

Exponential Lower Bounds for Planning in MDPs With Linearly-Realizable Optimal Action-Value Functions

Gellért Weisz

DeepMind, London, UK

University College London, London, UK

Philip Amortila

University of Illinois, Urbana-Champaign, USA

Csaba Szepesvári

DeepMind, Edmonton, Canada

University of Alberta, Edmonton, Canada

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We consider the problem of local planning in fixed-horizon and discounted Markov Decision Processes (MDPs) with linear function approximation and a generative model under the assumption that the optimal action-value function lies in the span of a feature map that is available to the planner. Previous work has left open the question of whether there exist sound planners that need only $\text{poly}(H, d)$ queries regardless of the MDP, where H is the horizon and d is the dimensionality of the features. We answer this question in the negative: we show that any sound planner must query at least $\min(e^{\Omega(d)}, \Omega(2^H))$ samples in the fixed-horizon setting and $e^{\Omega(d)}$ samples in the discounted setting. We also show that for any $\delta > 0$, the least-squares value iteration algorithm with $\tilde{O}(H^5 d^{H+1} / \delta^2)$ queries can compute a δ -optimal policy in the fixed-horizon setting. We discuss implications and remaining open questions.

1. Introduction

Much research in the theory of planning (and learning) in large-scale Markov Decision Processes (MDP) with function approximation revolves around the conditions that are necessary for query-efficient planning. In the setting we consider, a planner is given access to a feature map that maps state-action pairs to d -dimensional (feature) vectors, and interacts with a simulation model (also known as a generative model) of the MDP to find a “good action” at any given state. The planner is promised that the unknown optimal action-values at any state-action pair can be written as the inner product between the features at that state and an unknown parameter vector. As is well known, actions that are optimal to take in a state are those that maximize the optimal action-values at the state. Thus, if the planner can compute a good approximation to the unknown parameter vector, it could return a good action at the initial state (and perhaps even at all the states). The hope then is that regardless the size of the state and action spaces, a planner may be able to find a good action while only interacting with the simulator $\mathcal{O}(\text{poly}(H, d))$ times, where H is the horizon of the MDP.

Much has been written about planning (and learning) in the presence of (linear) function approximation with additional assumptions. Yet, the basic question:

Is realizability of the optimal action-value function enough to guarantee query-efficient learning?

had remained unanswered so far. In this paper, we answer this question in the negative by proving that for any $\eta \in \left(0, \frac{1}{2} - \frac{2}{\log_2(d-1)}\right]$, the worst-case query-complexity is at least

$$\Omega\left(\min\left\{e^{(d-1)^{2\eta}/8}, 2^{-H}d^{H(\frac{1}{2}-\eta)}\right\}\right).$$

In particular, this implies a lower bound of $\min(e^{\Omega(d)}, \Omega(2^H))$. Thus, in general, the query complexity of a planner which returns a “good” solution may be exponential in d or exponential in H (for a precise statement of this result, see Theorem 9). Since our results hold under the generative setting, by extension, they also apply to the more general online setting (where a simulator is not given). Our proof also translates into the discounted MDP setting, where we prove a query complexity of $e^{\Omega(d)}$. For ease of presentation, this paper primarily examines the fixed-horizon setting, presenting the relevant changes to prove the lower bound for the discounted MDP setting in Section 4.9.

Inspired by the recent work of Du et al. (2019a), our lower bound construction uses the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984) to create a large set of nearly-orthogonal feature vectors. At a high-level, each state of our construction has this large set of features available as actions (see Figure 2). The parameter of the optimal action-value function is then chosen to be the feature of one of these actions – by realizability this entails that one action per state will have higher value while the others are nearly identical. At the final stage of the MDP, this gap between the optimal action and its non-optimal counterparts will be exponentially small. The gap is then increased multiplicatively as the planner approaches the initial stage. There are additional subtleties needed to ensure realizability, which distinguishes our approach from the bandit-like lower bound construction of Du et al. (2019a).

We note that our lower bound construction is deterministic, save for random rewards obtained at the last time-step of each episode. This is in contrast to the result of Wen and Roy (2013, Theorem 1), which establishes that for fully deterministic MDPs, there exists a planner that chooses only at most d times suboptimal actions out of any number of episodes. Together with our construction, this suggests a stark separation between planning/learning in deterministic and in stochastic environments.

1.1. Related work

The idea of using function approximation to help solving large-scale MDPs originates in the 1960s (e.g., Bellman et al., 1963): These early works provided experimental evidence that in MDPs with large (or even infinite) state spaces, the optimal value function can be well approximated with the linear combination of a few basis functions, which in turn encouraged work to explore how such basis functions could be used to design efficient planning algorithms whose compute cost is *independent of the size of the state space* and depends mildly on the number of basis functions and the planning horizon. The seminal paper of Schweitzer and Seidmann (1985) gave general, “least-squares” versions of the basic dynamic programming methods (value iteration, policy iteration and linear programming) that relied on the basis functions. However, no analysis was provided.

In an independent line of work, Kearns et al. (2002) noticed that if a planner that is given simulator access to the MDP is asked to return a good action only at some fixed state (provided as part of the input to the planner), the query (and computational) complexity of planning can already be made independent of the size of the state space. However, without further extra structure (such as the presence of helpful basis functions), the query complexity of local planning turns out to be exponential in the planning horizon (Kearns et al., 2002).

Merging the two directions of research gives rise to the central question of efficient planning in MDPs, namely whether the aforementioned exponential dependence can be avoided by assuming the presence of “helpful” basis functions. When the action-value function of *all* policies are well-represented by some linear combination of the basis functions, “fitted policy iteration”-type algorithms have recently been shown to achieve polynomial query (and computational) complexity (Yang and Wang, 2019; Lattimore et al., 2020). These results extend to the case when the basis functions induce a worst-case approximation error $\varepsilon > 0$ over the policies, as long as the planner is required to return $\mathcal{O}(\sqrt{d}\varepsilon)$ -optimal policies only. Interestingly, demanding $\mathcal{O}(\varepsilon)$ -optimal policies worsens the query complexity to be *exponential*, namely to $\Omega(e^{\Omega(d)} \wedge 2^H)$ (Du et al., 2019a).

While these results give a (nearly) complete characterization of the query and computational complexity under the said assumption, they left open the question whether a similar result may hold true under the milder assumption that only the optimal value function is well-approximated by the basis functions. Positive results for this question have been obtained under a number of additional assumptions. In the online setting, Wen and Roy (2013) provides a low-regret guarantee for deterministic MDPs (when both the rewards and transitions are deterministic). This was later extended to “low-variance” MDPs by Du et al. (2019b) under an additional gap assumption, which requires knowledge of the minimum separation in value between any optimal action and the second best action. Their sample complexity also scales in the inverse gap. Further positive results have been obtained for MDPs with a linear reward function and a low-rank transition matrix (Jin et al., 2020) (a condition which entails the above assumption of Lattimore et al. (2020)), and for MDPs with low “Bellman ranks” (Jiang et al., 2017). In the planning setting, Du et al. (2019a) give a query complexity result for least-squares value iteration which scales as $\mathcal{O}(\text{poly}(H, d))$ provided that the inverse gap is treated as a fixed parameter and is itself $\mathcal{O}(\text{poly}(H, d))$, which is a restrictive condition. Additionally, Shariff and Szepesvári (2020) obtains polynomial bounds under q^* -realizability with the additional assumption that the features for all state-action pairs are inside the convex hull of the features at $\mathcal{O}(\text{poly}(H, d))$ state-action pairs.

Lastly, we highlight the concurrent work of Wang et al. (2020), which establishes a similar exponential lower bound for the analogous question in the setting of *offline RL*.¹ In their setting, they assume linear-realizability and an additional assumption related to the data-generation process. While their construction does not work in the generative setting, there are a number of interesting similarities, notably a scaling phenomenon which geometrically reduces values throughout the horizons.

The rest of the paper is organized as follows. In the next section, we provide preliminary definitions and introduce notation. Section 3 introduces the formal problem definition, which we call local planning under q^* -realizability with linear function approximation. The exponential lower bound, together with its proof, is presented in Section 4. Section 5 shows that a least-squares value iteration with $\mathcal{O}(H^5 d^{H+1} / \delta^2)$ queries is able to guarantee δ -optimal policies. The paper is concluded with a discussion of the results and the remaining open problems Section 6.

2. Preliminaries

Let \mathbb{R} denote the set of real numbers, and for a positive integer i , let $[i] = \{1, \dots, i\}$ be the set of integers from 1 to i . We let $\mathbb{N}_+ = \{1, 2, \dots\}$ denote the set of positive integers. We write $\mathcal{M}_1(\mathcal{X})$ for

1. In offline RL, the agent is not allowed to interact with the environment but is instead given a dataset of transition tuples drawn from a certain distribution.

the set of probability measures on a measurable space $(\mathcal{X}, \mathcal{F})$ where $\mathcal{F} \subset 2^{\mathcal{X}}$ is a σ -algebra over \mathcal{X} . Random quantities are denoted by capital letters, but some capital letters denote non-random quantities. We let $a \wedge b = \min(a, b)$. For a probability measure μ over a topological space, we let $\text{supp}(\mu)$ denote its support. We use $\|f\|_{\infty} = \sup_{x \in \mathcal{X}} |f(x)|$ to denote the supremum norm of a function $f : \mathcal{X} \rightarrow \mathbb{R}$. We use $\mathbb{I}\{A\}$ to denote the indicator of a set A .

We consider MDPs given by a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{Q})$, where \mathcal{S} and \mathcal{A} are the set of states and actions, respectively, $\mathcal{Q} = (P_a : \mathcal{S} \rightarrow \mathcal{M}_1(\mathbb{R} \times \mathcal{S}))_{a \in \mathcal{A}}$ are Markov transition kernels from \mathcal{S} to $\mathbb{R} \times \mathcal{S}$ (Puterman, 1994). We assume that \mathcal{S} is a measurable subset of some Euclidean space,² while we assume that \mathcal{A} is finite. An MDP describes an environment that an agent interacts with in a sequential manner in discrete time steps. At time t , the agent observes the state $S_t \in \mathcal{S}$, takes an action A_t to transition to state S_{t+1} while incurring the reward R_{t+1} where $(S_{t+1}, R_{t+1}) \sim Q_{A_t}(\cdot | S_t)$.

In the fixed, finite-horizon setting, the agent-environment interaction happens in episodes, with each episode lasting for $H > 0$ steps. For simplicity, we consider the variant where every episode starts with a fixed initial state $s_1 \in \mathcal{S}$. The goal of the agent is to maximize the total expected reward incurred in the episodes. Let $r_a(s) = \int Q_a(dr, \mathcal{S}|s)r$ and $P_a(ds'|s) = Q_a(\mathbb{R}, ds'|s)$. For $h \in [H]$, define \mathcal{S}_h as the set of states accessible from s_1 in $h - 1$ steps. Thus, $\mathcal{S}_1 = \{s_1\}$ and $\mathcal{S}_{h+1} = \{s' \in \mathcal{S} : \exists a \in \mathcal{A}, s \in \mathcal{S}_h \text{ s.t. } s' \in \text{supp}P_a(\cdot | s)\}$.

A memoryless policy in this setting takes the form $\pi = (\pi_h)_{h \in [H]}$ where $\pi_h : \mathcal{S}_h \rightarrow \mathcal{M}_1(\mathcal{A})$ is a probability kernel from the states in \mathcal{S}_h to actions. Given $h \in [H]$ and a state $s \in \mathcal{S}_h$, the value $v_h^\pi(s)$ of π for stage h and state s is defined to be the total expected reward incurred when π is deployed beginning at stage h from state s . Denoting by \mathbb{E}_π the expectation operator induced over state-action sequences $(S_1, A_1, \dots, S_H, A_H)$ by the interconnection of π and M , formally we have

$$v_h^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=h}^H r_{A_t}(S_t) \mid S_h = s \right],$$

We call $v^\pi = (v_h^\pi)_{h \in [H]}$ the **value function** of policy π . The value of the *optimal* policy π^\star is found to satisfy:

$$q_h^\star(s, a) = r_a(s) + \int v_{h+1}^\star(s') P_a(ds' | s), \quad (1)$$

$$v_h^\star(s) = \max_{a \in \mathcal{A}} q_h^\star(s, a), \quad (2)$$

with

$$v_{H+1}^\star(s) = 0, \quad s \in \mathcal{S}_{H+1}.$$

Here, $v^\star = (v_h^\star)_{h \in [H]}$ is called the optimal value function and $q^\star = (q_h^\star)_{h \in [H]}$ is called the optimal action-value function (we exclude v_{H+1}^\star , as this is identically zero). Without loss of generality, we will assume that $\mathcal{S} = \bigcup_{h=1}^{H+1} \mathcal{S}_h$. An optimal H -horizon behavior for an agent in the MDP is one where, for each stage $h \in [H]$ of an episode, for each state $s \in \mathcal{S}_h$, the agent takes an action that maximizes $q_h^\star(s, \cdot)$. Further, the solution to the above equations is unique (Puterman, 1994).

We will also need the notion of the **action gap** of an action a : At stage $h \in [H]$ and state $s \in \mathcal{S}_h$, the action gap of action a is defined to be $\Delta_h^\star(s, a) = v_h^\star(s) - q_h^\star(s, a)$. When needed, we use $\Delta_{h, M}^\star$

2. We only need a restriction on what these sets are so that we can refer to the set of all MDPs concerned. Clearly, we could allow much more generality here. This matters for the upper bound: for the lower bound finite sets are sufficient.

to indicate the dependence of the action gap on the MDP M . Finally, for $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and a memoryless policy π , we will use $f(s, \pi)$ as the shorthand for $\sum_{a \in \mathcal{A}} \pi(a|s) f(s, a)$. Similarly, we will also use P_π to denote the Markov transition kernel defined using $P_\pi(ds'|s) = \sum_{a \in \mathcal{A}} \pi(a|s) P_a(ds'|s)$.

3. Planning with features realizing q^*

In this section we introduce the formal problem definition. We consider the *local planning problem* in the presence of an MDP simulator (or “generative model” [Kearns et al. 2002](#)) and a feature map $\varphi = (\varphi_h)_{h \in [H]}$, $\varphi_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$, that, for some stage h , maps state-action pairs to d -dimensional vectors with some finite $d > 0$. In the local planning problem, the planner is given a stage $h \in [H]$ and an input state $s_{\text{in}} \in \mathcal{S}_h$, and must return a (possibly random) action $A \in \mathcal{A}$. Roughly, we ask that A has low suboptimality as measured by the action gap (a more precise statement is postponed to Definition 5). The planner has access to the set of states, the decomposition $\mathcal{S} = \cup_{h=1}^{H+1} \mathcal{S}_h$, the set of actions, and the feature map. Furthermore, they are promised that q^* lies in the span of the features. This promise is captured by the following assumption:

Assumption 1 (Realizability) *There exists a vector $\theta^* \in \mathbb{R}^d$ such that for any $h \in [H]$ and state-action pair $(s, a) \in \mathcal{S}_h \times \mathcal{A}$,*

$$q_h^*(s, a) = \langle \varphi_h(s, a), \theta^* \rangle. \quad (3)$$

If this holds, we say that the features φ realize the optimal action-value function q^ .*

Definition 2 *A pair (M, φ) is called q^* -realizable if M is an MDP whose optimal action-value function q^* is realized by the feature map φ .*

We further assume that the MDP has rewards bounded by $[0, 1]$ almost surely:

Assumption 3 (Bounded rewards) *For any time t , the reward R_{t+1} received after taking any action in any state, $\mathbb{P}(R_{t+1} \notin [0, 1]) = 0$.*

Definition 4 *We write $\mathcal{M}_{H,d}$ for the set of MDP-feature map pairs (M, φ) where M has bounded rewards, the horizon is H , φ has dimension d , and (M, φ) is q^* -realizable.*

Interaction between the planner and the simulator, implemented policy A planner is given a stage $h \in [H]$ and an input a state $s_{\text{in}} \in \mathcal{S}_h$. The planner can then access the MDP $M = (\mathcal{S}, \mathcal{A}, \mathcal{Q})$ by sending *queries* to the underlying simulator. A query is defined to be a state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. Given the query (s, a) , the simulator returns a pair $(R, S) \sim Q_a(\cdot, \cdot | s)$. Based on the result, the planner can then send in a new state-action pair, and so on and so forth. Thus, in general, the query will be random, as it depends on (random) data received previously by the planner from the simulator. For the sake of generality, we also allow the planner to inject extra randomness into the planning process. The planner eventually must decide to stop (say after $N < \infty$ queries) and output an action $A \in \mathcal{A}$. Formally, a planner together with $\varphi, \mathcal{S} \times \mathcal{A}, s_{\text{in}}$ and h determines two sequences of probability kernels, $\rho = (\rho_t)_{t \geq 1}$ and $\lambda = (\lambda_t)_{t \geq 1}$, where ρ_t is a probability kernel from $\mathcal{H}_{t-1} := (\mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S} \times \{0, 1\})^{t-1}$ to $\mathcal{S} \times \mathcal{A}$, while λ_t is a probability kernel from $\mathcal{H}_{t-1} \times (\mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S})$ to $\{0, 1\}$. For $t \geq 1$, λ_t determines whether the planner should continue with the queries. Without loss of generality, we use 1 to denote the choice that the planner continues. Similarly, for $t \geq 1$, ρ_t determines the query to be

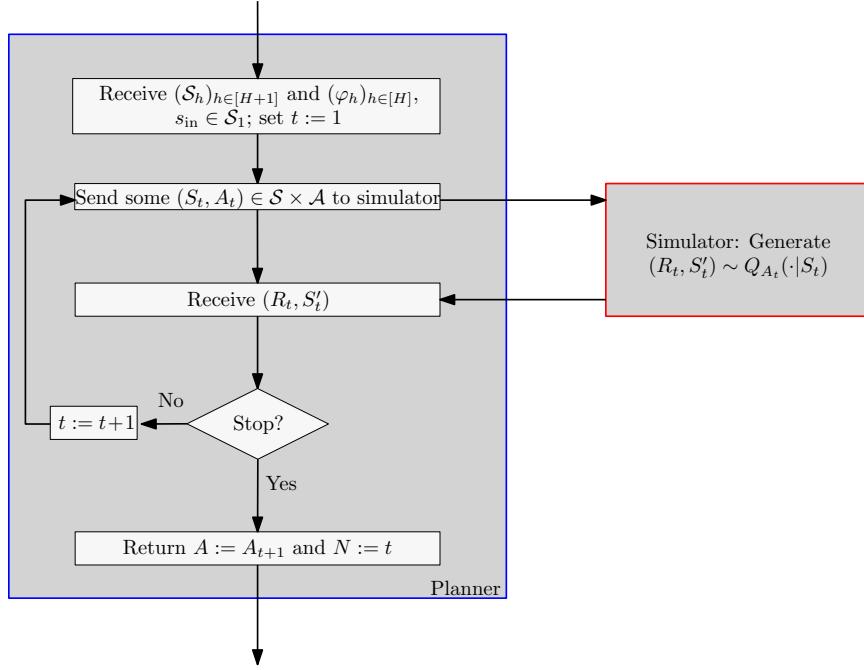


Figure 1: Data flow between the planner and the simulator

used in step t if the planner has not stopped in time step $t - 1$, and if it has just stopped, the action component sampled from ρ_t determines the action to be returned by the planner. The interaction between the planner and the simulator is shown in Fig. 1. In general, both the returned action A and the number of queries N are random. A planner is well-formed if $N < \infty$ holds with probability one for all MDPs that the planner is designed to interact with.

When the planner is well-formed, A is well-defined. We write $\pi_h(\cdot | s_{\text{in}})$ for the distribution of A . These distributions altogether define a memoryless policy $\pi = (\pi_h)_{h \in [H]}$: this is the policy that will be followed if the planner is called sequentially on the trajectory where the actions taken are those chosen by the planner. Thus, π is the policy that is effectively *implemented* by the planner. Note that the planner does not return π , nor does it return $\pi_h(\cdot | s_{\text{in}})$. In particular, when the action set is large, $\pi_h(\cdot | s_{\text{in}})$ may be too large to even write down, let alone π . Note also that $\pi_h(\cdot | s_{\text{in}})$ itself is not a random variable as this distribution already accounts for the randomness of the simulator and that of the planner.

The probability distribution induced over interaction sequences In what follows we will need a probability distribution \mathbb{P} over the infinitely long interaction sequences $\mathcal{H} = (S \times \mathcal{A} \times [0, 1] \times \mathcal{S} \times \{0, 1\})^{\mathbb{N}^+}$ that the interconnection between a planner and a simulator induces. Denote by $(S_1, A_1, R_1, S'_1, C_1, S_2, A_2, R_2, S'_2, C_2, \dots) \in \mathcal{H}$ such a sequence (formally, these are the coordinate functions from \mathcal{H} to the respective spaces) and for $t \geq 1$ introduce the abbreviation $H_{t-1} = (S_1, A_1, R_1, S'_1, C_1, \dots, S_{t-1}, A_{t-1}, R_{t-1}, S'_{t-1}, C_{t-1})$. The induced probability distribution \mathbb{P} satisfies the following properties: For any $t \geq 1$, \mathbb{P} -almost surely,

- Next query follows ρ_t : $\mathbb{P}((S_t, A_t) \in \cdot | H_{t-1}) = \rho_t(\cdot | H_{t-1})$;

- Data received follows Q : $\mathbb{P}((R_t, S'_t) \in \cdot | H_{t-1}, S_t, A_t) = Q_{A_t}(\cdot | S_t)$;
- Continuation follows λ_t : $\mathbb{P}(C_t = 1 | H_{t-1}, S_t, A_t, R_t, S'_t) = \lambda_t(1 | H_{t-1}, S_t, A_t, R_t, S'_t)$.

Note that with this notation, $N = \min_{t \geq 1} \mathbb{I}\{C_t = 0\}$. Also, to save on notation, without loss of generality, we let the output of the planner be A_{N+1} , so the distribution of the output is also determined by ρ . Since the planner takes as input φ , s_{in} and h , \mathbb{P} also depends on these quantities, in addition to depending on the MDP $M = (\mathcal{S}, \mathcal{A}, Q)$. We use $\mathbb{P}_{M, \varphi, s_{\text{in}}, h}^{\mathcal{P}}$ to signify this dependence when needed. We will also $\mathbb{E}_{M, \varphi, s_{\text{in}}, h}^{\mathcal{P}}$ to denote the corresponding expectation operator. The Ionescu-Tulcea theorem guarantees that \mathbb{P} with these properties exist (see [Ionescu Tulcea 1949](#), or Theorem 6.17 in the book of [Kallenberg \(2002\)](#)). Finally, note that we defined \mathbb{P} for sequences of infinite length regardless of the planner by assuming that ρ is defined for arbitrary histories, including those that have $C_t = 0$ possibly multiple times. Of course, there is no loss of generality here: statements concerning soundness and query complexity of planners are only concerned with histories up to the first time when $C_t = 0$. However, for our proofs, it will be convenient for probabilities to be assigned to events formed of sequences of infinite length. Note that with this notation, we have that for any $a \in \mathcal{A}$, $\pi_h(a | s_{\text{in}}) = \mathbb{P}_{M, \varphi, s_{\text{in}}, h}^{\mathcal{P}}(A_{N+1} = a)$.

Sound planners and query cost The goal of the planner is to arrive at a policy π that is nearly optimal regardless of stages and states. We define the *suboptimality gap* of a policy π by

$$\delta^{\pi} := \sup_{h \in [H], s \in \mathcal{S}_h} \{v_h^{\star}(s) - v_h^{\pi}(s)\}.$$

When the dependence on the MDP is needed, we use δ_M^{π} instead of δ^{π} . The following definition introduces two notions of soundness for the policies implemented by planners.

Definition 5 (δ -soundness and (δ, ζ) -transition-soundness) *Recall that $\mathcal{M}_{H,d}$ is the set of q^{\star} -realizable and bounded (M, φ) pairs with horizon H and dimension d (Definition 4).*

Let \mathcal{P} be a planner and $\pi_{\mathcal{P}, M, \varphi}$ be the policy which is implemented by the planner when given φ and upon interaction with the simulator of M . Let $\delta \geq 0$ and $\zeta \in [0, 1]$.

- We say \mathcal{P} is δ -sound if for any $(M, \varphi) \in \mathcal{M}_{H,d}$, the suboptimality of $\pi_{\mathcal{P}, M, \varphi}$ satisfies

$$\delta^{\pi_{\mathcal{P}, M, \varphi}} \leq \delta;$$

- We say that \mathcal{P} is (δ, ζ) -transition-sound if for any $(M, \varphi) \in \mathcal{M}_{H,d}$, and for all $h \in [H], s \in \mathcal{S}_h$

$$\sum_{a \in \mathcal{A}} \mathbb{I}\{\Delta_{h,M}^{\star}(s, a) \geq \delta\} \pi_h(a | s) \leq \zeta.$$

For the lower bound we will find it more convenient to consider (δ, ζ) -transition-soundness. The following proposition shows that these two notions of soundness are roughly equivalent:

Proposition 6 *For $\zeta \in (0, 1]$ and $(M, \varphi) \in \mathcal{M}_{H,d}$, a planner that is (δ, ζ) -transition-sound is also $H\delta + H(H+1)\zeta/2$ -sound, and a planner that is δ -sound is also $(\delta/\zeta, \zeta)$ -transition-sound.*

In particular, this means that if \mathcal{P}_δ is the set of all δ -sound planners (for some model class) and $\mathcal{P}_{(\delta,\zeta)}$ is the set of all (δ,ζ) -transition-sound planners then $\mathcal{P}_\delta \subset \mathcal{P}_{(\delta/\zeta,\zeta)}$ and $\mathcal{P}_{(\delta,\zeta)} \subset \mathcal{P}_{H\delta+H(H+1)\zeta/2}$. Thus, if for some \mathcal{P}' , $\mathcal{P}_{(\delta/\zeta,\zeta)} \cap \mathcal{P}' = \emptyset$ then we also have that $\mathcal{P}_\delta \cap \mathcal{P}' = \emptyset$.

Proof Note that the soundness of a planner only depends on how well the policy it implements is doing in the MDP. Hence, it suffices to show that for any MDP in $(M, \varphi) \in \mathcal{M}_{H,d}$ and policy π, δ and ζ the following hold:

1. If for any stage $h \in [H]$ and state $s \in \mathcal{S}_h$, $\sum_a \mathbb{I}\{\Delta_h^\star(s, a) \geq \delta\} \pi_h(a | s) \leq \zeta$ then $\delta^\pi \leq H\delta + H(H+1)\zeta/2$;
2. If $\delta^\pi \leq \delta$ then for any stage $h \in [H]$ and state $s \in \mathcal{S}_h$, $\sum_a \mathbb{I}\{\Delta_h^\star(s, a) \geq \delta/\zeta\} \pi(a | s) \leq \zeta$.

For the first claim fix $h \in [H]$, $s \in \mathcal{S}_h$. Then,

$$\begin{aligned}
 v_h^\star(s) - v_h^\pi(s) &= v_h^\star(s) - \mathbb{E}_\pi \left[\sum_{t=h}^H r_{A_t}(S_t) \mid S_h = s \right] \\
 &= v_h^\star(s) - \mathbb{E}_\pi \left[\sum_{t=h}^H q_t^\star(S_t, A_t) - v_{t+1}^\star(S_{t+1}) \mid S_h = s \right] \\
 &= \mathbb{E}_\pi \left[\sum_{t=h}^H v_t^\star(S_t) - q_t^\star(S_t, A_t) \mid S_h = s \right] \\
 &= \sum_{t=h}^H \mathbb{E}_\pi [\Delta_t^\star(S_t, A_t) \mathbb{I}\{\Delta_t^\star(S_t, A_t) < \delta\} \mid S_h = s] \\
 &\quad + \sum_{t=h}^H \mathbb{E}_\pi [v_t^\star(S_t) - q_t^\star(S_t, A_t) \mathbb{I}\{\Delta_t^\star(S_t, A_t) \geq \delta\} \mid S_h = s] \\
 &\leq \delta H + \zeta \frac{H(H+1)}{2},
 \end{aligned}$$

where the last step used that π is (δ, ζ) -transition-sound and that $v_t^\star(S_t) - q_t^\star(S_t, A_t) \leq v_t^\star(S_t) \leq H - t + 1$ due to the bounded rewards (since $(M, \varphi) \in \mathcal{M}_{H,d}$, Assumption 3 holds). The result follows by taking the supremum over h and s of both sides and noting that we get δ^π on the left-hand side.

To prove the second claim let π be such that $\delta^\pi \leq \delta$ and fix $h \in [H]$ and $s \in \mathcal{S}_h$. Let $A \sim \pi_h(\cdot | s)$. Then, $\delta^\pi \geq v_h^\star(s) - v_h^\pi(s) \geq \sum_{a \in \mathcal{A}} \pi_h(a | s) (v_h^\star(s) - q^\star(s, a)) = \mathbb{E}[v_h^\star(s) - q^\star(s, A)]$. By Markov's inequality, $\mathbb{P}(v_h^\star(s) - q^\star(s, A) \geq \mathbb{E}[v_h^\star(s) - q^\star(s, A)]/\zeta) \leq \zeta$, which is what we wanted to show. ■

Besides its soundness, the second figure of merit of a planner is the expected number of queries that was used for planning.

Definition 7 (Query cost of a planner) *The query cost of the planner, $C(\mathcal{P}, H, d)$, is the worst-case expected number of queries over any q^\star -realizable (M, φ) with horizon H and dimension d , and any input state:*

$$C(\mathcal{P}, H, d) = \sup_{(M, \varphi) \in \mathcal{M}_{H,d}} \sup_{s_{in}} \mathbb{E}_{M, \varphi, s_{in}, h}^{\mathcal{P}} [N],$$

where $\mathbb{E}_{M, \varphi, s_{in}, h}^{\mathcal{P}} [\cdot]$ is the expectation under the distribution of planner interactions as defined above.

In the following section, we give a lower bound on the query cost of any planner on the set of q^* -realizable (M, φ) pairs.

Remark 8 Note that in an episode of length H , the planner is called H times. The extreme of this is when a planner is used in an infinite horizon or discounted MDP, as in this case the planner would be called infinitely many times and could thus submit infinitely many queries to the simulator, even when the number of queries in each call is limited. Note also that the definition implicitly forbids planners that have a global memory where they collect information about the MDP they interact with (across calls with different (s_{in}, h) pairs). However, this is only to simplify the presentation, as our lower bound proof is based on the planner making a mistake at the first call, where such global memory would be empty.

4. Lower bound

This section is devoted to proving the main result of the paper:

Theorem 9 For any d and H large enough and any

$$0 < \eta \leq \frac{1}{2} - \frac{2}{\log_2(d-1)}, \quad (4)$$

any planner \mathcal{P} that is $\frac{9}{128}$ -sound on the set of q^* -realizable H -horizon local planning problems with linear function approximation with features in \mathbb{R}^d , the planner's worst-case query complexity $C(\mathcal{P}, H, d)$ satisfies

$$C(\mathcal{P}, H, d) = \Omega \left(\min \left\{ e^{(d-1)^{2\eta}/8}, 2^{-H} d^{H(\frac{1}{2}-\eta)} \right\} \right).$$

Thus, the lower bound says that the query complexity is either exponential in the dimension, or it is exponential in the planning horizon (when the dimension is large). While we gave the result using the $\Omega(\cdot)$ formalism to minimize clutter, in our proof we compute the lower bound in an explicit form. In fact, the lower bound is shown to hold for $d \geq 18$ and $H \geq 1$, although no attempt is made to optimize the lower bound on d . The above result implies the following:

Corollary 10 By choosing $\eta = \frac{1}{2} - \frac{2}{\log_2(d-1)}$,

$$C(\mathcal{P}, H, d) = \min(e^{\Omega(d)}, \Omega(2^H)).$$

4.1. Proof: Main ideas

First note that thanks to Proposition 6, it suffices to show the query lower bound for $(\delta/\zeta, \zeta)$ -transition-sound planners with some $\zeta > 0$.

At a high level, if α_H is the planner's "measuring accuracy" at the final stage H , we construct MDPs where all except some optimal action a have q^* -value in $[0, \alpha_H]$ at the last stage. All these values look like 0 to the planner. Inspired by the work of Du et al. (2019a), the idea of our construction will be to use the Johnson-Lindenstrauss lemma to construct features. This allows us to give action a the optimal value $d^{\frac{1}{2}-\eta}\alpha_H$. Unless a is played, the planner's measuring accuracy for stage $H-1$ is thus $\alpha_{H-1} = d^{\frac{1}{2}-\eta}\alpha_H$. At stage $H-1$, the same argument is repeated, until finally we derive a

suboptimality gap at the first stage that is exponentially larger than the measuring accuracy α_H . An important deviation from the construction of [Du et al. \(2019a\)](#) is that we need to ensure realizability with the growing gaps. To be able to do this, we let the optimal action leave the normal states (which like in their paper are arranged in a tree) and reach a stream of special states with no rewards and a straight path to the end. Then, to hide the identity of the optimal action, we need to have many actions. Note that the dynamics are deterministic and the rewards are also deterministic except for the last stage and the amount of randomness here is chosen carefully so that planners that focus on the last stage need many interactions before identifying the optimal action at the first stage.

The rigorous proof is based on constructing $k \approx e^{d^2n/8}$ MDPs, $(M_a)_{a \in \mathcal{A}}$ with a shared action set $\mathcal{A} = [k]$ and a shared state space \mathcal{S} and initial state $s_1 \in \mathcal{S}$ and feature map φ such that in MDP M_a , a is the optimal action in all states and in particular in state s_1 , (M_a, φ) are q^* -realizable for any $a \in \mathcal{A}$ and the rewards are in $[0, 1]$. The planner will face one of these MDPs, the identity of which is hidden. Given $n \approx k/4 \wedge (d/2)^{H(1/2-\eta)}$, which one should think of as an intended upper bound on the number of interactions between the planner and the MDP, the MDPs and the feature map are chosen so that the identity of a is difficult to establish in n planning steps while the action gap at the initial state is large. In particular, for any $a \in \mathcal{A}$,

$$\min_{a' \neq a} \Delta_{1, M_a}^*(s_1, a') \geq \frac{1}{4}. \quad (5)$$

This means that for any $\delta \leq 1/4$ and $\zeta \in (0, 1)$, a (δ, ζ) -transition-sound planner run on M_a needs to figure out the identity of a with at least $1 - \zeta$ probability. To hide the identity of a , we use random rewards in the last stage with low signal to noise ratio.

To argue that planners using at most n interactions have a hard time to identify the optimal action, a “test MDP” M_0 is constructed, which, by construction, is “symmetric” over the actions. This MDP is used to find out any bias a planner with “essentially no information” may have: If a planner is under-utilizing some action a in this MDP, we will show that the planner will fail on M_a in identifying a as the action to be taken at s_1 . To be able to show this, we make sure that M_0 shares the structure of the other MDPs apart from the fact that in it all actions behave the same (M_0 is invariant to permutations of the action set). As a result, letting $A_{1:n}$ denote the random sequence of actions taken by the planner \mathcal{P} , it will hold that the probability assigned to $a \notin A_{1:n}$ when \mathcal{P} is interacting in M_0 lower bounds the same probability under M_a :

$$\mathbb{P}_{M_a}^{\mathcal{P}}(a \notin A_{1:n}) \geq \frac{3}{4} \mathbb{P}_{M_0}^{\mathcal{P}}(a \notin A_{1:n}) \quad \text{for all planners } \mathcal{P} \text{ and } a \in [k]. \quad (6)$$

Here, $\mathbb{P}_M^{\mathcal{P}}$ is the probability distribution jointly induced by \mathcal{P} and M (with the input $\varphi, s_{\text{in}} = s_1$ and $h = 1$), as defined in [Section 3](#). To minimize clutter the dependence on φ, s_{in} and h is not shown. The following lemma shows that this construction allows one to give a lower bound on the query complexity of sound planners:

Lemma 11 *Let $d, H \geq 1$, $k \geq 2$ and let $n \leq \lfloor k/4 \rfloor$. Assume that there exists $(M_a, \varphi) \in \mathcal{M}_{H,d}$ for any $a \in \mathcal{A}$, and M_0 such that Eq. (5) and Eq. (6) hold. Take any $(\frac{1}{4}, \frac{9}{32})$ -transition-sound planner \mathcal{P} for the q^* -realizable fixed horizon planning problem. Then there exists $a \in \mathcal{A}$ such that $\mathbb{E}_{M_a}^{\mathcal{P}}[N] \geq \frac{9}{32}n$.*

To use this result, we will need to show that Eq. (5) and Eq. (6) can be satisfied for “large” values of n as suggested above.

Proof Let k and n be as in the statement of the lemma. Fix the planner \mathcal{P} whose input is $s_{\text{in}} = s_1$, φ and $h = 1$. Since $A_{1:n} = (A_1, \dots, A_n)$ has at most $n \leq k/4$ actions, by the pigeonhole principle, there exists $a \in \mathcal{A}$ such that $\mathbb{P}_{M_0}^{\mathcal{P}}(a \in A_{1:n}) \leq n/k \leq 1/4$. Take such an action a . We have $\mathbb{P}_{M_0}^{\mathcal{P}}(a \notin A_{1:n}) \geq 3/4$. This, together with Eq. (6) gives that

$$\mathbb{P}_{M_a}^{\mathcal{P}}(a \notin A_{1:n}) > \left(\frac{3}{4}\right)^2. \quad (7)$$

Let $A = A_{N+1}$ be the output of the planner. Since the planner is $(\frac{1}{4}, \zeta := \frac{9}{32})$ -transition-sound and $(M_a, \varphi) \in \mathcal{M}_{H,d}$,

$$\begin{aligned} \zeta &\geq \mathbb{P}_{M_a}^{\mathcal{P}}(\Delta_{1,M_a}^{\star}(s_1, A) \geq \frac{1}{4}) = \mathbb{P}_{M_a}^{\mathcal{P}}(a \neq A) \\ &\geq \mathbb{P}_{M_a}^{\mathcal{P}}(a \neq A_1, \dots, a \neq A_{N+1}, N \leq n-1) \geq \mathbb{P}_{M_a}^{\mathcal{P}}(a \neq A_1, \dots, a \neq A_n, N \leq n-1) \\ &= \mathbb{P}_{M_a}^{\mathcal{P}}(a \notin A_{1:n}, N \leq n-1), \end{aligned}$$

where the first equality used Eq. (5). From $P(A \cap B) = P(A) - P(B^c \cap A) \geq P(A) - P(B^c)$, we have

$$\zeta \geq \mathbb{P}_{M_a}^{\mathcal{P}}(a \notin A_{1:n}, N \leq n-1) \geq \mathbb{P}_{M_a}^{\mathcal{P}}(a \notin A_{1:n}) - \mathbb{P}_{M_a}^{\mathcal{P}}(N \geq n) > \left(\frac{3}{4}\right)^2 - \mathbb{E}_{M_a}^{\mathcal{P}}[N]/n,$$

where the last inequality follows from Eq. (7) and that from Markov's inequality, $\mathbb{P}_{M_a}^{\mathcal{P}}(N \geq n) \leq \mathbb{E}_{M_a}^{\mathcal{P}}[N]/n$. Reordering and plugging in the value of ζ gives

$$\mathbb{E}_{M_a}^{\mathcal{P}}[N]/n \geq \frac{1}{2} \left(\frac{3}{4}\right)^2.$$

The desired claim follows by algebra. ■

Organization of the rest of the section In the next section, we define the states, actions and the feature map. In Section 4.3 we define the transitions, the parameter vector and the rewards for the MDPs $(M_a)_a$. In this section we establish that these MDPs are well-defined and the rewards indeed lie in $[0, 1]$. This is followed by Section 4.4, where we show the realizability of the optimal action-value function by the features and the parameter vectors given earlier. In Section 4.5 we show that Eq. (5) is satisfied, along with some additional properties that we assumed along the way for the various constants involved in the construction. In Section 4.6 we define M_0 and in Section 4.7 we show that Eq. (6) is satisfied by M_0 and $(M_a)_a$. Finally, the formal proof of Theorem 9 based on Lemma 11 and the construction of the previous sections is given in Section 4.8.

4.2. States, actions and feature map

Let $d \geq 18$ and fix some $k > 0$ to be chosen later. We let $\mathcal{A} = [k]$. The set of states is \mathcal{S} is the disjoint union of two sets: The ‘‘game-over states’’ $\mathcal{F} = \{f_2, \dots, f_{H+1}\}$ and the ‘‘tree states’’, \mathcal{T} . The tree states, as the name suggests, *can be thought of* as nodes in a k -ary complete rooted tree of depth H , with the edges in the tree labeled by the actions. Any node in this tree can be identified with the

sequence of actions that, in the tree (but not necessarily in the MDPs to be constructed later), leads to the node. However, we only allow action sequences with non-repeated actions. Thus,

$$\mathcal{T} = \bigcup_{h=0}^{H-1} \left\{ (a_1, \dots, a_h) \in \mathcal{A}^h : \text{ for any } 1 \leq i \neq j \leq h, a_i \neq a_j \right\}.$$

Formally, we then have

$$\mathcal{S} = \{f_2, \dots, f_{H+1}\} \cup \mathcal{T} = \mathcal{F} \cup \mathcal{T},$$

where the union is disjoint. The level decomposition of the state space follows the tree structure. Denoting by \perp the empty sequence (the unique element of \mathcal{A}^0), $\mathcal{S}_1 = \{\perp\}$, $\mathcal{S}_2 = \{f_2\} \cup \mathcal{A}$, \dots , $\mathcal{S}_H = \{f_H\} \cup \mathcal{A}^{H-1}$, $\mathcal{S}_{H+1} = \{f_{H+1}\}$.

For convenience, we abbreviate $(a_1, \dots, a_i) \in \mathcal{A}^i$ as $a_{1:i}$. For $s = (a_1, \dots, a_i) \in \mathcal{A}^i$ and $a \in \mathcal{A}$ we let $sa \in \mathcal{A}^{i+1}$ be (a_1, \dots, a_i, a) . By abusing notation we also let $a_{1:0} = \perp$ denote the unique element of \mathcal{A}^0 . For convenience, the sequence notation is further abused by identifying the sequence with the underlying subset of actions. This allows us to write $a \in a_{1:i}$, which means $a \in \{a_1, \dots, a_i\}$. We also allow $a \in f_h$, which is defined to be false. For $s \in \mathcal{S}$, we let $|s|$ denote the level of state s , the unique index such that $s \in \mathcal{S}_{|s|}$. Specifically, for $s \in \mathcal{T}$ if $s \in \mathcal{A}^i$ then $|s| = i + 1$ and for $s \in \mathcal{F}$ with $f = f_h$ then $|s| = h$.

To construct the feature map φ corresponding to our MDP, we employ the next lemma, which is a consequence of the Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984; Dasgupta and Gupta, 2003; Du et al., 2019a):

Lemma 12 *For any $\gamma > 0$ and positive integer d' such that $d' \geq \lceil 8 \log(k)/\gamma^2 \rceil$, the ℓ^2 -unit sphere in $\mathbb{R}^{d'}$ has k distinct vectors $(v_a)_{a \in [k]}$ such that for all $a, b \in [k]$, $a \neq b$, it holds that $|\langle v_a, v_b \rangle| \leq \gamma$.*

Proof See the proof of (Du et al., 2019a, Lemma A.1). ■

To use the lemma we choose the tuning parameter $\eta > 0$ and set

$$\gamma = (d - 1)^{-\frac{1}{2} + \eta}. \quad (8)$$

Then, the conditions of the lemma are satisfied with $d' = d - 1$ and

$$k = \left\lfloor e^{\frac{(d-1)^{2\eta}}{8}} \right\rfloor. \quad (9)$$

While γ in general should be thought of as a small value, it will be useful to put a specific upper bound on it.³ As it turns out, the following constraint will be particularly useful:

$$\gamma \leq 1/4. \quad (10)$$

To satisfy this constraint, we restrict the range of η :

$$0 < \eta \leq \frac{1}{2} - \frac{2}{\log_2(d - 1)}. \quad (\text{A11})$$

3. γ is not to be confused with the discounting factor. When dealing with discounted MDPs (Section 4.9), we denote the discount factor by α .

Note that owing to $d \geq 17$, the range of η is nonempty and one can indeed verify that for η in this range, $\gamma \leq 1/4$ indeed holds. Here, and in what follows, we use the convention of putting the letter ‘A’ in front of an equation number to mark those relations that remain to be shown.

Let $(v_a)_i$ be the set of $d - 1$ -dimensional vectors on the ℓ^2 -sphere whose existence is guaranteed by Lemma 12. Let $(\sigma_{s,a})_{s \in \mathcal{T}, a \in \mathcal{A}, a \notin s}$ be positive constants to be chosen later. Let c_1, \dots, c_H be constants defined as follows:

$$c_h = \frac{1}{2} + \frac{1 + \gamma}{2} \sum_{l=1}^{H-h} \left(\frac{1 - \gamma}{2\gamma} \right)^l. \quad (12)$$

Note that empty sums are defined to be zero, hence $c_H = 1/2$. With this, the feature map $(\varphi_h)_{h \in [H]}$, $\varphi_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ is defined as follows: for $h \in [H]$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$,

$$\varphi_h(s, a) = \begin{cases} \mathbf{0}, & \text{if } s \in \mathcal{F} \text{ or } (s \in \mathcal{T} \text{ and } a \in s); \\ \left(c_h, \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \sigma_{s,a} v_a^\top \right)^\top, & \text{otherwise.} \end{cases} \quad (13)$$

Note that φ is well-defined: the second branch applies only if $s \in \mathcal{T}$. We will see that $\sigma_{s,a}$ satisfies

$$\gamma \leq \sigma_{s,a} \leq 1. \quad (\text{A14})$$

The ideas behind the constants c_h , scaling with $\sigma_{s,a}$, and why we require it to be within this range is explained once the MDP is fully defined, in Remark 13. The high level goal of this construction is to make sure that realizability holds, a^* is always optimal, and the bias term c_h will ensure that at the first stage we have a large action gap, while the identity of a^* remains hidden.

4.3. Transitions, parameter vector and rewards

We now construct a family of MDPs $(M_{a,\varepsilon})_{a \in \mathcal{A}, \varepsilon > 0}$ with state space \mathcal{S} and action set \mathcal{A} . Here, ε is a parameter whose value we will choose later (the role of ε is to allow some rewards in the MDP to be randomized with ‘signal-to-noise ratio’ $\mathcal{O}(\varepsilon)$).

For $a^* \in [k]$ and $\varepsilon > 0$ fixed, the transition and rewards in the MDP $M_{a^*,\varepsilon}$ are as follows: The state transitions are deterministic. Once a game-over state f_h is reached with $2 \leq h \leq H$, the agent can only transition to the next game-over state f_{h+1} regardless of the action taken. State f_{H+1} is an absorbing state: Any action taken here leads to f_{H+1} . Consider now a tree state $s \in \mathcal{T}$ at level $1 \leq h \leq H - 1$. Taking action $a \in \mathcal{A}$ in s leads to f_{h+1} if either $a = a^*$ or $a \in s$. Taking any other action leads to the next tree node, $sa \in \mathcal{A}^{h+1}$. When in a leaf node, that is when $s \in \mathcal{A}^{H-1}$, taking any action leads to f_{H+1} . Formally, letting $g : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ denote the function that gives the next state for any given state-action pair, we have

$$g(s, a) = \begin{cases} f_{H+1}, & \text{if } |s| \geq H; \\ f_{|s|+1}, & \text{if } |s| < H \text{ and } (s \in \mathcal{F} \setminus \{f_H\} \text{ or } a = a^* \text{ or } a \in s); \\ sa, & \text{otherwise.} \end{cases} \quad (15)$$

The reward structure is dictated by the choice of the parameter vector:

$$\theta^* = \varepsilon (1, v_{a^*}^\top)^\top. \quad (16)$$

Let the reward distribution given a state s and action a be $\mathcal{R}_a(\cdot|s)$. Then, using δ_x to denote the Dirac distribution with point mass at x ,

$$\mathcal{R}_a(\cdot|s) = \begin{cases} \delta_{\langle \varphi_{|s|}(s, a), \theta^* \rangle}, & \text{if } s \in \mathcal{T} \text{ and } a = a^*; \\ \text{Ber}(\mu_a(s)), & \text{if } s \in \mathcal{A}^{H-1} \text{ and } a \neq a^*; \\ \delta_0, & \text{otherwise.} \end{cases} \quad (17)$$

In words, the reward is deterministically zero when either the state is a game-over state or the state is a non-leaf tree state but the action is not a^* . When the action is a^* , the reward is deterministic and is equal to $\langle \varphi_{|s|}(s, a), \theta^* \rangle$. Finally, for any other action $a \neq a^*$ in a leaf tree state $s \in \mathcal{A}^{H-1}$, the reward is drawn from a Bernoulli distribution with mean $\mu_a(s)$, which is chosen as follows:

$$\mu_a(s) = \varepsilon \sigma_{s,a} \left(\frac{1-\gamma}{2\gamma} \right) \langle v_a, v_{a^*} \rangle + \varepsilon/2. \quad (18)$$

Note that the value of $\mu_a(s)$ is selected such that for $s \in \mathcal{A}^{H-1}$ and $a \neq a^*$, $\mu_a(s) = \langle \varphi_{|s|}(s, a), \theta^* \rangle$.

To make the rewards well-defined, we need $\mu_a(s) \in [0, 1]$. For this, let us show that

$$0 \leq \mu_a(s) \leq \varepsilon. \quad (19)$$

Let $x = \varepsilon \sigma_{s,a} \left(\frac{1-\gamma}{2\gamma} \right) \langle v_a, v_{a^*} \rangle$. Using that $|\sigma_{s,a}| \leq 1$ (by Eq. (A14)), and the upper bound in $\langle v_a, v_{a^*} \rangle$ for $a \neq a^*$ from Lemma 12, we get $|x| \leq \varepsilon \gamma \left(\frac{1}{2\gamma} - \frac{1}{2} \right) \leq \varepsilon/2$. This proves Eq. (19) under the bound on $\sigma_{s,a}$. Hence, the rewards are well-defined as long as

$$\varepsilon \leq 1 \quad (\text{A20})$$

holds.

Figure 2 illustrates this MDP. To summarize, the *only* times that rewards are given are either when a^* is played, or when any action is played in a non-game-over state at the *final* stage.

Note that by construction, there can only be one non-zero reward received in any episode, whether this reward is given as a result of playing a^* (case 1), or because we are in the final stage (case 2) (to satisfy realizability according to (Assumption 1), the expectation of it will be the inner product between the features and weight vector $(\langle \varphi_h(s, a), \theta^* \rangle)$ to be constructed momentarily. In case 1, this reward is deterministically a constant, whereas in case 2, it will be the sum of a constant and a Bernoulli with very low expectation. At a high level, the feature map does not depend on a^* , and unless a^* is played at some stage, none of the deterministic transitions depend on a^* , so the agent can only learn about a^* through the low-expectation Bernoulli rewards.

4.3.1. DEFINING $\sigma_{s,a}$ AND SATISFYING EQ. (A14)

We now define $(\sigma_{s,a})_{s \in \mathcal{T}, a \notin s}$. For $a \in \mathcal{A}$, let $\sigma_{\perp,a} = 1$, and for $h > 1$, $sa \in \mathcal{A}^{h-1} \cap \mathcal{T}$ with $a, a' \in \mathcal{A}$ and $a' \notin sa$, let

$$\sigma_{sa,a'} = \sigma_{s,a} \frac{1-\gamma}{2\gamma} \langle v_a, v_{a'} \rangle + \frac{1+\gamma}{2}. \quad (21)$$

To prove that Eq. (A14) holds for all $s \in \mathcal{T}$ and $a \in \mathcal{A}$ with $a \notin s$, we use induction on $h = |s|$. For $h = 1$, $\sigma_{s,a} = 1$ by definition. Let $h > 1$ now and assume that Eq. (A14) holds

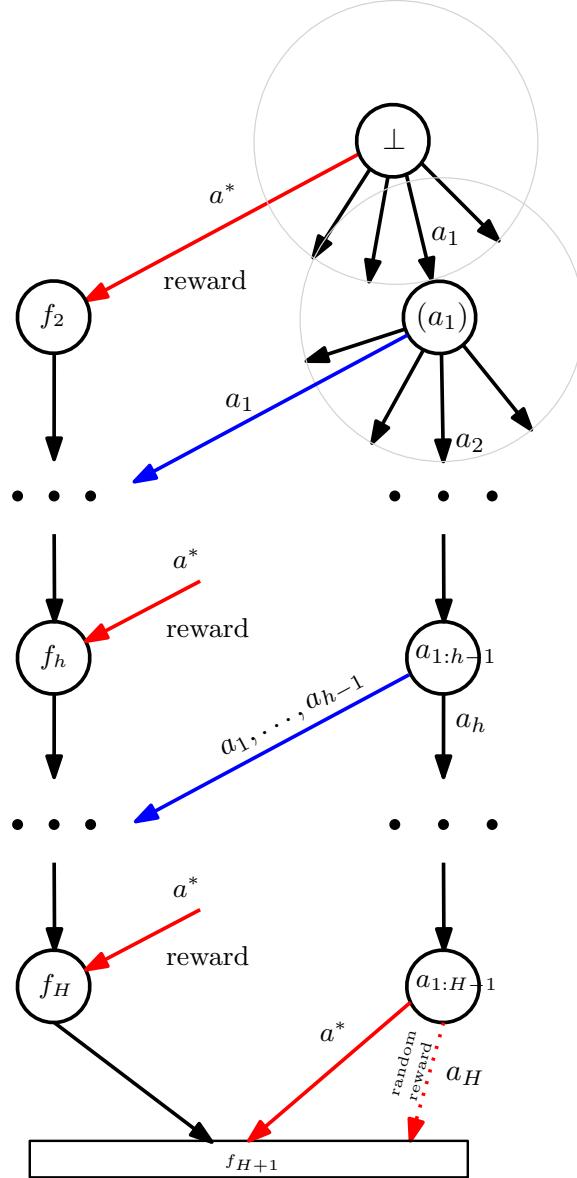


Figure 2: Illustration of the MDP M_{a^*} . The states on the right belong to the “tree”: These are uniquely determined by sequences of actions. The dynamics is deterministic. The initial state $s_1 = \perp$ and this is the state that is given as the input s_{in} to the planner. Action a^* , or actions that are repeated cause the next state to be a “game-over state”. In these states (labeled by f_2, \dots, f_{H+1} , no rewards are obtained and the next state in f_i is f_{i+1} regardless of the action (except for f_{H+1} , where the next state is f_{H+1}). The difference between taking a repeated and the action a^* is that a^* gives a positive (large) reward, while repeated actions do not give rewards. The rewards in the bottom ensure that realizability holds but they are random and make the detection of a^* hard. For more details, see the main text.

for any $s \in \mathcal{A}^{h-1} \cap \mathcal{T}$. Take any $s' \in \mathcal{A}^h \cap \mathcal{T}$ and write it as $s' = sa$ with $s \in \mathcal{A}^{h-1}$ and $a \in \mathcal{A}$. By assumption, $a \notin s$. Hence, by the induction hypothesis, $0 \leq \sigma_{s,a} \leq 1$. We have $\sigma_{sa,a'} = \sigma_{s,a} \langle v_a, v_{a'} \rangle \frac{1-\gamma}{2\gamma} + \frac{1+\gamma}{2}$. Since $a \neq a'$, $-\gamma \leq \langle v_a, v_{a'} \rangle \leq \gamma$. Thus, $-\gamma \leq \sigma_{s,a} \langle v_a, v_{a'} \rangle \leq \gamma$. Plugging this into the expression of $\sigma_{sa,a'}$ gives $\gamma \leq \sigma_{sa,a'} \leq 1$, proving the induction hypothesis for s' and a' and hence the desired claim follows by induction.

Remark 13 *The reason for choosing the value of $\sigma_{sa,a'}$ is to satisfy the Bellman equations as follows: if a suboptimal action $a \neq a^*$ is played at state s of stage h , $\varphi_{h+1}(sa, a')$ for any next-state action a' is scaled such that we pretend a' is the optimal action, and make $\langle \varphi_{h+1}(sa, a'), \theta^{a'} \rangle = \langle \varphi_h(s, a), \theta^{a'} \rangle$ (where $\theta^{a'}$ is the value that θ^* would take if a^* was a'). Crucially, the scaling does not depend on the optimal action, so reveals no information about it to the learner: any action can be the optimal one, and each $\sigma_{sa,a'}$ is calculated as if a' was the optimal one.*

We have to ensure $\sigma_{sa,a'}$ is in some range (Eq. (A14)), or else such scaling could change what the optimal action is at some stage, leading to $\langle \varphi_{h+1}(sa, a'), \theta^* \rangle > \langle \varphi_h(s, a), \theta^* \rangle$. Keeping $\sigma_{s,a}$ in the required range is indeed (indirectly) the role of c_h . In particular, if $c_h = 0$, the above scaling breaks the Bellman equations. Furthermore, to see why $c_h = 0$ fails for any scaling $\sigma_{s,a}$ that satisfies the Bellman equation, let us suppose our planner chooses an action a from the starting state such that $\langle v_a, v_{a^*} \rangle < 0$ (in our construction this holds for a significant fraction of actions). This leads to features $\varphi_{h+1}(sa, a')$ for any a' that also have a negative inner product with θ^* (due to the Bellman equation). This reveals much information about v_{a^*} , and therefore also about the identity of a^* , while our proof strongly relies on extremely little information being revealed about a^* .

4.3.2. BOUNDING THE SUPPORT OF RANDOM REWARDS AND SATISFYING EQ. (A20)

We show that Assumption 3, that is that the support of rewards lies in $[0, 1]$, is satisfied for $M_{a^*, \varepsilon}$.

In the definition of rewards, there are three cases: The reward is either identically zero or a Bernoulli (in which cases there is nothing to be proven), or it is a deterministic value. In the latter case, the optimal action is taken ($a = a^*$) in a tree state ($s \in \mathcal{T}$). In this case the rewards take on the value $r_{a^*}(s) = \langle \varphi_{|s|}(s, a^*), \theta^* \rangle$ deterministically. Plugging in the definitions,

$$\langle \varphi_{|s|}(s, a^*), \theta^* \rangle = \varepsilon c_{|s|} + \sigma_{s,a^*} \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-|s|+1}. \quad (22)$$

Notice that $\frac{1-\gamma}{2\gamma} \geq \frac{3}{2}$ (cf. Eq. (10)). Therefore, both terms are positive (hence $r_{a^*}(s) \geq 0$), and decreasing as $|s|$, the level of the state, increases. It remains to show the upper bound on these rewards. By monotonicity, the largest value in Eq. (22) will be less than one provided

$$\varepsilon \left(\frac{1}{2} + \frac{1+\gamma}{2} \sum_{l=1}^{H-1} \left(\frac{1-\gamma}{2\gamma} \right)^l + \left(\frac{1-\gamma}{2\gamma} \right)^H \right) \leq 1. \quad (23)$$

holds. As $\frac{1}{2} + \frac{1+\gamma}{2} \sum_{l=1}^{H-1} \left(\frac{1-\gamma}{2\gamma} \right)^l + \left(\frac{1-\gamma}{2\gamma} \right)^H < \sum_{l=0}^H \left(\frac{1-\gamma}{2\gamma} \right)^l$, this holds if

$$\varepsilon \sum_{l=0}^H \left(\frac{1-\gamma}{2\gamma} \right)^l \leq 1. \quad (\text{A24})$$

Finally note that Eq. (A24) implies that $\varepsilon \leq 1$ (i.e., Eq. (A20)).

4.4. Showing realizability

The goal of this section is to show that $(M_{a^*, \varepsilon}, \varphi)$ is q^* -realizable according to Definition 2 and the parameter θ^* in the realizability definition can be chosen as shown in Eq. (16). Together with Section 4.3.2, this implies that $(M_{a^*, \varepsilon}, \varphi) \in \mathcal{M}_{H,d}$.

Let the optimal action-value function of $M_{a^*, \varepsilon}$ be q^* . We show realizability (cf. Eq. (3)) by showing $q_h(s, a) := \langle \varphi_h(s, a), \theta^* \rangle$ satisfies the Bellman equations (Eqs. (1) and (2)). By uniqueness (Section 2), this implies $q_h = q^*$.

Take $s \in \mathcal{S}$, $a \in \mathcal{A}$, and let $h = |s|$. We prove the statement by a case-analysis. One of the following cases hold:

Case 1 s is a game-over state ($s \in \mathcal{F}$) or $a \in s$;

Case 2 s is not game-over ($s \in \mathcal{T}$) and $a \notin s$.

For Case 1, the Bellman equations trivially hold as $q_h(s, a) = 0$ and no reward is given (neither immediately, nor for the rest of the episode). We subdivide Case 2 into three sub-cases:

Case 2.1 $a = a^*$;

Case 2.2 $h = H$ and $a \neq a^*$;

Case 2.3 $h < H$ and $a \neq a^*$.

For Cases 2.1 and 2.2, there is an immediate reward given, the expectation of which matches $q_h(s, a)$ by definition. In both cases the next state, $g(s, a)$, is a game-over state. Then, thanks to Case 1, $v_{h+1}(g(s, a)) = 0$, so the Bellman equation is satisfied in these cases.

What is left to show is that the Bellman equation is satisfied for Case 2.3. In Case 2.3, $s \in \mathcal{T}$, $a \notin s$, $h < H$, and $a \neq a^*$. We show that the Bellman equation holds in two parts: first, we will show that a^* is the optimal next-state action:

$$a^* \in \arg \max_{a' \in \mathcal{A}} q_{h+1}(sa, a'). \quad (25)$$

Second, q_h at (s, a) satisfies the Bellman equation:

$$q_h(s, a) = \underbrace{r_a(s)}_{=0} + q_{h+1}(sa, a^*), \quad (26)$$

where we used Eq. (25).

To show Eq. (26), it suffices to show that

$$\langle \theta^*, \varphi_h(s, a) \rangle - \langle \theta^*, \varphi_{h+1}(sa, a^*) \rangle = 0.$$

Plugging in definitions (noting that under Case 2.3 $\varphi_h(s, a) \neq \mathbf{0}$ and $\varphi_{h+1}(sa, a^*) \neq \mathbf{0}$), we get that

$$\begin{aligned} & \langle \theta^*, \varphi_h(s, a) \rangle - \langle \theta^*, \varphi_{h+1}(sa, a^*) \rangle \\ &= \left(\varepsilon c_h + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \sigma_{s,a} \langle v_{a^*}, v_a \rangle \right) - \left(\varepsilon c_{h+1} + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} \sigma_{sa,a^*} \langle v_{a^*}, v_{a^*} \rangle \right) \\ &= \frac{1+\gamma}{2} \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \sigma_{s,a} \langle v_{a^*}, v_a \rangle - \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} \sigma_{sa,a^*} \\ &= \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} \left\{ \frac{1+\gamma}{2} + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right) \sigma_{s,a} \langle v_{a^*}, v_a \rangle - \sigma_{sa,a^*} \right\}. \end{aligned}$$

Plugging in the definition of σ_{sa,a^*} gives the desired result.

To prove Eq. (25), pick any $a' \in \mathcal{A}$ such that $a' \neq a^*$. Our aim is to show that $q_{h+1}(sa, a^*) \geq q_{h+1}(sa, a')$. If $a' \in sa$, then $q_{h+1}(sa, a') = 0$, and we are done because we have already shown that all the rewards are nonnegative (in fact, we have even shown that Assumption 3 holds) and thus $q_{h+1}(sa, a^*) \geq 0$. In the remaining case, and since sa is not a game-over state under Case 2.3, $\varphi_{h+1}(sa, a') \neq \mathbf{0}$ and $\varphi_{h+1}(sa, a^*) \neq \mathbf{0}$, and we can substitute their values as:

$$\begin{aligned} q_{h+1}(sa, a^*) - q_{h+1}(sa, a') &= \left(\varepsilon c_{h+1} + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} \sigma_{sa, a^*} \langle v_{a^*}, v_{a^*} \rangle \right) - \left(\varepsilon c_{h+1} + \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} \sigma_{sa, a'} \langle v_{a^*}, v_{a'} \rangle \right) \\ &= \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^{H-h} (\sigma_{sa, a^*} - \sigma_{sa, a'} \langle v_{a^*}, v_{a'} \rangle) . \end{aligned}$$

The right-hand side here is nonnegative as $|\langle v_{a^*}, v_{a'} \rangle| \leq \gamma$ (due to $a^* \neq a'$ and Lemma 12), $\sigma_{sa, a'} \leq 1$, and $\sigma_{sa, a^*} \geq \gamma$ (by Eq. (A14)). This finishes the proof that the Bellman equations hold in Case 2.3 and thus they hold for all state-action pairs. As such, $q_h = q_h^*$, which shows that q_h^* is indeed realizable.

4.5. Action gap lower bounds (Eq. (5)) and satisfying Eq. (A24)

Let us now turn to showing Eq. (5) with $s_1 = \perp$, that is, that for $a \neq a^*$, the action gaps

$$\Delta_1^*(\perp, a) = \langle \varphi_1(\perp, a^*) - \varphi_1(\perp, a), \theta^* \rangle$$

are bounded from below by $1/4$.

Plugging in the definitions of φ_1 and θ^* , we get

$$\begin{aligned} \langle \varphi_1(\perp, a^*) - \varphi_1(\perp, a), \theta^* \rangle &= \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^H (1 - \langle v_{a^*}, v_a \rangle) \geq \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^H (1 - \gamma) \\ &\geq \frac{3}{4} \varepsilon \left(\frac{1-\gamma}{2\gamma} \right)^H , \end{aligned}$$

where the first inequality follows from the choice of $(v_a)_a$ and the second follows because $\gamma \leq 1/4$ (cf. Eq. (10)). To get $\Delta_1^*(\perp, a) \geq \frac{1}{4}$, we set

$$\varepsilon = \frac{1}{3} \left(\frac{1-\gamma}{2\gamma} \right)^{-H} . \quad (27)$$

Let us now show that with this choice of ε , Eq. (A24) is also satisfied. For this let $x = \frac{1-\gamma}{2\gamma}$. As before, $x \geq 1.5$ due to $\gamma \leq \frac{1}{4}$. Hence,

$$\varepsilon \sum_{l=0}^H \left(\frac{1-\gamma}{2\gamma} \right)^l = \frac{1}{3} x^{-H} \sum_{l=0}^H x^l = \frac{1}{3} \sum_{l=0}^H x^{-l} < \frac{1}{3} \sum_{l=0}^{\infty} x^{-l} = \frac{1}{3} \cdot \frac{1}{1-x^{-1}} \leq \frac{1}{3} \cdot \frac{1}{1-\frac{2}{3}} = 1.$$

4.6. Constructing $M_{0,\varepsilon}$

Let $M_{0,\varepsilon}$ be an MDP with the same state and action spaces as $M_{a^*,\varepsilon}$. The transitions will be as with $M_{a^*,\varepsilon}$ too, except there is no special action a^* that leads to a game-over state. Formally, letting $g_0 : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ denote the function that gives the next state for any given state-action pair, we have

$$g_0(s, a) = \begin{cases} f_{H+1}, & \text{if } |s| \geq H; \\ f_{|s|+1}, & \text{if } |s| < H \text{ and } (s \in \mathcal{F} \setminus \{f_H\} \text{ or } a \in s); \\ sa, & \text{otherwise.} \end{cases}$$

Let all reward distributions for all states and actions be $\mathcal{R}_a(\cdot|s) = \delta_0$ (ie. rewards are always zero). Note that this differs from the rewards of $M_{a^*,\varepsilon}$ only when action a^* is played or when a Bernoulli reward is given by $M_{a^*,\varepsilon}$ at stage H .

4.7. Showing Eq. (6)

Fix an arbitrary planner \mathcal{P} and $a \in \mathcal{A}$ and let the distributions induced by the interconnection of \mathcal{P} and $M_{a,\varepsilon}$ (\mathcal{P} and $M_{0,\varepsilon}$) be denoted by \mathbb{P}_a (\mathbb{P}_0 , respectively). We claim that for any $n \geq 1$,

$$\mathbb{P}_a(a \notin A_{1:n}) \geq (1 - \varepsilon)^n \mathbb{P}_0(a \notin A_{1:n}) \quad (28)$$

holds. Fix $n \geq 1$. Notice that the support of the reward distributions in either $M_{a,\varepsilon}$ or $M_{0,\varepsilon}$ is finite. Let \mathcal{W} be this set. By the law of total probability,

$$\begin{aligned} \mathbb{P}_a(a \notin A_{1:n}) &= \sum_{a_{1:n} \in (\mathcal{A} \setminus \{a\})^n} \sum_{\substack{r_{1:n} \in \mathcal{W}^n \\ s_{1:n}, s'_{1:n} \in \mathcal{S}^n \\ c_{1:n} \in \{0,1\}^n}} \mathbb{P}_a(A_{1:n} = a_{1:n}, R_{1:n} = r_{1:n}, S_{1:n} = s_{1:n}, S'_{1:n} = s'_{1:n}, C_{1:n} = c_{1:n}) \end{aligned}$$

and the same holds for \mathbb{P}_0 . Hence, it suffices to show that for any fixed $a_{1:n} \in (\mathcal{A} \setminus \{a\})^n$, $r_{1:n} \in \mathcal{W}^n$, $s_{1:n}, s'_{1:n} \in \mathcal{S}^n$ and $c_{1:n} \in \{0,1\}^n$ such that $\mathbb{P}_0(A_{1:n} = a_{1:n}, R_{1:n} = r_{1:n}, S_{1:n} = s_{1:n}, S'_{1:n} = s'_{1:n}, C_{1:n} = c_{1:n}) \neq 0$,

$$\kappa := \frac{\mathbb{P}_a(A_{1:n} = a_{1:n}, R_{1:n} = r_{1:n}, S_{1:n} = s_{1:n}, S'_{1:n} = s'_{1:n}, C_{1:n} = c_{1:n})}{\mathbb{P}_0(A_{1:n} = a_{1:n}, R_{1:n} = r_{1:n}, S_{1:n} = s_{1:n}, S'_{1:n} = s'_{1:n}, C_{1:n} = c_{1:n})} \geq (1 - \varepsilon)^n$$

holds. By construction (see Section 3), both the numerator and the denominator factorizes into the product of n terms. In fact, in both the numerator and the denominator, the terms coming from the planner \mathcal{P} are identical and hence cancel. Let g be the transition function in $M_{a,\varepsilon}$. By our assumption that the probability in the denominator is nonzero and because $a \notin a_{1:n}$ we have $s'_t = g(s_t, a_t) = g_0(s_t, a_t)$. Hence,

$$\kappa = \prod_{t=1}^n \frac{\mathbb{P}_a(R_t = r_t \mid S_t = s_t, A_t = a_t)}{\mathbb{P}_0(R_t = r_t \mid S_t = s_t, A_t = a_t)}.$$

For $a' \neq a$, the reward distributions in M_a and M_0 only differ when $s \in \mathcal{A}^{H-1}$. Thus, the ratio above is one unless $s_t \in \mathcal{A}^{H-1}$. Now, since $\mathbb{P}_0(R_t = r_t \mid S_t = s_t, A_t = a_t) > 0$, $r_t = 0$. Therefore,

$\mathbb{P}_a(R_t = r_t | S_t = s_t, A_t = a_t) \geq 1 - \varepsilon$ by Eq. (19). Hence, $\kappa \geq (1 - \varepsilon)^n$, as required, finishing the proof of Eq. (28).

It remains to be shown that Eq. (6) holds for a suitable choice of n . In fact, we choose

$$n = \left\lfloor \frac{k}{4} \wedge \left(\frac{1}{\varepsilon} - 1 \right) / 3.5 \right\rfloor. \quad (29)$$

Note that by Eq. (A20), the second term in the minimum above is nonnegative. On the one hand, this choice of n implies that $n \leq \lfloor k/4 \rfloor$, as required by Lemma 11. Furthermore, $n \leq \left(\frac{1}{\varepsilon} - 1 \right) / 3.5$ and hence, $1 - \varepsilon \geq 1 - \frac{1}{1+3.5n}$. Hence,

$$(1 - \varepsilon)^n \geq \left(1 - \frac{1}{1 + 3.5n} \right)^n \geq \lim_{n \rightarrow \infty} \left(1 - \frac{1}{1 + 3.5n} \right)^n > 3/4.$$

4.8. Proving Theorem 9

Let us now collect the choice of the parameters. We have for $d \geq 18$, $0 < \eta \leq \frac{1}{2} - \frac{2}{\log_2(d-1)}$ (cf. Eq. (A11)):

$$\gamma = (d-1)^{-\frac{1}{2}+\eta} \quad (30)$$

$$k = \left\lfloor e^{\frac{(d-1)^{2\eta}}{8}} \right\rfloor, \text{ and} \quad (31)$$

$$\varepsilon = \frac{1}{3} \left(\frac{1-\gamma}{2\gamma} \right)^{-H} \quad (32)$$

(cf. Eqs. (9), (8), and (27)). We have shown that the conditions of Lemma 11 are satisfied when n is set as shown in Eq. (29). Thus, the theorem follows from the conclusion of this lemma and Proposition 6, once we lower bound n . For this we have

$$\begin{aligned} n &= \lfloor k/4 \wedge (\varepsilon^{-1} - 1)/3.5 \rfloor \geq \left(\exp \left((d-1)^{2\eta}/8 \right) / 4 \wedge \left(3 \left(\frac{1-\gamma}{2\gamma} \right)^H - 1 \right) / 3.5 \right) - 2 \\ &= \left(\exp \left((d-1)^{2\eta}/8 \right) / 4 \wedge \frac{3}{3.5} \left(\frac{1}{2}(d-1)^{\frac{1}{2}-\eta} - \frac{1}{2} \right)^H - \frac{1}{3.5} \right) - 2 \\ &= \Omega \left(\min \left\{ \exp \left((d-1)^{2\eta}/8 \right), 2^{-H} d^{H(\frac{1}{2}-\eta)} \right\} \right), \end{aligned}$$

finishing the proof of the Theorem 9. ■

4.9. Lower bound for the discounted MDP case

The construction and proof presented can be applied with minor modifications to the discounted MDP setting, which we briefly discuss here. In this setting, instead of maximizing the expected sum of rewards over a finite horizon H (ie. $\mathbb{E}_\pi [\sum_{t=h}^H r_{A_t}(S_t) | S_h = s]$), the goal of the agent is to maximize the expected total discounted reward for some “discount factor” $0 < \alpha < 1$: $\mathbb{E}_\pi [\sum_{t=h}^\infty \alpha^{t-h} r_{A_t}(S_t) | S_h = s]$. Usually, this discount factor is close to 1, and to make our results translate into this setting, we further assume that

$$\frac{2}{3} \leq \alpha < 1.$$

The Bellman equation Eq. (1) is redefined to reflect this as:

$$q_h^\star(s, a) = r_a(s) + \alpha \int v_{h+1}^\star(s') P_a(ds'|s). \quad (33)$$

Fix some $H \geq 1$ integer. For some action a^\star , let the MDP $M'_{a^\star, \varepsilon}$ have the same states, actions, and transition structure (Eq. (15)) as $M_{a^\star, \varepsilon}$. Choose parameters γ, k, ε as before (Eqs. (30), (31), and (32)). We define the feature map for this MDP φ' for $h \in [H]$, $s \in \mathcal{S}$, $a \in \mathcal{A}$ as:

$$\varphi'_h(s, a) = \alpha^{-h+1} \varphi_h(s, a),$$

where $\varphi_h(s, a)$ is defined as before (Eq. (13)), noting that for $h > H$, the agent is always in the absorbing stage f_{H+1} , so we simply define $\varphi'_h(s, a) = \mathbf{0}$ for any $h > H$. Let us redefine the final-stage rewards of Eq. (18) to accommodate the discounting:

$$\mu_a(s) = \alpha^{-H+1} \left(\varepsilon \sigma_{s, a} \left(\frac{1-\gamma}{2\gamma} \right) \langle v_a, v_{a^\star} \rangle + \varepsilon/2 \right).$$

Note that as before, we have $\mu_a(s) = \langle \varphi'_{|s|}(s, a), \theta^\star \rangle$ for $s \in \mathcal{A}^{H-1}$ and $a \neq a^\star$. Let us otherwise keep the reward distribution the same, according to Eq. (17), using the new definition of $\mu_a(s)$. It is easy to see that realizability in $M'_{a^\star, \varepsilon}$ still holds, as we negated the effect of discounting in the Bellman equation with our new definitions.

Let us turn to showing that the rewards are still in $[0, 1]$ and that $\mu_s(a) \in [0, 1]$. Notice that by the same argument as before, all rewards and $\mu_s(a)$ are non-negative, so we only need to show boundedness from above by 1. Furthermore, as there can only be up to one non-zero reward in any episode, it suffices to show an upper bound on $q_h^\star(s, a)$ for any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $h = |s|$:

$$\langle \varphi'_h(s, a), \theta^\star \rangle = \varepsilon \alpha^{-h+1} \left(c_h + \sigma_{s, a} \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \langle v_a, v_{a^\star} \rangle \right) \leq 1.$$

In two parts, first we show that

$$\begin{aligned} \varepsilon \alpha^{h-1} c_h &= \varepsilon \alpha^{-h+1} \left(\frac{1}{2} + \frac{1+\gamma}{2} \sum_{l=1}^{H-h} \left(\frac{1-\gamma}{2\gamma} \right)^l \right) < \varepsilon \alpha^{-h+1} \frac{1+\gamma}{2} \left(\sum_{l=0}^{H-h} \left(\frac{1-\gamma}{2\gamma} \right)^l \right) \\ &\leq \varepsilon \alpha^{-h+1} \frac{1+\gamma}{2} 2 \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} = \varepsilon \left(\alpha \left(\frac{1-\gamma}{2\gamma} \right) \right)^{-h+1} (1+\gamma) \left(\frac{1-\gamma}{2\gamma} \right)^H \\ &= \frac{1}{3} \left(\alpha \left(\frac{1-\gamma}{2\gamma} \right) \right)^{-h+1} (1+\gamma) \leq \frac{5}{12} (1.5\alpha)^{-h+1} < \frac{1}{2}, \end{aligned}$$

where we repeatedly used that $\left(\frac{1-\gamma}{2\gamma} \right) \geq 1.5$ (due to $\gamma \leq \frac{1}{4}$) and $\alpha \geq \frac{2}{3}$, and substituted the value of ε . Second, we show for the second term of the sum that

$$\begin{aligned} \varepsilon \alpha^{-h+1} \sigma_{s, a} \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \langle v_a, v_{a^\star} \rangle &\leq \varepsilon \alpha^{-h+1} \left(\frac{1-\gamma}{2\gamma} \right)^{H-h+1} \\ &= \varepsilon \left(\alpha \left(\frac{1-\gamma}{2\gamma} \right) \right)^{-h+1} \left(\frac{1-\gamma}{2\gamma} \right)^H \leq \frac{1}{3} (1.5\alpha)^{-h+1} < \frac{1}{2}, \end{aligned}$$

where we used the bound on $\sigma_{s,a}$ (Eq. (A14)).

The discounting also affects Eqs. (19) and (28), which respectively become:

$$0 \leq \mu_a(s) \leq \alpha^{-H+1} \varepsilon$$

$$\mathbb{P}_a(a \notin A_{1:n}) \geq (1 - \alpha^{-H+1} \varepsilon)^n \mathbb{P}_0(a \notin A_{1:n}).$$

Carrying this change forward in the calculations, by a choice of $\eta = \frac{1}{2} - \frac{2}{\log_2(d-1)}$, analogously to Corollary 10, we arrive at

$$C_\alpha(\mathcal{P}, d) = \min(e^{\Omega(d)}, \Omega((2\alpha)^H)),$$

where $C_\alpha(\mathcal{P}, d)$ is defined for the α -discounted setting analogously to Definition 7 (with the difference of taking the supremum over any q^* -realizable (M, φ) with discount factor α instead of horizon H). This holds for any $H \geq 1$, so due to our assumption of $\alpha \geq \frac{2}{3}$, we can take the limit as $H \rightarrow \infty$, to get that

$$C_\alpha(\mathcal{P}, d) = e^{\Omega(d)}.$$

5. Upper bound

The single-stage setting (i.e., when $H = 1$) under q^* -realizability is effectively a ‘‘linear bandit’’ problem (Lattimore and Szepesvári, 2020). In this case δ -good actions (i.e., actions a with $q^*(s, a) \geq v^*(s) - \delta$) can be found regardless of the cardinality of $\mathcal{S} \times \mathcal{A}$ using $\mathcal{O}(\text{poly}(d)/\delta^2)$ queries and computation time by simply choosing $\tilde{\mathcal{O}}(d)$ state-action pairs that provide maximal information about any other state-action pairs (via a so-called *approximate G-optimal design*, cf. Proposition 14), followed by estimating the unknown linear parameter using a least-squares estimator.

In this section, we consider the ‘‘naïve’’ planning algorithm which treats the finite-horizon planning problem in MDPs as a sequence of H single-stage problems, and applies the above procedure recursively. More specifically, the problem can be seen as a sequence of misspecified linear bandits, with the misspecification of any given level being the estimation error of the previous level. We provide an upper bound on the query complexity of the planner which implements this algorithm – we find that, perhaps unsurprisingly, the estimation errors compound multiplicatively over the different stages: by level $h \in [H]$, the learned hypothesis will roughly have an error of $\mathcal{O}((\sqrt{d})^{H-h+1})$. The complementary lower bound of Section 4 tells us that in a certain sense this is the best that a planner can do (i.e., some multiplicative compounding of errors is unavoidable).

In more detail, our algorithm first estimates the optimal value functions $q_H^*(s, a) = r_a(s)$ using a least-squares estimator and design points from $(s, a) \in \mathcal{S}_H \times \mathcal{A}$. Recursively, once the estimate f_{h+1} is computed, level h can be treated as a single-stage problem with immediate rewards $\mu^{(h)}$ so that $\mu_a^{(h)}(s) = r_a(s) + \int \max_{a'} f_{h+1}(s', a') \mathcal{P}_a(ds' | s)$. This gives a misspecified linear bandit, where the misspecification error is the estimation error of the preceding level, $\|f_{h+1} - q_{h+1}^*\|_\infty$. Readers familiar with the literature recognize that the algorithm described is known as the least-squares value iteration algorithm. The pseudo-code makes reference to Proposition 14, which we present after the algorithm. Below, we write $\Pi_H(x) := \max(\min(x, H), -H)$ for the clipping operator.

We say that $\mu : \mathcal{X} \rightarrow \mathbb{R}$ is an ε -realizable linear function for feature map $\varphi' : \mathcal{X} \rightarrow \mathbb{R}^d$ if there exists a vector $\theta^* \in \mathbb{R}^d$ such that

$$\mu(x) = \langle \varphi(x), \theta^* \rangle + \varepsilon_x, \quad |\varepsilon_x| \leq \varepsilon \quad \forall x \in \mathcal{X}.$$

Algorithm 1 Least-squares value iteration with G -optimal design

- Let $f_{H+1} \equiv 0$
- for $h = H$ down to 1:
 - (i) Compute the experimental design $\rho_h \in \mathcal{M}_1(S_h \times \mathcal{A})$ satisfying the assumptions of Proposition 14
 - (ii) For each $z = (s, a) \in \text{supp}(\rho_h)$, collect n samples of $(R_i(z), S'_i(z)) \sim Q_a(\cdot, \cdot | s)$ and calculate the empirical average $\hat{\mu}_h(s, a) = \frac{1}{n} \sum_{i=1}^n R_i(z) + \max_{a'} f_{h+1}(S'_i(z), a')$.
 - (iii) Calculate the least-squares estimator $\hat{\theta}_h$ appearing in Proposition 14 using $\rho = \rho_h$ and $r = \hat{\mu}_h$, and set $f_h(s, a) = \Pi_H \left(\langle \varphi_h(s, a), \hat{\theta}_h \rangle \right)$.
- Return $(f_h)_h$.

To perform the least-squares estimation, we follow the approach of Lattimore et al. (2020) and make use of the *Kiefer-Wolfowitz* theorem to sample only from a few state-action pairs per horizon. The main result we use is the following, which bounds the error of producing a least-squares estimator over a certain distribution of points with small support:

Proposition 14 *Assume that $\{\varphi'(x) : x \in \mathcal{X}\} \subset \mathbb{R}^d$ is compact. There exists a distribution $\rho \in \mathcal{M}_1(\mathcal{X})$ whose support has at most $4d \log \log d + 16$ points such that for any $r, \mu : \mathcal{X} \rightarrow \mathbb{R}$, the vector*

$$\hat{\theta} = G(\rho)^{-1} \sum_{x \in \text{supp}(\rho)} \rho(x) r(x) \varphi'(x), \quad \text{where } G(\rho) = \sum_{x \in \text{supp}(\rho)} \rho(x) \varphi'(x) \varphi'(x)^\top$$

satisfies

$$\sup_{x \in \mathcal{X}} |\langle \varphi'(x), \hat{\theta} \rangle - \mu(x)| \leq \varepsilon + (\varepsilon + \delta) \sqrt{2d}$$

provided that $\mu : \mathcal{X} \rightarrow \mathbb{R}$ is an ε -realizable linear function with feature map φ' and $\|r - \mu\|_\infty \leq \delta$.

The distribution ρ appearing in the above proposition is called a *near-optimal experimental design*. Note that the result only needs r to be specified at the points in the support of ρ , a property that we use in our algorithm.

Proof This is essentially Proposition 4.5 of Lattimore et al. (2020) with the difference that here we allow infinite \mathcal{X} . As the Kiefer-Wolfowitz theorem applies regardless the cardinalities of the sets involved, the result continues to hold in our case. \blacksquare

The following result gives an upper bound on the estimation error and query complexity of this algorithm.

Proposition 15 *Let $f_h : S_h \times \mathcal{A} \rightarrow \mathbb{R}$, for $h \in [H]$, be the functions returned by Algorithm 1. Consider the planner \mathcal{P} that chooses actions greedily based on $f = (f_h)$. Let $(M, \varphi) \in \mathcal{M}_{H,d}$ be q^* -realizable and bounded (Definition 4). Then setting $n = \tilde{\Theta}(H^4(2d)^H / \delta^2)$, \mathcal{P} is a δ -sound planner that uses at most*

$$C(\mathcal{P}, H, d) = \tilde{O} \left(\frac{H^5(2d)^{H+1}}{\delta^2} \right)$$

queries.

In fact, the proof shows that the planner is not only a local δ -sound planner, but it is also a **global** δ -sound planner. That is, no matter the state, after the said number of queries, the planner can output the actions of a δ -optimal policy without any further interactions with the simulator (and the cost of this computation is $O(d|\mathcal{A}|)$). We also note that the upper bound is exponential in H , as is the second term of the lower bound. Still there remains a gap between the lower and upper bounds.

For the proof, we will need a finite-horizon analogue of Corollary 2 from [Singh and Yee \(1994\)](#). For any $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, let $\pi_f(s) = \arg \max_a f(s, a)$ denote a policy that is greedy with respect to f .

Lemma 16 *Given any $(f_h)_h$ with $f_h : \mathcal{S}_h \times \mathcal{A} \rightarrow \mathbb{R}$, the H -stage policy π that at stage $h \in [H]$ is greedy with respect to f_h ($\pi_h = \pi_{f_h}$) satisfies $\max_h \|v_h^\star - v_h^\pi\|_\infty \leq 2 \sum_{h=1}^H \|q_h^\star - f_h\|_\infty$.*

Proof Let $s \in \mathcal{S}_1$ and let π^\star be an optimal policy. From the Bellman equations for π^\star and π ,

$$\begin{aligned} v_1^\star(s) - v_1^\pi(s) &= q_1^\star(s, \pi^\star) - q_1^\star(s, \pi_{f_1}) + q_1^\star(s, \pi_{f_1}) - q_1^\pi(s, \pi_{f_1}) \\ &= q_1^\star(s, \pi^\star) - q_1^\star(s, \pi_{f_1}) + \langle P_{\pi_{f_1}}(s), v_2^\star - v_2^\pi \rangle \\ &\leq q_1^\star(s, \pi^\star) - f_1(s, \pi^\star) + f_1(s, \pi_{f_1}) - q_1^\star(s, \pi_{f_1}) + \langle P_{\pi_{f_1}}(s), v_2^\star - v_2^\pi \rangle \\ &\leq 2\|q_1^\star - f_1\|_\infty + \|v_2^\star - v_2^\pi\|_\infty, \end{aligned}$$

where the first inequality follows from that π_{f_1} is greedy with respect to f_1 . An inductive argument completes the proof. \blacksquare

We also need Hoeffding's inequality:

Lemma 17 (Hoeffding's inequality) *Let $(X_i)_{i \in [m]}$ be i.i.d. random variables from the unit interval, $[0, 1]$. Letting μ denote their common mean, for any $\zeta \in (0, 1]$ it holds with probability $1 - \zeta$ that*

$$\left| \frac{1}{m} \sum_{i=1}^m X_i - \mu \right| \leq \sqrt{\frac{\log(2/\zeta)}{2m}}.$$

Proof (of Proposition 15). We first derive an error bound on how well the optimal action-value function is approximated by the functions computed. Let $m = \lceil 4d \log \log d + 16 \rceil$ be the maximum cardinality of the support of ρ_h . Recall that $f_{H+1} = 0$. For $h \in [H+1]$, let $g_h(s) = \max_a f_h(s, a)$ and

$$\mu_h(s, a) = r_a(s) + \langle P_a(s), g_{h+1} \rangle.$$

Fix $h \in [H]$, $z = (s, a) \in \text{supp}(\rho_h)$,

$$\hat{\mu}_h(s, a) - \mu_h(s, a) = \frac{1}{n} \sum_{i=1}^n \underbrace{R_i(z) - r_a(s) + g_{h+1}(S'_i(z)) - \langle P_a(s), g_{h+1} \rangle}_{X_i}.$$

Note that $X_i \in [-H, H]$ as rewards are positive (Assumption 3), and the output of f_h is clipped at $[-H, H]$ (cf. Step (iii) of Algorithm 1). Also note that X_i has mean zero. Indeed,

$$\mathbb{E}[X_i] = \mathbb{E}[\mathbb{E}[X_i | g_{h+1}]] = \mathbb{E}[R_i(z) - r_a(s)] + \mathbb{E}[\mathbb{E}[g_{h+1}(S'_i(z)) - \langle P_a(s), g_{h+1} \rangle | g_{h+1}]] = 0.$$

Hence, by a union bound and Hoeffding's inequality, with probability at least $1 - \zeta$, we have

$$|\hat{\mu}_h(s, a) - \mu_h(s, a)| \leq H \sqrt{\frac{2}{n} \log \left(\frac{2Hm}{\zeta} \right)} := \beta \quad \forall h \in [H], \forall (s, a) \in \text{supp}(\rho_h). \quad (34)$$

In the rest of the proof we work on the event where the inequalities in the above display hold (i.e., the inequalities below hold with probability at least $1 - \zeta$). By backwards induction, we will show that the error at level h satisfies

$$\|f_h - q_h^*\|_\infty \leq \beta \left((2 + \sqrt{2d})^{H-h+1} - 1 \right) := \varepsilon_h \quad (35)$$

Starting with stage H , by realizability of (M, φ) , we can apply Proposition 14 with $\delta = \beta$, $\varepsilon = 0$, $\mu = r = q_H^*$, and $\varphi' = \varphi_H$ to get

$$\|f_H - q_H^*\|_\infty \leq \beta \sqrt{2d},$$

as desired. Now let $1 \leq h < H$ and assume the claim is true for level $h+1$; we will now show that it is also true for level h . We claim that μ_h is ε_{h+1} -realizable under φ_h . Indeed, from the definition of μ_h , the realizability of q_h^* and the Bellman equation,

$$\begin{aligned} |\mu_h(s, a) - q_h^*(s, a)| &= \left| \int P_a(ds'|s) \left\{ \max_{a'} f_{h+1}(s', a') - \max_{a'} q_{h+1}^*(s', a') \right\} \right| \\ &\leq \|f_{h+1} - q_{h+1}^*\|_\infty \leq \varepsilon_{h+1}, \end{aligned}$$

where the last inequality follows by the induction hypothesis. Applying Proposition 14 again, with $\varepsilon = \varepsilon_{h+1}$, $\delta = \beta$, $\mu = \mu_h$, and $\varphi' = \varphi_h$ gives:

$$\begin{aligned} \|f_h - q_h^*\|_\infty &\leq \|f_h - \mu_h\|_\infty + \varepsilon_{h+1} \\ &\leq \varepsilon_{h+1} (2 + \sqrt{2d}) + \beta \sqrt{2d} \\ &= \beta \left((2 + \sqrt{2d})^{H-h} - 1 \right) (2 + \sqrt{2d}) + \beta \sqrt{2d} \\ &= \beta \left((2 + \sqrt{2d})^{H-h+1} - 1 \right), \end{aligned}$$

as desired. Letting π be the policy implemented by the planner, we bound its suboptimality using Lemma 16: $\|v_1^* - v_1^\pi\|_\infty \leq 2H\|q_1^* - f_1\|_\infty = 2H\beta \left((2 + \sqrt{2d})^H - 1 \right)$. By boundedness of rewards (Assumption 3), on the error event that has probability ζ at most, the policy computed incurs a total cost of at most H compared to the optimal policy, it holds that π is at least $2H\beta \left((2 + \sqrt{2d})^H - 1 \right) + H\zeta$ -optimal. Setting $\zeta = \delta/2$ and setting n so that $2H\beta \left((2 + \sqrt{2d})^H - 1 \right) \leq \delta/2$ also holds, we get that with

$$n = \left\lceil 32H^4 \log \left(\frac{4Hm}{\delta} \right) \left((2 + \sqrt{2d})^H - 1 \right)^2 \delta^{-2} \right\rceil$$

queries per the mH support points, the planner \mathcal{P} is δ -sound. Plugging in the value of m and simplifying gives the desired result. \blacksquare

As a side note we remark that the proof avoids a union bound over the value functions. This works because the random next states that are used in the random value function computed are chosen independently of the random value function.

Note that not accounting for the computation of the distributions $(\rho_h)_h$, the total computation cost of a naive implementation of the planner for a single stage $h \in [H]$ is $\mathcal{O}(nm|\mathcal{A}|d + md + d^3) = \tilde{\mathcal{O}}(H^4d^2|\mathcal{A}| + d^3)$, making the total compute cost $\tilde{\mathcal{O}}(H^5d^2|\mathcal{A}| + Hd^3)$. When a quadratic optimization oracle is available over the set $\{\varphi_h(s, a) : (s, a) \in \mathcal{S}_h \times \mathcal{A}\}$, a version of the Frank-Wolfe algorithm can be used to approximately compute ρ_h in $\text{poly}(d)$ -time (Lattimore and Szepesvári, 2020, Notes 3-4, Chapter 21.2).

6. Discussion

In this paper we have shown an exponential lower bound on the query complexity of local planning with linear function approximation in both fixed-horizon and discounted MDPs, under the assumption that the optimal action-value function is realizable by the features. We have also given an upper bound for the fixed-horizon setting, which, in some regimes of the parameters is relatively close to the lower bound. Closing the gap between the lower and upper bounds remains an interesting open problem. Since the upper bound applies to global planning and not only local planning, there is a possibility that the query complexity of local and global planning in this specific setting are the same. It would also be interesting to refute or validate this conjecture.

Our lower bound construction critically relied on allowing the action set to be exponentially large in the dimension. We remark that the large action set does not preclude a polynomial sample complexity – in particular, Lattimore et al. (2020) give a polynomial upper bound regardless of the number of actions, under the assumption that the state-action value function q^π for any memoryless policy π is realizable. Note that in the single-stage/linear bandit setting (with $H = 1$), this assumption becomes equivalent to the realizability of q^* . However, for $H > 1$, our exponential sample complexity lower bound shows a strong separation between these two realizability assumptions. The difficulty of planning (and learning) thus cannot be attributed to the large action set alone. Yet, it is intriguing that our construction relies heavily on the action set being large: it remains an interesting open question to resolve the worst-case query complexity of q^* -realizable planning with linear function approximation when the action set is of constant size.

7. Acknowledgements

We thank Nan Jiang for insightful discussions, and Roshan Shariff for helpful feedback on an earlier draft. 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

R. Bellman, R. Kalaba, and B. Kotkin. Polynomial approximation – a new computational technique in dynamic programming: Allocation processes. *Mathematics of Computation*, 17(8):155–161, 1963.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

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.

C. Ionescu Tulcea. Mesures dans les espaces produits. *Atti Accademia Nazionale Lincei Rend*, 7: 208–211, 1949.

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.

William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

O. Kallenberg. *Foundations of modern probability*. Springer-Verlag, 2002.

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.

T. Lattimore and Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

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.

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

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.

S. P. Singh and R. C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

Ruosong Wang, Dean P. Foster, and Sham M. Kakade. What are the statistical limits of offline rl with linear function approximation?, 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 Yang and Mengdi Wang. Sample-optimal parametric q -learning using linearly additive features. In *ICML*, pages 6995–7004, 2019.