On Query-efficient Planning in MDPs under Linear Realizability of the Optimal State-value Function

Gellért Weisz
DeepMind, London, UK
University College London, London, UK

Philip Amortila
University of Illinois, Urbana-Champaign, USA

Barnabás Janzer
University of Cambridge, Cambridge, UK

Yasin Abbasi-Yadkori
DeepMind, London, UK

Nan Jiang
University of Illinois, Urbana-Champaign, USA

Csaba Szepesvári
DeepMind, London, UK
University of Alberta, Edmonton, Canada

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We consider the problem of local planning in fixed-horizon Markov Decision Processes (MDPs) with a generative model under the assumption that the optimal value function lies close to the span of a feature map. The generative model provides a restricted, "local" access to the MDP: The planner can ask for random transitions from previously returned states and arbitrary actions, and the features are also only accessible for the states that are encountered in this process. As opposed to previous work (e.g. Lattimore et al. (2020)) where linear realizability of *all* policies was assumed, we consider the significantly relaxed assumption of a single linearly realizable (deterministic) policy. A recent lower bound by Weisz et al. (2020) established that the related problem when the action-value function of the optimal policy is linearly realizable requires an exponential number of queries, either in H (the horizon of the MDP) or d (the dimension of the feature mapping). Their construction crucially relies on having an exponentially large action set. In contrast, in this work, we establish that $\mathrm{poly}(H, d)$ planning *is* possible with state value function realizability whenever the action set has a constant size. In particular, we present the TensorPlan algorithm which uses $\mathrm{poly}((dH/\delta)^A)$ simulator queries to find a δ-optimal policy relative to *any* deterministic policy for which the value function is linearly realizable with some bounded parameter (with a known bound). This is the first algorithm to give a polynomial query complexity guarantee using only linear-realizability of a single competing value function. Whether the computation cost is similarly bounded remains an interesting open question. We also extend the upper bound to the near-realizable case and to the infinite-horizon discounted MDP setup. The upper bounds are complemented by a lower bound which states that in the infinite-horizon episodic setting, planners that achieve constant suboptimality need exponentially many queries, either in the dimension or the number of actions.

1. Introduction

We are concerned with the problem of *planning* in large Markov Decision Processes (MDPs) using a *simulator* (or generative model), with a query complexity—that is, the number of calls to the simulator—independent of the size of the state space. While such a result is possible by running a Monte-Carlo tree search algorithm if we are concerned with finding out a good action with some computation every time a state is encountered, these methods require *exponential* in the planning horizon H number of queries, which, in the worst-case and without further assumptions, is unavoidable (Kearns et al., 2002).

In the hope of avoiding such exponential dependence, we consider the setting when the simulator gives the planner access to a feature mapping that maps states to d-dimensional vectors. The idea is to use the linear combination of features with some fixed parameter vector to approximate value functions in the MDP (e.g. Schweitzer and Seidmann, 1985). A basic question, then, is when and how such (linear) function approximation schemes enable query-efficient learning. One minimal assumption, which we consider here, is that the optimal value function $v^\star$ can be represented as a linear function of the feature mapping and an unknown d-dimension parameter. Finding this d-dimensional coefficient would then grant access to $v^\star$, and choosing a near-optimal action for a given state is then possible using low-cost one-step lookahead planning.

Despite that the number of unknowns is substantially reduced to d, it is not clear at all whether this setting is tractable. In fact, in a closely related setting where the optimal *state-action* value function $q^\star$ is linear, Weisz et al. (2020) have recently shown that the query complexity is still exponential in $\Omega(\min(H, d))$. Crucially, their construction relies on having exponentially many actions, leaving open the possibility that a small action set will enable polynomial query complexity.

In our setting, where $v^\star$ rather than $q^\star$ is realizable, finding a near-optimal action amongst A actions trivially requires $\Omega(A)$ queries, even when $v^\star$ is known. Thus, in order to enable polynomial-time learning, we consider the setting of *small actions sets*. To summarize, the central question we address in this paper is the following:

Is a polynomial query complexity achievable under linear realizability of $v^\star$, when the number of actions is $A = O(1)$?

We provide a positive result to this question in the *fixed-horizon* setting, where our algorithm TENSORPLAN enjoys a per-call query complexity $\mathrm{poly}((dH/\delta)^A)$, where H is the horizon and δ is the suboptimality target that the policy induced by continuously running the planning algorithm at every state encountered needs to satisfy. Given an input state at the beginning of the horizon, in its initialization phase, TENSORPLAN uses simulations to estimate the parameters of $v^\star$. In this and subsequent calls, given an input state, the estimated $v^\star$ is used by another procedure that uses additional simulations to compute one-step lookahead action-value estimates. We prove that the resulting policy loses at most δ total expected reward compared to optimality, regardless of the choice of the initial state, while the number of queries both for the initialization and the subsequent steps stays below the quoted polynomial bound.

In fact, TENSORPLAN works in a more general setting – it will automatically compete with the *best deterministic policy* whose value function is realizable by the features. This recovers the previously mentioned "classic" setting: when $v^\star$ is realizable the best deterministic policy is an optimal policy $\pi^\star$. Loosely, the initialization phase of our algorithm works in the following way: The algorithm keeps track of list of critical data that is used to refine a hypothesis set that contains

those d-dimensional parameter vectors that (may) induce a value function for some deterministic policy. Call these parameter vectors consistent. The algorithm refines its hypothesis set in a number of phases. For this, at the beginning of a phase, it chooses a parameter vector from the hypothesis set that maximizes the total predicted value at the initial state; an "optimistic choice". Next, the algorithm runs a fixed number of tests to verify that the parameter vector chosen gives a value function of some policy. If this consistency is satisfied, it also follows that the predicted value is almost as high as the actual value of the parameter-induced policy. As such, by its optimistic choice, the parameter vector gives rise to the policy whose value function is linearly realizable and whose value is the highest in the initial state. When the test fails, the hypothesis set is shrunk by expanding the list of critical data with data from the failed test. To show that the hypothesis set shrinks rapidly, we introduce a novel tensorization device which lifts the consistency checking problem to a d^A-dimensional Euclidean space where the tests become linear. This tensorization device allows us to prove that at most $O(d^A)$ constraints can be added (in the noise-free case) if there exist a deterministic policy with linearly realizable value function. We note that with minor modifications to the inputs and analysis, the query complexity guarantees of TENSORPLAN translate to the discounted MDP setting.

To complement the query complexity upper bound of TENSORPLAN, we show that a lower bound of $\Omega\left(2^{\min\{d,A/2\}}/d\right)$ applies for any planner with a constant suboptimality, albeit this lower bound is available only for the *infinite-horizon* episodic setting with a total cost criterion. The hard examples in the lower bound are (navigation) MDPs with deterministic dynamics and costs, and $\Theta(d)$ diameter and $\Theta(d)$ actions. Thus the three differences between the lower bound's setting and that of TENSORPLAN's upper bound are that the lower bound is in the *undiscounted infinite-horizon*, total *cost* setting and the number of actions *grows with* d, albeit only linearly.

The rest of the paper is structured as follows. The upcoming section (Section 2) introduces notations, definitions, and the formal problem definition. Section 3 gives the exponential lower bound for the infinite-horizon setting. Section 4 presents the TENSORPLAN algorithm for efficient planning in the finite-horizon setting, and states the query complexity guarantee (Theorem 4.2), as well as an extension of this result to the near-realizable case (Theorem 4.4) and infinite-horizon discounted case (Theorem 4.5). Section 5 discusses related work and we conclude in Section 6.

2. Preliminaries

We write $\mathbb{N}_+ = \{1, 2, \dots\}$ for the set of positive integers, $\mathbb{R}$ for the set of real numbers, and for $i \in \mathbb{N}_+$, $[i] = \{1, \dots, i\}$ for the set of integers from 1 to i. Given a measurable space $(\mathcal{X}, \Sigma)$, we write $\mathcal{M}_1(\mathcal{X})$ for the set of probability measures on that space (the σ-algebra will be understood from context). We write $\mathrm{supp}(\mu)$ for the support of a distribution μ.

We recall the notation and the most important facts about Markov Decision Processes (MDPs). For further details (and proofs) the reader is referred to the book of Puterman (1994). In this paper an MDP is given by a tuple $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, Q)$, where $(\mathcal{S}, \Sigma)$ is a measurable state space, $\mathcal{A}$ is a set of actions and for each $(s, a) \in \mathcal{S} \times \mathcal{A}$, $Q_{s,a} \in \mathcal{M}_1([0,1] \times \mathcal{S})$ is a probability measure on rewards and next-state transitions received upon taking action a at state s. Note that it follows that the random rewards are bounded in $[0, 1]$. We denote by r_{sa} the expected reward when using action a in state s: $r_{sa} = \int r Q_{sa}(dr, ds')$. Further, we let $P_{sa} \in \mathcal{M}_1(\mathcal{S})$ denote the distribution of the next-state: $P_{sa}(ds') = \int_{r \in \mathbb{R}} Q_{sa}(dr, ds')$. We assume that $\mathcal{A}$ is finite, and thus without loss of generality we let $\mathcal{A} = [A]$ for some integer $A \geq 2$.

In the **fixed-horizon setting** with horizon $H \geq 1$ the agent (a decision maker) interacts with the MDP in an H-step sequential process as follows: The process is initialized at a random initial state $S_1 \in \mathcal{S}$. In step $h \in [H]$, the agent first observes the current state $S_h \in \mathcal{S}$, then chooses an action $A_h \in \mathcal{A}$ based on the information available to it. The MDP then gives a reward R_h and transitions to a next-state S_{h+1}, where $(R_h, S_{h+1}) \sim Q_{S_h, A_h}$. After time-step H, the episode terminates.

The goal of the agent is to maximize the total expected reward $\sum_{h\in[H]} R_h$ for the episode by choosing the actions based on the observed past states and actions in the episode. A *(memoryless) policy* π takes the form $(\pi^{(h)})_{h\in[H]}$ where $\forall h \in [H]$, $\pi^{(h)} : \mathcal{S} \to \mathcal{M}_1(\mathcal{A})$. A *deterministic policy* π further satisfies that for any $h \in [H]$ and $s \in \mathcal{S}$ there exists $a \in \mathcal{A}$ such that $\pi^{(h)}(s) = \delta_a$ where δ is the Dirac delta distribution. Given a memoryless policy π, a state $s \in \mathcal{S}$ and step $h \in [H]$ within an episode, the value $v_h^\pi(s)$ is defined as the total expected reward incurred until the end of the episode when the MDP is started from s in step h and π is followed throughout. Writing $\mu f = \int f(s')\mu(ds')$ for the expected value of a measurable function $f : \mathcal{S} \to \mathbb{R}$ with respect to $\mu \in \mathcal{M}_1(\mathcal{S})$, these values are known to satisfy

$$v_h^\pi(s) = r_\pi(s) + P_\pi(s) v_{h+1}^\pi, \qquad s \in \mathcal{S},$$

where $v_{H+1}^\pi = 0$, $r_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) r_{sa}$, and $P_\pi(s)(ds') = \sum_{a\in\mathcal{A}} \pi(a|s) P_{sa}(ds')$. The maximum value achievable from a state $s \in \mathcal{S}$ when in step $h \in [H]$ is denoted by $v_h^\star(s)$. We also define $v_{H+1}^\star(s) = 0$, for convenience. We let $v^\star = (v_h^\star)_{h\in[H+1]}$ and call $v^\star$ the optimal value function. It is known that $v^\star$ satisfies the recursive *Bellman optimality equations*:

$$v_h^\star(s) = \max_{a\in\mathcal{A}} \left\{ r_{sa} + P_{sa} v_{h+1}^\star \right\}, \qquad s \in \mathcal{S}. \tag{1}$$

As is well known, the policy that in state $s \in \mathcal{S}$ chooses an action that maximizes the right-hand side of Eq. (1), is optimal. It also follows that there is always at least one optimal deterministic memoryless policy.

By taking $H \to \infty$, we obtain the **infinite-horizon total reward setting**. Here, policies are described by infinite sequences of probability kernels mapping histories to actions. The interconnection of a policy and an MDP puts a probability distribution over the space of infinitely long histories and the value of a policy π in state $s \in \mathcal{S}$ is the total expected reward incurred when the policy is started from state s. Note that the total expected reward may be undefined, or take on $\mathbb{R} \cup \{\pm\infty\}$. A policy is **admissible** if this value $v^\pi(s)$ is well-defined and not $-\infty$ no matter the initial state (we do not mind policies that generate infinite reward, but mind policies which generate infinite cost). In what follows we only consider MDPs in the infinite horizon setting where there exist at least one admissible policy. The optimal value $v^\star(s)$ of a state $s \in \mathcal{S}$, similarly to the finite-horizon case, is defined as the largest value that can be obtained by some admissible policy.

2.1. Featurized MDPs, feature map compatible optimal values

As noted earlier, we provide the planner with a feature mapping which captures the optimal value function. In the finite-horizon setting this translates to the existence of some $\theta^\star$ such that

$$v_h^\star(s) = \left\langle \varphi_h(s), \theta^\star \right\rangle, \qquad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{2}$$

We also consider the nearly-realizable case, where for some "misspecification" parameter $\eta \geq 0$ there exists some $\theta^\star$ such that

$$\left| v_h^\star(s) - \left\langle \varphi_h(s), \theta^\star \right\rangle \right| \leq \eta, \qquad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{3}$$

The parameter $\theta^\star$ is unknown to the planner in both cases. Here, $\varphi_h : \mathcal{S} \to \mathbb{R}^d$ is the so-called feature map. As will be described in more details in the next section, the planner is given *local access* to the feature map. That is, the planner can access $\varphi_h(s)$ for all the states $s \in \mathcal{S}$ that it has previously encountered while interacting with the simulator, but has no access to the features of other states. For convenience, in the finite horizon-setting we will also define $\varphi_{H+1}(s) = \mathbf{0}$ for all $s \in \mathcal{S}$, regardless of the other maps. An MDP together with a feature map $\varphi = (\varphi_h)_{h\in[H]}$ on its state-space is called a ***featurized MDP***. When Eq. (2) holds we say that $v^\star$ ***is (linearly) realizable by the feature map*** φ.

In this paper we consider a setting that relaxes linear realizability of the optimal value function. To define this setting we need the notion of ***v-linearly realizable policies***:

Definition 2.1 (v-linearly realizable policies)**.** *We say that a policy π is v-linearly realizable with misspecification $\eta \geq 0$ under the feature map $\varphi = (\varphi_h)_{h\in[H]}$ if there exists some $\theta \in \mathbb{R}^d$ such that its value function satisfies $\left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \leq \eta$ for all $h \in [H]$ and $s \in \mathcal{S}$. Furthermore, if θ satisfies $\|\theta\|_2 \leq B$ we say that π is B-boundedly v-linearly realizable with misspecification η under φ.*

In what follows we will be concerned with designing a planning algorithm that, given local access to a feature map, competes with the best v-linearly realizable memoryless deterministic (MLD) policy under that feature map (if one exists) in the following sense: For $B > 0$ and $\eta \geq 0$, define the function $v^\circ_{B,\eta} : \mathcal{S} \to \mathbb{R}$ as

$$v^\circ_{B,\eta}(s) = \sup\left\{v_1^\pi(s) \ : \pi \text{ is MLD and is } B\text{-boundedly } v\text{-linearly realizable with misspecification } \eta \text{ given } \varphi\right\}. \tag{4}$$

We call will $v^\circ_{B,\eta}$ the ***φ-compatible optimal value function*** *at scale B and misspecification η*. Note that if there are no v-linearly realizable policies with misspecification η in an MDP, $v^\circ_{B,\eta}(s) \equiv -\infty$ for each state $s \in \mathcal{S}$ of the MDP. Competing with the best v-linearly realizable MLD policy (at scale $B > 0$ and misspecification $\eta \geq 0$) means the ability to generate actions of a policy whose value function is close to $v^\circ_{B,\eta}$ (for the fully formal definition, see the next section). Note that if the optimal value function of an MDP is linearly realizable with parameter vector $\theta^\star$ and misspecification η' then for any $B \geq \|\theta^\star\|_2$, $v^\circ_{B,\eta} = v^\star$ for any $\eta \geq \eta'$. Hence, the setting we introduce generalizes the one where the optimal value function is exactly or near-realizable with B-bounded parameter vectors.

In the **infinite-horizon setting**, the constraint equivalent to linear realizability of the optimal value function is that with some $\theta^\star \in \mathbb{R}^d$, again unknown to the planner,

$$\left|v^\star(s) - \left\langle \varphi(s), \theta^\star\right\rangle\right| \leq \eta, \qquad \text{for all } s \in \mathcal{S},$$

where now the feature map is $\varphi : \mathcal{S} \to \mathbb{R}^d$, i.e., with no dependence on the step within the episode. Featurized MDPs then are defined by joining an MDP with such a (homogeneous) feature map. The other notions introduced above have also natural counterparts. To save space, these are not introduced explicitly, trusting that the reader can work out the necessary modifications to the definitions without introducing any ambiguity.

2.2. Local Planning

In the ***fixed-horizon local planning*** problem, a ***planner*** is given an input state and is tasked with computing a near-optimal action for that state while interacting with a black-box that simulates the MDP. In the ***(state-)featurized local planning*** problem, the black-box also returns the feature-vector

of the next state. Access to the black-box is provided by means of calling a function SIMULATE, whose semantics is essentially as just described, but will be further elaborated on below.

More formally, the planner needs to "implement" a function, which we call `GetAction` and whose semantics, in the context of fixed-horizon MDPs, is as follows:

Definition 2.2 (`GetAction`$(d, A, H,$ SIMULATE$, s, h, \varphi_h(s), \delta, B)$)**.** *The meaning of inputs is as follows: d is the dimension of the underlying feature map, A is the number of actions, H is the episode length,* SIMULATE *is a function that provides access to the oracle that simulates the MDP, s is the state where an action is needed at stage $h \in [H]$, $\delta > 0$ is a suboptimality target, and B is the parameter vector bound. This function needs to return an action in $\mathcal{A}$ with the intent that this is a "good action" to be used at stage h when the state is s.*

Given a featurized MDP and a planner as described above, the planner ***induces a (randomized, possibly memoryful) policy***, which is the policy that results from calling `GetAction` along a trajectory and following its recommended actions. If the initial state is $S_1 = s_0 \in \mathcal{S}$, the first action taken by this policy is $A_1 =$ `GetAction`$(\dots, S_1, 1, \dots)$, the second is $A_2 =$ `GetAction`$(\dots, S_2, 2, \dots)$ where $S_2 \sim P_{S_1,A_1}$, etc. If `GetAction` does not save data between the calls, the resulting policy would be memoryless, but this is not a requirement. *In fact, we require that* `GetAction` *is first called with $h = 1$ and then $h = 2$, etc.* A practical planner which is used across multiple episodes can also save data between episodes. In this case `GetAction` can be called with $h = 1$ after being called with $h = H$, designating the start of a new episode. For now, we assume that this is not the case, as this allows for cleaner definitions.[1]

Inside `GetAction` the planner can issue any number of calls to SIMULATE. The function SIMULATE takes as inputs a state-stage-action triplet (s, h, a). In response, SIMULATE returns a triplet $(R, S', \varphi_{h+1}(S'))$ where (R, S') is a "fresh" random draw from $Q(s, a)$. For generality the simulator is also allowed some inaccuracy, in the sense that it returns $([R + \Lambda_{sa}]_0^1, S', \varphi_{h+1}(S'))$ where $\Lambda_{sa} \in \mathbb{R}$ is a constant satisfying $|\Lambda_{sa}| \le \lambda$, for some $\lambda \ge 0$ that we call the simulator's accuracy, and $[x]_0^1 = \max(0, \min(1, x))$ (ie. inaccurate rewards are clipped in $[0, 1]$). Neither Λ_{sa} nor λ are known to the planner. The planner can only access states that it is given access to either when `GetAction` is called, or returned by a call to SIMULATE. The same holds for the features of the states. We note that this is essentially the same setting as what is called *sampling with state revisiting* by Li et al. (2021).

The ***quality of a planner*** is, on one hand, assessed based on the quality of the policy that it induces and, on the other hand, by its ***worst-case (per-episode) query-cost***, which is defined as the largest total query-cost (ie. number of calls to SIMULATE made by `GetAction`) encountered while running the planner for the H stages of an episode, starting at stage $h = 1$. Our lower bounds in Section 3 will be given in terms of the ***worst-case (per-state) query-cost***, which is defined as the largest query-cost encountered during any single call to `GetAction`. Note that the switch from per-episode to per-state cost only strengthens the lower bound as the worst-case per-episode query-cost is at least as large as the worst-case per-state query-cost.

Definition 2.3 (Sound planner)**.** *Let $B, \delta > 0$, $\lambda, \eta \ge 0$, $H \ge 1$. A planner is (δ, B)-**sound** with simulator accuracy λ and misspecification η if for any featurized H-horizon MDP $(\mathcal{M}, \varphi)$ with rewards bounded in $[0, 1]$ and with 1-bounded feature maps (i.e. for all $h \in [H]$, $s' \in S$,*

1. Jumping a bit ahead of ourselves, if we cared about long-run average per-state query-complexity, one could perhaps do better by allowing planners to save data between episodes.

$\|\varphi_h(s')\|_2 \le 1$*), the (random) H-horizon policy π that the planner induces while interacting with the λ-accurate simulation oracle satisfies*

$$v_1^\pi(s) \ge v_1^\circ(s) - \delta \quad \textit{for all } s \in \mathcal{S}\,, \tag{5}$$

where v^π is the H-horizon value function of π in $\mathscr{M}$ and $v^\circ = v^\circ_{B,\eta}$ is the H-horizon φ-compatible optimal value function of $\mathscr{M}$ (cf. Equation (4)).

The reader interested in exploring alternatives to the protocol described here is referred to Appendix A, where we also discuss "nuances" like how to work with "arbitrary" state spaces.

Further notations For $v \in \mathbb{R}^d$, and $a \le b \le d$ positive integers, let $v_{a:b} \in \mathbb{R}^{b-a+1}$ be the vector corresponding to the entries with indices in $\{a, a+1, \dots, b\}$, i.e., $(v_{a:b})_i = v_{a+i-1}$. For $x \in \mathbb{R}, v \in \mathbb{R}^d$, let $\overline{xv} \in \mathbb{R}^{d+1}$ denote the concatenation of x and v such that $(\overline{xv})_1 = x$ and $(\overline{xv})_{2:d+1} = v$. We write $\otimes : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \mapsto \mathbb{R}^{d_1 \times d_2}$ for the *tensor product* of two vectors, defined as $(u \otimes v)_{i \in [d_1], j \in [d_2]} = u_i \cdot v_j$ For a larger set of vectors $(u^{(1)}, \dots u^{(n)})$, $n \in \mathbb{N}_+$, we write

$$\left(\otimes_{i\in[n]} u^{(i)}\right)_{j_1, j_2, \dots, j_n} = \prod_{i \in [n]} (u^{(i)})_{j_i} \qquad u^{(i)} \in \mathbb{R}^{d_i}.$$

We will frequently view tensors in $\mathbb{R}^{\times_{i\in[n]} d_i}$ as vectors in $\mathbb{R}^{\prod_{i\in[n]} d_i}$ via the usual isomorphism. Notationally, this is given by the *vectorize* operation. Letting $\flat : [d_1] \times \cdots \times [d_n] \mapsto [\prod d_i]$ be any bijection between indices, we define: $(\text{vectorize}(T))_{\flat(j_1,\dots,j_n)} = T_{j_1, j_2, \dots, j_n}$, for $T \in \mathbb{R}^{\times_{i\in[n]} d_i}$. Let us define the inner product of two compatible tensors to be

$$\left\langle \otimes_{i\in[n]} u^{(i)}, \otimes_{i\in[n]} v^{(i)} \right\rangle = \left\langle \text{vectorize}(\otimes_{i\in[n]} u^{(i)}), \text{vectorize}(\otimes_{i\in[n]} v^{(i)}) \right\rangle .$$

The key property which we need is that inner product between the two tensors is then seen to be:

$$\left\langle \otimes_{i\in[n]} u^{(i)}, \otimes_{i\in[n]} v^{(i)} \right\rangle = \prod_{i\in[n]} \left\langle u^{(i)}, v^{(i)} \right\rangle \tag{6}$$

3. Exponential query complexity for infinite-horizon problems

In this section, we show that in infinite-horizon episodic problems, the query complexity of $(\frac{1}{2}, \sqrt{d})$-sound planners is exponential in the dimension of the feature map d, even with no misspecification (ie. $\eta = 0$):

Theorem 3.1. *Fix $d > 1$. For any local planner P whose worst-case per-state query-cost is at most β_{sim} there exists a featurized MDP with d-dimensional features and $2d - 1$ actions, such that the optimal value function of the MDP is exactly realizable with the features (Eq. 2) and takes values in the $[-2, 0]$ interval, the 2-norm of feature vectors is $O(1)$, the 2-norm of the parameter vector of the optimal value function is $O(\sqrt{d})$ and the suboptimality δ of the policy induced by P in the MDP satisfies*

$$\delta \ge \frac{2^{d-2}}{d(\beta_{sim} + 1)} - 1\,.$$

In particular, to guarantee $\delta \le 0.5$, the query-cost must satisfy $\beta_{sim} \ge \Omega\left(2^{\min\{d, A/2\}}/d\right)$.

We provide a proof sketch only. The full proof is given in Appendix B.

Sketch. Consider an MDP whose state space is $\{-1,0,1\}^d$, with deterministic dynamics: In every state, there are at most $2d$ actions to increment or decrement one coordinate (1 or -1 can only be changed to 0, 0 can be changed to either -1 or $+1$) and there is an additional action that does not change the state. Instead of rewards, it will be more convenient to consider costs. The cost everywhere is 1, except for staying put in a special state, $s^\star$, where the cost is zero. Thus, the optimal policy at any state takes the shortest path to $s^\star$, which gives that $v^\star(s)$ is the ℓ^1 distance between s and $s^\star$. Restricting $s^\star$ to take values in $\{-1,1\}^d$, setting $\varphi(s) = \overline{ds}$, we have $v^*(s) = \left\langle \varphi(s), \overline{(-1)s^\star} \right\rangle$. A planner that takes less than 2^{d-1} queries has no better than $1/2$ chance of discovering $s^\star$ (which is one of 2^d possibilities). If a planner does not discover $s^\star$, the actions it takes are uninformed. Hence, for any planner that uses a small query budget, one can hide $s^\star$ and force the planner to wander around in the MDP for a long time. One can then scale costs to finish the proof. ■

4. Efficient planning for the finite-horizon setting

In this section, we present TENSORPLAN (Algorithm 1) and prove its soundness (cf. Definition 2.3) and efficiency (Theorem 4.2). We start with a high-level description of the main ideas underlying the planner. Initially, we only prove soundness for exact realizability (ie. $\eta = 0$), which we later generalize in Theorem 4.4.

The planner belongs to the family of generate-and-test algorithms. To describe it, let $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, Q)$ denote the MDP that the planner interacts with and let $\varphi = (\varphi_h)_{h\in[H]}$ be the underlying feature map. Further, let $\Theta^\circ \subset \mathbb{R}^d$ be the set which collects the parameter vectors of the value functions of B-boundedly v-linearly realizable DML policies with misspecification $\eta = 0$ (Definition 2.1). That is, Θ° is such that for any $\theta \in \Theta^\circ$, $\|\theta\|_2 \le B$ and for some DML policy π of $\mathcal{M}$,

$$v_h^\pi(s) = \langle \varphi_h(s), \theta \rangle \qquad \text{for all } h \in [H] \text{ and } s \in \mathcal{S}. \tag{7}$$

Let s_0 be the state which the planner is called for. The algorithm will maintain a subset Θ of $\mathbb{R}^d$ such that, with high probability, $\Theta^\circ \subset \Theta$. The set is initialized to the ℓ^2-ball of radius B, which obviously satisfies this constraint. Given the set Θ of admissible parameter vectors and $s_0 \in \mathcal{S}$, the planner finds the *optimistic* parameter vector $\theta^+ = \arg\max_\Theta \langle \varphi_1(s_0), \theta \rangle$ from the set Θ. Let us write $v_h(s;\theta) \coloneqq \langle \varphi_h(s), \theta \rangle$. If $\theta \in \Theta^\circ$ then for any $h \in [H]$ and $s \in \mathcal{S}$, since the policies defining Θ° are deterministic, it follows that there exists an action $a \in \mathcal{A}$ such that

$$v_h(s;\theta) = r_{sa} + P_{sa} v_{h+1}(\cdot;\theta)\,. \tag{8}$$

For any θ, let π_θ denote the policy which chooses the action satisfying the above equation when in state s and stage h (when there is no action that satisfies the consistency condition Eq. (8), the policy can choose any action).

To test whether $\theta^+ \in \Theta^\circ$, the algorithm aims to "roll out" π_{θ^+}. By this, we mean that upon encountering a state s in stage h in such a rollout, the algorithm checks whether there is an action a that satisfies Eq. (8). If such an action is found, it is sent to the simulator, which responds with the next state. If no such action is found, the test fails – this means that $\theta^+ \notin \Theta^\circ$. When this happens, the data corresponding to the transition where the test failed is used to refine the set of admissible

parameter vectors and a new admissible set Θ' is established. Assuming that the test failed at stage $h^\star$ and state $s^\star$, this new set is

$$\Theta' = \{\theta \in \Theta \,:\, \exists a \in \mathcal{A} \text{ s.t. } Eq.\ (8) \text{ holds with } s = s^\star \text{ and } h = h^\star\}\,.$$

Then the testing of θ^+ is abandoned, Θ is updated to Θ', and the process is repeated. Clearly, $\Theta^\circ \subset \Theta'$ still holds, so $\Theta^\circ \subset \Theta$ also holds after the update.

When a rollout continues up to the end of the episode without failure, the algorithm is given some evidence that $\theta^+ \in \Theta^\circ$, but this evidence is weak. This is because the states encountered in a rollout are random, and the trajectory generated may just happen to avoid the "tricky" states where the consistency test would fail. Luckily though, if the algorithm keeps testing with multiple rollouts and the tests do not fail for a sufficiently large (but not too large) number of such rollouts, this can be taken as evidence that π_{θ^+} is indeed a good policy in starting state s_0. It may happen that θ^+ is still not in Θ°, but the value of π_{θ^+} cannot be low.

This is easy to see, if for the moment we add a further, (seemingly) stronger test. This test checks whether $v_1(s_0;\theta^+)$ correctly predicts the value of π_{θ^+} in state s_0. To this end, the test simply takes the average sum of rewards along the rollouts. If we detect that $v_1(s_0;\theta^+)$ is not sufficiently close to the measured average value, the test fails. If this strengthened test does not fail either then this is strong evidence that $v_1^{\pi_{\theta^+}}(s_0)$ is as high as $v_1(s_0;\theta^+)$. Now, since $\Theta^\circ \subset \Theta$ holds throughout the execution of the algorithm, $v_1(s_0;\theta^+) \geq \max_{\theta\in\Theta^\circ} v_1^{\pi_\theta}(s_0) = v_B^\circ(s)$ (since we pick θ^+ optimistically), and hence policy π_{θ^+} can successfully compete with the best v-linearly realizable policy in $\mathscr{M}$ under φ and at scale B (Eq. (4)).

To complete the description of the algorithm, there are three outstanding issues. The first is that due to the randomizing simulation oracle, for any given state $s \in \mathcal{S}$, one can only check whether Eq. (8) holds up to some fixed accuracy and only with high probability. Luckily, this does not cause any issues – when the tests fail, the parameters can be set so that $\Theta^\circ \subset \Theta$ is still maintained.

The second issue is whether the algorithm is efficient. (So far we have been concerned only with soundness.) This is addressed by "tensorizing" the consistency test. For $\psi : \mathcal{S} \to \mathbb{R}^d$, we let $P_{sa}\psi = \int \psi(s')P_{sa}(ds')$. Using Eq. (6) we then observe that the existence of an action such that Eq. (8) holds is equivalent to:

$$\begin{aligned} 0 &= \prod_{a\in\mathcal{A}} r_{sa} + \langle P_{sa}\varphi_{h+1} - \varphi_h(s), \theta\rangle = \prod_{a\in\mathcal{A}} \left\langle \overline{r_{sa}\,(P_{sa}\varphi_{h+1} - \varphi_h(s))}, \overline{1\theta} \right\rangle \\ &= \left\langle \otimes_{a\in\mathcal{A}} \overline{r_{sa}\,(P_{sa}\varphi_{h+1} - \varphi_h(s))}, \otimes_{a\in\mathcal{A}} \overline{1\theta} \right\rangle\,. \end{aligned}$$

Now, defining $M_\theta = \otimes_{a\in\mathcal{A}} \overline{1\theta}$ and $T_s = \otimes_{a\in\mathcal{A}} \overline{r_{sa}\,(P_{sa}\varphi_{h+1} - \varphi_h(s))}$, we see that $\theta \in \Theta^\circ$ is equivalent to that $\langle T_s, M_\theta\rangle = 0$ holds for all $s \in \mathcal{S}$. Testing a parameter vector at some state is equivalent to checking whether M_θ is orthogonal to T_s. Clearly, the maximum number of tests that can fail before identifying an element of Θ° is at most d^A, the dimension of M_θ. Since our tests are noisy, we use an argument based on eluder dimensions (which allow imperfect measurements) to complete our efficiency proof (Russo and Van Roy, 2014).

The final issue is really an optimization opportunity. In our proposed algorithm we do not separately test if the value estimates at s_0 are close to the empirical return over the rollouts, and instead rely only on the consistency tests. This can be done since, when consistency holds, the expected total reward in an episode is close to the predicted value. This follows from a telescoping

argument. Let $S_1 = s_0, A_1, S_2, A_2, \ldots, S_H, A_H, S_{H+1}$ be the state-action pairs in a rollout where the tests do not fail, and note that

$$v_1^{\pi_{\theta^+}}(s_0) = \mathbb{E}_{\pi_{\theta^+}}\left[\sum_{h=1}^{H} r_{S_h,A_h}\right] = \mathbb{E}_{\pi_{\theta^+}}\left[\sum_{h=1}^{H} v_h(S_h;\theta^+) - v_{h+1}(S_{h+1};\theta^+)\right] = v_1(s_0;\theta^+),$$

where the first equality uses the definition of $v^{\pi_{\theta^+}}$, the second equality uses $r_{S_h,A_h} = v_h(S_h;\theta^+) - P_{S_hA_h}v_{h+1}(\cdot;\theta^+)$, and the last equality uses that $v_{H+1} \equiv 0$. When measurements are noisy, a similar telescoping argument gives that with high probability, $v_1^{\pi_{\theta^+}}(s_0)$ is almost as high as $v_1(s_0;\theta^+)$ when consistency tests do not fail for a number of rollouts.

4.1. The TENSORPLAN algorithm

The pseudocode of `GetAction` of TENSORPLAN is shown in Algorithm 1.

Algorithm 1 TENSORPLAN.`GetAction`

Inputs: $d, A, H,$ SIMULATE$, s, h, \varphi_h(s), \delta, B$
if $h = 1$ **then** ▷ Initialize global θ^+
  TENSORPLAN.`Init`$($
    $d, A, H,$ SIMULATE$, s, \varphi_1(s), \delta)$
end if
$\overline{\Delta}_\cdot \leftarrow$ APPROXTD$(s, h, \varphi_h(s), A, n_2,$ SIMULATE$)$
Access θ^+ saved by TENSORPLAN.`Init`
return $\arg\min_{a\in[A]} \left|\left\langle \overline{\Delta}_a, \overline{1\theta^+}\right\rangle\right|$

Algorithm 2 APPROXTD

Inputs: $s, h, \varphi_h(s), A, n,$ SIMULATE
for $a = 1$ to A **do**
  for $l = 1$ to n **do**
    $(R_l, S'_l, \varphi_{h+1}(S'_l)) \leftarrow ($
      SIMULATE$(s, h, a))$
    $\tilde{\Delta}_l \leftarrow \overline{R_l\left(\varphi_{h+1}(S'_l) - \varphi_h(s)\right)}$
  end for
  $\Delta_a := \frac{1}{n}\sum_{l\in[n]} \tilde{\Delta}_l$
end for
return $(\Delta_a)_{a\in[A]}$

The main workhorse of TENSORPLAN is the initialization routine, TENSORPLAN.`Init` (Algorithm 3), which generates a global variable $\theta^+ \in \mathbb{R}^d$ that is an estimate for the parameter of the best realizable value function v_B°. Within an episode, this parameter is used by the current and subsequent calls to `GetAction`. In particular, given θ^+, `GetAction` approximately implements π_{θ^+} of the previous section. For this, `GetAction` calls APPROXTD[2] (Algorithm 2), which produces an estimate of $\overline{r_{sa}(P_{sa}\varphi_{h+1} - \varphi_h(s))}$ for all actions $a \in \mathcal{A}$.

The `Init` function uses

$$\mathrm{Sol}(\Delta_1, \ldots, \Delta_\tau) = \left\{\theta \in \mathbb{R}^d \;:\; \|\theta\|_2 \le B, \forall i \in [\tau] \;:\; \left|\left\langle \Delta_i, \otimes_{a\in[A]}\overline{1\theta}\right\rangle\right| \le \frac{H^A\varepsilon}{2\sqrt{E_d}}\right\}. \tag{9}$$

where ε is a function of the target suboptimality and $E_d = \tilde{O}\left(d^A A\right)$, defined in Eq. (15), is an upper bound on the the eluder dimension of a tensorized clipped-linear function class (cf. Eq. (16)). The $\mathrm{Sol}(\cdot)$ set stands for the successfully refined sets Θ of the previous section and its arguments $\Delta_i \in \mathbb{R}^{(d+1)^A}$ correspond to estimates of $\otimes_{a\in\mathcal{A}}\overline{r_{sa}(P_{sa}\varphi_{h+1} - \varphi_h(s))}$ for the various states s and stages h where the algorithm detects a failure of the consistency test it runs. Estimates of these in `Init` are obtained by calls to APPROXTD.

2. Thusly named since $\left\langle \overline{r_{sa}(P_{sa}\varphi_{h+1} - \varphi_h(s))}, \overline{1\theta}\right\rangle$ corresponds to the "temporal difference" error of value function v_θ at state-action pair (s,a) (Sutton, 1988).

Algorithm 3 TENSORPLAN.`Init`

1: **Inputs:** $d, A, H,$ SIMULATE, $s_0, \varphi_1(s_0), \delta$
2: $X \leftarrow \{\}$ ▷ X is a list
3: Initialize $\zeta, \varepsilon, n_1, n_2, n_3$ via equations (10), (11), (12), (13), (14), respectively.
4: **for** $\tau = 1$ to $E_d + 2$ **do**
5: Choose any $\theta_\tau \in \arg\max_{\theta \in \mathrm{Sol}(X)} \langle \varphi_1(s_0), \theta \rangle$ ▷ Optimistic choice
6: `CleanTest` $\leftarrow$ `true`
7: **for** $t = 1$ to n_1 **do** ▷ n_1 rollouts with θ_τ-induced policy
8: $S_{\tau t1} = s_0$ ▷ Initialize rollout
9: **for** $j = 1$ to H **do** ▷ Stages in episode
10: $\overline{\Delta}_{\tau tj,\cdot} \leftarrow$ APPROXTD$(S_{\tau tj}, j, \varphi_j(S_{\tau tj}), A, n_2,$ SIMULATE$)$
11: **if** `CleanTest` and $\min_{a \in [A]} \left| \left\langle \overline{\Delta}_{\tau tja}, \overline{1\theta_\tau} \right\rangle \right| > \frac{\delta}{4H}$ **then** ▷ Consistency failure?
12: $\hat{\Delta}_{\tau tj,\cdot} \leftarrow$ APPROXTD$(S_{\tau tj}, j, \varphi_j(S_{\tau tj}), A, n_3,$ SIMULATE$)$ ▷ Refined data
13: X.append $\left(\otimes_{a \in [A]} \hat{\Delta}_{\tau tja} \right)$ ▷ Save failure data
14: `CleanTest` $\leftarrow$ `false` ▷ Not clean anymore
15: **end if**
16: $A_{\tau tj} \leftarrow \arg\min_{a \in [A]} \left| \left\langle \overline{\Delta}_{\tau tja}, \overline{1\theta_\tau} \right\rangle \right|$ ▷ Find most consistent action
17: $(R_{\tau tj}, S_{\tau tj+1}, \varphi_{j+1}(S_{\tau tj+1})) \leftarrow$ SIMULATE$(S_{\tau tj}, j, A_{\tau tj})$ ▷ Roll forward
18: **end for**
19: **end for**
20: **if** `CleanTest` **then break** ▷ Success?
21: **end for**
22: Save into global memory $\theta^+ \leftarrow \theta_\tau$

Note that `Init` as described continues to generate rollout data even after a consistency test fails. This is clearly superfluous and in an optimized implementation one could break out of the test loop to generate the next candidate immediately after a failure happens. The only reason the algorithm is described in the way it is done here is because this allows for a cleaner analysis: every policy will have access to data from n_1 rollouts, even if the policy fails a consistency test.

Remark 4.1. *The reader might wonder why* TENSORPLAN *follows the most consistent action in Line 7 of* `GetAction`*, instead of the best action according to its θ^+, which would be* $\arg\max_{a \in [A]} \left\langle \overline{\Delta}_a, \overline{1\theta^+} \right\rangle$*. Indeed, a practical implementation might adopt this, together with the same change to Line 16 of* `Init`*, and a strengthening of the consistency test of* `Init`*'s Line 11 to require that* the best *action (according to θ_τ) be consistent, instead of* any *action. This test would fail if* $\left| \max_{a \in [A]} \left\langle \overline{\Delta}_{\tau tja}, \overline{1\theta_\tau} \right\rangle \right| > \frac{\delta}{4H}$*. One might hope that this strengthened consistency test improves sample efficiency, and indeed the proofs go through (giving the same query complexity bounds), albeit with a significant weakening of the final guarantee: this version of* TENSORPLAN *could* only *compete with optimal policies that are realizable, instead of the best of* all *realizable DML policies.* TENSORPLAN*, as presented, is able to compete with the latter, with its only source of pressure to do well coming from the optimistic choice of θ_τ in Line 5 of* `Init`*.*

The following theorem gives a query complexity guarantee on using TENSORPLAN to find a near-optimal policy. The precise values of $\zeta, \varepsilon, n_1, n_2,$ and n_3 mentioned in the theorem can be found

in Appendix C. For the theorem statement recall that B is the bound on the 2-norm of value-function parameter vectors that the algorithm competes with.

Theorem 4.2. *For any $\delta > 0$ and $B > 0$, there exists values of $\zeta, \varepsilon, n_1, n_2$, and n_3 such that the* TENSORPLAN *algorithm (Algorithm 1) is (δ, B)-sound (Definition 2.3) with misspecification $\eta = 0$ and simulator accuracy $\lambda \leq \varepsilon/(4\sqrt{E_d}) = \tilde{O}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A / \sqrt{A}\right)$ for the H-horizon planning problem with worst-case per-episode query-cost*

$$\tilde{O}\left(d^A A^3 B^2/\delta^2 \left(H^3 B^2 dA/\delta^2 + d^A A^2 H^{4A+2} 12^{2A}/\delta^{2A}\right)\right) = \text{poly}\left((dH/\delta)^A, B\right).$$

Corollary 4.3. *When the optimal value function $v^\star$ is linearly realizable with the given feature map with misspecification $\eta = 0$, then* TENSORPLAN*, given access to a simulator with accuracy $\lambda \leq \varepsilon/(4\sqrt{E_d})$ induces a policy π within the budget constraints of Theorem 4.2 for which $v_1^\pi(s_0) \geq v_1^\star(s_0) - \delta$.*

Proof (of Theorem 4.2). We provide here a very brief sketch, and defer the full proof to Appendix C. The proof proceeds in a few steps. First, fix any starting state $s_0 \in \mathcal{S}$ and any $\theta^\circ \in \Theta^\circ$. Appendix C.1 establishes that despite the simulator's inaccuracy, the estimates $\hat{\Delta}$ and $\overline{\Delta}$ are close to their respective expected values (Lemma C.1) and that $\left\langle \overline{\Delta}, \theta^\circ \right\rangle$ is close to its expected value (Lemma C.2). This entails that θ° does not get eliminated from the solution set (Lemma C.3). In Appendix C.2, we use the eluder dimension to bound the maximal length of X (essentially, the list of states where consistency is broken). It follows that, with high probability, the iteration over τ will be exited in Line 20 with `CleanTest` being true for $\tau \leq E_d + 1$. The last subsection (Appendix C.3) bounds the suboptimality of the policy induced by θ^+ in terms of the inner product between θ^+ and the measured TD vectors (Lemma C.6). We then bound these suboptimalities by the desired suboptimality (Corollary C.7) and finally establish in Corollary C.8 that the policy induced by the planner is δ-optimal compared to $v_1(s_0; \theta^\circ)$. Since this argument holds for any $s_0 \in \mathcal{S}$ and $\theta^\circ \in \Theta^\circ$, the planner is (δ, B)-sound according to Definition 2.3. ∎

Our next theorem generalizes the previous results to the misspecified case (ie. $\eta > 0$) by trading off simulator accuracy for misspecification. Formally, we provide a reduction to the realizable case and run TENSORPLAN with a slightly modified simulation oracle SIMULATE′ which requires no additional information beyond that provided by the original simulator. The proof is deferred to Appendix D. The main idea of the proof is to define an alternate MDP with an expanded state space where states are indexed by which stage they belong to so that the misspecification error of a target policy can be "pushed" into the rewards of the new MDP. This way, the target policy will not have misspecification errors. The simulator for the new MDP still reports the rewards from the original MDP, but this is allowed since the previous result was stated for the case when the simulator introduces (small) errors when reporting the rewards.

Theorem 4.4. *For any $\delta, B > 0$,* TENSORPLAN *is (δ, B)-sound with misspecification $\eta \leq \varepsilon/(12\sqrt{E_d})$ and simulator accuracy $\lambda \leq \varepsilon/(12\sqrt{E_d})$ with worst-case per-episode query-cost* $\text{poly}\left((dH/\delta)^A, B\right)$*, when run with input $\delta' = 0.98\delta$ and simulation oracle* SIMULATE′*.*

4.2. Discounted MDPs

In the discounted MDP setting, instead of maximizing the expected value of the reward $\sum_{h\in[H]} R_h$ over a horizon H, the goal of the agent is to maximize the expected value of the discounted total reward, $\sum_{h\in\mathbb{N}_+} \gamma^{h-1} R_h$, over an infinite horizon, where $0 \le \gamma < 1$ is a fixed discount factor, given to the agent. The value function for a policy π, $v^\pi : \mathcal{S} \to \mathbb{R}$ is defined as $v^\pi(s) = r_\pi(s) + \gamma P_{sa} v^\pi$. The stage index h is dropped from the feature mapping ($\varphi : \mathcal{S} \to \mathbb{R}^d$), and the definition of v-linearly realizable policies (Definition 2.1) changes from requiring $\left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \le \eta$ to requiring

$$|v^\pi(s) - \langle \varphi(s), \theta\rangle| \le \eta \qquad \text{for all } s \in \mathcal{S}\,.$$

Soundness is otherwise defined identically to the H-horizon case, except for swapping the value function to v^π. Importantly, value guarantees are only required for the initial state the planner is called with, and not for every state that the planner ever encounters. As the episodes are infinitely long in this setting, we use the per-state (instead of per-episode) query-cost.

We use a reduction of the discounted case to the finite-horizon case with an "effective horizon" $H_{\gamma,\delta}$. Our next theorem shows that the guarantees of TENSORPLAN in the $H_{\gamma,\delta}$-horizon setting transfer to the discounted setting if it is run with a slightly modified simulation oracle $\text{SIMULATE}^{\gamma,\delta}$, which once again does not require any additional information beyond that of the original simulation oracle. As this is a reduction, the input h given to TENSORPLAN's `GetAction` should be incremented for each transition, exactly as in the finite-horizon case. The definition of $H_{\gamma,\delta}$ and $\text{SIMULATE}^{\gamma,\delta}$, as well as the proof can be found in Appendix E.

Theorem 4.5. *For any $\delta, B > 0$,* TENSORPLAN *is (δ, B)-sound for discounted MDPs with discount factor $0 \le \gamma < 1$, with misspecification $\eta \le \varepsilon/(24\sqrt{E_d})$ and simulator accuracy $\lambda \le \varepsilon/(12\sqrt{E_d})$, with worst-case per-state query-cost* $\text{poly}\left(\left(dH_{\gamma,\delta}/\delta\right)^A, B\right)$*, when run with input $\delta' = 0.98\delta$ and simulation oracle* $\text{SIMULATE}^{\gamma,\delta}$.

5. Related work

Planning with generative models The local planning problem was introduced by Kearns et al. (2002), who noticed that a planner which is given a simulator and an input state and asked to return a good action can do so with computation/query time independent of the size of the state space. However, this runtime is exponential in H. Munos (2014) gives algorithms that use optimism to improve on this exponential runtime in benign cases. With linear features, a negative result of Du et al. (2019a) (see also Van Roy and Dong (2019); Lattimore et al. (2020)) states that an exponential in $\min\{H, d\}$ runtime remains for any planner with constant suboptimality, even if the feature map nearly realizes the action-value functions of *all* policies but the approximation error is $\varepsilon = \Omega(\sqrt{H/d})$. For target suboptimality $O(\sqrt{d}\varepsilon)$, assuming access to the solution of a feature-map-dependent optimal design problem, Lattimore et al. (2020) gives a planner with polynomial computational (and query) complexity. These results are complemented by the lower bound of Weisz et al. (2020), showing that an exponential lower bound still holds when only $q^\star$ is realizable even if there are no approximation errors. When only the optimal value function is well-represented, Shariff and Szepesvári (2020) give an algorithm for the case where the features are contained in the convex hull of a "core set" of feature vectors. Their planning algorithm, which builds on top of Lakshminarayanan et al. (2017), has computational and query cost that scales polynomially in the size of the core set and the other

relevant quantities. A similar approach appears in Zanette et al. (2019). By contrast, we only provide a bound only on the query complexity of our algorithm, but our query complexity is independent of the size of the core set, whose size, in general, is uncontrolled by the other quantities.

Online learning Any online learning algorithm that controls regret can also be used for local planning by recommending the most frequently used action at the start state. Of the sizable literature on online learning with linear function approximation (Jiang et al., 2017; Du et al., 2019b; Jin et al., 2020; Wang et al., 2019; Yang and Wang, 2019; Ayoub et al., 2020; Modi et al., 2020; Wang et al., 2020b; Zanette et al., 2020), the most relevant are the works of Wen and Roy (2013); Jiang et al. (2017). Both works give algorithms for the online setting with realizable function approximation, and are based on the principle of optimism. The algorithm of Wen and Roy (2013) is restricted to MDPs with deterministic rewards and deterministic transitions, and guarantees that at most d trajectories will be suboptimal. Their proof is based on a similar eluder dimension argument. On the other hand, the algorithm of Jiang et al. (2017) is restricted to the case when a complexity measure called the Bellman rank is low. In fact, our agnostic guarantee (see Definition 2.1) is related to a similar agnostic guarantee of their algorithm (see their Appendix A.2), where optimism at the initial state allows them to compete with the best policy whose state-value function is realizable. Despite the similarities, neither the algorithm nor the analysis applies to our setting.

6. Conclusions and discussion

We presented TENSORPLAN, a provably efficient algorithm for local planning in finite-horizon MDPs which only requires linear realizability of $v^\star$. When the action set is small (i.e. $O(1)$), TENSORPLAN is the first algorithm that enjoys polynomial query complexity without further assumptions. Our results are also complemented by an exponential lower bound for the analogous problem in the infinite-horizon setting and an extension of the positive result to the near-realizable as well as the discounted setting.

In contrast to ADP-type algorithms (Schweitzer and Seidmann, 1985), our algorithm does not use value fitting. In fact, without stronger assumptions such as a core set, ADP algorithms appear to be susceptible to an exponential blow-up of errors (Tsitsiklis and Van Roy, 1996; Dann et al., 2018; Zanette et al., 2019; Wang et al., 2020a; Weisz et al., 2020). For the same reason, our algorithm works with a weaker simulation oracle that provides access only to states that have been encountered previously. Learning via local consistency ("bootstrapping") also allows us to provide a more agnostic guarantee, which automatically matches the best realizable value function. However, our lower bound suggests that such bootstrapping procedures are inefficient for the infinite-horizon setting, at least when the number of actions is larger (scaling with the feature space dimensionality). In offline RL, this issue was recently highlighted for the discounted setting in Zanette (2020).

There are several directions for future work. The first would be to understand the computational efficiency of our algorithm, or to find a computationally efficient alternative. The second would be to understand whether the exponential dependence of the query complexity on the number of actions is strictly necessary. Lastly, it remains to be seen whether polynomial query complexity is possible under $q^\star$ realizability with a small number of actions.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This work was done while the authors were visiting the Simons Institute for the Theory of Computing. PA gratefully acknowledges funding from the Natural Sciences and Engineering Research Council (NSERC). CS gratefully acknowledges funding from the Canada CIFAR AI Chairs Program, Amii and NSERC.

References

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. *arXiv preprint arXiv:1803.00606*, 2018.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2019a.

Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. Provably efficient Q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, pages 8060–8070, 2019b.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49:193–208, 2002.

Chandrashekar Lakshminarayanan, Shalabh Bhatnagar, and Csaba Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic control*, 63(4):1185–1191, 2017.

Tor Lattimore, Csaba Szepesvári, and Gellért Weisz. Learning with good feature representations in bandits and in RL with a generative model. In *ICML*, pages 9464–9472, 2020.

Gen Li, Yuxin Chen, Yuejie Chi, Yuantao Gu, and Yuting Wei. Sample-efficient reinforcement learning is feasible for linearly realizable mdps with limited revisiting. *arXiv preprint arXiv:2105.08024*, 2021.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 2010–2020. PMLR, 2020.

Rémi Munos. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Paul J. Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, September 1985.

Roshan Shariff and Csaba Szepesvári. Efficient planning in large MDPs with weak linear function approximation. *arXiv preprint arXiv:2007.06184*, 2020.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, 1996.

Benjamin Van Roy and Shi Dong. Comments on the Du-Kakade-Wang-Yang lower bounds. *arXiv preprint arXiv:1911.07910*, 2019.

Ruosong Wang, Dean P. Foster, and Sham M. Kakade. What are the statistical limits of offline RL with linear function approximation?, 2020a.

Ruosong Wang, Ruslan Salakhutdinov, and Lin F Yang. Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*, 2020b.

Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Gellert Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions. *arXiv preprint arXiv:2010.01374*, 2020.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*, pages 3021–3029, 2013.

Lin F Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. *arXiv preprint arXiv:1902.04779*, 2019.

Andrea Zanette. Exponential lower bounds for batch reinforcement learning: Batch RL can be exponentially harder than online RL. *arXiv preprint arXiv:2012.08005*, 2020.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Limiting extrapolation in linear approximate value iteration. In *Advances in Neural Information Processing Systems*, pages 5615–5624, 2019.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. *arXiv:2003.00153 [cs]*, Mar 2020. arXiv: 2003.00153.

Appendix A. Discussion of the Interaction Protocol

The astute reader may wonder about whether the above description unduly restricts what state-spaces the planners can deal with: After all, the planner needs to be given "states". This issue can be resolved by introducing state-identifiers: The planner and the simulator communicate by passing each other identifiers of states, rather than states themselves. The simulator then needs to be prepared to translate the identifiers to its internal state representation. This way, the planner can interact uniformly with MDPs of all sorts without any restriction on what their state-spaces are.

The reader may also wonder about whether access to states can be altogether avoided. In a way, as we shall see, our planner does not need the full power of the above interface. In particular, for planning within an episode starting at some initial state s_0, it is sufficient if the oracle *(i)* has an internal state and provides an interface to: *(ii)* reset its internal state to the initial state s_0, *(iii)* forward the internal state to a random next state by feeding it an action; *(iv)* obtain data of the form (f, X) where $X = (R_{a,i} + \Lambda_{sa}, f'_{a,i})_{a\in\mathcal{A}, i\in[n]}$ and *where the value of n is provided as an input*, $f = \varphi_h(s)$ and for $a \in \mathcal{A}, i \in [n]$, $f'_{a,i} = \varphi_{h+1}(S'_{a,i})$ where $(R_{a,i}, S'_{a,i})_i \sim Q(s,a)$ and $|\Lambda_{sa}| \le \lambda$, provided that the oracle's internal state is $S = s$ in stage h.

The reader may also be tempted to think that an even weaker interface that replaces *(iv)* with the ability to receive $f = \varphi_h(s)$ provided that the oracle's internal state is $S = s$ in stage h would be sufficient. Clearly, this is too weak in the sense that even the query complexity of the one-step lookahead calculation on the right-hand side of the Bellman optimality equation (Eq. (1)) would be uncontrolled.

Appendix B. Proof of Theorem 3.1

Proof. Fix the planner p and $d > 0$. For convenience, we will describe the case when the feature-space dimension is $d + 1$. We define a family of featurized MDPs: The MDP mentioned in the theorem statement will be a member of this family. The MDPs in the family are all deterministic and they share the dynamics (and as such both a state space and action space). The state space is a regular d-dimensional grid with 3^d points, say, $\mathcal{S} = \{-1, 0, 1\}^d$. The actions correspond to moving between neighboring states of the grid, or staying in put. Thus, there are at most $2d + 1$ actions: in each state: Incrementing, or decrementing a coordinate, or not changing anything. For convenience, we will use costs in the description of the MDPs. The MDPs in the family differ only in the costs assigned to transitions. In the family there will be 2^d MDPs, each MDP defines the problem of getting to a goal state $s^* \in \{-1, 1\}^d$ by taking the fewest number of actions. This is achieved by setting the cost of each action to $1/d$ except that the cost of action 'stay' provided that this action is taken in a "goal state" in which case the cost is set to zero. We denote the resulting MDP by $M_{s^\star}$. Fix $s^\star$. The optimal value function (negative cost) in $M_{s^\star}$ is $v^\star(s) = -\frac{1}{d}\|s - s^\star\|_1$, which indeed takes values in the $[-2, 0]$ interval. Since for $x \in \{-1, 0, 1\}$, $x^\star \in \{-1, 1\}$, $|x - x^\star| = 1 - xx^\star$, choosing $\varphi(s) = \frac{1}{d}\overline{ds}$ we see that with $\theta^\star = \overline{(-1)\, s^\star}$, $v^\star(s) = \left\langle \varphi(s), \theta^\star \right\rangle$, i.e., the optimal value function is realizable under φ. Note that the MDPs in the family not only share the dynamics, but they also share the feature map.

To find the MDP within this family on which planner p is far from optimal we choose $s^\star$ in an adversarial manner. For this, we define a new MDP, $M_\varnothing$, which shares the dynamics with the previously described MDPs except that here all actions have a cost of $1/d$. This MDP is used for "testing" the planner. For an MDP M either from the above family or $M_\varnothing$, starting with $S_1 = s_0 := (0, \dots, 0)$, let $S_1, Q_1, A_1, S_2, Q_2, A_2, \dots, S_n, Q_n, A_n, \dots$ be the infinite sequence of random elements that describe the data available to the planner while it is used in M. In particular, S_1 is the state passed to the planner in the first call of `GetAction`, $Q_1 \in \cup_{p \ge 0} (\mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S})^p$ collects the queries sent and the responses received, A_1 is the action return by this call of `GetAction`, S_2 is the state the MDP transitions to from state S_1 on the effect of action A_1, etc. Let $S'_1, S'_2, S'_3, \dots$ collect the *distinct* states in this data in the order that they are encountered. In particular, $S'_1 = S_1$ and if $Q_1 = (\tilde{S}_1, \tilde{A}_1, \tilde{R}_1, \tilde{S}'_2, \dots, \tilde{S}_{N_1}, \tilde{A}_{N_1}, \tilde{R}_{N_1}, \tilde{S}'_{N_1})$ with some $N_1 \ge 0$ then (since necessarily $\tilde{S}_1 = S_1$), $S'_2 = \tilde{S}'_2$ unless $\tilde{S}'_2 = S_1$, etc.

Let $S'_{1:m} = (S'_1, \dots, S'_m)$. Slightly abusing notation, we treat this sequence as the set of the elements in it, when convenient. Let $\mathbb{P}_\varnothing$ denote the probability distribution over the above data induced by interconnecting p and $M_\varnothing$ and let $\mathbb{P}_{s^\star}$ be the same when the MDP is $M_{s^\star}$.

An elementary argument shows that $\min_{s \in \{-1,1\}^d} \mathbb{P}_\varnothing(s \in S_{1:m}) \le m/2^d$. Indeed, $2^d \min_{s \in \{-1,1\}^d} \mathbb{P}_\varnothing(s \in S_{1:m}) \le \sum_{s \in \{-1,1\}^d} \mathbb{P}_\varnothing(s \in S_{1:m}) \le \sum_s \sum_{i=1}^m \mathbb{P}_\varnothing(s = S_i) = m$. Let $s^\star$ be a minimizer of the above probability when $m = 2^{d-1}$. Consider the event $\mathcal{E} = \{s^\star \notin S_{1:2^{d-1}}\}$. By the above, $\mathbb{P}_\varnothing(\mathcal{E}) \ge 2^{d-1}/2^d = 0.5$. Then $\mathbb{P}_{s^\star}(\mathcal{E}) = \mathbb{P}_\varnothing(\mathcal{E})$. This follows because for either of these probability measures, $\mathbb{P}(\mathcal{E}) = \sum_{s^\star \notin s_{1:m}} \mathbb{P}(S'_{1:m} = s_{1:m})$ and the probabilities in the sum are identical for $\mathbb{P}_{s^\star}$ and $\mathbb{P}_\varnothing$ since the MDPs only differ in the cost assigned to the 'stay put' action used *at state* $s^\star$ and the sum is restricted for state-sequences that avoid $s^\star$.

For each call of `GetAction`, at most $\beta_{\text{sim}} + 1$ distinct states are encountered. Therefore, `GetAction` is called at least $\lfloor 2^{d-1}/(\beta_{\text{sim}} + 1) \rfloor$ times during the time the first 2^{d-1} distinct states are encountered. On the event $\mathcal{E}$, during this time, $s^\star$ is not encountered and thus the cost of each action executed is $1/d$. Since the optimal cost to get to $s^\star$ from s_0 is one and all costs are nonnegative, on $\mathcal{E}$, the total cost incurred while following the actions taken by the planner is at least $\delta_0 = \lfloor 2^{d-1}/(d(\beta_{\text{sim}} + 1)) \rfloor - 1$ more than the optimal cost. Since $\mathbb{P}_{s^\star}(\mathcal{E}) \ge 0.5$, the suboptimality δ of the planner is at least $0.5\delta_0$. ∎

Appendix C. Proof of Theorem 4.2

To prove that TENSORPLAN (Algorithm 1) is (δ, B)-sound (Definition 2.3) for the H-dimensional planning problem, we fix $\delta > 0$, $B > 0$, $H > 1$, a featurized MDP $(\mathscr{M}, \varphi)$ with 1-bounded feature maps, a suboptimality target $0 < \delta < H$, and a (starting) state state $s_0 \in \mathcal{S}$. We assume that $\delta < H$ as otherwise, for $\delta \ge H$, Eq. (5) trivially holds due to the rewards being bounded in $[0, 1]$ (and therefore the values in $[0, H]$).

The precise values of hyperparameters used in TENSORPLAN will be set to:

$$\zeta = \frac{1}{4H}\delta \tag{10}$$

$$\varepsilon = \left(\frac{\delta}{12H^2}\right)^A \Big/ \left(1 + \frac{1}{2\sqrt{E_d}}\right) \tag{11}$$

$$n_1 = \left\lceil \frac{32(1+2B)^2}{\delta^2} \log \frac{E_d + 1}{\zeta} \right\rceil \tag{12}$$

$$n_2 = \left\lceil \frac{1867H^2(B+1)^2(d+1)}{2\delta^2} \log(4(E_d+1)n_1 HA(d+1)/\zeta) \right\rceil \tag{13}$$

$$n_3 = \left\lceil \max\left\{ n_2, \frac{32(H+1)^2 E_d}{\varepsilon^2} \log((2(E_d+1)n_1 HA))/\zeta \right\} \right\rceil \tag{14}$$

We assume $H > 1$ for simplicity of presentation, as for $H = 1$ the same analysis will apply, replacing H with $H + 1$ in the above display for ε.

Denote by τ^+ the final value of τ at the end of TENSORPLAN.`Init`. For the proof let $\mathbb{P}$ denote the probability distribution induced by the interconnection of TENSORPLAN with the MDP when the initial state of the episode is s_0 and the planner is used for the H steps. In particular, $\mathbb{P}$ is defined over some measurable space $(\Omega, \mathbb{P})$ that carries the random variables S_1, A_1, S_2, A_2, ..., A_H, S_{H+1}, where $S_1 = s_0$, $S_i \sim P_{A_{i-1}}(S_{i-1})$ for $i > 1$, and for $j \in [H]$, A_j is the action returned by `GetAction` when called with S_j and $h = j$. $(\Omega, \mathbb{P})$ also carries the random variables $\hat{\Delta}$, $\overline{\Delta}$, $\tilde{\Delta}$, and $(S_{\tau tj}, A_{\tau tj})_{\tau \le E_d+2, t \in [n_1], j \in [H]}$ of the TENSORPLAN algorithm. For the latter, assume for now that TENSORPLAN.`Init` does not break out from the loop over τ when the test fails, but that it keeps running, so that we can refer to $(S_{\tau tj}, A_{\tau tj})$ even for $\tau > \tau^+$. Note that all other quantities that appear in TENSORPLAN can be written as a function of these. We denote the expectation operator underlying $\mathbb{P}$ by $\mathbb{E}$.

C.1. Concentration bounds

This section establishes concentration bounds on the estimated difference vectors $\hat{\Delta}$ and $\overline{\Delta}$, and then establishes that the true parameter is unlikely to be eliminated from the solution set.

Lemma C.1. *If the simulator's accuracy $\lambda \le \frac{\varepsilon}{4\sqrt{E_d}}$, then with n_2 samples for $\overline{\Delta}$ and n_3 samples for $\hat{\Delta}$, with probability greater than $1 - \zeta$, for all $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 \le B$, for all $\tau \in [E_d + 1]$, $t \in [n_1]$, $j \in [H]$ and action $a \in [A]$, $\overline{\Delta}_{\tau tja}$ and $\hat{\Delta}_{\tau tja}$ satisfy*

$$\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta} \right\rangle\right| \le \delta/(12H) \qquad \text{and} \qquad \left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta} \right\rangle\right| \le \delta/(12H),$$

where $\Delta(S_{\tau tj}, a) = \overline{r_{S_{\tau tj}, a}(P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj}))}$.

Proof. We show this for $\overline{\Delta}_{\tau tja}$, i,e., that the first inequality holds with probability at least $1 - \zeta/2$. As $n_3 \ge n_2$, by a similar argument this statement holds for $\hat{\Delta}_{\tau tja}$ too, and a union bound on the failure probability finishes the proof. Let us refer here to the measurements $\tilde{\Delta}_l$ done by APPROXTD called in Line 10 in Algorithm 3 as $(\tilde{\Delta}_{\tau tjal})_{l \in [n_2]}$. By the bounded rewards (the simulator's rewards are clipped in $[0, 1]$ despite its inaccuracy), triangle inequality, and the assumption that $\forall h \in [H + 1], s \in \mathcal{S}$, $\|\varphi_h(s)\|_2 \le 1$, we have that $\|\tilde{\Delta}_{\tau tjal}\|_\infty \le \|\tilde{\Delta}_{\tau tjal}\|_2 \le 3$. Since $\overline{\Delta}_{\tau tja}$ is the

average of n_2 independent identically distributed bounded samples of the distribution of $\tilde{\Delta}_{\tau tjal}$, which has expectation $\Delta'(S_{\tau tj}, a) = \overline{\left(\left[r_{S_{\tau tj},a} + \Lambda_{S_{\tau tj},a}\right]_0^1\right)\left(P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj})\right)}$, we can apply Hoeffding's inequality for each component $i \in [d+1]$ of the vector:

$$\mathbb{P}\left(\left|\left(\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\right)_i\right| > \delta/\left(\frac{72}{5}H(B+1)\sqrt{d+1}\right)\right) \leq 2\exp\left(-\frac{2n_2\delta^2}{\left(\frac{72}{5}\right)^2 H^2(B+1)^2(d+1)3^2}\right)$$

Setting $n_2 = \left\lceil \frac{1867H^2(B+1)^2(d+1)}{2\delta^2}\log(4(E_d+1)n_1HA(d+1)/\zeta)\right\rceil$ allows this probability to be bounded by $\zeta/(2(E_d+1)n_1HA(d+1))$. A union bound over $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$, $a \in [A]$, and $i \in [d+1]$ achieves the $\zeta/2$ failure probability bound. Under this high-probability event we have that $\|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\|_\infty \leq \delta/\left(\frac{72}{5}H(B+1)\sqrt{d+1}\right)$, so $\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a), \overline{1\theta}\right\rangle\right| \leq \|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\|_\infty \|\overline{1\theta}\|_1 \leq \|\overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a)\|_\infty \|\overline{1\theta}\|_2\sqrt{d+1} \leq \delta/(\frac{72}{5}H)$. By the triangle inequality:

$$\left|\left\langle \overline{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta}\right\rangle\right| \leq \left|\left\langle \overline{\Delta}_{\tau tja} - \Delta'(S_{\tau tj}, a), \overline{1\theta}\right\rangle\right| + \lambda \leq \delta/H\left(\frac{5}{72} + \frac{1}{72}\right) = \delta/(12H),$$

as $\lambda \leq \frac{\varepsilon}{4\sqrt{E_d}} \leq \delta/(12H)/4/(1+\frac{1}{2})$. ∎

Lemma C.2. *If the simulator's accuracy $\lambda \leq \frac{\varepsilon}{4\sqrt{E_d}}$, then with n_3 samples for $\hat{\Delta}$, with probability at least $1-\zeta$, for all $\tau \in [E_d+1]$, $t \in [n_1]$, $j \in [H]$ and action $a \in [A]$,*

$$\left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta^\circ}\right\rangle\right| \leq \frac{\varepsilon}{2\sqrt{E_d}}$$

where $\Delta(S_{\tau tj}, a) = \overline{r_{S_{\tau tj},a}(P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj}))}$.

Proof. Let us refer here to the measurements $\tilde{\Delta}_l$ done by APPROXTD called in Line 12 in Algorithm 3 as $(\tilde{\Delta}_{\tau tjal})_{l\in[n_3]}$. Since $\theta^\circ \in \Theta^\circ$, θ° satisfies Eq. (7) for some policy. Furthermore, due to the bounded rewards, horizon H, and the simulator's clipping of rewards into $[0,1]$ (despite its inaccuracy), and the bounded values (of any state for any policy) in $[0,H]$, we have that $\left\langle \tilde{\Delta}_{\tau tjal}, \overline{1\theta^\circ}\right\rangle \in [-(H+1),(H+1)]$. Since $\hat{\Delta}_{\tau tja}$ is the average of n_3 independent identically distributed bounded samples of the distribution of $\tilde{\Delta}_{\tau tjal}$, which has expectation $\Delta'(S_{\tau tj}, a) = \overline{\left(\left[r_{S_{\tau tj},a} + \Lambda_{S_{\tau tj},a}\right]_0^1\right)\left(P_{S_{\tau tj}a}\varphi_{j+1} - \varphi_j(S_{\tau tj})\right)}$, we can apply Hoeffding's inequality:

$$\mathbb{P}\left(\left|\left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ}\right\rangle - \left\langle \Delta'(S_{\tau tj}, a), \overline{1\theta^\circ}\right\rangle\right| > \frac{\varepsilon}{4\sqrt{E_d}}\right) \leq 2\exp\left(-\frac{n_3\varepsilon^2}{32(H+1)^2E_d}\right).$$

Setting $n_3 = \left\lceil \max\left\{n_2, \frac{32(H+1)^2E_d}{\varepsilon^2}\log((2(E_d+1)n_1HA))/\zeta\right\}\right\rceil$ allows this probability to be bounded by $\zeta/((E_d+1)n_1HA)$. By the triangle inequality, under the high-probability event, the desired bound with Δ instead of Δ' is guaranteed as:

$$\left|\left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ}\right\rangle - \left\langle \Delta(S_{\tau tj}, a), \overline{1\theta^\circ}\right\rangle\right| \leq \left|\left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ}\right\rangle - \left\langle \Delta'(S_{\tau tj}, a), \overline{1\theta^\circ}\right\rangle\right| + |\Lambda_{S_{\tau tj},a}| \leq 2\frac{\varepsilon}{4\sqrt{E_d}}$$

A union bound over $\tau \in [E_d + 1]$, $t \in [n_1]$, $j \in [H]$, and $a \in [A]$ achieves the desired probability bound. ∎

Lemma C.3 ($\theta^\circ \in \mathrm{Sol}(X)$)**.** *For $\tau \in [E_d + 1]$, let $X_{\le\tau}$ denote the first τ elements of X, where X is defined in Line 2 of Algorithm 3. Then, with probability at least $1 - \zeta$ we have that $\forall \tau \in [E_d + 1]$, $\theta^\circ \in Sol(X_{\le\tau})$.*

Proof. As in Lemma C.2, by MDP reward boundedness, $\left|\left\langle \Delta(S_{\tau tj}, a), \overline{1\theta^\circ} \right\rangle\right| \le H$ for any $\Delta(S_{\tau tj}, a)$. Let $A^\circ_{\tau tj}$ be the action satisfying Eq. (8) for θ° in state $S_{\tau tj}$. Then we have that $\left\langle \Delta(S_{\tau tj}, A^\circ_{\tau tj}), \overline{1\theta^\circ} \right\rangle = 0$. Thus, using Lemma C.2, with probability at least $1 - \zeta$, for all $\tau \in [E_d + 1]$, $t \in [n_1]$, $j \in [H]$, $a \in [A]$,

$$\begin{aligned}
\left|\left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ} \right\rangle\right| &= \left|\left\langle \Delta(S_{\tau tj}, a), \overline{1\theta^\circ} \right\rangle + \left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta^\circ} \right\rangle\right| \\
&\le \left|\left\langle \Delta(S_{\tau tj}, a), \overline{1\theta^\circ} \right\rangle\right| + \left|\left\langle \hat{\Delta}_{\tau tja} - \Delta(S_{\tau tj}, a), \overline{1\theta^\circ} \right\rangle\right| \\
&\le \mathbb{I}\{a \ne A^\circ_{\tau tj}\} H + \frac{\varepsilon}{2\sqrt{E_d}},
\end{aligned}$$

where $\mathbb{I}\{S\}$ is the indicator of a set S. We can then bound the product across $a \in [A]$ as

$$\prod_{a \in [A]} \left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ} \right\rangle \le \left(H + \frac{\varepsilon}{2\sqrt{E_d}}\right)^{A-1} \frac{\varepsilon}{2\sqrt{E_d}} = \left(1 + \frac{\varepsilon}{2\sqrt{E_d}H}\right)^{A-1} H^{A-1} \frac{\varepsilon}{2\sqrt{E_d}},$$

and

$$\begin{aligned}
\left(1 + \frac{\varepsilon}{2\sqrt{E_d}H}\right)^{A-1} &\le 1 + (2^{A-1} - 1)\frac{\varepsilon}{2\sqrt{E_d}H} < 1 + 2^A \varepsilon \\
&< 1 + 2^A \frac{\delta^A}{(12H^2)^A} = 1 + \left(\frac{2\delta}{12H^2}\right)^A < 2 \le H,
\end{aligned}$$

so $\prod_{a \in [A]} \left\langle \hat{\Delta}_{\tau tja}, \overline{1\theta^\circ} \right\rangle < H^A \frac{\varepsilon}{2\sqrt{E_d}}$. Let $\tau \in [E_d + 1]$. The τ^{th} element added to X will be $\otimes_{a \in [A]} \hat{\Delta}_{\tau tja}$ computed in Line 12 of Algorithm 3 for some $\tau \in [E_d + 1]$, $t \in [n_1]$, $j \in [H]$, so $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$ according to Eq. (9). ∎

C.2. Eluder dimension

This subsection uses the eluder dimension to bound the maximal number of iterations. For $\Theta \in \mathbb{R}^{(d+1)^A}$ and $x \in \mathbb{R}^{(d+1)^A}$, let

$$f_\Theta(x) = \langle \mathrm{clip}(x), \Theta \rangle,$$

where $\mathrm{clip}(x) = \frac{x}{\|x\|_2} \min\{\|x\|_2, 3^A\}$. Notice the similarity between these functions and the form of the constraints we use in Eq. (9) to define the set of parameter vectors $\mathrm{Sol}(\cdot)$ consistent with our observations. Let

$$\mathcal{F}^+ = \{f_\Theta : \Theta \in \mathbb{R}^{(d+1)^A}, \|\Theta\|_2 \le (B+1)^A\}$$

and

$$E_d = \left\lfloor 3(d+1)^A \frac{e}{e-1} \ln\left\{3 + 3\left(\frac{2(B+1)^A 3^A}{H^A\varepsilon}\right)^2\right\} + 1\right\rfloor = \tilde{O}\left(d^A A\right). \tag{15}$$

By Russo and Van Roy (2014), $\dim_E(\mathcal{F}^+, H^A\varepsilon)$, the **eluder dimension** of $\mathcal{F}^+$ at scale $H^A\varepsilon$ is the length τ of the longest **eluder sequence** $x_1, \ldots, x_\tau$, such that for some $\varepsilon' \geq H^A\varepsilon$, for each $l \in [\tau]$,

$$w_l := \sup\left\{|f_1(x_l) - f_2(x_l)| \;:\; \sqrt{\sum_{i=1}^{l-1}(f_1(x_i) - f_2(x_i))^2} \leq \varepsilon',\; f_1, f_2 \in \mathcal{F}^+\right\} > \varepsilon'.$$

Also by Russo and Van Roy (2014) (Appendix C.2), $\dim_E(\mathcal{F}^+, H^A\varepsilon) \leq E_d$. Now let

$$\mathcal{F} = \{f_\Theta \;:\; \exists\theta \in \mathbb{R}^d.\, \|\theta\|_2 \leq B,\, \Theta = \text{flatten}(\otimes_{a\in[A]}\overline{1\theta})\}. \tag{16}$$

Since $\|\theta\|_2 \leq B$ implies $\|\text{flatten}(\otimes_{a\in[A]}\overline{1\theta})\|_2 \leq (B+1)^A$, $\mathcal{F} \subseteq \mathcal{F}^+$, and so $\dim_E(\mathcal{F}, H^A\varepsilon) \leq \dim_E(\mathcal{F}^+, H^A\varepsilon) \leq E_d$.

Lemma C.4. *With probability at least $1 - 2\zeta$, at any point in the execution of Algorithm 3, the sequence $X_{\leq E_d+1}$ is an eluder sequence for $\mathcal{F}$ at scale $H^A\varepsilon$.*

Proof. Let us assume the event under which $\theta^\circ \in \text{Sol}(X_{\leq\tau})$ for $\tau \in [E_d + 1]$, which has probability at least $1-\zeta$ by Lemma C.3. Let us also assume the high-probability event of Lemma C.1. Let $\varepsilon' = H^A\varepsilon$. The empty sequence is trivially an eluder sequence. By induction, assume for some $\tau \in [E_d + 1]$ that $X_{\leq\tau-1}$ is an eluder sequence. Let $\bar{\theta}^\circ = \text{flatten}(\otimes_{a\in[A]}\overline{1\theta^\circ})$ and let $\bar{\theta}_j = \text{flatten}(\otimes_{a\in[A]}\overline{1\theta_j})$.

$$\begin{aligned}
w_\tau &= \sup\left\{|f_1(X_\tau) - f_2(X_\tau)| \;:\; \sqrt{\sum_{i=1}^{\tau-1}(f_1(X_i) - f_2(X_i))^2} \leq H^A\varepsilon,\; f_1, f_2 \in \mathcal{F}\right\} \\
&\geq \sup\left\{|f_1(X_\tau) - f_2(X_\tau)| \;:\; \forall i \in [\tau-1].\, |(f_1(X_i) - f_2(X_i))| \leq \frac{H^A\varepsilon}{\sqrt{E_d}},\; f_1, f_2 \in \mathcal{F}\right\} \\
&\geq \left|f_{\bar{\theta}_\tau}(X_\tau) - f_{\bar{\theta}^\circ}(X_\tau)\right| > (\delta/(4H))^A - \left|f_{\bar{\theta}^\circ}(X_\tau)\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right) - \frac{H^A\varepsilon}{2\sqrt{E_d}} = H^A\varepsilon,
\end{aligned}$$

where the first line expands the definition of w_τ, the second comes from proving that $\forall i \in [\tau - 1].\, |(f_1(X_i) - f_2(X_i))| \leq \frac{H^A\varepsilon}{\sqrt{E_d}}$ implies $\sqrt{\sum_{i=1}^{\tau-1}(f_1(X_i) - f_2(X_i))^2} \leq H^A\varepsilon$. We show this by assuming the former and letting $v \in \mathbb{R}^{\tau-1}$ be $v_i = f_1(X_i) - f_2(X_i)$, and then $\|v\|_2 \leq \|v\|_\infty\sqrt{\tau-1} \leq H^A\varepsilon$ as $\tau - 1 \leq E_d$ by the induction assumption.

The last line comes from substituting $f_1 = f_{\bar{\theta}_\tau}$ and $f_2 = f_{\bar{\theta}^\circ}$. For this we have to show that $f_{\bar{\theta}_\tau}, f_{\bar{\theta}^\circ} \in \mathcal{F}$, and that $\forall i \in [\tau-1]$, $\left|(f_{\bar{\theta}_\tau}(X_i) - f_{\bar{\theta}^\circ}(X_i))\right| \leq \frac{H^A\varepsilon}{\sqrt{E_d}}$. The former holds by definition (as $\|\theta^\circ\|_2 \leq B$ and $\|\theta_\tau\|_2 \leq B$ as $\theta_\tau \in \text{Sol}(X_{\leq\tau-1})$). For the latter, we use that $\theta^\circ, \theta_\tau \in \text{Sol}(X_{\leq\tau-1})$, so for either $\bar{\theta} \in \{\bar{\theta}^\circ, \bar{\theta}_\tau\}$, $\forall i \in [\tau-1].\, \left|f_{\bar{\theta}}(X_i)\right| \leq \left|\left\langle X_i, \bar{\theta}\right\rangle\right| \leq \frac{H^A\varepsilon}{2\sqrt{E_d}}$, so by the triangle inequality, $\forall i \in [\tau-1].\, \left|f_{\bar{\theta}_\tau}(X_i) - f_{\bar{\theta}^\circ}(X_i)\right| \leq \frac{H^A\varepsilon}{\sqrt{E_d}}$. Finally, it is left to show that $\left|f_{\bar{\theta}_\tau}(X_\tau) - f_{\bar{\theta}^\circ}(X_\tau)\right| > H^A\varepsilon$. For some $t \in [n_1]$, $j \in [H]$, $X_\tau = \otimes_{a\in[A]}\hat{\Delta}_{\tau tja}$. Since $\|\hat{\Delta}_{\tau tja}\|_2 \leq 3$,

$\|X_\tau\|_2 \le 3^A$, so $\forall f_{\bar\theta} \in \mathcal{F}$, $f_{\bar\theta}(X_\tau) = \left\langle \mathrm{clip}(X_\tau), \bar\theta \right\rangle = \left\langle X_\tau, \bar\theta \right\rangle$. Furthermore, because the algorithm added $(X_{\tau a})_a = (\hat\Delta_{\tau tja})_a$ in Line 13, $\min_{a\in[A]} \left|\left\langle \overline{\Delta}_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| > \delta/(4H) = 3H\varepsilon^{1/A}\left(1 + \frac{1}{2\sqrt{E_d}}\right)^{1/A}$. Under the assumed high-probability event of Lemma C.1, for $a \in [A]$, since $\tau \in [E_d + 1]$ and $t \in [n_1]$, by Lemma C.1 and the triangle inequality, $\left|\left\langle \overline{\Delta}_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| - \left|\left\langle \hat\Delta_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| \le 2\delta/(12H)$, so $\min_{a\in[A]} \left|\left\langle \hat\Delta_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| > \delta/(12H) = H\varepsilon^{1/A}\left(1 + \frac{1}{2\sqrt{E_d}}\right)^{1/A}$, therefore $\prod_{a\in[A]} \left|\left\langle \hat\Delta_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right)$. We finish by bounding $\left|f_{\bar\theta^\circ}(X_\tau)\right| \le \frac{H^A\varepsilon}{2\sqrt{E_d}}$ as $\theta^\circ \in \mathrm{Sol}(X_{\le\tau})$ by our high-probability assumption, so by the triangle inequality, and noting that $f_{\bar\theta_\tau}(X_\tau) = \prod_{a\in[A]} \left\langle \hat\Delta_{\tau tja}, \overline{1\theta}_\tau \right\rangle$, we have that $\left|f_{\bar\theta_\tau}(X_\tau) - f_{\bar\theta^\circ}(X_\tau)\right| \ge \prod_{a\in[A]} \left|\left\langle \hat\Delta_{\tau tja}, \overline{1\theta}_\tau \right\rangle\right| - \left|f_{\bar\theta^\circ}(X_\tau)\right| > H^A\varepsilon\left(1 + \frac{1}{2\sqrt{E_d}}\right) - \frac{H^A\varepsilon}{2\sqrt{E_d}}$. ■

By definition of the eluder dimension, we then have:

Corollary C.5. *With probability at least* $1 - 2\zeta$, $\tau^+ \le \dim_E(\mathcal{F}, H^A\varepsilon) + 1 \le E_d + 1$.

Proof. Assume the high-probability statements of Lemma C.4 hold and that $\tau^+ > \dim_E(\mathcal{F}, H^A\varepsilon) + 1$. Take $X_{\le \dim_E(\mathcal{F}, H^A\varepsilon)+1}$ which is of length $\dim_E(\mathcal{F}, H^A\varepsilon) + 1$. Also, $\dim_E(\mathcal{F}, H^A\varepsilon) + 1 \le E_d + 1$. Therefore, by Lemma C.4, $X_{\le \dim_E(\mathcal{F}, H^A\varepsilon)+1}$ is an eluder sequence for $\mathcal{F}$ at scale $H^A\varepsilon$ of length $> \dim_E(\mathcal{F}, H^A\varepsilon)$, which is a contradiction. ■

C.3. Value bound

Denote by π_{TP} the policy induced by TENSORPLAN.

Lemma C.6. *With probability* $1 - 2\zeta$, *if* $\tau^+ \in [E_d + 1]$, $v_1^{\pi_{TP}}(s_0) \ge \langle \varphi_1(s_0), \theta^+ \rangle - \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]} \left\langle \overline{\Delta}_{\tau^+ tjA_{\tau^+ tj}}, \overline{1\theta^+} \right\rangle - \frac{1}{2}\delta$.

Proof. Let us denote the state we reach after H steps (once the episode is over) by S_{H+1} in the following. For $\psi : \mathcal{S} \to \mathbb{R}^d$, we let $P_{sa}\psi = \int \psi(s')P_{sa}(ds')$. Recall that under $\mathbb{P}$, the random variables $S_1 = s_0, A_1, S_2, A_2, \ldots, A_H, S_{H+1}$ have the distribution of an episode in the MDP that

starts from s_0 and follows the policy π_{TP} induced by TENSORPLAN.

$$
\begin{aligned}
v_1^{\pi_{\mathrm{TP}}}(s_0) &= \mathbb{E}\sum_{j\in[H]} r_{S_j,A_j} = \mathbb{E}\left\langle \overline{\left(\sum_{j\in[H]} r_{S_j,A_j}\right)\varphi_{H+1}(S_{H+1})}, \overline{1\theta^+}\right\rangle \\
&= \mathbb{E}\left[\left\langle \varphi_1(s_0), \theta^+\right\rangle + \sum_{j\in[H]} \left\langle \overline{r_{S_j,A_j}(\varphi_{j+1}(S_{j+1}) - \varphi_j(S_j))}, \overline{1\theta^+}\right\rangle\right] \\
&= \left\langle \varphi_1(s_0), \theta^+\right\rangle + \sum_{j\in[H]} \mathbb{E}\left\langle \overline{r_{S_j,A_j}(\varphi_{j+1}(S_{j+1}) - \varphi_j(S_j))}, \overline{1\theta^+}\right\rangle \\
&= \left\langle \varphi_1(s_0), \theta^+\right\rangle + \sum_{j\in[H]} \mathbb{E}\left[\left\langle \overline{r_{S_j,A_j}(P_{S_jA_j}\varphi_{j+1} - \varphi_j(S_j))}, \overline{1\theta^+}\right\rangle\right] \\
&\geq \left\langle \varphi_1(s_0), \theta^+\right\rangle + \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]}\left[\left\langle \overline{r_{S_{\tau^+tj},A_{\tau^+tj}}(P_{S_{\tau^+tj}A_{\tau^+tj}}\varphi_{j+1} - \varphi_j(S_{\tau^+tj}))}, \overline{1\theta^+}\right\rangle\right] - \frac{1}{4}\delta \\
&\geq \left\langle \varphi_1(s_0), \theta^+\right\rangle + \frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]}\left\langle \overline{\Delta}_{\tau^+tjA_{\tau^+tj}}, \overline{1\theta^+}\right\rangle - \frac{1}{2}\delta,
\end{aligned}
$$

where in the first line we used that $\varphi_{H+1}(S_{H+1}) = \mathbf{0}$, in the second that $s_0 = S_1$, in the third that s_0 is fixed so can be moved out of the expectation, and in the fourth we used the tower rule for expectations. In the fifth line we replace the outer expectation with an average of rollouts by the algorithm that is close to the expectation with high probability, while we also switched to the variable notation used in Algorithm 3. More specifically, we use the fact that for all $h \in [H+1]$, $s \in \mathcal{S}$, and $\tau \in [E_d + 1]$, we have that $\|\varphi_h(s)\|_2 \leq 1$ and $\|\theta^+\|_2 \leq B$, $\left|\left\langle \overline{r_{S_{\tau tj},A_{\tau tj}}(P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj}))}, \overline{1\theta^+}\right\rangle\right| \leq 1 + 2B$ (as rewards are bounded in $[0,1]$). We can therefore apply Hoeffding's inequality:

$$
\begin{aligned}
&\mathbb{P}\Bigg(\frac{1}{n_1}\sum_{t\in[n_1]}\sum_{j\in[H]}\Big[\left\langle \overline{r_{S_{\tau tj},A_{\tau tj}}(P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj}))}, \overline{1\theta^+}\right\rangle \\
&\qquad - \mathbb{E}\left\langle \overline{r_{S_{\tau tj},A_{\tau tj}}(P_{S_{\tau tj}A_{\tau tj}}\varphi_{j+1} - \varphi_j(S_{\tau tj}))}, \overline{1\theta^+}\right\rangle\Big] > \delta/4\Bigg) \\
&\leq \exp\left(-\frac{n_1\delta^2}{32(1+2B)^2}\right) \leq \frac{\zeta}{E_d+1},
\end{aligned}
$$

if $n_1 = \left\lceil 32(1+2B)^2/\delta^2 \log\frac{E_d+1}{\zeta}\right\rceil$. With an union bound, the probability that any of these bounds fail for any $\tau \in [E_d+1]$ is upper bounded by ζ. We can therefore apply this bound for $\tau = \tau^+$, noting that

$$
\mathbb{E}\left\langle \overline{r_{S_{\tau^+tj},A_{\tau^+tj}}(P_{S_{\tau^+tj}A_{\tau^+tj}}\varphi_{j+1} - \varphi_j(S_{\tau^+tj}))}, \overline{1\theta^+}\right\rangle = \mathbb{E}\left\langle \overline{r_{S_j,A_j}(P_{S_jA_j}\varphi_{j+1} - \varphi_j(S_j))}, \overline{1\theta^+}\right\rangle .
$$

This is because $\theta^+ = \theta_{\tau^+}$, so for all $t \in [n_1]$, the episode $\left(S_{\tau^+t1}, A_{\tau^+t1}, \ldots, A_{\tau^+tH}, S_{\tau^+t,H+1}\right)$ is distributed identically to the episode $(S_1, A_1, S_2, A_2, \ldots, A_H, S_{H+1})$. Finally, in the sixth line we replace the remaining expectation with the average measured by the algorithm, which is close to the expectation with high probability (Lemma C.1) for $\tau \in [E_d+1], t \in [n_1], j \in [H], a \in [A]$. By a union bound, this adds another ζ to the probability that our bound does not hold. ■

Corollary C.7. *With probability at least* $1-3\zeta$, $v_1^{\pi_{TP}}(s_0) \geq \langle \varphi_1(s_0), \theta^+ \rangle - \frac{3}{4}\delta$.

Proof. Under the high-probability event of Corollary C.5, $\tau^+ \leq E_d+1$. From the proof of Lemma C.6:

$$v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \left\langle \varphi_1(s_0), \theta^+ \right\rangle + \frac{1}{n_1} \sum_{t\in[n_1]} \sum_{j\in[H]} \left\langle \overline{\Delta}_{\tau t j A_{\tau t j}}, \overline{1\theta^+} \right\rangle - \frac{1}{2}\delta \geq \left\langle \varphi_1(s_0), \theta^+ \right\rangle - \frac{3}{4}\delta$$

where we use the fact that, since $\tau^+ \leq E_d + 1$, we exited the τ loop as `CleanTest` was true in Line 20, so for τ^+, all $t \in [n_1]$ the path in Line 16 was chosen (otherwise we would have finished with a larger τ^+). This directly bounds the inner product of interest. Taking a union bound over the underlying high-probability events, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \langle \varphi_1(s_0), \theta^+ \rangle - \frac{3}{4}\delta$ holds with probability at least $1-3\zeta$. ∎

Corollary C.8. $v_1^{\pi_{TP}}(s_0) \geq v_1(s_0;\theta^\circ) - \delta$.

Proof. Assume all high-probability events introduced so far, which hold with probability at least $1-3\zeta$. By Corollary C.5, $\tau^+ \leq E_d + 1$. By Lemma C.3, $\theta^\circ \in \mathrm{Sol}(X_{\leq \tau^+})$. Since θ^+ was chosen optimistically in Line 5, $\langle \varphi_1(s_0), \theta^+ \rangle \geq \langle \varphi_1(s_0), \theta^\circ \rangle = v_1(s_0;\theta^\circ)$. By Corollary C.7, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \langle \varphi_1(s_0), \theta^+ \rangle - \frac{3}{4}\delta \geq v_1(s_0;\theta^\circ) - \frac{3}{4}\delta$. Therefore, with probability at least $1-3\zeta = 1-\frac{1}{4H}\delta$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq v_1(s_0;\theta^\circ) - \frac{3}{4}\delta$, so $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \left(1-\frac{1}{4H}\delta\right)\left(v_1(s_0;\theta^\circ) - \frac{3}{4}\delta\right) \geq v_1(s_0;\theta^\circ) - \delta$ (using that due to bounded rewards, $v_1^{\pi_{\mathrm{TP}}}(s_0) \leq H$).

∎

C.4. Final bound

We can now combine all the ingredients together to get the final result.

Theorem 4.2. *For any* $\delta > 0$ *and* $B > 0$, *there exists values of* $\zeta, \varepsilon, n_1, n_2$, *and* n_3 *such that the* TENSORPLAN *algorithm (Algorithm 1) is* (δ, B)*-sound (Definition 2.3) with misspecification* $\eta = 0$ *and simulator accuracy* $\lambda \leq \varepsilon/(4\sqrt{E_d}) = \tilde{O}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A/\sqrt{A}\right)$ *for the* H*-horizon planning problem with worst-case per-episode query-cost*

$$\tilde{O}\left(d^A A^3 B^2/\delta^2 \left(H^3 B^2 dA/\delta^2 + d^A A^2 H^{4A+2} 12^{2A}/\delta^{2A}\right)\right) = \mathrm{poly}\left((dH/\delta)^A, B\right).$$

Proof. Fix $\delta > 0$ and $B > 0$. By Corollary C.8, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq v_1(s_0;\theta^\circ) - \delta$ for any $\theta^\circ \in \Theta^\circ$. Denoting by $v^\circ = v_B^\circ$ the H-horizon φ-compatible optimal value function of $\mathscr{M}$, $v_1^{\pi_{\mathrm{TP}}}(s_0) \geq \sup_{\theta^\circ\in\Theta^\circ} v_1(s_0;\theta^\circ) - \delta = v_1^\circ(s_0) - \delta$ by definition, proving soundness. In each episode, *Line 5* in TENSORPLAN.`GetAction` is called H times, and TENSORPLAN.`Init` is called once. The former results in Hn_2A calls to the simulator. We turn our attention to the query complexity of TENSORPLAN.`Init`. The loop variable τ of `Init` goes up to $E_d + 2$ so $\tau^+ \leq E_d + 2$. Line 10 can therefore be called at most $(E_d+2)n_1HA$ times, each performing n_2 interactions with the simulator. Line 17 can be called at most $(E_d+2)n_1H$ times, each performing 1 interaction with the simulator. Line 12 can be called at most $(E_d+2)n_1A$ times, each performing n_3 interactions with the simulator. Using that $E_d = \tilde{O}\left(d^A A\right)$, $\lambda = \tilde{O}\left(\left(\frac{\delta}{12\sqrt{d}H^2}\right)^A/\sqrt{A}\right)$. Furthermore, using that $n_1 = \tilde{O}\left(B^2A/\delta^2\right)$, $n_2 = \tilde{O}\left(H^2B^2dA/\delta^2\right)$, $n_3 = \tilde{O}\left(d^A A^2 H^2/\varepsilon^2 + H^2B^2dA/\delta^2\right) =$

$\tilde{O}\left(d^A A^2 H^{4A+2} 12^{2A}/\delta^{2A} + H^2 B^2 dA/\delta^2\right)$, the (worst-case per-episode) query-cost of TENSORPLAN (along any episode) is

$$\tilde{O}\left(Hn_2 A + E_d n_1 A\left(Hn_2 + n_3\right)\right) = \tilde{O}\left(E_d n_1 A\left(Hn_2 + n_3\right)\right) = \tilde{O}\left(d^A A^3 B^2/\delta^2\left(Hn_2 + n_3\right)\right)$$
$$= \tilde{O}\left(d^A A^3 B^2/\delta^2\left(H^3 B^2 dA/\delta^2 + d^A A^2 H^{4A+2} 12^{2A}/\delta^{2A}\right)\right). \qquad \blacksquare$$

Appendix D. Proof of Theorem 4.4

Theorem 4.4. *For any $\delta, B > 0$,* TENSORPLAN *is (δ, B)-sound with misspecification $\eta \le \varepsilon/(12\sqrt{E_d})$ and simulator accuracy $\lambda \le \varepsilon/(12\sqrt{E_d})$ with worst-case per-episode query-cost* poly $\left((dH/\delta)^A, B\right)$, *when run with input $\delta' = 0.98\delta$ and simulation oracle* SIMULATE$'$.

Proof. Fix $\delta > 0$, $H \ge 1$, $\eta = \varepsilon/(12\sqrt{E_d})$ and $\lambda = \varepsilon/(12\sqrt{E_d})$. We assume that $\delta < H$ as soundness trivially holds otherwise. Let $(\mathscr{M}, \varphi)$ be any featurized MDP with 1-bounded feature maps and rewards bounded in $[0,1]$. Let SIMULATE be the λ-accurate simulation oracle for $(\mathscr{M}, \varphi)$. We will shortly define a slightly modified simulation oracle SIMULATE$'$ corresponding to a featurized MDP $(\mathscr{M}', \varphi')$ derived from $(\mathscr{M}, \varphi)$. This oracle will simply use the data returned from calls to SIMULATE while we will claim that it is a simulator for $(\mathscr{M}', \varphi')$ with inaccuracy not more than $\varepsilon/(4\sqrt{E_d})$.

Denote by π_{TP} the policy while TENSORPLAN interacts with the simulator SIMULATE$'$. By the correspondence between the two MDPs, π_{TP} can be interpreted as a policy of $\mathscr{M}$. We will then prove that for all states $s \in \mathcal{S}$ of $\mathscr{M}$,

$$v_1^{\pi_{\mathrm{TP}}}(s) \ge v_1^\circ(s) - \delta\,,$$

where $v^{\pi_{\mathrm{TP}}}$ is the H-horizon value function of TENSORPLAN's policy π_{TP} in $\mathscr{M}$ and $v^\circ = v^\circ_{B,\eta}$ is the H-horizon φ-compatible optimal value function of $\mathscr{M}$ (cf. Equation (4)).

Let $\Pi^\circ_{B,\eta}$ be the set of memoryless, deterministic (MLD) policies that are B-boundedly v-linearly realizable with misspecification η and features φ. Then, by definition, $v^\circ_{B,\eta}(s) = \sup_{\pi \in \Pi^\circ_{B,\eta}} v_1^\pi(s)$. It is enough to prove that for any $\pi \in \Pi^\circ_{B,\eta}$,

$$v_1^{\pi_{\mathrm{TP}}}(s) \ge v_1^\pi(s) - \delta\,.$$

Fix a $\theta \in \mathbb{R}^d$ such that $\left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \le \eta$ for all $h \in [H]$ and $s \in \mathcal{S}$. Such a θ exists by definition. We now construct an alternative featurized MDP $(\mathscr{M}', \varphi')$ that will mimic $\mathscr{M}$, but with slightly different rewards and an expanded state-space. The main point of introducing this MDP is that the value function of π (when "used" in $\mathscr{M}'$) will be realizable with $\eta = 0$. The function SIMULATE$'$ will be defined to act as a simulator for $(\mathscr{M}', \varphi')$. Then we will use an extension Theorem 4.2 to argue that TENSORPLAN induces a policy that can compete with π in $\mathscr{M}'$ and hence, by the correspondence between the two MDPs, it also competes with π in $\mathscr{M}$. The required extension of Theorem 4.2 is as follows:

Claim D.1. *The conclusions of Theorem 4.2 remain valid with the following two changes:*

(i) The rewards in the MDP are allowed to belong to $[-2, 2]$*;*

(ii) A set $\mathcal{S}_1 \subset \mathcal{S}$ is fixed and the requirement of soundness is redefined so that the initial state chosen at the beginning of an episode must belong to $\mathcal{S}_1$ while v-realizability (cf. Definition 2.1) of a policy π is redefined so that instead of $\max_{h\in[H]} \sup_{s\in\mathcal{S}} \left|v_h^\pi(s) - \langle\varphi_h(s),\theta\rangle\right| \le \eta$ we require $\max_{h\in[H]} \sup_{s\in\mathcal{S}_h} \left|v_h^\pi(s) - \langle\varphi_h(s),\theta\rangle\right| \le \eta$ where $\mathcal{S}_h \subset \mathcal{S}$ is defined as the set of states that can be reached with positive probability in $\mathscr{M}$ from some state in $\mathcal{S}_1$ and action sequence of length $h-1$.

Proof. For *(i)* note that shifting the rewards does not impact the proof, while the range of rewards scales the query cost quadratically (this comes from the use of Hoeffding's inequality, where ranges of temporal difference errors appear, which scale linearly with the range of rewards). For *(ii)* we only need to check that if θ° is a parameter vector of a policy with the modified definition, this parameter vector will not be eliminated by TENSORPLAN. A quick look at the proof of Lemma C.3 confirms that this is the case. Indeed, TENSORPLAN constructs data for checking consistency at stage h only with states that it reaches through $h-1$, or h actions from the initial state it is given. Therefore the states that appear with φ_h always belong to $\mathcal{S}_h$. As such, Lemma C.3 continues to hold true, and the result follows. ∎

Let us now return to the definition of $\mathscr{M}' = (\mathcal{S}', \Sigma', [A], Q')$ and φ'. We let the states of $\mathscr{M}'$ be $\mathcal{S}' = \mathcal{S}\times[H] \cup \{\bot\}$, that is, the state space of $\mathscr{M}'$ contains H copies of each state, and a final absorbing state $\bot$. The intention is that only states of the form $(s, h+1)$ are accessible from states of the form (s,h). We let Σ' to be the smallest σ algebra under which $\{\bot\}$ and all the sets of the form $S\times\{h\}$ are measurable where $S\in\Sigma$ and $h\in[H]$. We let $\varphi'_h((s,\cdot)) = \varphi_h(s)$ and $\varphi'_h(\bot) = \mathbf{0}$, a d-dimensional vector of all zeros.

The transition kernel Q' in $\mathscr{M}'$ will follow that in $\mathscr{M}$, with the appropriate modification to create the promised "levelled" structure, while the rewards are modified to "cancel out the misspecification" of policy π. That is, for $h < H$, from state $(s,h)\in\mathcal{S}'$ taking action $a\in\mathcal{A}$, kernel Q' gives $(R + z(s,h), (S', h+1))$ where $(R,S')\sim Q_{sa}$ and

$$z(s,h) = \mathbb{E}_{a'\sim\pi^{(h)}(s)}\left[\langle\varphi_h(s) - P_{sa'}\varphi_{h+1}, \theta\rangle - r_{sa'}\right] .$$

From state $(s,H)\in\mathcal{S}'$ or $\bot$, any action leads deterministically to $\bot$ while incurring zero reward.

Notice that any $(s',h)\in\mathcal{S}'$ can only be reached after exactly h steps when starting from some other state $(s,1)$, $s\in\mathcal{S}$. Furthermore, denoting by r' the immediate rewards in $\mathscr{M}'$, we have $r'_{(s,h),a} = r_{sa} + z(s,h)$. Note that $|z(s,h)|\le 2\eta$, since $\left|v_h^\pi(s) - \langle\varphi_h(s),\theta\rangle\right| \le \eta$ for all $h\in[H]$ and $s\in\mathcal{S}$, and $\mathbb{E}_{a'\sim\pi_h(s)}\left[v_h^\pi(s) - P_{sa'}v_{h+1}^\pi - r_{sa'}\right] = 0$ by the Bellman equation. Hence, the rewards in $\mathscr{M}'$ are supported on $[-2\eta, 1+2\eta]\subset[-2,2]$ (as $\eta < 1/2$).

For any $(s,h)\in\mathcal{S}'$, let $\bar{v}'_h((s,h)) = \left\langle\varphi'_h((s,h)),\theta\right\rangle = \langle\varphi_h(s),\theta\rangle$. We claim that $\bar{v}'_h$ satisfies the Bellman equation of π when policy π in $\mathscr{M}'$ is taken as a policy of $\mathscr{M}'$ with the understanding that in state (s,h) and stage h, following π means using $\pi_h(s)$, while in stage $h'\ne h$, an arbitrary action

can be taken. Indeed, for any $(s,h) \in \mathcal{S}'$ we have

$$\begin{aligned}
\bar{v}'_h((s,h)) = \langle \varphi_h(s), \theta \rangle &= E_{a\sim\pi^{(h)}(s)}\left[r_{sa} + \langle \varphi_h(s) - P_{sa}\varphi_{h+1}, \theta\rangle - r_{sa} + \langle P_{sa}\varphi_{h+1}, \theta\rangle\right] \\
&= E_{a\sim\pi^{(h)}(s)}\left[r_{sa} + E_{a'\sim\pi^{(h)}(s)}\left[\langle \varphi_h(s) - P_{sa'}\varphi_{h+1}, \theta\rangle - r_{sa'}\right] + \langle P_{sa}\varphi_{h+1}, \theta\rangle\right] \\
&= E_{a\sim\pi^{(h)}(s)}\left[r'_{(s,h),a} + \langle P_{sa}\varphi_{h+1}, \theta\rangle\right] \\
&= E_{a\sim\pi^{(h)}(s)}\left[r'_{(s,h),a} + P'_{(s,h),a}\bar{v}'_{h+1}\right] \\
&= r'_\pi((s,h)) + P'_\pi((s,h))\bar{v}'_{h+1}\,,
\end{aligned}$$

where P' is the transition kernel in $\mathscr{M}'$ and P'_π (r'_π) is the corresponding kernel (respectively, reward function) induced by π. Since v'^π also satisfies this equation and $\bar{v}'_{H+1} = v'^\pi_{H+1} = \mathbf{0}$, it follows that for any $(s,h) \in \mathcal{S}'$, $v'^\pi_h((s,h)) = \bar{v}'_h((s,h)) = \left\langle \varphi'_h((s,h)), \theta \right\rangle$. Now, define $\mathcal{S}'_1 = \mathcal{S} \times \{1\}$. Then, $\mathcal{S}'_h$, the set of states reachable in $\mathscr{M}'$ with positive probability from $\mathcal{S}'_1$ with an action sequence of length $h-1$, is easily seen to be a subset of $\mathcal{S} \times \{h\}$. Therefore, policy π is v-realizable with $\eta' = 0$ in the sense of the definition of v-realizability given in Part *(ii)* of Claim D.1.

For state and action $s \in \mathcal{S}, a \in [A]$, recall that SIMULATE$(s,h,a)$ is implemented by a λ-accurate simulator for $(\mathscr{M}, \varphi)$, and that the state transitions of $\mathscr{M}$ and $\mathscr{M}'$ are the same apart from that in the latter the stage counter is incremented in each transition. Hence, we define SIMULATE$'$ as follows: SIMULATE$'((s,h),h',a)$ for $(s,h) \in \mathcal{S}'$ calls $(R, S', \varphi_{h'+1}(S')) \leftarrow$ SIMULATE(s,h',a) and returns $(R, (S', h+1), \varphi_{h'+1}(S'))$ for $h < H$ and $(R, \perp, \mathbf{0})$ otherwise. We also let SIMULATE$'(\perp,\cdot,\cdot)$ deterministically returns $(0, \perp, \mathbf{0})$.

Let π' be a policy of $\mathscr{M}'$ that is induced by a planner interacting with $\mathscr{M}'$ using SIMULATE$'$ where the episode starts in $\mathscr{M}'$ are restricted to $\mathcal{S}'_1$. Then, on the one hand, π' can be seen as a policy in $\mathscr{M}$: For a history in $\mathscr{M}$, one just needs to add the respective stage counters to the states in the history and then use π' to return an action.

Now note that the reward distribution of $\mathscr{M}'$ is shifted by up to 2η compared to the reward distribution of $\mathscr{M}$. The distribution of the simulator's rewards $[R_{sa} + \Lambda_{sa}]_0^1$ are shifted by up to $\Lambda_{sa} \le \lambda$ compared to the reward distribution of $\mathscr{M}$, so by the triangle inequality it is shifted by up to $2\eta + \lambda$ compared to the reward distribution of $\mathscr{M}'$. Since $2\eta + \lambda = \varepsilon/(4\sqrt{E_d})$, using the reward of the simulator call SIMULATE(s,h',a) as the output of SIMULATE$'((s,h),h',a)$ ensures SIMULATE$'$ is a simulator for $(\mathscr{M}', \varphi')$ with inaccuracy $\varepsilon/(4\sqrt{E_d})$.

Therefore, applying Claim D.1 with $\eta' = 0$, $\lambda' = \varepsilon/(4\sqrt{E_d})$, and $\delta' = 0.98\delta$, TENSORPLAN is (δ', B)-sound for $\mathscr{M}'$ and initial states from $\mathcal{S}'$ when run with the simulator SIMULATE$'$, with worst-case per-episode query-cost poly $\left((dH/\delta)^A, B\right)$. Thus, for all $(s,1) \in \mathcal{S}'$ (ie. all $s \in \mathcal{S}$), $v'^{\pi_{\mathrm{TP}}}_1((s,1)) \ge v'^{\circ}_1((s,1)) - 0.98\delta$, where $v'^\circ = v'^\circ_{B,0}$ is the H-horizon φ-compatible optimal value function of $\mathscr{M}'$. As π is v-linearly realizable in MDP $\mathscr{M}'$ with no misspecification, $v'^\circ_1((s,1)) \ge v'^\pi_1((s,1))$, so $v'^{\pi_{\mathrm{TP}}}_1((s,1)) \ge v'^\pi_1((s,1)) - 0.98\delta$. As the state transition distributions of $\mathscr{M}$ and $\mathscr{M}'$ are the same except for the stage counter incrementation in $\mathscr{M}'$, the distribution of any policy π in $\mathscr{M}$ producing an episode $(S_1, A_1, S_2, A_2 \ldots, S_H, A_H)$ is the same as the distribution of π in $\mathscr{M}'$ producing the episode $((S_1,1), A_1, (S_2,2), A_2, \ldots, (S_H,H), A_H)$. Furthermore, the rewards of $\mathscr{M}'$ are shifted by up to 2η. Therefore, the H-horizon value functions $v'^{\pi'}_1(s)$ and $v'^{\pi'}_1((s,1))$ for any π' differ by at most $2H\eta$, and thus by treating π_{TP} as a policy of both $\mathscr{M}$ and $\mathscr{M}'$, we have

$$v^{\pi_{\mathrm{TP}}}_1(s) \ge v'^\pi_1((s,1)) - 0.98\delta - 2H\eta \ge v^\pi_1(s) - 0.98\delta - 4H\eta \ge v^\pi_1(s) - \delta\,,$$

as $4H\eta = \frac{H\varepsilon}{3\sqrt{E_d}} \le \frac{H\frac{\delta}{12H^2}/(1+0.5)}{3\sqrt{E_d}} \le \frac{\delta}{18H}/3 \le 0.02\delta$. ■

We note in passing that the result as stated could be (slightly) strengthened and simplified: Since SIMULATE′ generates the same data (with some redundancy) as SIMULATE, using TENSORPLAN on $(\mathscr{M}', \varphi')$ via SIMULATE′ produces the same policy in $\mathscr{M}$ as using it directly on $(\mathscr{M}, \varphi)$ via SIMULATE. Thus, SIMULATE′ is only needed for the proof; the conclusion of the result applies when TENSORPLAN directly uses SIMULATE with a near-realizable featurized MDP.

By reiterating the arguments of Claim D.1 in the context of Theorem 4.4, we get the following claim, which will be needed in the next section:

Claim D.2. *The conclusions of Theorem 4.4 remain valid with the following two changes:*

(i) The rewards in the MDP are allowed to belong to $[-2, 2]$*;*

(ii) A set $\mathcal{S}_1 \subset \mathcal{S}$ *is fixed and the requirement of soundness is redefined so that the initial state chosen at the beginning of an episode must belong to* $\mathcal{S}_1$ *while v-realizability (cf. Definition 2.1) of a policy* π *is redefined so that instead of* $\max_{h\in[H]} \sup_{s\in\mathcal{S}} \left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \le \eta$ *we require* $\max_{h\in[H]} \sup_{s\in\mathcal{S}_h} \left|v_h^\pi(s) - \langle \varphi_h(s), \theta\rangle\right| \le \eta$ *where* $\mathcal{S}_h \subset \mathcal{S}$ *is defined as the set of states that can be reached with positive probability in* $\mathscr{M}$ *from some state in* $\mathcal{S}_1$ *and action sequence of length* $h-1$.

Appendix E. Proof of Theorem 4.5

Theorem 4.5. *For any* $\delta, B > 0$, TENSORPLAN *is* (δ, B)*-sound for discounted MDPs with discount factor* $0 \le \gamma < 1$*, with misspecification* $\eta \le \varepsilon/(24\sqrt{E_d})$ *and simulator accuracy* $\lambda \le \varepsilon/(12\sqrt{E_d})$*, with worst-case per-state query-cost* $\mathrm{poly}\left(\left(dH_{\gamma,\delta}/\delta\right)^A, B\right)$*, when run with input* $\delta' = 0.98\delta$ *and simulation oracle* $\textsc{Simulate}^{\gamma,\delta}$.

Proof. Fix a suboptimality target $\delta > 0$. We assume that $\delta < H$ as soundness trivially holds otherwise. Fix $\eta = \varepsilon/(24\sqrt{E_d})$ and $\lambda = \varepsilon/(12\sqrt{E_d})$; proving soundness and the query-cost bound for these values implies the same results for smaller η or λ. Let $(\mathscr{M}, \varphi)$ be a featurized MDP in the discounted setting with 1-bounded feature maps and rewards bounded in $[0, 1]$. Take a λ-accurate simulation oracle SIMULATE for $(\mathscr{M}, \varphi)$. Let

$$H_{\gamma,\delta} = \left\lceil \frac{\log\left((1-\gamma)\eta\right)/\log\gamma}{1-\gamma} \right\rceil .$$

In the remainder of the proof we shorten $H_{\gamma,\delta}$ and will just use H (i.e., in what follows $H = H_{\gamma,\delta}$). We now construct a featurized, fixed-horizon MDP $(\mathscr{M}', \varphi^{\gamma,\delta})$ with horizon H. Let the states of $\mathscr{M}'$ be $\mathcal{S}' = \mathcal{S} \times [H] \cup \{\perp\}$, that is, the state space contains H copies of each state, and an additional state $\perp$, which will play the role of a final, absorbing state. The σ algebra for $\mathcal{S}'$ is constructed as in the proof of Theorem 4.4 (we omit the definition). The action set of $\mathscr{M}'$ remains $[A]$. The kernel Q' is inherited from $\mathscr{M}$, again, with the appropriate modification to create the promised "levelled" structure, while the rewards are modified to accommodate discounting: That is, for $h < H$, from state $(s, h) \in \mathcal{S}'$ taking action $a \in \mathcal{A}$, kernel Q' gives $(\gamma^{h-1}R, (S', h+1))$ where $(R, S') \sim Q_{sa}$. From state $(s, H) \in \mathcal{S}'$ or $\perp$, any action leads deterministically to $\perp$ while incurring zero reward. In words, states with associated stage $h < H$ lead to respective states with associated stage $h+1$, and the episode is

terminated after H steps by transitioning to the absorbing state $\perp$. By letting r' denote the immediate expected rewards in $\mathscr{M}'$, for state $(s,h) \in \mathcal{S}'$ and action a we have $r'_{(s,h),a} = \gamma^{h-1} r_{sa}$.

Let $\varphi_h^{\gamma,\delta}((s,\cdot)) = \gamma^{h-1}\varphi(s)$ and $\varphi_h^{\gamma,\delta}(\perp) = \mathbf{0}$, a d-dimensional vector of all zeros. We define $\textsc{Simulate}^{\gamma,\delta}$ as follows: $\textsc{Simulate}^{\gamma,\delta}$ is a simulation oracle for $(\mathscr{M}', \varphi^{\gamma,\delta})$ so that for $(s,h) \in \mathcal{S}'$ with $h < H$, $h' \in [H]$ and $a \in [A]$, $\textsc{Simulate}^{\gamma,\delta}((s,h),h',a)$ first gets $(R,S,\varphi(S)) \leftarrow \textsc{Simulate}(s,h',a)$ to return $\left(\gamma^{h-1}R, (S,h+1), \varphi^{\gamma,\delta}_{h'+1}((S,h+1))\right)$, while it returns $\left(\gamma^{H-1}R, \perp, \mathbf{0}\right)$ when $h = H$. Finally, $\textsc{Simulate}^{\gamma,\delta}(\perp,\cdot,\cdot)$ deterministically returns $(0,\perp,\mathbf{0})$. As $\gamma < 1$, the inaccuracy of $\textsc{Simulate}^{\gamma,\delta}$ is at most the inaccuracy of $\textsc{Simulate}$, which is at most λ, by assumption.

Next, we prove that the value function of the discounted MDP $\mathscr{M}$ is close to the corresponding values of its H-horizon counterpart $\mathscr{M}'$. For this, we first need to agree on a way of transporting policy between $\mathscr{M}$ and $\mathscr{M}'$. This is done as follows: Let α be a function that maps histories in $\mathscr{M}$ to histories in $\mathscr{M}'$ by adding stage counters to them. Let α^{-1} be the "inverse", which simply drops stage indices from histories of $\mathscr{M}'$. For any h history of $\mathscr{M}$, $\alpha^{-1}(\alpha(h)) = h$, while $\alpha(\alpha^{-1}(h')) = h'$ holds for all histories h' of $\mathscr{M}'$ whose start state is from $\mathcal{S}'_1 = \mathcal{S} \times \{1\}$ and where the states in the history do not include $\perp$. If π' is any (possibly memoryful) policy of $\mathscr{M}'$, following π' in $\mathscr{M}$ means that given some history h of $\mathscr{M}$, the action $A \sim \pi'(\cdot|\alpha(h))$ should be taken. Conversely, using a policy π of $\mathscr{M}$ in $\mathscr{M}'$ means that given a history h', $A \sim \pi(\cdot|\alpha^{-1}(h'))$ should be taken. This way, we can view a policy of either $\mathscr{M}$ or $\mathscr{M}'$ as a policy of the other MDP.

Now take any policy π of $\mathscr{M}$ and take any $(s,h) \in \mathcal{S}'$. As π is also a policy of $\mathscr{M}'$, we can talk about its value function in $\mathscr{M}'$, which we denote by v'^{π}. By definition, $v'^{\pi}_h((s,h_0)) = \mathbb{E}'_{\pi,(s,h_0)}[\sum_{h'=1}^{H-h+1} r'_{(S_{h'},h_0+h'-1),A_{h'}}]$, where $\mathbb{E}'_{\pi,s'}$ denotes the expectation operator underlying the distribution $\mathbb{P}'_{\pi,s'}$ over state-action trajectories induced by the interconnection of π and $\mathscr{M}'$ given the initial state $s' \in \mathcal{S}'$. Similarly, we will use $\mathbb{E}_{\pi,s}$ to denote this operator when the MDP is $\mathscr{M}$ and the initial state is $s \in \mathcal{S}$, and we let $\mathbb{P}_{\pi,s}$ denote the underlying distribution. With this note that

$$\mathbb{P}'_{\pi,(s,h)}(U \times V) = \mathbb{P}_{\pi,s}(\alpha(U \times V)) \tag{17}$$

holds for any measurable subset U of $(\mathcal{S} \times [H] \times [A])^{H-h+1}$ and where $V = (\mathcal{S} \times [H] \times [A])^{\mathbb{N}_+}$ is the set of all histories. We claim that the following holds:

$$\left|v'^{\pi}_h((s,h)) - \gamma^{h-1}v^{\pi}(s)\right| \le \eta\,. \tag{18}$$

We calculate

$$\begin{aligned}
&\left|v'^{\pi}_h((s,h)) - \gamma^{h-1}v^{\pi}(s)\right| \\
&\quad= \left|\mathbb{E}'_{\pi,(s,h)}\left[\sum_{h'=1}^{H-h+1} r'_{(S_{h'},h+h'-1),A_{h'}}\right] - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=1}^{\infty}\gamma^{h'-1}r_{S_{h'},A_{h'}}\right]\right| \\
&\quad= \left|\gamma^{h-1}\mathbb{E}'_{\pi,(s,h)}\left[\sum_{h'=1}^{H-h+1}\gamma^{h'-1}r_{S_{h'},A_{h'}}\right] - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=1}^{H-h+1}\gamma^{h'-1}r_{S_{h'},A_{h'}}\right] - \gamma^{h-1}\mathbb{E}_{\pi,s}\left[\sum_{h'=H-h+2}^{\infty}\gamma^{h'-1}r_{S_{h'},A_{h'}}\right]\right| \\
&\quad= \left|-\gamma^{H}\mathbb{E}_{\pi,s}\left[\sum_{h'=H-h+2}^{\infty}\gamma^{h'-(H-h+2)}r_{S_{h'},A_{h'}}\right]\right| \qquad \text{(by Eq. (17))} \\
&\quad\le \gamma^H\sum_{i=0}^{\infty}\gamma^i = \frac{\gamma^H}{1-\gamma} \le \frac{\gamma^{\log((1-\gamma)\eta)/\log\gamma}}{1-\gamma} = \eta\,,
\end{aligned}$$

where in the last line used the fact that rewards are bounded in $[0,1]$. Now, notice that if π was a policy of $\mathscr{M}'$, Eq. (17) would still hold true, and as such, Eq. (18) also holds for π.

Take any policy π that is v-linearly realizable in $\mathscr{M}$ with misspecification η under the feature map φ. By definition, there exists a $\theta \in \mathbb{R}^d$ such that $|v^\pi(s) - \langle \varphi(s), \theta \rangle| \le \eta$ for all $s \in \mathcal{S}$ (ie. for all $(s,h) \in \mathcal{S}'$). By Eq. (18) and the triangle inequality, for all $(s,h) \in \mathcal{S}'$,

$$\begin{aligned}\left| v_h'^{\pi}((s,h)) - \left\langle \varphi_h^{\gamma,\delta}((s,h)), \theta \right\rangle \right| &= \left| v_h'^{\pi}((s,h)) - \gamma^{h-1} \langle \varphi(s), \theta \rangle \right| \\ &\le \left| v_h'^{\pi}((s,h)) - \gamma^{h-1} v^\pi(s) \right| + \gamma^{h-1} |v^\pi(s) - \langle \varphi(s), \theta \rangle| \le 2\eta \,.\end{aligned}$$

Therefore any such policy π is v-linearly realizable in MDP $\mathscr{M}'$ with misspecification 2η under the feature map $\varphi^{\gamma,\delta}$ for the respective stage h for each state $(s,h) \in \mathcal{S}'$.

Therefore we can apply Claim D.2 for featurized MDP $(\mathscr{M}', \varphi^{\gamma,\delta})$, initial set $\mathcal{S} \times \{1\}$, and λ-accurate simulator $\textsc{Simulate}^{\gamma,\delta}$, with misspecification $\eta' = 2\eta$ and $\delta' = 0.98\delta$, which guarantees that TensorPlan is (δ', B)-sound for MDP $\mathscr{M}'$ when run with this simulator and features. Furthermore, it has a worst-case per-state query-cost poly $\left((dH/\delta)^A, B\right)$. Denote by π_{TP} the policy induced by TensorPlan while interacting with the simulator $\textsc{Simulate}^{\gamma,\delta}$. We then have that π_{TP} satisfies

$$v_1'^{\pi_{\text{TP}}}((s,1)) \ge v_1'^{\circ}((s,1)) - 0.98\delta \,,$$

where $v'^{\pi_{\text{TP}}}$ is the H-horizon value function of π_{TP} in $\mathscr{M}'$ and $v'^{\circ} = v'^{\circ}_{B,2\eta}$ is the H-horizon $\varphi^{\gamma,\delta}$-compatible optimal value function of $\mathscr{M}'$ under misspecification 2η (cf. Equation (4)). Similarly, let $v^\circ = v^\circ_{B,\eta}$ be the discounted φ-compatible optimal value function of $\mathscr{M}$ under misspecification η. Let $\Pi'^{\circ}_{B,2\eta}$ be the set of MLD policies that are B-boundedly v-linearly realizable in MDP $\mathscr{M}'$ with misspecification 2η and features $\varphi^{\gamma,\delta}$, and let $\Pi^\circ_{B,\eta}$ be the set of MLD policies that are B-boundedly v-linearly realizable in $\mathscr{M}$ with misspecification η and features φ. Then, by definition, $v'^{\circ}_{B,2\eta}(s) = \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v_1'^{\pi}((s,1))$ and $v^\circ_{B,\eta}(s) = \sup_{\pi \in \pi'^{\circ}_{B,\eta}} v^\pi(s)$.

As we have seen, $\pi \in \Pi^\circ_{B,\eta}$ implies $\pi \in \Pi'^{\circ}_{B,2\eta}$, in other words, $\Pi^\circ_{B,\eta} \subseteq \Pi'^{\circ}_{B,2\eta}$. For any policy π Eq. (18) applies with any $(s,1) \in \mathscr{M}'$, and therefore

$$\begin{aligned}v^\circ_{B,\eta}(s) &= \sup_{\pi \in \Pi^\circ_{B,\eta}} v^\pi(s) \le \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v^\pi(s) \le \sup_{\pi \in \Pi'^{\circ}_{B,2\eta}} v_1'^{\pi}((s,1)) + \eta \\ &= v'^{\circ}_{B,2\eta}((s,1)) + \eta \le v_1'^{\pi_{\text{TP}}}((s,1)) + 0.98\delta + \eta \le v'^{\pi_{\text{TP}}}(s) + 0.98\delta + 2\eta \,,\end{aligned}$$

where the last inequality used again Eq. (18) with π_{TP}. Lastly we use that $\eta = \frac{\varepsilon}{24\sqrt{E_d}} \le \frac{\frac{\delta}{12H^2}/(1+0.5)}{24\sqrt{E_d}} \le \frac{\delta}{18H}/24 < 0.01\delta$ to obtain $v^\circ_{B,\eta}(s) \le v_1'^{\pi_{\text{TP}}}((s,1)) + \delta$, which establishes that TensorPlan's policy π_{TP} is (δ, B)-sound for the featurized MDP $(\mathscr{M}, \varphi)$ in the discounted setting, with misspecification $\eta \le \frac{\varepsilon}{24\sqrt{E_d}}$ and simulator accuracy $\lambda \le \frac{\varepsilon}{12\sqrt{E_d}}$. ∎