *is at most ϵ-suboptimal. Moreover, the number of episodes required is at most,*

$$\tilde{\mathcal{O}}\left(\frac{MH^6K^2}{\epsilon^3}\log(N/\delta)\log(1/\delta)\right).$$

This result uses the $\tilde{\mathcal{O}}$ notation to suppress logarithmic dependence in all parameters except for N and δ. The precise dependence on all parameters can be recovered by examination of our proof and is shortened here simply for clarity. See Appendix C for the full proof of the result.

This theorem states that CTXMDPLEARN produces a policy that is at most ϵ-suboptimal using a number of episodes that is polynomial in all relevant parameters. To our knowledge, this is the first polynomial sample complexity bound for reinforcement learning with infinite observation spaces, without prohibitively strong assumptions. We also believe this is the first finite-sample guarantee for RL with function approximation.

Since Contextual-MDPs generalize both contextual bandits and MDPs, it is worth comparing the sample complexity bounds.

1. In contextual bandits, we have $M = H = 1$ so that the sample complexity of CTXMDPLEARN is $\tilde{\mathcal{O}}(\frac{K^2}{\epsilon^3}\log(N/\delta)\log(1/\delta))$, in contrast with known $\tilde{\mathcal{O}}(\frac{K}{\epsilon^2}\log(N/\delta))$ results.
2. Prior results establish the sample complexity for learning layered episodic MDPs with deterministic transitions is $\tilde{\mathcal{O}}(\frac{MK\text{poly}(H)}{\epsilon^2}\log(1/\delta))$ [6, 23].

Both comparisons show our sample complexity bound may be suboptimal in its dependence on K and ϵ. Looking into our proof, the additional factor of K comes from collecting observations to estimate the value of future states, while the additional $1/\epsilon$ factor arises from trying to identify a previously unexplored state. In contextual bandits, these issues do not arise since there is only one state, while in tabular MDPs they can be trivially resolved as the states are observed. Thus, with minor modifications, DFS-LEARN can avoid these dependencies for both special cases. In addition, our bound disagrees with the MDP results in the dependence on the policy complexity $\log(N)$; which we believe is unavoidable when working with rich observation spaces.

Finally, our bound depends on the number of states M in the worst case, but the algorithm actually uses a more refined notion. Since the states are unobserved, the algorithm considers two states distinct only if they have reasonably different value functions, meaning learning on one does not lead to consensus on the other. Thus a more distribution-dependent analysis defining states through the function class is a promising avenue for future work.

4 Discussion

This paper introduces Contextual-MDPs, a new model in which it is possible to design and analyze principled reinforcement learning algorithms engaging in global exploration. As a first step, we develop CTXMDPLEARN and show that it learns near-optimal behavior in deterministic-transition Contextual-MDPs with polynomial sample complexity. This is the first polynomial sample complexity bound for RL with general function approximation. However, there are major open questions:

1. Do polynomial sample bounds for Contextual-MDPs with stochastic transitions exist?
2. Can we design an algorithm for learning Contextual-MDPs that is both computationally and statistically efficient? The sample complexity of our algorithm is logarithmic in the size of the function class $\mathcal{F}$ but uses an intractably slow enumeration of these functions.

Good answers to both of these questions may yield new practical reinforcement learning algorithms.

Acknowledgements

We thank Akshay Balsubramani and Hal Daumé III for formative discussions, and we thank Tzu-Kuo Huang for a careful reading of an early draft of this paper.

References

[1] Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert E Schapire. Contextual bandit learning with predictable rewards. In *AISTATS*, 2012.

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SICOMP*, 2002.

[3] Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. Reinforcement learning of pomdp's using spectral methods. *arXiv:1602.07764*, 2016.

[4] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *ICML*, 1995.

[5] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 2003.

[6] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *NIPS*, 2015.

[7] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *UAI*, 2011.

[8] Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *ICML*, 2015.

[9] Nicholas K Jong and Peter Stone. Model-based exploration in continuous state spaces. In *Abstraction, Reformulation, and Approximation*, 2007.

[10] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, 2002.

[11] Sham Kakade, Michael Kearns, and John Langford. Exploration in metric state spaces. In *ICML*, 2003.

[12] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *MLJ*, 2002.

[13] Michael J Kearns, Yishay Mansour, and Andrew Y Ng. Approximate planning in large pomdps via reusable trajectories. In *NIPS*, 1999.

[14] Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *MLJ*, 2002.

[15] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2008.

[16] Lihong Li and Michael L Littman. Reducing reinforcement learning to kwik online regression. *Ann. Math AI*, 2010.

[17] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *ISAIM*, 2006.

[18] Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *NIPS*, 2001.

[19] Yishay Mansour. Reinforcement learning and mistake bounded algorithms. In *COLT*, 1999.

[20] Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. In *UAI*, 1999.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[22] Theodore J Perkins and Doina Precup. A convergent form of approximate policy iteration. In *NIPS*, 2002.

[23] Spyros Reveliotis and Theologos Bountourelis. Efficient pac learning for episodic tasks with acyclic state spaces. *DEDS*, 2007.

[24] Satinder Singh, Michael R James, and Matthew R Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *UAI*, 2004.

[25] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *ICML*, 2006.

[26] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1999.

[27] John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE TAC*, 1997.

A The Lower Bounds

Theorem 2 (Lower bound for best arm identification in stochastic bandits)**.** *For any $K \geq 2$ and $\epsilon \leq \sqrt{1/8}$, there exists a multi-armed bandit problem for which the best arm $i^\star$ is ϵ better than all others, but for which any algorithm that produces an estimate $\hat{i}$ of the best arm must have $\mathbb{P}[\hat{i} \neq i^\star] \geq 1/3$ unless the number of samples collected T is at least $\frac{K}{72\epsilon^2}$.*

Proof. The proof is essentially obtained from the regret lower bound for stochastic multi-armed bandits from Auer et al. [2]. Since we want the lower bound for best arm identification instead of regret, we include a full proof for completeness.

Following Auer et al. [2], we define the lower bound instance as a random mixture of a family of multi-armed bandit problems with K arms each. There are K problems in the family, and each one is parametrized by the optimal arm $i^\star$. For the $i^{\star\text{th}}$ problem, arm $i^\star$ produces rewards drawn from $\text{Ber}(1/2+\epsilon)$ while all other arms produce rewards $\text{Ber}(1/2)$. Let $\mathbb{P}_{i^\star}$ denote the reward distribution for the bandit problem with optimal arm $i^\star$, so that $\mathbb{P}_{i^\star}(\cdot|a = i^\star) = \text{Ber}(1/2+\epsilon)$ and $\mathbb{P}_{i^\star}(\cdot|a \neq i^\star) = \text{Ber}(1/2)$. Let $\mathbb{P}_0$ denote the reward distribution where all arms receive $\text{Ber}(1/2)$ rewards.

Since the environment is stochastic, any randomized algorithm can be thought of as a distribution over deterministic ones, and it therefore suffices to consider only deterministic algorithms. A deterministic algorithm can be specified as a sequence of mappings $\psi_t : \{0,1\}^t \to [K]$ with the interpretation of ψ_T as the estimate of the best arm. Note that ψ_0 is the first arm chosen, which does not depend on any of the observations. The algorithm can be specified this way since the sequence of actions played can be inferred by the sequence of observed rewards. Let $\mathbb{P}_{i^\star,\psi}$ denote the distribution over all T rewards when $i^\star$ is the optimal arm and actions are selected according to ψ. We are interested in bounding the error event $\mathbb{P}_{i^\star,\psi}[\psi_T \neq i^\star]$.

We first prove,

$$\mathbb{P}_{i^\star,\psi}[\psi_T = i^\star] - \mathbb{P}_{0,\psi}[\psi_T = i^\star] \leq \frac{1}{2}\sqrt{\mathbb{E}_{0,\psi}[N_{i^\star}] \log \frac{1}{1-4\epsilon^2}}$$

where N_i is the number of times ψ plays action i over the course of T rounds. N_i is a random variable since it depends on the sequence of observations, and here we take expectation with respect to $\mathbb{P}_0$.

To prove this statement, notice that,

$$|\mathbb{P}_{i^\star,\psi}[\psi_T = i^\star] - \mathbb{P}_{0,\psi}[\psi_T = i^\star]| \leq \|P_{i^\star,\psi} - P_{0,\psi}\|_{\text{TV}} \leq \sqrt{\frac{1}{2} KL(P_{0,\psi}||P_{i^\star,\psi})} \,.$$

The first inequality is by definition of the total variation distance, while the second is Pinsker's inequality. We are left to bound the KL divergence. To do so, we introduce notation for sequences. For any $t \in \mathbb{N}$, we use $r_{1:t} \in \{0,1\}^t$ to denote a binary sequence of length t. The KL divergence is

$$\begin{aligned}
KL(P_{0,\psi}||P_{i^\star,\psi}) &= \sum_{r_{1:T}\in\{0,1\}^T} P_{0,\psi}(r_{1:T}) \log\left(\frac{P_{0,\psi}(r_{1:T})}{P_{i^\star,\psi}(r_{1:T})}\right) \\
&= \sum_{t=1}^{T} \sum_{r_{1:t}\in\{0,1\}^t} P_{0,\psi}(r_{1:t}) \log\left(\frac{P_{0,\psi}(r_t|r_{1:t-1})}{P_{i^\star,\psi}(r_t|r_{1:t-1})}\right) \\
&= \sum_{t=1}^{T} \sum_{r_{1:t-1}:\psi_{t-1}(r_{1:t-1})=i^\star} P_{0,\psi}(r_{1:t-1}) \left(\sum_{x\in\{0,1\}} P_{0,\psi}(x) \log\left(\frac{P_{0,\psi}(x|\psi_{t-1}=i^\star)}{P_{i^\star,\psi}(x|\psi_{t-1}=i^\star)}\right)\right) \\
&= \sum_{t=1}^{T} \sum_{r_{1:t-1}:\psi_{t-1}(r_{1:t-1})=i^\star} P_{0,\psi}(r_{1:t-1}) \left(\frac{1}{2}\log\left(\frac{1/2}{1/2-\epsilon}\right) + \frac{1}{2}\log\left(\frac{1/2}{1/2+\epsilon}\right)\right) \\
&= \left(-\frac{1}{2}\log(1-4\epsilon^2)\right) \sum_{t=1}^{T} \sum_{r_{1:t-1}:\psi_{t-1}(r_{1:t-1})=i^\star} P_{0,\psi}(r_{1:t-1}) \\
&= \left(-\frac{1}{2}\log(1-4\epsilon^2)\right) \sum_{t=1}^{T} \mathbb{P}_{0,\psi}[\psi_t = i^\star]
\end{aligned}$$

To arrive at the second line we use the chain rule for KL-divergence. The third line is based on the fact that if $\psi_{t-1} \neq i^\star$, then the log ratio is zero, since the two conditional distributions are identical. The remaining lines are straightforward. This proves the sub-claim, which follows the same argument as as Auer et. al [2].

To prove the final result, we take expectation over the problem $i^\star$.

$$\begin{aligned}
\frac{1}{K}\sum_{i^\star=1}^{K} \mathbb{P}_{i^\star,\psi}[\psi_T = i^\star] &\leq \frac{1}{K}\sum_{i^\star=1}^{K} \mathbb{P}_{0,\psi}[\psi_T = i^\star] + \frac{1}{2K}\sum_{i^\star=1}^{K}\sqrt{\mathbb{E}_{0,\psi}[N_{i^\star}]\log\frac{1}{1-4\epsilon^2}} \\
&\leq \frac{1}{K} + \frac{1}{2}\sqrt{\frac{-\log(1-4\epsilon^2)}{K}\mathbb{E}_{0,\psi}\sum_{i^\star=1}^{K} N_{i^\star}} \leq \frac{1}{K} + \frac{1}{2}\sqrt{\frac{-\log(1-4\epsilon^2)T}{K}}
\end{aligned}$$

If $4\epsilon^2 \leq 1/2$ then $-\log(1-4\epsilon^2) \leq 8\epsilon^2$. This follows by the Taylor expansion of $-\log(1-x)$,

$$-\log(1-x) = \sum_{i=1}^{\infty}\frac{x^i}{i} \leq x\left(\sum_{i=0}^{\infty}\frac{2^{-i}}{i+1}\right) \leq x\sum_{i=0}^{\infty} 2^{-i} = 2x$$

The inequality here uses the assumption that $x \leq 1/2$.

Thus, whenever $\epsilon \leq \sqrt{1/8}$ and $T \leq \frac{K}{72\epsilon^2}$, this number is smaller than $2/3$, since we restrict to the cases where $K \geq 2$. This is the success probability, so the failure probability is at least $1/3$, which proves the result. □

A.1 The construction

Here we design a family of POMDPs that we used for both lower bounds. As with multi-armed bandits above, the lower bound will be realized by using a POMDP that places a uniform distribution over this family of problems. Fix H and K and pick a single $x_h \in \mathcal{X}$ for each level $h \in [H]$ so that $x_h \neq x_{h'}$ for all pairs $h \neq h'$. For each level there are two states g_h and b_h for "good" and "bad." The observation marginal distribution $D_{g_h} = D_{b_h}$ is concentrated on x_h for each level h, so the observations provide no information about the underlying state. Rewards for all levels except for $h = H$ are zero.

Each POMDPs in the family corresponds to a path $p^\star = (a_1^\star, \ldots, a_H^\star) \in K^H$. The transition function for the POMDP corresponding to the path $p^\star$ is,

$$\begin{aligned} \Gamma(g_h, a_h^\star) &= g_{h+1} \\ \Gamma(g_h, a) &= b_{h+1} \text{ if } a \neq a_h^\star \\ \Gamma(b_h, a) &= b_{h+1} \;\forall\, a \end{aligned}$$

The reward is drawn from $\mathrm{Ber}(1/2 + \epsilon)$ if the last state is g_H and if the last action is $a_H^\star$. For all other outcomes the reward is drawn from $\mathrm{Ber}(1/2)$.

Clearly all of the models in this family are distinct, and there are K^H such models. Moreover, since the observations x_h provide no information and only the final reward is not identically zero, no information is received until the full sequence of actions is selected. More formally, for any two policies π, π', the KL divergence between the distributions of observations and rewards produced by the two policies is exactly the KL divergence between the final rewards produced by the two policies. In terms of the final rewards, the problem is equivalent to a multi-armed bandit problem with K^H arms, where the optimal arm gets a $\mathrm{Ber}(1/2 + \epsilon)$ reward while all other arms get a $\mathrm{Ber}(1/2)$ reward. Thus, identifying a policy that is no-more than ϵ suboptimal in this POMDP is information-theoretically equivalent to identifying the best arm in the stochastic bandit problem in Theorem 2 with K^H arms. Applying that lower bound gives a sample complexity bound of $\Omega(K^H/\epsilon^2)$.

A.2 Proving both lower bounds

To verify both lower bounds in Propositions 1 and 2, we construct the policy and regressor sets. For Proposition 1, we need a set of reactive policies such that finding the optimal policy has a large sample complexity. To this end, we use the set of all K^H mappings from the H observations to actions. Specifically, each policy π is identified with a sequence of H actions $(a_1, \ldots, a_H)$ and has $\pi(x_h) = a_h$. These policies are reactive by definition since they do not depend on any previous history, or state of the world. Clearly there are K^H such policies, and each policy is optimal for exactly one POMDP defined above, namely π_p is optimal for the POMDP corresponding to the path p. Furthermore, in the POMDP defined by p, we have $V(\pi_p) = 1/2 + \epsilon$, whereas $V(\pi) = 1/2$ for every other policy. Consequently, finding the best policy in the class is equivalent to identifying the best arm in this family of problems. Taking a uniform mixture of problems in the family as before, we reason that this requires at least $\Omega(K^H/\epsilon^2)$ trajectories.

For Proposition 2, we use a similar construction. For each path $p = (a_1, \ldots, a_H)$, we associate a regressor f_A with,

$$f_p(\rho) = \frac{1}{2} + \epsilon \mathbf{1}[\rho \text{ is a prefix of } p]$$

Here we use ρ to denote the history of the interaction, which can be condensed to a sequence of actions since the observations provide no information.

Clearly for each POMDP, parameterized by p, f_p correctly maps the history to future reward, since the observations are useless here, meaning that the POMDP is realizable for this regressor class. Relatedly, π_{f_p} is the optimal policy for the POMDP with optimal sequence p. Moreover, there are precisely K^H regressors.

Algorithm 4 CTXMDPLEARN $(\mathcal{F}, \epsilon, \delta)$

$\mathcal{F} \leftarrow$ DFS-LEARN$(\varnothing, \mathcal{F}, \epsilon, \delta/2)$.
Let $\hat{V}^\star = \hat{V}^f(\varnothing, \pi_f)$ for any $f \in \mathcal{F}$.
$f \leftarrow$ EXPLORE-ON-DEMAND$(\mathcal{F}, \hat{V}^\star, \epsilon, \delta/2)$.
Return π_f.

Algorithm 5 DFS-LEARN $(p, \mathcal{F}, \epsilon, \delta)$

Set $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$ and $\epsilon_{\text{test}} = 20(H - |p| - 5/4)\sqrt{K}\phi$.
for $a \in \mathcal{A}$ **do**
  if Not CONSENSUS$(p \circ a, \mathcal{F}, \epsilon_{\text{test}}, \phi, \frac{\delta/2}{MKH})$ **then**
    $\mathcal{F} \leftarrow$ DFS-LEARN$(p \circ a, \mathcal{F}, \epsilon, \delta)$. # Recurse
  end if
end for
$\hat{\mathcal{F}} \leftarrow$ TD-ELIM $\left(p, \mathcal{F}, \phi, \frac{\delta/2}{MH}\right)$. # Learn in state p.
Return $\hat{\mathcal{F}}$.

As before, the learning objective requires identifying the optimal policy and hence the optimal path, which requires $\Omega(K^H/\epsilon^2)$ trajectories.

B Full Algorithm Pseudocode

It is more natural to break the algorithm into more components for the analysis. This lets us focus on each component in isolation.

Pseudocode for the compartmentalized version of the algorithm is displayed in Algorithm 4 with subroutines displayed as Algorithms 5, 6, 7, and 8. The algorithm should be invoked as CTXMDPLEARN$(\mathcal{F}, \epsilon, \delta)$ where $\mathcal{F}$ is the given class of regression functions, ϵ is the target accuracy and δ is the target failure probability. The two main components of the algorithm are the DFS-LEARN and EXPLORE-ON-DEMAND routines. DFS-LEARN ensures proper invocation of the training step, TD-ELIM, by verifying a number of preconditions, while EXPLORE-ON-DEMAND finds regions of the search tree for which training must be performed.

It is easily verified that this is an identical description of the algorithm.

C The Full Analysis

The proof of the theorem hinges on analysis of the the subroutines. We turn first to the TD-ELIM routine, for which we show the following guarantee. Recall the definition,

$$V^f(p, \pi_f) = \mathbb{E}_{x \sim D_p} f(x, \pi_f(x)).$$

Theorem 3 (Guarantee for TD-ELIM)**.** *Consider running* TD-ELIM *at path p with regressors $\mathcal{F}$, parameters ϕ, δ and with $n_{train} = 24\log(4N/\delta)/\phi^2$. Suppose that the following are true:*

1. ***Estimation Precondition:*** *We have access to estimates $\hat{V}^f(p \circ a, \pi_f)$ for all $f \in \mathcal{F}, a \in \mathcal{A}$ such that, $|\hat{V}^f(p \circ a, \pi_f) - V^f(p \circ a, \pi_f)| \le \phi$.*

Algorithm 6 CONSENSUS$(p, \mathcal{F}, \epsilon_{\text{test}}, \phi, \delta)$

Set $n_{\text{test}} = 2\log(2N/\delta)/\phi^2$.
Collect n_{test} observations $x_i \sim D_p$.
Compute Monte-Carlo estimates for each value function,

$$\hat{V}^f(p, \pi_f) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} f(x_i, \pi_f(x_i)) \qquad \forall f \in \mathcal{F}$$

if $|\hat{V}^f(p, \pi_f) - \hat{V}^g(p, \pi_g)| \leq \epsilon_{\text{test}}$ for all $f, g \in \mathcal{F}$ **then**
  return `true`
end if
Return `false`.

Algorithm 7 TD-ELIM$(p, \mathcal{F}, \phi, \delta)$

Require estimates $\hat{V}^f(p \circ a, \pi_f), \forall f \in \mathcal{F}, a \in \mathcal{A}$.
Set $n_{\text{train}} = 24\log(4N/\delta)/\phi^2$
Collect n_{train} observations (x_i, a_i, r_i) where $x_i \sim D_p$, a_i is chosen uniformly at random, and $r_i = r_i(a_i)$.
Update $\mathcal{F}$ to

$$\left\{ f \in \mathcal{F} : \tilde{R}(f) \leq \min_{f' \in \mathcal{F}} \tilde{R}(f') + 2\phi^2 + \frac{22\log(2N/\delta)}{n_{\text{train}}} \right\},$$

$$\text{with } \tilde{R}(f) = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (f(x_i, a_i) - r_i - \hat{V}^f(p \circ a_i, \pi_f))^2$$

Return $\mathcal{F}$.

2. ***Bias Precondition:*** *For all $f, g \in \mathcal{F}$ and for all $a \in \mathcal{A}$, $|V^f(p \circ a, \pi_f) - V^g(p \circ a, \pi_g)| \leq \tau_1$.*

Then the following hold simultaneously with probability at least $1 - \delta$:

1. *$f^\star$ is retained by the algorithm.*
2. ***Bias Bound**:*

$$|V^f(p, \pi_f) - V^g(p, \pi_g)| \leq 8\phi\sqrt{K} + 2\phi + \tau_1 \tag{6}$$

3. ***Instantaneous Risk Bound:***

$$V^\star(p) - V^{f^\star}(p, \pi_f) \leq 4\phi\sqrt{2K} + 2\phi + 2\tau_1 \tag{7}$$

4. ***Estimation Bound**: Regardless of whether the preconditions hold, we have estimates $\hat{V}^f(p, \pi_f)$ with,*

$$|\hat{V}^f(p, \pi_f) - V^f(p, \pi_f)| \leq \frac{\phi}{\sqrt{12}} \tag{8}$$

The last three bounds hold for all surviving $f, g \in \mathcal{F}$.

Algorithm 8 EXPLORE-ON-DEMAND $(\mathcal{F}, \hat{V}^\star, \epsilon, \delta)$

Set $\epsilon_{\text{demand}} = \epsilon/2, n_1 = \frac{32\log(6MH/\delta)}{\epsilon^2}$ and $n_2 = \frac{8\log(3MH/\delta)}{\epsilon}$.
while `true` **do**
  Fix a regressor $f \in \mathcal{F}$.
  Collect n_1 trajectories according to π_f and estimate $V(\pi_f)$ via a Monte-Carlo estimate $\hat{V}(\pi_f)$.
  If $|\hat{V}(\pi_f) - \hat{V}^\star| \leq \epsilon_{\text{demand}}$, return π_f.
  Otherwise update $\mathcal{F}$ by calling DFS-LEARN $(p, \mathcal{F}, \epsilon, \delta/(3MH^2 n_2))$ on each of the $H-1$ prefixes p of each of the first n_2 paths collected for the Monte-Carlo estimate.
end while

The theorem shows that, as long as we call TD-ELIM with the two preconditions, then $f^\star$, the optimal regressor, always survives. It also establishes a number of other properties about the surviving functions, namely that they agree on the value of this path (the bias bound) and that the associated policies take good actions from this path (the instantaneous risk bound). Note that the instantaneous risk bound is *not* a cumulative risk bound. The second term on the left hand side is the reward achieved by behaving like π_f for one action but then behaving optimally afterwards. The proof is deferred to Appendix E

Analysis of the CONSENSUS subroutine requires only standard concentration-of-measure arguments.

Theorem 4 (Guarantee for CONSENSUS)**.** *Consider running* CONSENSUS *on path p with $n_{test} = 2\log(2N/\delta)/\phi^2$ and $\epsilon_{test} \geq 2\phi + \tau_2$, for some $\tau_2 > 0$.*

1. *With probability at least $1-\delta$, we have estimates $\hat{V}^f(p, \pi_f)$ with $|\hat{V}^f(p,\pi_f) - V^f(p,\pi_f)| \leq \phi$ $\forall f \in \mathcal{F}$.*
2. *If $|V^f(p,\pi_f) - V^g(p,\pi_g)| \leq \tau_2, \forall f, g \in \mathcal{F}$, under the event in (1), the algorithm returns* `true`*.*
3. *If the algorithm returns* `true`*, then under the event in (1), we have $|V^f(p,\pi_f) - V^g(p,\pi_g)| \leq 2\phi + \epsilon_{test}$ $\forall f, g \in \mathcal{F}$.*

Appendix F provides the proof.

Analysis of both the DFS-LEARN and EXPLORE-ON-DEMAND routines requires a careful inductive argument. We first consider the DFS-LEARN routine.

Theorem 5 (Guarantee for DFS-LEARN)**.** *Consider running* DFS-LEARN *on path p with regressors $\mathcal{F}$, and parameters ϵ, δ. With probability at least $1-\delta$, for all h and all $s_h \in \mathcal{S}_h$ for which we called* TD-ELIM*, the conclusions of Theorem 3 hold with $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$ and $\tau_1 = 20(H-h)\sqrt{K}\phi$. If T is the number of times the algorithm calls* TD-ELIM*, then the number of episodes executed by the algorithm is at most,*

$$\mathcal{O}\left(\frac{TH^4K^2}{\epsilon^2}\log(NMKH/\delta)\right).$$

Moreover, $T \leq MH$ for any execution of DFS-LEARN.

The proof details are deferred to Appendix G.

A simple consequence of Theorem 5 is that we can estimate $V^\star$ accurately once we have called DFS-LEARN on $\varnothing$.

Corollary 1 (Estimating $V^\star$)**.** *Consider running* DFS-LEARN *at $\varnothing$ with regressors $\mathcal{F}$, and parameters ϵ, δ. Then with probability at least $1-\delta$, the estimate $\hat{V}^\star$ satisfies,*

$$|\hat{V}^\star - V^\star| \leq \epsilon/8.$$

Moreover the algorithm uses at most,

$$\mathcal{O}\left(\frac{MH^5K^2}{\epsilon^2}\log\left(\frac{NMHK}{\delta}\right)\right)$$

trajectories.

Proof. Since we ran DFS-LEARN at $\varnothing$, we may apply Theorem 5. By specification of the algorithm, we certainly ran TD-ELIM at $\varnothing$, which is at level $h = 1$, so we apply the conclusions in Theorem 3. In particular, we know that $f^\star \in \mathcal{F}$ and that for any surviving $f \in \mathcal{F}$,

$$\begin{aligned}|\hat{V}^f(p,\pi_f) - V^\star| &= |\hat{V}^f(p,\pi_f) - V^f(p,\pi_f) + V^f(p,\pi_f) - V^{f^\star}(p,\pi_{f^\star})| \\ &\le \frac{\phi}{\sqrt{12}} + 8\phi\sqrt{K} + 2\phi + 20(H-1)\sqrt{K}\phi \le \epsilon/8.\end{aligned}$$

The last bound follows from the setting of ϕ and τ_1. Since our estimate $\hat{V}^\star$ is $\hat{V}^f(p,\pi_f)$ for some surviving f, we guarantee estimation error at most $\epsilon/8$.

As for the sample complexity, Theorem 5 shows that the total number of executions of TD-ELIM can be at most MH, which is our setting of T. $\square$

Finally we turn to the EXPLORE-ON-DEMAND routine.

Theorem 6 (Guarantee for EXPLORE-ON-DEMAND)**.** *Consider running* EXPLORE-ON-DEMAND *with regressors $\mathcal{F}$, estimate $\hat{V}^\star$ and parameters ϵ, δ and assume that $|\hat{V}^\star - V^\star| \le \epsilon/8$. Then with probability at least $1-\delta$,* EXPLORE-ON-DEMAND *terminates after at most,*

$$\tilde{\mathcal{O}}\left(\frac{MH^6K^2}{\epsilon^3}\log(N/\delta)\log(1/\delta)\right)$$

trajectories and it returns a policy π_f with $V^\star - V(\varnothing, \pi_f) \le \epsilon$.

See Appendix H for details.

D Proof of Theorem 1

The proof of the main theorem follows from straightforward application of Theorems 5 and 6. First, since we run DFS-LEARN at the root, $\varnothing$, the bias and estimation bounds in Theorem 3 apply at $\varnothing$, so we guarantee accurate estimation of the value $V^\star$ (See Corollary 1). This is required by the EXPLORE-ON-DEMAND routine, but at this point, we can simply apply Theorem 6, which is guaranteed to find a ϵ-suboptimal policy and also terminate in MH iterations. Combining these two results, appropriately allocating the failure probability δ evenly across the two calls, and accumulating the sample complexity bounds establishes Theorem 1.

E Proof of Theorem 3

The proof of Theorem 3 is quite technical, and we compartmentalize into several components. We begin with several technical lemmas. Throughout we will use the preconditions of the theorem, which we reproduce here.

Condition 1. For all $f \in \mathcal{F}$ and $a \in \mathcal{A}$, we have estimates $\hat{V}^f(p \circ a, \pi_f)$ such that,

$$|\hat{V}^f(p \circ a, \pi_f) - V^f(p \circ a, \pi_f)| \le \phi.$$

Condition 2. For all $f, g \in \mathcal{F}$ and $a \in \mathcal{A}$ we have,

$$|V^f(p \circ a, \pi_f) - V^g(p \circ a, \pi_g)| \le \tau_1.$$

We will make frequent use of the parameters ϕ and τ_1 which are specified by these two conditions, and explicit in the theorem statement.

Recall the notation,

$$V^f(p, \pi_g) = \mathbb{E}_{x \sim D_p} f(x, \pi_g(x))$$

which will be used heavily throughout the proof.

As notational convenience, we will suppress dependence on the distribution D_p, since we are considering one invocation of TD-ELIM and we always roll into path p. This means that all (observation, reward) tuples will be drawn from D_p. Secondly it will be convenient to introduce the shorthand $V^f(p) = V^f(p, \pi_f)$ and similarly for the estimates. Finally, we will further shorten the value functions for paths $p \circ a$ by defining,

$$V_a^f = \mathbb{E}_{x \sim D_{p \circ a}} f(x, \pi_f(x)) = V^f(p \circ a, \pi_f).$$

We will also use $\hat{V}_a^f$ to denote the estimated versions which we have access to according to Condition 1.

Lastly, our proof makes extensive use of the following random variable, which is defined for a particular regressor $f \in \mathcal{F}$

$$Y(f) \triangleq (f(x,a) - r(a) - \hat{V}^f(p \circ a))^2 - (f^\star(x,a) - r(a) - \hat{V}^{f^\star}(p \circ a))^2.$$

Here $(x, r) \sim D_p$ and $a \in \mathcal{A}$ is drawn uniformly at random as prescribed by Algorithm 7. We use $Y(f)$ to denote the random variable associated with regressor f, but sometimes drop the dependence on f when it is clear from context.

To proceed, we first compute the expectation and variance of this random variable.

Lemma 1 (Properties of TD Squared Loss)**.** *Assume Condition 1 holds. Then for any $f \in \mathcal{F}$, the random variable Y satisfies,*

$$\mathbb{E}_{x,a,r}[Y] = \mathbb{E}_{x,a}\left[(f(x,a) - \hat{V}^f(p \circ a) - f^\star(x,a) + V^{f^\star}(p \circ a))^2\right] - \mathbb{E}_{x,a}\left[(\hat{V}^{f^\star}(p \circ a) - V^{f^\star}(p \circ a))^2\right]$$
$$\operatorname*{Var}_{x,a,r}[Y] \le 32\mathbb{E}_{x,a}[Y] + 64\phi^2$$

Proof. For further shorthand, denote $f = f(x,a), f^\star = f^\star(x,a)$ and recall the definition of V_a^f and $\hat{V}_a^f$.

$$\begin{aligned}
&\mathbb{E}_{x,a,r} Y \\
&= \mathbb{E}_{x,a,r}\left[(f - \hat{V}_a^f - r(a))^2 - (f^\star - \hat{V}_a^{f^\star} - r(a))^2\right] \\
&= \mathbb{E}_{x,a,r}\left[(f - \hat{V}_a^f)^2 - 2r(a)(f - \hat{V}_a^f - f^\star + \hat{V}_a^{f^\star}) - (f^\star - \hat{V}_a^{f^\star})^2\right]
\end{aligned}$$

Now recall that $\mathbb{E}[r(a)|x,a] = f^*(x,a) - V_a^{f^\star}$ by the definition of f^*, which allows us to further obtain

$$
\begin{aligned}
&\mathbb{E}_{x,a,r} Y \\
&= \mathbb{E}_{x,a}\left[(f - \hat{V}_a^f)^2 - 2(f^\star - V_a^{f^\star})(f - \hat{V}_a^f) + 2(f^\star - \hat{V}_a^{f^\star} + \hat{V}_a^{f^\star} - V_a^{f^\star})(f^\star - \hat{V}_a^{f^\star}) - (f^\star - \hat{V}_a^{f^\star})^2\right] \\
&= \mathbb{E}_{x,a}\left[(f - \hat{V}_a^f)^2 - 2(f^\star - V_a^{f^\star})(f - \hat{V}_a^f) + (f^\star - \hat{V}_a^{f^\star})^2 + 2(\hat{V}_a^{f^\star} - V_a^{f^\star})(f^\star - \hat{V}_a^{f^\star})\right] \\
&= \mathbb{E}_{x,a}\left[(f - \hat{V}_a^f)^2 - 2(f^\star - V_a^{f^\star})(f - \hat{V}_a^f) + (f^\star - V_a^{f^\star} + V_a^{f^\star} - \hat{V}_a^{f^\star})^2 + 2(\hat{V}_a^{f^\star} - V_a^{f^\star})(f^\star - \hat{V}_a^{f^\star})\right] \\
&= \mathbb{E}_{x,a}\left[(f - \hat{V}_a^f - f^\star + V_a^{f^\star})^2 + 2(V_a^{f^\star} - \hat{V}_a^{f^\star})(f^\star - V_a^{f^\star}) + (V_a^{f^\star} - \hat{V}_a^{f^\star})^2 + 2(\hat{V}_a^{f^\star} - V_a^{f^\star})(f^\star - \hat{V}_a^{f^\star})\right] \\
&= \mathbb{E}_{x,a}\left[(f - \hat{V}_a^f - f^\star + V_a^{f^\star})^2 - (V_a^{f^\star} - \hat{V}_a^{f^\star})^2\right]
\end{aligned}
$$

For the second claim, notice that we can write,

$$Y = (f - \hat{V}_a^f - f^\star + \hat{V}_a^{f^\star})(f - \hat{V}_a^f + f^\star - \hat{V}_a^{f^\star} - 2r(a)),$$

so that,

$$Y^2 \le 16(f - \hat{V}_a^f - f^\star + \hat{V}_a^{f^\star})^2.$$

This holds because all quantities in the second term are bounded in $[0,1]$. Therefore,

$$
\begin{aligned}
\mathrm{Var}(Y) &\le \mathbb{E}[Y^2] \\
&\le 16\mathbb{E}_{x,a}\left[(f(x,a) - \hat{V}_a^f - f^\star(x,a) + \hat{V}_a^{f^\star})^2\right] \\
&= 16\mathbb{E}_{x,a}\left[(f(x,a) - \hat{V}_a^f - f^\star(x,a) + V_a^{f^\star} + \hat{V}_a^{f^\star} - V_a^{f^\star})^2\right] \\
&\le 32\mathbb{E}_{x,a}\left[(f(x,a) - \hat{V}_a^f - f^\star(x,a) + V_a^{f^\star})^2\right] + 32\phi^2 \\
&\le 32\mathbb{E}_{x,a} Y + 64\phi^2
\end{aligned}
$$

The first inequality is straightforward, while the second inequality is from the argument above. The third inequality uses the fact that $(a+b)^2 \le 2a^2 + 2b^2$ and the fact that for each a, the estimate $\hat{V}_a^{f^\star}$ has absolute error at most ϕ (By Condition 1). The last inequality adds and subtracts the term involving $(V_a^{f^\star} - \hat{V}_a^{f^\star})^2$ to obtain $\mathbb{E}_{x,a} Y$. □

The next step is to relate the empirical squared loss to the population squared loss, which is done by application of Bernstein's inequality.

Lemma 2 (Squared Loss Deviation Bounds)**.** *Assume Condition 1 holds. With probability at least* $1 - \delta/2$, *where* δ *is a parameter of the algorithm,* $f^\star$ *survives the filtering step of Algorithm 7 and moreover, any surviving* f *satisfies,*

$$\mathbb{E}Y(f) \le 10\phi^2 + \frac{120\log(2N/\delta)}{n_{train}}.$$

Proof. We will apply Bernstein's inequality on the centered random variable,

$$\sum_{i=1}^{n_{\text{train}}} Y_i(f) - \mathbb{E}Y_i(f),$$

and then take a union bound over all $f \in \mathcal{F}$. Here the expectation is over the n_{train} samples (x_i, a_i, r_i) where $(x_i, r) \sim D_p$, a_i is chosen uniformly at random, and $r_i = r(a_i)$. Notice that since actions are chosen uniformly at random, all terms in the sum are identically distributed, so that $\mathbb{E}Y_i(f) = \mathbb{E}Y(f)$.

To that end, fix one $f \in \mathcal{F}$ and notice that $|Y - \mathbb{E}Y| \le 8$ almost surely, as each quantity in the definition of Y is bounded in $[0, 1]$, so each of the four terms can be at most 4, but two are non-positive and two are non-negative in $Y - \mathbb{E}Y$. We will use Lemma 1 to control the variance. Bernstein's inequality implies that, with probability at least $1 - \delta$,

$$\begin{aligned}\sum_{i=1}^{n_{\text{train}}} \mathbb{E}Y_i - Y_i &\le \sqrt{2\sum_i \mathrm{Var}(Y_i)\log(1/\delta)} + \frac{16\log(1/\delta)}{3} \\ &\le \sqrt{64\sum_i \left(\mathbb{E}(Y_i) + 2\phi^2\right)\log(1/\delta)} + \frac{16\log(1/\delta)}{3}\end{aligned}$$

The first inequality here is Bernstein's inequality while the second is based on the variance bound in Lemma 1.

Now letting $X = \sqrt{\sum_i (\mathbb{E}(Y_i) + 2\phi^2)}$, $Z = \sum_i Y_i$ and $C = \sqrt{\log(1/\delta)}$, the inequality above is equivalent to,

$$\begin{aligned}&X^2 - 2n_{\text{train}}\phi^2 - Z \le 8XC + \frac{16}{3}C^2 \\ &\Rightarrow X^2 - 8XC + 16C^2 - Z \le 2n_{\text{train}}\phi^2 + 22C^2 \\ &\Rightarrow (X - 4C^2) - Z \le 2n_{\text{train}}\phi^2 + 22C^2 \\ &\Rightarrow -Z \le 2n_{\text{train}}\phi^2 + 22C^2\end{aligned}$$

Using the definition of $-Z$, this last inequality implies that,

$$\sum_{i=1}^{n_{\text{train}}}(f^\star(x_i, a_i) - r_i(a_i) - \hat{V}^{f^\star}(p \circ a_i))^2 \le \sum_{i=1}^{n_{\text{train}}}(f(x_i, a_i) - r_i(a_i) - \hat{V}^{f}(p \circ a_i))^2 + 2n_{\text{train}}\phi^2 + 22\log(1/\delta)$$

Via a union bound over all $f \in \mathcal{F}$, rebinding $\delta \leftarrow \delta/(2N)$, and dividing through by n_{train}, we have,

$$\tilde{R}(f^\star) \le \min_{f \in \mathcal{F}} \tilde{R}(f) + 2\phi^2 + \frac{22\log(2N/\delta)}{n_{\text{train}}}$$

Since this is precisely the threshold used in filtering regressors, we ensure that $f^\star$ survives.

Now for any other surviving regressor f, we are ensured that Z is upper bounded. Specifically we have,

$$\begin{aligned}(X - 4C)^2 &\le Z + 2n_{\text{train}}\phi^2 + 22C^2 \le 4n_{\text{train}}\phi^2 + 44C^2 \\ \Rightarrow X^2 &\le (\sqrt{4n_{\text{train}}\phi^2 + 44C^2} + 4C)^2 \\ &\le 8n_{\text{train}}\phi^2 + 120C^2\end{aligned}$$

This proves the claim since $X^2 = n_{\text{train}}\mathbb{E}Y(f) + 2n_{\text{train}}\phi^2$ (Recall that the Y_is are identically distributed). □

This deviation bound allows us to establish the three claims in Theorem 3. We start with the estimation error claim, which is straightforward.

Lemma 3 (Estimation Error). *Let $\delta \in (0,1)$. Then with probablity at least $1-\delta$, for all $f \in \mathcal{F}$ that are retained by the Algorithm 7, we have estimates $\hat{V}^f(p,\pi_f)$ with,*

$$|\hat{V}^f(p,\pi_f) - V^f(p,\pi_f)| \leq \sqrt{\frac{2\log(2N/\delta)}{n_{train}}}.$$

Proof. The proof is a consequence of Hoeffding's inequality and a union bound. Clearly the Monte Carlo estimate,

$$\hat{V}^f(p,\pi_f) = \frac{1}{n_{\text{train}}}\sum_{i=1}^{n_{\text{train}}} f(x_i,\pi_f(x_i)),$$

is unbiased for $V^f(p,\pi_f)$ and the centered quantity is bounded in $[-1,1]$. Thus Hoeffding's inequality gives precisely the bound in the lemma. $\square$

Next we turn to the claim regarding bias.

Lemma 4 (Bias Accumulation). *Assume Conditions 1 and 2 hold. In the same $1-\delta/2$ event in Lemma 2, for any pair $f,g \in \mathcal{F}$ retained by Algorithm 7, we have,*

$$V^f(p,\pi_f) - V^g(p,\pi_g) \leq 2\sqrt{K}\sqrt{11\phi^2 + \frac{120\log(2N/\delta)}{n_{train}}} + 2\phi + \tau_1$$

Proof. We start by expanding definitions,

$$V^f(p,\pi_f) - V^g(p,\pi_g) = \mathbb{E}_{x\sim D_p}[f(x,\pi_f(x)) - g(x,\pi_g(x))]$$

Now, since g prefers $\pi_g(x)$ to $\pi_f(x)$, it must be the case that $g(x,\pi_g(x)) \geq g(x,\pi_f(x))$, so that,

$$\begin{aligned}
V^f(p,\pi_f) - V^g(p,\pi_g) \leq & \mathbb{E}_{x\sim D_p} f(x,\pi_f(x)) - g(x,\pi_f(x)) \\
= \quad & \mathbb{E}_{x\sim D_p}[f(x,\pi_f(x)) - \hat{V}^f(p\circ\pi_f(x),\pi_f) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star})] \\
& - \mathbb{E}_{x\sim D_p}[g(x,\pi_f(x)) - \hat{V}^g(p\circ\pi_f(x),\pi_g) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star})] \\
& + \mathbb{E}_{x\sim D_p}[\hat{V}^f(p\circ\pi_f(x),\pi_f) - \hat{V}^g(p\circ\pi_f(x),\pi_g)]
\end{aligned}$$

This last equality is just based on adding and subtracting several terms. The first two terms look similar, and we will relate them to the squared loss. For the first, by Lemma 1, we have that for each $x \in \mathcal{X}$,

$$\begin{aligned}
&\mathbb{E}_{r,a|x}[Y(f)] + \mathbb{E}_{a|x}[(\hat{V}^{f^\star}(p\circ a,\pi_{f^\star}) - V^{f^\star}(p\circ a,\pi_{f^\star}))^2] \\
&= \mathbb{E}_{a|x}\left[(f(x,a) - \hat{V}^f(p\circ a,\pi_f) - f^\star(x,a) + V^{f^\star}(p\circ a,\pi_{f^\star}))^2\right] \\
&\geq \frac{1}{K}\left[(f(x,\pi_f(x)) - \hat{V}^f(p\circ\pi_f(x),\pi_f) - f^\star(x,\pi_f(x)) + V^{f^\star}(p\circ\pi_f(x),\pi_{f^\star}))^2\right]
\end{aligned}$$

The equality is Lemma 1 while the inequality follows from the fact that each action, in particular $\pi_f(x)$, is played with probability $1/K$ and the quantity inside the expectation is non-negative. Now by Jensen's inequality the first term can be upper bounded as,

$$
\begin{aligned}
&\mathbb{E}_{x\sim D_p}[f(x,\pi_f(x)) - \hat{V}^f(p\circ\pi_f(x),\pi_f) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star})] \\
\leq&\sqrt{\mathbb{E}_{x\sim D_p}[(f(x,\pi_f(x)) - \hat{V}^f(p\circ\pi_f(x),\pi_f) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star}))^2]} \\
=&\sqrt{K\mathbb{E}_{x\sim D_p}\left[\frac{1}{K}(f(x,\pi_f(x)) - \hat{V}^f(p\circ\pi_f(x),\pi_f) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star}))^2\right]} \\
\leq&\sqrt{K\left(\mathbb{E}_{x,a,r}[Y(f)] + \mathbb{E}_{x,a}[(\hat{V}^{f^\star}(p\circ a,\pi_{f^\star}) - V^{f^\star}(p\circ a,\pi_{f^\star}))^2]\right)} \\
&\leq \sqrt{K}\sqrt{\mathbb{E}Y(f)+\phi^2} \\
&\leq \sqrt{K}\sqrt{11\phi^2 + \frac{120\log(N/\delta)}{n_{\text{train}}}},
\end{aligned}
$$

where the last step follows from Lemma 2. This bounds the first term in the expansion of $V^f(p,\pi_f) - V^g(p,\pi_g)$. Now for the term involving g, we can apply essentially the same argument,

$$
\begin{aligned}
&-\mathbb{E}_{x\sim D_p}[g(x,\pi_f(x)) - \hat{V}^g(p\circ\pi_f(x),\pi_g) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star})] \\
&\leq \sqrt{\mathbb{E}_{x\sim D_p}[(g(x,\pi_f(x)) - \hat{V}^g(p\circ\pi_f(x),\pi_g) - f^\star(x,\pi_f(x)) + \hat{V}^{f^\star}(p\circ\pi_f(x),\pi_{f^\star}))^2]} \\
&\leq \sqrt{K}\sqrt{11\phi^2 + \frac{120\log(N/\delta)}{n_{\text{train}}}}
\end{aligned}
$$

Summarizing, the current bound we have is,

$$
V^f(p,\pi_f) - V^g(p,\pi_g) \leq 2\sqrt{K}\sqrt{11\phi^2 + \frac{120\log(N/\delta)}{n_{\text{train}}}} + \mathbb{E}_{x\sim D_p}[\hat{V}^f(p\circ\pi_f(x),\pi_f) - \hat{V}^g(p\circ\pi_f(x),\pi_g)] \tag{9}
$$

The last term is easily bounded by the preconditions in the statement of Theorem 3. For each a, we have,

$$
\begin{aligned}
&\hat{V}^f(p\circ a,\pi_f) - \hat{V}^g(p\circ a,\pi_g) \\
&\leq |\hat{V}^f(p\circ a,\pi_f) - V^f(p\circ a,\pi_f)| + |V^f(p\circ a,\pi_f) - V^g(p\circ a,\pi_g)| + |V^g(p\circ a,\pi_g) - \hat{V}^g(p\circ a,\pi_g)| \\
&\leq 2\phi + \tau_1,
\end{aligned}
$$

from Conditions 1 and 2. Consequently

$$
\begin{aligned}
&\mathbb{E}_{x\sim D_p}[\hat{V}^f(p\circ\pi_f(x),\pi_f) - \hat{V}^g(p\circ\pi_f(x),\pi_g)] \\
&= \sum_{a\in\mathcal{A}}\mathbb{E}_x\left[\mathbf{1}[\pi_f(x)=a](\hat{V}^f(p\circ a,\pi_f) - \hat{V}^g(p\circ a,\pi_g))\right] \\
&\leq 2\phi + \tau_1
\end{aligned}
$$

This proves the claim. □

Lastly, we must show how the squared loss relates to the risk, which helps establish the last claim of the theorem. The proof is similar to that of the bias bound but has subtle differences that require reproducing the argument.

Lemma 5 (Instantaneous Risk Bound)**.** *Assume Conditions 1 and 2 hold. In the same $1-\delta/2$ event in Lemma 2, for any regressor $f \in \mathcal{F}$ retained by Algorithm 7, we have,*

$$V^{f^\star}(p,\pi_{f^\star}) - V^{f^\star}(p,\pi_f) \le \sqrt{2K}\sqrt{11\phi^2 + \frac{120\log(2N/\delta)}{n_{train}}} + 2(\phi+\tau_1).$$

Proof.

$$\begin{aligned}V^{f^\star}(p,\pi_{f^\star}) - V^{f^\star}(p,\pi_f) &= \mathbb{E}_x[f^\star(x,\pi_{f^\star}(x)) - f^\star(x,\pi_f(x))]\\ &\le \mathbb{E}_x[f^\star(x,\pi_{f^\star}(x)) - f(x,\pi_{f^\star}(x)) + f(x,\pi_f(x)) - f^\star(x,\pi_f(x))]\end{aligned}$$

This follows since f prefers its own action to that of $f^\star$, so that $f(x,\pi_f(x)) \ge f(x,\pi_{f^\star}(x))$. For any observation $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, define,

$$\Delta_{x,a} = (f(x,a) - \hat{V}^f(p\circ a,\pi_f) - f^\star(x,a) + V^{f^\star}(p\circ a,\pi_{f^\star}))$$

so that we can write,

$$\begin{aligned}&V^{f^\star}(p,\pi_{f^\star}) - V^{f^\star}(p,\pi_f)\\ &\le \mathbb{E}_x[\Delta_{x,\pi_f(x)} - \Delta_{x,\pi_{f^\star}(x)} + (\hat{V}^f(p\circ\pi_f(x)) - V^{f^\star}(p\circ\pi_f(x)) - \hat{V}^f(p\circ\pi_{f^\star}(x)) + V^{f^\star}(p\circ\pi_{f^\star}(x)))]\end{aligned}$$

The term involving both Δs can be bounded as in the proof of Lemma 4. For any $x \in \mathcal{X}$

$$\begin{aligned}&\mathbb{E}_{r,a|x}Y(f) + \mathbb{E}_{a|x}[(\hat{V}^{f^\star}(p\circ a) - V^{f^\star}(p\circ a))^2]\\ &= \mathbb{E}_{a|x}\left[(f(x,a) - \hat{V}^f(p\circ a) - f^\star(x,a) + V^{f^\star}(p\circ a))^2\right]\\ &\ge \frac{\Delta^2_{x,\pi_f(x)} + \Delta^2_{x,\pi_{f^\star}(x)}}{K} \ge \frac{(\Delta_{x,\pi_{f^\star}(x)} - \Delta_{x,\pi_f(x)})^2}{2K}\end{aligned}$$

Thus,

$$\begin{aligned}\mathbb{E}_x[\Delta_{x,\pi_f(x)} - \Delta_{x,\pi_{f^\star}(x)}] &\le \sqrt{2K\mathbb{E}\frac{(\Delta_{x,\pi_f(x)} - \Delta_{x,\pi_{f^\star}(x)})^2}{2K}}\\ &\le \sqrt{2K}\sqrt{\mathbb{E}Y(f) + \phi^2} \le \sqrt{2K}\sqrt{11\phi^2 + \frac{120\log(2N/\delta)}{n_{\text{train}}}}\end{aligned}$$

We are left to bound the residual term,

$$\begin{aligned}&(\hat{V}^f(p\circ\pi_f(x)) - V^{f^\star}(p\circ\pi_f(x)) - \hat{V}^f(p\circ\pi_{f^\star}(x)) + V^{f^\star}(p\circ\pi_{f^\star}(x)))\\ &\le \left|V^f(p\circ\pi_f(x)) - V^{f^\star}(p\circ\pi_f(x)) - V^f(p\circ\pi_{f^\star}(x)) + V^{f^\star}(p\circ\pi_{f^\star}(x))\right| + 2\phi\\ &\le 2(\phi+\tau_1)\end{aligned}$$

□

Notice that Lemma 5 above controls the quantity $V^{f^\star}(p, \pi_{f^\star}) - V^{f^\star}(p, \pi_f)$ which is the difference in values of the optimal behavior from p and the policy that first acts according to π_f and then behaves optimally thereafter. This is *not* the same as acting according to π_f for all subsequent actions. We will control this cumulative risk $V^\star(p) - V(p, \pi_f)$ in the second phase of the algorithm.

Proof of Theorem 3: Equipped with the above lemmas, we can proceed to prove the theorem. By assumption of the theorem, Conditions 1 and 2 hold, so all lemmas are applicable. Apply Lemma 3 with failure probability $\delta/2$, where δ is the parameter in the algorithm, and apply Lemma 2, which also fails with probability at most $\delta/2$. A union bound over these two events implies that the failure probability of the algorithm is at most δ.

Outside of this failure event, all three of Lemmas 3, 4, and 5 hold. If we set $n_{\text{train}} = 24\log(4N/\delta)/\phi^2$ then these four bounds give,

$$\begin{aligned}
|\hat{V}^f(p, \pi_f) - V^f(p, \pi_f)| &\leq \frac{\phi}{\sqrt{12}} \\
|V^f(p, \pi_f) - V^g(p, \pi_g)| &\leq 8\phi\sqrt{K} + 2\phi + \tau_1 \\
V^{f^\star}(p, \pi_{f^\star}) - V^{f^\star}(p, \pi_f) &\leq 4\phi\sqrt{2K} + 2\phi + 2\tau_1.
\end{aligned}$$

These bounds hold for all $f, g \in \mathcal{F}$ that are retained by the algorithm. Of course by Lemma 2, we are also ensured that $f^\star$ is retained by the algorithm.

This proves the four claims in the theorem.

F Proof of Theorem 4

This result is a straightforward application of Hoeffding's inequality. We collect n_{test} observations $x_i \sim D_p$ by rolling into p and use the Monte Carlo estimates,

$$\hat{V}^f(p, \pi_f) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} f(x_i, \pi_f(x_i))$$

By Hoeffding's inequality, via a union bound over all $f \in \mathcal{F}$, we have that with probability at least $1 - \delta$,

$$\left|\hat{V}^f(p, \pi_f) - V^f(p, \pi_f)\right| \leq \sqrt{\frac{2\log(2N/\delta)}{n_{\text{test}}}}$$

Setting $n_{\text{test}} = 2\log(2N/\delta)/\phi^2$, gives that our empirical estimates are at most ϕ away from the population versions.

Now for the first claim, if the population versions are already within τ_2 of each other, then the empirical versions are at most $2\phi + \tau_2$ apart by the triangle inequality,

$$\begin{aligned}
|\hat{V}^f(p, \pi_f) - \hat{V}^g(p, \pi_g)| &\leq |\hat{V}^f(p, \pi_f) - V^f(p, \pi_f)| + |V^f(p, \pi_f) - V^g(p, \pi_g)| + |V^g(p, \pi_g) - \hat{V}^g(p, \pi_g)| \\
&\leq 2\phi + \tau_2.
\end{aligned}$$

This applies for any pair $f, g \in \mathcal{F}$ whose population value predictions are within τ_2 of each other. Since we set $\epsilon_{\text{test}} \geq 2\phi + \tau_2$ in Theorem 4, this implies that the procedure returns `true`.

For the second claim, if the procedure returns `true`, then all empirical value predictions are at most ϵ_{test} apart, so the population versions are at most $2\phi + \epsilon_{\text{test}}$ apart, again by the triangle inequality. Specifically, for any pair $f, g \in \mathcal{F}$ we have,

$$\begin{aligned}|V^f(p,\pi_f) - V^g(p,\pi_g)| &\le |V^f(p,\pi_f) - \hat{V}^f(p,\pi_f)| + |\hat{V}^f(p,\pi_f) - \hat{V}^g(p,\pi_g)| + |\hat{V}^g(p,\pi_g) - V^g(p,\pi_g)| \\ &\le 2\phi + \epsilon_{\text{test}}.\end{aligned}$$

Both arguments apply for all pairs $f, g \in \mathcal{F}$, which proves the claim.

G Proof of Theorem 5

Assume that all calls to TD-ELIM and CONSENSUS operate successfully, i.e., we can apply Theorems 3 and 4 on any path p for which the appropriate subroutine has been invoked. We will bound the number of calls and hence the total failure probability.

Recall that ϵ is the error parameter passed to DFS-LEARN and that we set $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$.

We first argue that in all calls to TD-ELIM, the estimation precondition is satisfied. To see this, notice that by design, the algorithm only calls TD-ELIM at path p after the recursive step, which means that for each a, we either ran TD-ELIM on $p \circ a$ or CONSENSUS returned `true` on $p \circ a$. Since both Theorems 3 and 4 guarantee estimation error of order ϕ, the estimation precondition for path p holds. This argument applies to all paths p for which we call TD-ELIM, so that the estimation precondition is always satisfied.

We next analyze the bias term, for which proceed by induction. To state the inductive claim, we define the notion of an *accessed path*. We say that a path p is *accessed* if either (a) we called TD-ELIM on path p or (b) we called CONSENSUS on p and it returned `true`.

The induction is on the number of actions remaining, which we denote with η. At time point h there are $H - h + 1$ actions remaining.

Inductive Claim: For all accessed paths p with η actions remaining and any pair $f, g \in \mathcal{F}$ of surviving regressors,

$$|V^f(p,\pi_f) - V^g(p,\pi_g)| \le 20\eta\sqrt{K}\phi$$

Base Case: The claim clearly holds when $\eta = 0$ since there are zero actions remaining and all regressors estimate future reward as zero.

Inductive Step: Assume that the inductive claim holds for all accessed paths with $\eta - 1$ actions remaining. Consider any accessed path p with η actions remaining. Since we access the path p, either we call TD-ELIM or CONSENSUS returns `true`. If we call TD-ELIM, then we access the paths $p \circ a$ for all $a \in \mathcal{A}$. Therefore by the inductive hypothesis, we have already filtered the regressor class so that for all $a \in \mathcal{A}, f, g \in \mathcal{F}$, we have,

$$|V^f(p \circ a,\pi_f) - V^g(p \circ a,\pi_f)| \le 20(\eta-1)\sqrt{K}\phi.$$

We will therefore instantiate $\tau_1 = 20(\eta-1)\sqrt{K}\phi$ in the bias precondition of Theorem 3. We also know that the estimation precondition is satisfied with parameter ϕ. Therefore, the bias bound of Theorem 3 shows that, for all $f, g \in \mathcal{F}$ retained by the algorithm,

$$\begin{aligned}|V^f(p,\pi_f) - V^g(p,\pi_g)| &\le 8\phi\sqrt{K} + 2\phi + \tau_1 \\ &\le 10\phi\sqrt{K} + 20(\eta-1)\phi\sqrt{K} \le 20(\eta - \frac{1}{2})\phi\sqrt{K} \qquad (10)\end{aligned}$$

Thus the inductive step holds in this case.

The other case we must consider is if CONSENSUS returns `true`. Notice that for a path p with η actions to go, we call CONSENSUS with parameter $\epsilon_{\text{test}} = 20(\eta - 1/4)\sqrt{K}\phi$. We actually invoke the routine on path p when we are currently processing a path p' with $h+1$ actions to go (i.e., $p = p' \circ a$ for some $a \in \mathcal{A}$), so we set ϵ_{test} in terms of $H - |p'| - 5/4 = H - |p \circ a| - 1/4 = \eta - 1/4$. ($|p|$ is actually one less than the level of the state reached by rolling in with p.) Then, by Theorem 4, we have the bias bound,

$$\begin{aligned}|V^f(p, \pi_f) - V^g(p, \pi_f)| &\leq 2\phi + 20(\eta - 1/4)\sqrt{K}\phi \\ &\leq 20\eta\sqrt{K}\phi\end{aligned}$$

Thus we have established the inductive claim.

Verifying preconditions for Theorem 3: To apply the conclusions of Theorem 3 at some state s, we must verify that the preconditions hold, with the appropriate parameter settings, before we executed the algorithm. We saw above that the estimation precondition always holds with parameter ϕ, assuming successful execution of all subroutines. The inductive argument also shows that the bias precondition also holds with $\tau_1 = 20(\eta - 1)\sqrt{K}\phi$ for a state $s \in \mathcal{S}_{H-\eta+1}$ that we called TD-ELIM on. Thus, both preconditions are satisfied at each execution of TD-ELIM, so the conclusions of Theorem 3 apply at any state s for which we have executed the subroutine. Note that the precondition parameters that we use here, specifically τ_1, depend on the actions-to-go η.

Substituting the level h for the actions-to-go η gives $\tau_1 = 20(H - h)\sqrt{K}\phi$ at level h.

Sample Complexity: We now bound the number of calls to each subroutine, which reveals how to allocate the failure probability and gives the sample complexity bound. Again assume that all calls succeed.

First notice that if we call CONSENSUS on some state s with η actions-to-go for which we have already called TD-ELIM, then CONSENSUS returns `true` (assuming all calls to subroutines succeed). This follows because TD-ELIM guarantees that the population predicted values are at most $20(\eta - 1/2)\sqrt{K}\phi$ apart (Eq. 10), which becomes the choice of τ_2 in application of Theorem 4. This is valid since,

$$2\phi + 20(\eta - 1/2)\sqrt{K}\phi \leq 20(\eta - 1/4)\sqrt{K}\phi = \epsilon_{\text{test}},$$

so that the precondition for Theorem 4 holds. Since actions-to-go uniquely identify the level, at any level h, we can call TD-ELIM at most one time per state $s \in \mathcal{S}_h$. In total, this yields MH calls to TD-ELIM.

Next, since we only make recursive calls when we execute TD-ELIM, we expand at most M paths per level. This means that we call CONSENSUS on at most MK paths per level, since the fan-out of the tree is K. Thus, the number of calls to CONSENSUS is at most MKH.

By our setting δ in the subroutine calls (i.e. $\delta/(2MKH)$ in calls to CONSENSUS and $\delta/(2MH)$ in calls to TD-ELIM), and by Theorems 3 and 4, the total failure probability is therefore at most δ.

Each execution of TD-ELIM requires n_{train} trajectories while executions of CONSENSUS require n_{test} trajectories. Since before each execution of TD-ELIM we always perform K executions of CONSENSUS, if we perform T executions of TD-ELIM, the total sample complexity is bounded by,

$$\begin{aligned}T(n_{\text{train}} + Kn_{\text{test}}) &\leq (3 \times 10^6)\frac{TH^4K}{\epsilon^2}\log(8NMH/\delta) + (3 \times 10^5)\frac{TH^4K^2}{\epsilon^2}\log(4NMKH/\delta) \\ &= \mathcal{O}\left(\frac{TH^4K^2}{\epsilon^2}\log\left(\frac{NMHK}{\delta}\right)\right),\end{aligned}$$

Recall that the total number of executions of TD-ELIM can be no more than MH, by the argument above.

H Analysis for EXPLORE-ON-DEMAND

The first part of the algorithm essentially computes the value $V^\star$ at the root of the search tree, but does not ensures good performance of retained policies. To do the latter, and to establish a PAC-guarantee, we run the EXPLORE-ON-DEMAND procedure.

Throughout the proof, we assume that $|\hat{V}^\star - V^\star| \le \epsilon/8$. We will ensure that the first half of the algorithm execution guarantees this. Let $\mathcal{E}$ denote the event that all Monte-Carlo estimates $\hat{V}(\varnothing, \pi_f)$ are accurate and all calls to DFS-LEARN succeed (so that we may apply Theorem 5). By accurate, we mean,

$$|\hat{V}(\varnothing, \pi_f) - V(\varnothing, \pi_f)| \le \epsilon/8.$$

Formally, $\mathcal{E}$ is the intersection over all executions of DFS-LEARN of the event that the conclusions of Theorem 5 apply for this execution and the intersection over all iterations of the loop in EXPLORE-ON-DEMAND of the event that the Monte Carlo estimate $\hat{V}(\varnothing, \pi_f)$ is within $\epsilon/8$ of $V(\varnothing, \pi_f)$. We will bound this failure probability, i.e. $\mathbb{P}[\bar{\mathcal{E}}]$, toward the end of the proof.

Lemma 6 (Risk bound upon termination)**.** *If $\mathcal{E}$ holds, then when* EXPLORE-ON-DEMAND *terminates, it outputs a policy π_f with $V^\star - V(\pi_f) \le \epsilon$.*

Proof. The proof is straightforward,

$$\begin{aligned} V^\star - V(\pi_f) &\le |V^\star - \hat{V}^\star| + |\hat{V}^\star - \hat{V}(\pi_f)| + |\hat{V}(\pi_f) - V(\pi_f)| \\ &\le \epsilon/8 + \epsilon/2 + \epsilon/8 = 3\epsilon/4 \le \epsilon \end{aligned}$$

The first bound follows by assumption on $\hat{V}^\star$ while the second comes from the definition of ϵ_{demand} and the third holds under event $\mathcal{E}$. □

Lemma 7 (Termination Guarantee)**.** *If $\mathcal{E}$ holds, then when* EXPLORE-ON-DEMAND *selects a policy that is at most $\epsilon/4$-suboptimal, it terminates.*

Proof. We must show that the test succeeds, for which we will apply the triangle inequality,

$$\begin{aligned} |\hat{V}^\star - \hat{V}(\pi_f)| &\le |\hat{V}^\star - V^\star| + |V^\star - V(\pi_f)| + |V(\pi_f) - \hat{V}(\pi_f)| \\ &\le \epsilon/8 + \epsilon/4 + \epsilon/8 \le \epsilon/2 = \epsilon_{\text{demand}}, \end{aligned}$$

And therefore the test is guaranteed to succeed. Again the last bound here holds under event $\mathcal{E}$. □

At some point in the execution of the algorithm, let L denote the set of learned states. Learned states are ones for which we have successfully called TD-ELIM, so that we may apply Theorem 3. Since we only ever call TD-ELIM through DFS-LEARN, the fact that these calls to TD-ELIM succeeded is implied by the event $\mathcal{E}$. A slightly tighter definition of L, which is sufficient for our purposes is

$$L(\mathcal{F}) = \bigcup_h \left\{ s \in \mathcal{S}_h : \max_{f \in \mathcal{F}} V^\star(s) - V^{f^\star}(s, \pi_f) \le 4\phi\sqrt{2K} + 2\phi + 40(H-h)\sqrt{K}\phi \right\}.$$

The only property we will use from Theorem 3 is the instantaneous risk bound, which is what this alternative definition of L provides.

For a policy π_f, let $q^{\pi_f}[s \to \bar{L}]$ denote the probability that when behaving according to π_f starting from state s, we visit an unlearned state. We now show that $q^{\pi_f}[\varnothing \to \bar{L}]$ is related to the risk of the policy π_f.

Lemma 8 (Policy Risk)**.** *Define L to be the set of states that have had* TD-ELIM *called on them and define $q^{\pi_f}[s \to \bar{L}]$ accordingly. Assume that $\mathcal{E}$ holds and let f be a surviving regressor, so that π_f is a surviving policy. Then,*

$$V^\star - V(\varnothing, \pi_f) \leq q^{\pi_f}[\varnothing \to \bar{L}] + 40\sqrt{K}\phi H^2.$$

Proof. Recall that under event $\mathcal{E}$, we can apply the conclusions of Theorem 3 with $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$ and $\tau_1 = 20(H-h)\sqrt{K}\phi$ for any h and state $s \in \mathcal{S}_h$ for which we have called TD-ELIM. Our proof proceeds by creating a recurrence relation through application of Theorem 3 and then solving the relation. Specifically, we want to prove the following inductive claim.

Inductive Claim: For a state $s \in L$ with η actions to go,

$$V^\star(s) - V(s, \pi_f) \leq 40\phi\sqrt{K}\eta^2 + q^{\pi_f}[s \to \bar{L}]$$

Base Case: With zero actions to go, all policies achieve zero reward and no policies visit $\bar{L}$ from this point, so the inductive claim trivially holds.

Inductive Step: For the inductive hypothesis, consider some state s at level h, for which TD-ELIM has successfully been called. There are $\eta = H - h + 1$ actions to go. By Theorem 5, we know that,

$$V^\star(s) - V^{f^\star}(s, \pi_f) \leq 4\phi\sqrt{2K} + 2\phi + 2\tau_1,$$

with $\tau_1 = 20(H-h)\phi\sqrt{K}$. This bound is clearly at most $40\eta\phi\sqrt{K}$. Now,

$$\begin{aligned} V^\star(s) - V(s, \pi_f) &= V^\star(s) - V^{f^\star}(s, \pi_f) + V^{f^\star}(s, \pi_f) - V(s, \pi_f) \\ &\leq 40\eta\phi\sqrt{K} + \mathbb{E}_{(x,r)\sim D_s} r(\pi_f(x)) + V^\star(s \circ \pi_f(x)) - r(\pi_f(x)) - V(s \circ \pi_f(x), \pi_f). \end{aligned}$$

Let us focus on just the second term, which is equal to,

$$\begin{aligned} &\mathbb{E}_{x\sim D_s}\left[(V^\star(s \circ \pi_f(x)) - V(s \circ \pi_f(x), \pi_f))\left(\mathbf{1}[\Gamma(s, \pi_f(x)) \in L] + \mathbf{1}[\Gamma(s, \pi_f(x)) \notin L]\right)\right] \\ &\leq \sum_{s' \in L} \mathbb{P}_{x\sim D_s}[\Gamma(s, \pi_f(x)) = s']\left(V^\star(s') - V(s', \pi_f)\right) + \mathbb{P}_{x\sim D_s}[\Gamma(s, \pi_f(x)) \notin L] \end{aligned}$$

Since all of the recursive terms above correspond only to states $s' \in L$, we may apply the inductive hypothesis, to obtain the bound

$$\begin{aligned} &40\eta\phi\sqrt{K} + \sum_{s' \in L} \mathbb{P}_{x\in D_s}[\Gamma(s, \pi_f(x)) = s']\left(40(h-1)^2\phi\sqrt{K} + q^{\pi_f}[s' \to \bar{L}]\right) + \mathbb{P}_{x\sim D_s}[\Gamma(s, \pi_f(x)) \notin L] \\ &\leq 40\eta\phi\sqrt{K} + 40(\eta-1)^2\phi\sqrt{K} + q^{\pi_f}[s \to \bar{L}] \\ &\leq 40\phi\sqrt{K}\eta^2 + q^{\pi_f}[s \to \bar{L}] \end{aligned}$$

Thus we have proved the inductive claim. Applying at the root of the tree gives the result. □

Recall that we set $\phi = \frac{\epsilon}{320H^2\sqrt{K}}$ in DFS-LEARN. This ensures that $40H^2\phi\sqrt{K} \leq \epsilon/8$, which means that if $q^{\pi_f}[\varnothing \to \bar{L}] = 0$, then we ensure $V^\star - V(\varnothing, \pi_f) \leq \epsilon/8$.

Lemma 9 (Each non-terminal iteration makes progress)**.** *Assume that $\mathcal{E}$ holds. If π_f is selected but fails the test, then with probability at least $1 - \exp(-\epsilon n_2/8)$, at least one of the n_2 trajectories collected visits a state $s \notin L$.*

Proof. First, if π_f fails the test, we know that,

$$\epsilon_{\text{demand}} < |\hat{V}(\varnothing, \pi_f) - \hat{V}^\star| \leq \epsilon/4 + |V(\varnothing, \pi_f) - V^\star|$$

which implies that,

$$\epsilon/4 < V^\star - V(\varnothing, \pi_f)$$

On the other hand Lemma 8, shows that,

$$V^\star - V(\varnothing, \pi_f) \leq q^{\pi_f}[\varnothing \to \bar{L}] + 40H^2\sqrt{K}\phi$$

Using our setting of ϕ, and combining the two bounds gives,

$$\epsilon/4 < q^{\pi_f}[\varnothing \to \bar{L}] + \epsilon/8 \Rightarrow q^{\pi_f}[\varnothing \to \bar{L}] > \epsilon/8$$

Thus, the probability that all n_2 trajectories miss $\bar{L}$ is,

$$\begin{aligned}\mathbb{P}[\text{all trajectories miss } \bar{L}] &= (1 - q^{\pi_f}[\varnothing \to \bar{L}])^{n_2} \\ &\leq (1 - \epsilon/8)^{n_2} \leq \exp(-\epsilon n_2/8).\end{aligned}$$

Thus we must hit $\bar{L}$ with substantial probability. □

H.1 Proof of Theorem 6

Again for now assume that $\mathcal{E}$ holds. First of all, by Lemma 6, we argued that if EXPLORE-ON-DEMAND terminates, then it outputs a policy that satisfies the PAC-guarantee. Moreover, by Lemma 7, we also argued that if EXPLORE-ON-DEMAND selects a policy that is at most $\epsilon/4$ suboptimal, then it terminates. Thus the goal of the proof is to show that it quickly finds a policy that is at most $\epsilon/4$ suboptimal.

Every execution of the loop in EXPLORE-ON-DEMAND either passes the test or fails the test at level ϵ_{demand}. If the test succeeds, then Lemma 6 certifies that we have found an ϵ-suboptimal policy, thus establishing the PAC-guarantee. If the test fails, then Lemma 9 guarantees that we call DFS-LEARN on a state that was not previously trained on. Thus at each non-terminal iteration of the loop, we call DFS-LEARN and hence TD-ELIM on at least one state $s \notin L$, so that the set of learned states grows by at least one. By Lemma 8 and our setting of ϕ, if we have called TD-ELIM on all states at all levels, then we guarantee that all surviving policies have risk at most $\epsilon/8$. Thus the number of iterations of the loop is bounded by at most MH since that is the number of unique states in the Contextual-MDP.

Bounding $\mathbb{P}[\bar{\mathcal{E}}]$: Since we have bounded the total number of iterations, we are now in a position to assign failure probabilities and bound the event $\mathcal{E}$. Actually we must consider not only the event $\mathcal{E}$ but also the event that all iterations where the test fails visit some state $s \notin L$. Call this new event $\mathcal{E}'$ which is the intersection of $\mathcal{E}$ with the event that all unsuccessful iterations visit $\bar{L}$.

We have δ probability to allocate, and we perform at most MH iterations. Thus in each iteration we may allocate $\delta/(MH)$ probability. There are three types of events required: (1) the initial Monte Carlo estimates $\hat{V}(\varnothing, \pi_f)$ must be close to $V(\varnothing, \pi_f)$, (2) the failure probability in Lemma 9 must be small, and (3) all Hn_2 calls to DFS-LEARN at this iteration must succeed. Naively, we allocate $1/3$ of the available failure probability to each.

For the initial Monte-Carlo estimate, by Hoeffding's inequality, we know that,

$$|\hat{V}(\varnothing, \pi_f) - V(\varnothing, \pi_f)| \leq \sqrt{\frac{\log(6MH/\delta)}{2n_1}}.$$

We want this bound to be at most $\epsilon/8$ which requires:

$$n_1 \geq \frac{32\log(6MH/\delta)}{\epsilon^2}$$

For the second event, Lemma 9 fails with probability at most $\exp(-n_2\epsilon/8)$. And for this to be smaller than $\delta/(3MH)$ we require,

$$n_2 \geq \frac{8\log(3MH/\delta)}{\epsilon}.$$

Both conditions on n_1 and n_2 are met by our choices in the algorithm specification.

Finally, for each of the Hn_2 calls to DFS-LEARN, we set the parameter to be $\delta/(3MH^2n_2)$, so that by Theorem 5, we may apply Theorem 3 at all states that we have called TD-ELIM on.

In total, if we set, $n_1 = \frac{32\log(6MH/\delta)}{\epsilon^2}$ and $n_2 = 8\log(3MH/\delta)/\epsilon$ in EXPLORE-ON-DEMAND and if EXPLORE-ON-DEMAND always call DFS-LEARN with parameter $\delta/(3MH^2n_2)$ we guarantee that the total failure probability for this subroutine is at most δ.

Sample Complexity: It remains to bound the sample complexity for the execution of EXPLORE-ON-DEMAND. We do at most MH iterations, and in each iteration we use n_1 trajectories to compute Monte-Carlo estimates, contributing an MHn_1 to the sample complexity. We also call DFS-LEARN on each of the Hn_2 prefixes collected during each iteration so that there are at most MH^2n_2 calls to DFS-LEARN in total. Naively, each call to DFS-LEARN takes at most $O(\frac{MH^5K^2}{\epsilon^2}\log(n_2NMKH/\delta))$ episodes, leading to a crude sample complexity bound of,

$$\tilde{\mathcal{O}}\left(\frac{M^2H^7K^2}{\epsilon^3}\log(N/\delta)\log(1/\delta)\right).$$

Recall that the $\tilde{\mathcal{O}}$ notation suppresses all logarithmic factors except those involving N and δ.

This bound can be significantly improved using a more careful argument. Apart from the first call to TD-ELIM in each application of DFS-LEARN, the total number of additional calls to TD-ELIM is bounded by MH since once we call TD-ELIM on a state, CONSENSUS always returns `true`.

Each call to TD-ELIM requires $n_{\text{train}} + Kn_{\text{test}}$ samples (because we always call CONSENSUS on all direct descendants before), and the total number of calls is at most,

$$MH^2n_2 + MH = \mathcal{O}\left(\frac{MH^2}{\epsilon}\log(MH/\delta)\right)$$

With our settings of n_{train} and n_{test}, the sample complexity is therefore at most,

$$\mathcal{O}\left(\frac{MH^6K^2}{\epsilon^3}\log(MHKN/(\epsilon\delta))\log(MH/\delta)\right)$$
$$= \tilde{\mathcal{O}}\left(\frac{MH^6K^2}{\epsilon^3}\log(N/\delta)\log(1/\delta)\right).$$

This concludes the proof of Theorem 6.