The Power and Limitations of Function Approximation for Efficient Reinforcement Learning

Philip Amortila

April 2023

Abstract

Modern Reinforcement Learning (RL) applications require the introduction of *function approximation* to generalize knowledge across large, complicated environments. From a theoretical perspective, there has been an interest in leveraging function approximation to develop RL algorithms with statistical complexities independent of the size of the state space (which is enormous in most applications of interest). All of these works require certain assumptions about the environments encountered or the quality of the approximation scheme. In this thesis, I will be studying when it is and isn't possible to achieve this goal under different function approximation settings and under various "minimal" assumptions. I will also argue that the necessary assumptions are too stringent, and will be interested in re-examining canonical notions of optimality and efficiency to study what remains possible in more realistic settings.

Contents

Part I
Setting the stage

1 Introduction

This thesis investigates theoretical aspects of Reinforcement Learning (RL), the branch of AI concerned with sequential decision-making in a unknown and changing environment. Recent advances in RL have demonstrated impressive empirical successes in complicated environments such as the game of Go [1], autonomous navigation [2], and protein folding problems [3]. Due to its generality, RL has the potential to be applied to even more critical applications, notably in the fields of healthcare [4] and climate change [5]. The key to all of these advances is the merging of RL algorithms with high-powered function approximators (in particular, deep neural networks). Despite the success of these advancements, the theoretical understanding of such function approximation schemes and their uses in RL is still relatively nascent.

1.1 Why is function approximation necessary?

The typical framework for RL is the Markov Decision Process (MDP) [6]. A *policy* in an MDP is a mapping from histories of states to distributions over actions. The goal of an RL agent is to maximize its expected sum of lifetime rewards by finding the *optimal* policy. In the RL setting, we assume that the MDP is unknown. Thus, the agent must find a good policy while simultaneously learning about (or "exploring") the environment. There is now a mature theory of provably efficient RL methods *without function approximation* [7].[1] These approaches for learning in MDPs have statistical and computational complexities which are polynomial in the size of the state and action spaces, and one can show that this is unavoidable in general. Intuitively, without further structure, each state-action pair must be checked at least once [8]. The guarantee of efficiency is crucial for certain applications, but the dependence on the state space size is clearly too large in most applications of interest; for example, the game of Go has more states than the number of atoms in the universe [9].

[1] This setting is typically called the *tabular* setting.

This establishes the need for approximation. The predominant approach is to introduce a *function class* $\mathcal{F}$ which will be used to model objects of interest. These may be value functions, policies, or MDPs themselves (these objects will be defined more formally in Section 2). With the help of the function class, one can avoid having to learn a separate value at each state-action pair, and instead rely on the learned function to "generalize" knowledge to unseen states. For this thesis, we will assume that the learner is simply *given* this function class, and the separate interesting question of designing or learning this function class is left on the table.

1.2 Central question, and objectives of this thesis

This thesis studies questions which will help us understand the following broad question:

Question 1.1 (Informal)**.** *When (and how) can function approximation enable algorithms that recover "good" solutions to the MDP in a provably-"efficient" manner?.*

We note, in particular, two terms which have not been well-defined in this question. Firstly, what is a "good" solution? And secondly, which complexities count as "efficient"? There are canonical (and commonly-accepted) answers for these questions:

"Definition" 1.2 (Canonical objectives)**.** *Respectively:*

- *"goodness": the algorithm should recover a near-optimal policy, upto some given suboptimality ε*
- *"efficiency": the algorithm should have (worst-case) complexities that are polynomial only in 1) the "complexity" of the function class, 2) the horizon of the MDP, and 3) $1/\varepsilon$.*

In particular, the complexities of the algorithm should be independent of the size of the state-action space (although we may occasionally accept dependence on the size of the action space). In the sequel we will refer to these definitions of goodness and efficiency as "canonical".

We need to assume *some* "connection" between the MDP and the function class, otherwise recovering the optimal policy will be just as difficult as when one is not given a function class.[2] Unfortunately, we will see that obtaining these guarantees will require restrictive assumptions in general. In this thesis, we are interested in both

1. the "minimal" amount of assumptions required to obtain the above goal, and
2. re-examinining these definitions.

The objective of the first item is the usual (important) one: we want our algorithms to come with guarantees that apply in as many cases as possible. Assumptions that are hard to verify, unlikely to hold, or difficult to enforce can limit the applicability of RL for important real-life applications where provable guarantees are needed. Of course, there may be several non-overlapping answers for which assumptions can be called minimal. For

[2]For example we can imagine the case where the function class only contains the 0 function.

example, assuming determinism of the MDP is orthogonal from assuming representational properties from the function class, which is orthogonal from assuming stronger interaction protocols (e.g. the presence of a "simulator" or the presence of some expert advice).

As for the second item: we will see several times that achieving the canonical goal will be impossible without some restrictive and arguably unrealistic assumptions. We argue in this thesis that it is also useful to flip the question. Namely, we should think about which assumptions we are willing to tolerate, and then to study which *relaxed* guarantees are possible by modifying one or both of the above goals. This implies, in particular, that in this more "realistic" setting, we will obtain worse statistical rates or a lower quality solution. Of course, we will also need to argue that these worse guarantees are the best that can be done in this new problem setting.

This thesis will roughly be split in two halves, corresponding to each of the above items. We approach this broad task by studying a variety of concrete learning problems in RL where these goals remain poorly understood. A central theme of this thesis will be to examine both what is and isn't possible under various function approximation settings, which will require a combination of positive results ("this task is solvable by some algorithm with complexity at most x") and negative ones ("no algorithm that solves this task can do so with complexity smaller than y"). The positive results will give rise to new algorithms that provably work in more general (thus more practical) settings. The negative results will tell us what we can hope to expect, and their proofs may explain the failure modes of RL with function approximation while motivating natural additional assumptions.

1.3 Overview of the proposal

We begin in Section 2 by covering some of the necessary background.

The second part of this proposal (Part II) will be focusing on the special case of *linear function approximation*. Despite the apparent simplicity of the linear structure (and its introduction at least since the 60s [10]), it's interplay with the RL problem remained poorly understood. We investigated, in a series of three papers, a central open question which asked whether realizability of the optimal value function alone was sufficient for sample-efficient learning. We showed that the answer is a) no in general (Section 3), but yes if b) the number of actions is "small" (Section 4), or c) a "small" amount of expert advice is available (Section 5).

The last part of this proposal (Part III) will more broadly aim to study relaxations of the canonical objectives (Definition 1.2). Section 6 begins by tackling the misspecified setting and derives optimal estimators for the problem of offline linear value function estimation. Section 7 proposes a generalized notion of regret which could meaningfully be used in settings where optimality is unachievable. Section 8 proposes to leverage recent connections between offline RL and online RL to derive complexity measures that characterize the hardness of learning with general realizable function classes.

2 Preliminaries

Notation We use typical asymptotic notation $\mathcal{O}$ and Ω. We use $\tilde{\mathcal{O}}$ and $\tilde{\Omega}$ when ignoring logarithmic factors. We use $\texttt{poly}(\cdot)$ to denote arbitrary polynomial quantities in $(\cdot)$. For $N \in \mathbb{N}$, we write $[N] = \{1, \ldots, N\}$. We write $\text{Dists}(\mathcal{X})$ for the set of probability measures on a space $\mathcal{X}$ (the σ-algebras will simply be the power set when on discrete spaces or the Borel σ-algebra when on $\mathbb{R}^n$).

Markov Decision Processes A Markov Decision Process (MDP) is a formulation that captures optimal decision-making in a changing and stochastic environment [6]. In this thesis we will only consider MDPs with finite state and action spaces. Most of our results will hold for both the *finite-horizon* setting, where there is a horizon H after which each episode terminates, and the *infinite-horizon* setting, where episodes do not terminate but rewards are discounted geometrically by a discount factor γ. For simplicity we will firstly give the exposition for the discounted setting, and then discuss the differences with the finite-horizon setting.

Definition 2.1 (Discounted infinite-horizon MDP)**.** *A discounted infinite-horizon MDP is defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma, \mu_0 \rangle$, where*

- $\mathcal{S} = [S]$ *is a finite state space*
- $\mathcal{A} = [A]$ *is a finite action space*
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \text{Dists}([0, 1])$ *is a stochastic reward function with expectation* $r(s, a)$
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \text{Dists}(\mathcal{S})$ *is a transition function*
- $\gamma \in [0, 1)$ *is a discount factor*
- $\mu_0 \in \text{Dists}(\mathcal{S})$ *is a starting distribution*

As is standard, we have assumed that the reward function is almost-surely bounded in the interval $[0, 1]$. The objective of the agent is to maximize its *return*, which is defined as the expected discounted sum of rewards, i.e. the quantity $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$. This expectation is taken over the distribution generated by interleaving the actions taken by the agent and the rewards and transitions resulting from those actions. Note that by reward-boundedness the return is bounded by $1/(1-\gamma)$. The strategy of the agent can be formalized as a *policy*, which is a mapping from histories of states to distributions over actions. A memoryless (or Markov) policy is a mapping $\pi : \mathcal{S} \to \text{Dists}(\mathcal{A})$. Memoryless policies will henceforth simply be called *policies*. A policy defines two fundamentals functions called the *value function* $v^\pi : \mathcal{S} \to [0, 1/(1-\gamma)]$ and the *action-value function* $q^\pi : \mathcal{S} \times \mathcal{A} \to [0, 1/(1-\gamma)]$. These are defined via

$$v^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s] \quad \& \quad q^\pi(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s, A_0 = a] \tag{1}$$

The value function v^π (resp. q^π) can be written as S-dimensional (resp. $S \cdot A$-dimensional) vectors. Value functions satisfy the recursion

$$v^\pi(s) = \mathbb{E}_{a\sim\pi(s)}\left[r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(s,a)}[v^\pi(s')]\right] \coloneqq \mathcal{T}^\pi v^\pi(s) \tag{2}$$

$$q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(s,a),a'\sim\pi}[q^\pi(s',a')] \coloneqq \mathcal{T}^\pi q^\pi(s,a) \tag{3}$$

In equations (2) and (3) we have also defined the policy-specific *Bellman evaluation operators* $\mathcal{T}^\pi$ which apply the right-hand-side of the above equations.[3]

An optimal policy is defined as a policy which simultaneously maximizes the return at all states. It is a fundamental theorem about MDPs [6] that optimal policies exist, are memoryless, and that there exists a deterministic optimal policy. For simplicity we refer to one of the deterministic optimal policies as *the* optimal policy, and denote it by $\pi^\star$. Its value functions are denoted by $v^\star$ and $q^\star$. The *optimal* value functions satisfy the recursion

$$v^\star(s) = \max_a\{r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(s,a)}[v^\star(s')]\} \coloneqq \mathcal{T} v^\star(s) \tag{4}$$

$$q^\star(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(s,a)}[\max_{a'} q^\star(s',a')] \coloneqq \mathcal{T} q^\star(s,a). \tag{5}$$

Equations (4) and (5) have also defined the *Bellman optimality operators* $\mathcal{T}$ (or simply Bellman operators). It is easy to see that the Bellman operators $\mathcal{T}^\pi$ and $\mathcal{T}$ have their respective value functions as fixed points. Since the Bellman operators are L_∞-contractive (with Lipshitz constant $\gamma < 1$), this fixed point is unique by the Banach fixed-point theorem.

Finite-horizon MDPs In the finite-horizon setting, the definition of the MDPs are identical except that the discount factor is instead replaced by $H \in \mathbb{N}$, and the return is defined as the expected sum of rewards along trajectories of length H, $\mathbb{E}[\sum_{t=1}^H r_t]$. The "semantics" for this model are that after H steps, the agent is reset to a new state sampled from the starting distribution. We can assume without loss of generality that $\mathcal{S}$ is a disjoint union of per-horizon state spaces, i.e. $\mathcal{S} = \cup_{h\in[H]}\mathcal{S}_h$. Furthermore, in the finite-horizon setting, the objects of interest may be non-stationary, meaning that they depend on the current step $h \in [H]$. A (non-stationary) policy $\pi = (\pi_1, \cdots, \pi_H)$ prescribes a sequence of actions $\pi_h : \mathcal{S}_h \to \mathrm{Dists}(\mathcal{A})$, and its value function is $v^\pi(s) = \mathbb{E}[\sum_{h'=h}^H r(s_{h'}, a_{h'}) \mid s_h = s, a_{h'} \sim \pi_{h'}(s_{h'})]$, where $s \in \mathcal{S}_h$. The action-value function $q^\pi(s,a)$ is defined similarly, save that the first action taken is a and the proceeding actions follow π. The definitions of the optimal value functions and the Bellman operators follow similarly.

Sample complexities in RL (the PAC objective) In RL, we assume that the MDP is unknown to the agent, and thus they must "explore" the MDP and collect enough samples to deduce the optimal policy. We measure the efficiency of an algorithm as the number of samples required to recover a policy with a probably-approximately-correct value function:

[3] We do not distinguish the Bellman operators on $\mathbb{R}^S$ and those on $\mathbb{R}^{S\cdot A}$, since it will be clear by type-checking which one is being applied.

Objective 2.2 (PAC-MDP)**.** *The objective is to find a policy $\hat{\pi}$ such that*

$$v^{\hat{\pi}}(\mu_0) \geq v^{\star}(\mu_0) - \varepsilon \quad \textit{with probability } \geq 1 - \delta, \tag{6}$$

where we have introduce the shorthand $v(\mu_0) = \mathbb{E}_{s\sim\mu_0}[v(s)]$. [4]

We refer to such a policy as ε-suboptimal (or sometimes ε-optimal). The *sample complexity* of an algorithm $\mathbb{A}$ is the worst-case number of samples $n(\mathbb{A}, \varepsilon, \delta)$ required to find a policy satisfying (6).

Interaction protocols At different times we will be considering several different interaction protocols. The weakest one is the *offline* setting, where the agent is only given a fixed dataset $\{(s_i, a_i, r_i, s_i', a_i')\}_{i=1}^n$ collected i.i.d. from the MDP in the following manner:

$$s_i \overset{\texttt{i.i.d.}}{\sim} \mu,\ a_i \sim \pi_b(s_i), r_i \sim \mathcal{R}(s_i, a_i),\ s_i' \sim \mathcal{P}(s_i, a_i), a_i' \sim \pi_b(s_i'). \tag{7}$$

In particular, no further interaction with the MDP is allowed. The policy π_b is called the *behaviour* policy.

By contrast, the *online* setting is the case where the agent can interact with the MDP. [5] Here, from any state s, the agent can take action a, which will gives a random reward $r \sim \mathcal{R}(s, a)$ and transitions the agent to state $s' \sim \mathcal{P}(s, a)$. The initial state is sampled from μ_0. In the finite-horizon setting, trajectories terminate after H steps and the agent is placed at a new initial state.

We will occasionally be assuming stronger interaction protocols, namely that the agent has the ability to "simulate" transitions in the MDP.

Assumption 2.3 (Resets)**.** *Like in the online setting, except that after experiencing a transition (s, a, r, s') in the MDP, the agent can return to the state s. Syntactically, this is done via the* RESET() *function.*

Assumption 2.4 (Generative model)**.** *The learner is equipped with a "simulator" which can query transitions (s, a, r, s') from the MDP starting from any state action pair. Syntactically, this is done via* SIMULATE(s, a).

Part II
Is linearity of optimal values sufficient for sample-efficient RL?

We begin our investigations into the theory of RL with function approximation by studying the seemingly-simple case of *linear function approximation*. In linear function approximation, we model policies, values functions, or MDPs themselves as linear functions of some

[4] In the literature, $v^{\pi}(\mu_0)$ is also denoted by $J(\pi)$.

[5] Not to be confused with the setting of "online learning", i.e. where the data arrives in an adversarial order [11].

given feature mapping. For reasons that will become clear, we will be focusing on the case of approximating value functions (which we call linear *value* approximation to distinguish from the more general setting). In this setting, the learner is provided with a *feature mapping* $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ (for modelling action-value functions) or $\psi : \mathcal{S} \to \mathbb{R}^d$ (for modelling state-value functions). The features can be used to approximate value functions as linear functions via $\hat{q}(s,a) = \langle \varphi(s,a), \hat{\theta} \rangle$. We can also write the linear prediction in vector form as $\hat{q} = \Phi\hat{\theta}$, where $\Phi \in \mathbb{R}^{(S\cdot A)\times d}$ is the feature matrix whose $(s,a)^{\text{th}}$ row is $\varphi(s,a)$.

The idea of using function approximation to help solving large-scale MDPs originates in the 1960s [e.g., 10]: 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 [12] 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.

It was not until the field began tackling the question of sample-efficiency in RL [13, 14, 15] that attention has been turned towards the problem of sample-efficient learning with the help of linear features. As previously mentioned, some assumptions are required if one wants to speed up the efficiency of recovering the optimal policy (again, one can imagine the case where the features all map to 0). A *realizability* assumption in the linear setting posits that the object of interest can be written as a linear function of the feature map. We define below what it means for policies, value functions or MDPs to be linear.

Definition 2.5 (Linear realizability)**.** *Given a feature map $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we say that $\pi^\star$ is linearly-realizable if*

$$\exists \theta \text{ s.t. } \pi^\star(s) \in \operatorname{argmax}_a \{\varphi(s,a), \theta\}. \tag{8}$$

We say that $q^\star$ is linearly-realizable if

$$\exists \theta^\star \text{ s.t } q^\star(s,a) = \langle \varphi(s,a), \theta^\star \rangle. \tag{9}$$

We say that all value functions q^π are linearly-realizable if

$$\forall \pi, \exists \theta_\pi \text{ s.t. } q^\pi(s,a) = \langle \varphi(s,a), \theta_\pi \rangle. \tag{10}$$

We say that the MDP is linearly-realizable (or simply a linear MDP) if

$$\exists \theta_r \,\&\, \mu : \mathcal{S} \to \mathbb{R}^d \text{ s.t. } r(s,a) = \langle \varphi(s,a), \theta_r \rangle \,\&\, P(s' \mid s,a) = \langle \varphi(s,a), \mu(s') \rangle. \tag{11}$$

When these are assumptions do not hold, we can measure the degree to which they are violated by looking at the distance (as measured by some metric) of the best linear approximation (e.g. for $q^\star$-realizability we may choose to define the approximation error as $\inf_\theta \|\Phi\theta - q^\star\|_\infty$). It is relatively straightforward to show that these realizability assump-

tions form a hierarchy, namely that:

$$(11) \implies (10) \implies (9) \implies (8)$$

It was shown by Du et al. [16] that Equation (8) is insufficient for sample-efficient RL, even if one strengthens this assumption to include an $\Omega(1)$-sized "margin" between the best action and the second-best action. On the other hand, it was shown by Jin et al. [17] and Lattimore et al. [18] respectively that Equation (11) and (10) *are* sufficient for sample-efficient RL (for the latter, under the generative model setting). This left open the basic question:

Question 2.6. Is realizability of the optimal action-value function enough to guarantee sample-efficient learning?

We are operating under the canonical objective (Definition 1.2), meaning that we aspire to find an algorithm with $\texttt{poly}(d, H, 1/\varepsilon)$ complexity for recovering a ε-suboptimal policy.

Related works Positive results for this question have been obtained under a number of additional assumptions. In the online setting, Wen and Roy [19] 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 [20] under an additional "gap assumption", which requires knowledge of the minimum separation (over all states) between the value of the optimal action and that of the second best action. Their sample complexity also scales in the inverse of the gap, meaning that we can make it arbitrarily large by adding a new action which is arbitrarily close to optimal. Several positive results have been obtained for linear MDPs [17, 21, 22, 23]. With a generative model, Du et al. [24] give a query complexity result for the least-squares value iteration algorithm which scales as $\mathcal{O}(\texttt{poly}(H, d))$ provided that the inverse gap is a known parameter and is itself $\mathcal{O}(\texttt{poly}(H, d))$. Shariff and Szepesvári [25] and Zanette et al. [26] obtain polynomial bounds under $v^\star$-realizability or $q^\star$-realizability with the additional assumption that all features lie inside the convex hull of at most $\mathcal{O}(\texttt{poly}(H, d))$ of the feature vectors. Du et al. [27] gives a positive result under the condition that *both* $q^\star$ and $v^\star$ are linearly-realizable. Zanette et al. [28] gives a positive result under the condition that the features satisfy a so-called *Bellman completeness* condition, which posits that $\mathcal{T} q_\theta$ is itself linear for all linear q_θ (this assumption is implied by (11) and implies (9)). All of the above assumptions are strictly stronger than just $q^\star$-realizability. A separate line of work [29, 30, 27, 31] establishes several complexity measures (e.g. the Bellman rank) for learning under arbitrary realizable function classes, which subsumes linear function classes. However, their bounds additionally scale with these complexity measures, and $q^\star$-linearity does not guarantee that these complexity measures will be $\texttt{poly}(d, H)$.

On the side of negative results, Du et al. [16] gave a partial answer to question 2.6 by ruling out the case where the optimal value function is not realizable but is instead approximately realizable, provided that the approximation error is large enough. More precisely, they showed that if this approximation error is larger than $\Omega(\sqrt{H/d})$, then there is an exponential lower bound of $\Omega(\min\{e^d, 2^H\})$. In fact, this lower bounds continues to hold even if we assume that *all policies* can be approximately realized with error greater than

$\Omega(\sqrt{H/d})$. However, this still left open the question whether realizability was sufficient in the presence of no (or of more moderate) error.

3 An Exponential Lower Bound [Completed, ALT '21]

Our first contribution [32] was a negative answer to the above question (Question 2.6). Namely, we give an *information-theoretic lower bound* on the sample complexity of learning with only linear optimal value functions. Let us recap the assumption.

Assumption 3.1 (Linear $q^\star$ realizability)**.** *Given $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, $\exists \theta^\star \in \mathbb{R}^d$ s.t.*

$$q^\star(s,a) = \langle \varphi(s,a), \theta^\star \rangle \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}.$$

We further assume that $\max_{s,a} \|\varphi(s,a)\|_2 \leq 1$ and that $\|\theta^\star\|_2 \leq B$ for some known $B \in \mathbb{R}$.

The main theorem is the following.

Theorem 3.2 (Lower bound under $q^\star$-realizability [32])**.** *For all d and H larger than some constants, there exists a family of MDPs with horizon H and features of dimension d satisfying Assumption 3.1 such that any algorithm which recovers a policy satisfying $v^{\hat{\pi}}(s_0) \geq v^\star(s_0) - \frac{9}{128}$ will need a worst-case sample complexity of at least*

$$\Omega(\min\{e^d, 2^H\}) \tag{12}$$

This continues to hold even if the learner is equipped with a generative model (Assumption 2.4).

A (slightly weaker) lower bound also holds in the discounted setting.

Corollary 3.2.1 (Lower bound under $q^\star$-realizability, discounted version)**.** *For any $\gamma \geq \frac{2}{3}$, the sample complexity is lower bounded by $e^{\Omega(d)}$.*

In other words, we showed that there exists a family of MDPs whose optimal value functions are linear but where any algorithm that returns a good solution will require at least exponentially many samples either in d or in H. The lower bound holds for all possible algorithms that claim to solve this problem, and applies even when the agent is given a simulator of the environment. Thus, the lower bound also applies for the more realistic "online" setting. This also implies hardness for learning with weaker function approximation, such as with approximate realizability or realizability with function classes that subsume linear functions (such as generalized linear models or neural networks).

Proof sketch We briefly sketch the main ideas for the lower bound construction in the finite-horizon setting, but refer the reader to the paper for more precise definitions and proofs of the following claims. See Figure 1 for an illustration. There are a large number of actions, $A \approx e^d$. One of these actions is optimal, and is denoted by $a^\star$. (In reality, we construct A different MDPs which are identical save for the identity of the optimal action). The state space is an A-ary tree, and the transitions are deterministic (thus, we can

identify each state by the sequence of actions it took to get there). We construct a *Johnson-Lindenstrauss* set of vectors [33], i.e. an exponentially large set of vectors $\{v_i\}$ which all satisfy $\|v_i\|_2 = 1$ and $|\langle v_i, v_j\rangle| \lesssim 1/2$. We assign each of the actions one of these vectors ($v_i = v_{a_i}$), and roughly define the features $\varphi(s_{h+1}, a_{h+1}) = \varphi(a_{1:h}, a_{h+1}) \approx \prod_{i=1}^{h} \langle v_{a_i}, v_{a_{i+1}}\rangle$, i.e. the inner product between all the vectors for the actions that it took to reach that state. By near-orthogonality, we notice that the magnitude of the feature vectors roughly halves at each iteration. The optimal parameter is taken to be the vector for $a^\star$, $v_{a^\star}$.

The reward function is identically zero unless: i) $a^\star$ is played, or ii) any action is played at the final stage. In both cases, the reward given is the value of $q^\star$, which is linear. Thus, since the features geometrically shrink, the magnitude of the rewards does as well. In particular, at horizon H the rewards are of size $\approx 2^{-H}$. We further make the terminal rewards Bernoulli random variables.

It remains to show linearity of $q^\star$ and statistical hardness of the MDP. The first point is deferred to the paper. For the second, the intuition is that the learner must find $a^\star$, and thus is forced to either a) try all of the actions to find the optimal one, or b) use the simulator to query rewards from the last stage to deduce the identity of $\theta^\star$ (this can be done, in principle, since the rewards are linear). The first case evidently leads to an exponential lower bound, since $A \approx e^d$. For the second case, recall that the rewards are Bernoulli with mean $\approx 2^{-H}$, and thus it will take at least 2^H samples before the learner can observe the mean of the random variable to get information about $\theta^\star$.

Discussion & subsequent work Our work was published at the ALT 2021 conference, where it received the Best Student Paper Award. Our result was covered in tutorials on the Theory of RL given at the Simons Institute (UC Berkeley) [link], FOCS 2020 [link], and COLT 2021 [link], and it is now included in textbooks [34] and RL theory lecture notes [link 1, link 2, link 3].

It is interesting to compare our result with several related but simpler settings. Our bound entails three *exponential separations*, i.e. three pairs of settings where the statistical complexity is polynomial in one and exponential in the other: 1) realizability of the value function for every policy [35] vs. realizability of only the optimal value function, 2) linearly-realizable bandits (i.e. single-horizon problems) [36] vs. longer-horizon problems, and 3) fully deterministic MDPs [19] vs. MDPs with deterministic transitions and stochastic rewards. In all of the above settings, linear realizability of the optimal value function alone entails the existence of a $\texttt{poly}(d, H, \frac{1}{\varepsilon})$ algorithm.

Our construction was adapted (and simplified) for the online setting (without a simulator) by Wang et al. [37], where they showed that the lower bound continues to hold even if we assume a constant-sized gap between the optimal value of the best action and that of the second-best action (the gap in our construction is exponentially small). The simplified construction was also used in [31]. Follow-up works by a subset of the authors [38] improves the lower bound by modifying the construction so that it holds for polynomially-sized action sets.

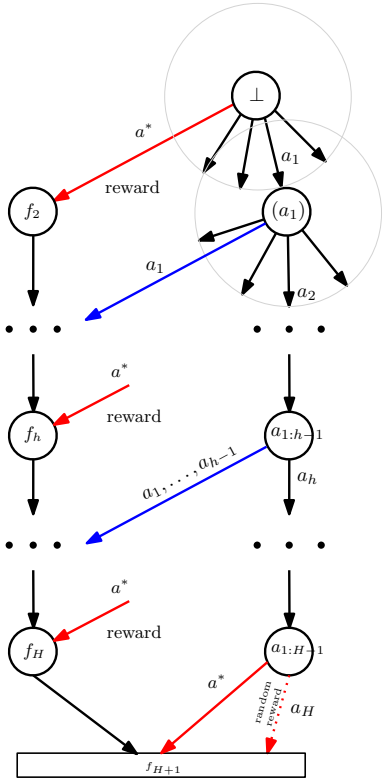

Figure 1: Illustration of the construction for Theorem 3.2. Figure taken from [32]. The states on the right belong to an A-ary tree. The dynamics are deterministic. Action $a^\star$ or actions that are repeated transition the agent to a "game-over state", where no further reward is given. Action $a^\star$ always gives a (large) positive reward. All actions at stage H are positive to ensure realizability, but they are stochastic with exponentially small means and thus hard to detect.

4 A Positive Result When There Are Few Actions [Completed, COLT '21]

Motivation & problem setting We recall that the construction in our lower bound (Theorem 3.2) had an exponentially large action set. Our next question was whether this was necessary for hardness, or whether the construction could be improved to hold for smaller action sets. A more general version of this question is to ask whether $\texttt{poly}(d, H, A, 1/\varepsilon)$ was possible. (Recall that A is the size of the action set). This guarantee is less desirable as the size of the action set may be extremely large (and, in the limit, even continuous). Indeed, in the three simpler settings just-mentioned (bandits, q^π-realizability for all π, and deterministic MDPs), sample-efficient learning is possible regardless of the size of the action set: the key is that all the actions are highly correlated via the feature mapping and realizability of $q^\star(s, a)$. Our previous lower bound states that this is no longer true once we remove those simplifying assumptions. Nevertheless, this possibility of $\texttt{poly}(d, H, A, 1/\varepsilon)$-learning was not ruled out since, in the setting where the number of actions is also a parameter of interest, the lower bound becomes linear in A. Our second paper [39] finds that, under

linear realizability, efficient learning is possible whenever the action set is "small", namely $\mathcal{O}(1)$.

Overview of contributions This paper considers the slightly different setting of $v^\star$-realizability. Formally:

Assumption 4.1 (Linear $v^\star$ realizability)**.** *Given $\psi : \mathcal{S} \to \mathbb{R}^d$, $\exists \theta^\star \in \mathbb{R}^d$ s.t.*

$$v^\star(s) = \langle \psi(s), \theta^\star \rangle \quad \forall s \in \mathcal{S}.$$

We further assume that $\max_s \|\psi(s)\|_2 \leq 1$ and that $\|\theta^\star\|_2 \leq B$ for some known $B \in \mathbb{R}$.

We also assume the same interaction protocol as the lower bound, which is that the learner is equipped with a generative model (Assumption 2.4). In fact, for this upper bound, we can relax this assumption to the setting of resets (Assumption 2.3). Our work derives the novel TensorPlan algorithm, which is sample-efficient whenever the action set is $\mathcal{O}(1)$ (i.e. a constant independent of d or H). More formally:

Theorem 4.2 (Upper bound under $v^\star$-realizability [39])**.** *Under $v^\star$-linearity (Assumption 4.1) and the presence of resets (Assumption 2.3), there exists an algorithm (*TensorPlan*) which returns a ε-suboptimal policy after*

$$\texttt{poly}\left(\left(\frac{dH}{\varepsilon}\right)^A, B\right) \tag{13}$$

interactions from the MDP.

When A is a constant (e.g. 2), this is the first *statistically* efficient algorithm when using only the linear realizability of the optimal value function. In addition, our algorithm works in a more robust setting: it will automatically approximate the *best* policy whose value function is linear and furthermore extends to the case where said value function is *approximately* linear (albeit with small misspecification, $\lesssim 1/\sqrt{d^A A}$). The algorithm also can be shown to work (with the same complexity) for $q^\star$-realizability assuming that the transitions in the MDP are deterministic. Unfortunately, while statistically efficient, it is not known whether our algorithm can be made *computationally* efficient as well. While well-behaved for small action sets, this algorithm's general complexity is an exponential function in the number of actions, which is not affordable in general.

Our algorithm is based on the principle of *optimism under uncertainty*, where the learner explores by assuming the world is as nice as can be, in combination with a "local consistency checking" mechanism to select hypotheses which satisfy the optimal value function equations at the states seen so far. We bound the maximal number of trajectories that the algorithm can execute via a novel tensorization scheme which could be of independent interest. For ease of presentation, we will give a more complete description of the algorithm in Section 5 after explaining the proceeding results.

Discussion, and subsequent work We have seen that TensorPlan is statistically efficient (in d, H, and $1/\varepsilon$) when $A = \mathcal{O}(1)$, but has an exponential-in-A complexity in general.

On the other hand, we have seen that when $A = \mathcal{O}(\exp(d))$ then sample efficiency is impossible. This still left open the possibility of efficient learning when $A = \texttt{poly}(d, H)$, or more generally the possibility of $\texttt{poly}(d, H, A, 1/\varepsilon)$-efficient learning. Both of these questions were ruled out in subsequent work by Weisz et al. [40], who showed that even when $A \approx \min\{d^{1/4}, H^{1/2}\}$, a (highly non-trivial) modification of the construction from Section 3 yields an exponential lower bound of $2^{\Omega(\min\{d^{1/4}, H^{1/2}\})} \approx 2^A$. Thus, some exponential dependence on the size of the action set is unavoidable in general. Our next work investigated the same learning setting but in the presence of some additional "side-information", which we will see can remove this exponential barrier.

5 A Positive Result Using A Few Expert Demonstrations [Completed, NeurIPS '22]

Motivation & problem setting We notice that the MDPs we constructed for the lower bound in Section 3 relied solely on hiding the identity of a single optimal action, and that this action can be played from any state. Thus, solving this class of MDPs is tantamount to discovering the identity of a single action. This property continues to hold in all of the subsequent lower bounds for RL with realizable optimal value functions [37, 31]. This suggests a particular structure which is present in $q^\star$-realizable MDPs, and we hypothesized that a small amount additional "side information" (to help reveal the identity of this action) would be highly beneficial for the learner. We investigated this intuition by formalizing the problem of learning with $v^\star$-realizable features in the presence of some expert advice. This strengthened interaction protocol is a stronger assumption compared to the two previous papers (which only assumed a generative model or resetting capabilities), but we will see that it enables polynomial sample complexities while making surprisingly-few inquiries to the expert. In particular, we do not need to make stronger representational assumptions from the features beyond $v^\star$-realizability, nor do we need to make structural assumptions on the MDPs.

The problem setting is the following. We firstly assume that the learner has interactive access to an expert policy π° which is deterministic.

Assumption 5.1 (Interactive expert)**.** *There is an oracle which can be queried at the current state s, which returns the action $\pi^\circ(s)$. Syntactically, the oracle is queried via the* $\textsc{oracle}(s)$ *function.*

We denote by $v^\circ := v^{\pi^\circ}$ the value function of the expert policy π°. We next assume that this value function is linear:

Assumption 5.2 (v°-linearity, with bounded features)**.** *The value function v° of the expert is linear with known features $\psi \in \mathbb{R}^d$, i.e.*

$$v^\circ(s) = \langle \psi(s), \theta^\circ \rangle, \ \forall s \in \mathcal{S}, \tag{14}$$

for some unknown $\theta^\circ \in \mathbb{R}^d$. We further assume that $\|\psi(s)\|_2 \leq 1 \, \forall s$ and that $\|\theta^\circ\|_2 \leq B$ for some known $B \in \mathbb{R}^d$.

The third assumption is that, as before, the agent has access to resets (Assumption 2.3). Our objective is the usual PAC-MDP objective (Objective 2.2), except that we generalize it so that the learner is instead tasked with competing with v°, the value function of the expert. Namely, we wish for our algorithm to recover a policy $\hat{\pi}$ which satisfies

$$v^{\hat{\pi}}(\mu_0) \geq v^\circ(\mu_0) - \varepsilon \quad \text{with probability } \geq 1 - \delta,$$

When $\pi^\circ = \pi^\star$, this is of course the traditional objective. The reason for this generalization is that π° may be arbitrarily uninformative if one wishes to compete with $\pi^\star$.[6]

Overview of contributions We derive a statistically and computationally efficient algorithm (DELPHI) which only requires a small number of demonstrations from the expert policy in addition to some polynomial amount of independent exploratory samples.

Theorem 5.3 (Upper bound under v°-linearity and expert advice)**.** *Suppose Assumptions 5.1, 5.2, and 2.3 hold. Then the* DELPHI *algorithm will recover a policy $\hat{\pi}$ such that $v^{\hat{\pi}}(\mu_0) \geq v^\circ(\mu_0) - \varepsilon$ with probability $\geq 1 - \delta$, using*

$$\mathcal{O}(d \ln(B/\varepsilon)) \text{ oracle calls } \& \ \tilde{\mathcal{O}}(d^2 H^5 A B^4 / \varepsilon^4) \text{ samples from the MDP.}$$

Furthermore this algorithm is computationally efficient.

Compared to pure RL approaches, this corresponds to an *exponential improvement* in sample complexity with surprisingly-little expert input. Ignoring logarithmic factors, the amount of oracle calls required is simply linear in d, the dimension of the feature mapping. Furthermore, this amount is significantly smaller than prior IL approaches which require the same dependence on d but additionally required at least linear factors of both H and $1/\varepsilon$.

DELPHI will solve for a parameter $\hat{\theta}$ such that an induced policy $\pi_{\hat{\theta}}$ (defined via Eq. (**??**)) will compete with the expert's value function. Deploying the policy $\pi_{\hat{\theta}}$ also requires the RESET functionality, since expectations must be estimated from a small number of samples at each state encountered. This is a consequence of our assumption that v° is linear (rather than, e.g., q° or the MDP itself), since selecting actions based on state-value functions will always require one-step look-aheads.

Towards establishing the optimality of our algorithm, we study the capabilities of expert-augmented learners which have bounded exploration budgets. The question of optimality is fairly complex, as the algorithms have adaptive access to two different resources that they may interleave. We choose to focus on minimizing the number of *expert queries*, since these may be significantly more costly than allowing the agent to explore on their own. Thus, we ask:

Question 5.4. *What is the minimal number of expert queries required by any algorithm which has access to a polynomially-bounded exploration budget (in terms of $d, H,$ and A)?*

[6] For instance, we can imagine augmenting any MDP with a "dummy" action that achieves no reward, and defining π° to always take that action.

The reason that we place such a bound on the exploration budget is that we are minimizing the expert cost over all "sample-efficient" algorithms. In particular, without such a bound, we are competing against algorithms that may exhaustively explore all states and solve the MDP without referring to the expert. We choose to study arbitrary polynomial complexities (rather than, say, the samples required by DELPHI) since it is a more fundamental question about the limits of exploration in the presence of linearly-realizable features. Our main lower bound is to show that any such learner will require at least $\tilde{\Omega}(\sqrt{d})$ oracle calls to recover a policy competing with the expert's value function.

Theorem 5.5. *There exists a family of MDPs, feature maps, and expert policies satisfying Assumptions 5.1, 5.2, and 2.3, such that any algorithm with* $\texttt{poly}(d, H, A)$ *exploration budget will need at least*

$$\tilde{\Omega}(\sqrt{d})$$

oracle calls to recover a policy such that $v^{\hat{\pi}}(s_0) \geq v^{\circ}(s_0) - 0.01$.

We also study the weaker setting where only the expert's *policy* is linear (as in Equation (8)), and show that in this setting the lower bound increases to $\Omega(d)$, matching our upper bound up to logarithmic factors. It is a particularly interesting open question to resolve the gap between the oracle complexity required by DELPHI ($\tilde{\mathcal{O}}(d)$) and the one obtained from our lower bound ($\tilde{\Omega}(\sqrt{d})$). We suspect that the looseness is on the lower bound side since the construction is quite intricate, although if the lower bound is tight, then this implies the intriguing possibility that we can find a different algorithm which uses only $\tilde{\mathcal{O}}(\sqrt{d})$ oracle calls (potentially at the cost of a higher exploration complexity).

Intuition for DELPHI We give an overview of the DELPHI algorithm. The full pseudo-code is found in Algorithm 1 (with Algorithm 2 used as a sub-routine). Recall that the expert policy π° satisfies $v^{\circ} = \mathcal{T}^{\pi^{\circ}} v^{\circ}$, and that this fixed point is unique. We say that a candidate value function is *consistent* at a state s if it satisfies the Bellman equation at that state, i.e. if $v(s) = \mathcal{T}^{\pi^{\circ}} v(s)$. Note that we need consistency to hold globally (i.e. at all states) in order to ensure that $v = v^{\pi^{\circ}}$. Our methodology is based on ensuring that consistency on a small number of well-chosen states will guarantee global consistency.

DELPHI is inspired by the previous TENSORPLAN algorithm [39]. As in TENSORPLAN, DELPHI proceeds via a "guess and check" procedure: at every iteration, we pick the *optimistic* linear parameter which is consistent on the past expert data that we have seen. Letting $v_\theta(\cdot) := \langle \psi(\cdot), \theta \rangle$ for any $\theta \in \mathbb{R}^d$, this means that at all states s_i where the expert has previously been queried we verify that $v_\theta(s_i) \approx \mathcal{T}^{\pi^{\circ}} v_\theta(s_i)$. Letting Θ_t denote the parameters that are consistent at iteration t, the optimistic linear parameter is the one which solves $\max_{\theta \in \Theta_t} v_\theta(\mu_0)$. We then check whether this choice of parameter is consistent, by playing $n_{\texttt{rollout}}$ rollouts of length H with a policy derived from the parameter. More specifically, for any θ the policy π_θ takes the form

$$\pi_\theta(s) = \operatorname{argmin}_a \left| \left(\hat{r}(s,a) + \hat{\mathbb{E}}_{s,a}[v_\theta(s')] \right) - v_\theta(s) \right|,$$

After a certain number of rollouts, one of two things happen: either this policy encounters a state where there is no consistent action (i.e. the above minimum has a large value),

or we only encounter states that are consistent.[7] In the first case, we query the oracle for its expert action and use the transition for that action to update the parameter set. In the second case, we derive that if no inconsistencies are observed for several rollouts, then our "virtual value" v_θ is close to the true value under π_θ. In particular:

Lemma 5.6 (Consistency implies accurate prediction (informal))**.** *If $n_{\texttt{rollout}}$ rollouts of π_θ have occured without any inconsistencies, then $v^{\pi_\theta}(s_0) \approx v_\theta(s_0)$.*

Using that θ was optimistic ($v_\theta \geq \max_{\theta' \in \Theta_t} v_{\theta'}$, realizability of $v^\circ = v_{\theta^\circ}$, and the fact that θ° does not get eliminated from our "version space" Θ_t, this implies that we are optimal.

The only thing left to argue is that the number of iterations (i.e. the number of times that we can continue finding new parameters which are not globally consistent) is small. For this, we recall the *direct product* $\oplus$, which corresponds to "concatenating" two vectors, i.e. for any two vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$, we have $u \oplus v = (u_1, \ldots, u_n, v_1, \ldots, v_m)^\top \in \mathbb{R}^{n+m}$. Using linearity of v°, it turns out roughly d inconsistencies are sufficient to bound the iteration complexity. To see this, note that we can re-write the Bellman equation for any $v_\theta(\cdot) = \langle \psi(\cdot), \theta \rangle$ as:

$$\begin{aligned} v_\theta(s) = \mathcal{T}^{\pi^\circ} v_\theta(s) &\iff 0 = r(s, \pi^\circ(s)) + \langle \mathbb{E}_{s' \sim P(s,\pi^\circ)}[\psi(s')] - \psi(s), \theta \rangle \\ &\iff 0 = \langle \Delta_{s,\pi^\circ(s)}, 1 \oplus \theta \rangle, \end{aligned} \tag{16}$$

where we have introduced the notation $\Delta_{s,a} \coloneqq r(s,a) \oplus (\mathbb{E}[\psi(s')] - \psi(s))$ and used linearity of expectation, linearity of inner products, the definition of v_θ, and the definition of the direct product. We call the vector $\Delta_{s,a}$ the *temporal difference* (TD) vector for (s,a). Equation (16) is precisely an orthogonality constraint in $d+1$ dimensions. Since the parameter θ_t which is chosen at time t was consistent on past data, it is orthogonal to the previous $t-1$ TD vectors which have been generated from interactions with the oracle. If we happen to find a state which has no consistent action, then the TD vector corresponding to the expert action at that state must not be in the span of the previous expert TD vectors (otherwise it would be consistent). It follows that the iteration complexity is at most $d+1$, since there are at most $d+1$ linearly independent vectors in $\mathbb{R}^{d+1}$. We use the Eluder dimension [41] to generalize this argument to the case where the expectations are estimated.

Algorithm 2 `measureTD`

Inputs: $s, a, \psi(\cdot), n,$ RESET()
for $i = 1$ to n **do**
    Play action a at s, receive sample R_l and S'_l from MDP
    $\Delta_i \leftarrow R_l \oplus (\psi(S'_l) - \psi(s))$
    RESET()
end for
return $\hat{\Delta}_{s,a} := \frac{1}{n} \sum_{i \in [n]} \Delta_i$

DELPHI VS. TENSORPLAN DELPHI is inspired by TENSORPLAN, and we can now describe the differences. In the expert-less setting, TENSORPLAN instead aims to ensure that $v(s) =$

[7] Due to sampling errors, we tolerate some small amount of inconsistency during these checks.

Algorithm 1 DELPHI

Inputs: $s_0, \varphi,$ `sub-optimality` $\varepsilon_{\texttt{target}},$ `confidence` $\delta,$ `parameter bound` B
$\Theta_0 \leftarrow \texttt{Ball}_{\ell_2}(B)$ ▷ Θ_t : current consistent parameters
Initialize hyperparameters E_d, $n_{\texttt{eval}}$,$n_{\texttt{rollout}}$, and $\varepsilon_{\texttt{tol}}$ [see paper for precise values]
for $t = 1$ to $E_d + 1$ **do**
  Pick $\theta_t \in \operatorname{argmax}_{\theta \in \Theta_{t-1}} \left(v_\theta(s_0) \coloneqq \theta^\top \psi(s_0)\right)$ ▷ Optimistic choice over Θ_{t-1}
  `consistent` $\leftarrow$ `true`
  for $m = 1$ to $n_{\texttt{rollout}}$ **do** ▷ $n_{\texttt{rollout}}$ number of rollouts with θ_t-induced policy
    $S_{t,m,h} = s_0$ ▷ Initialize rollout
    for $h = 1$ to H **do**
      for $a \in [A]$ **do** ▷ For each action
        $\hat{\Delta}_{S_{t,m,h},a} \leftarrow \texttt{measureTD}(S_{t,m,h}, a, n_{\texttt{eval}})$ ▷ Measure TD vector at (s, a)
      end for
      if $\min_a \left|\langle \hat{\Delta}_{S_{t,m,h},a}, 1 \oplus \theta_t \rangle\right| > \varepsilon_{\texttt{tol}}$ **then** ▷ No consistent action
        `consistent` $\leftarrow$ `false`
        $a_t^\circ \leftarrow \textsc{oracle}(S_{t,m,h})$ ▷ Query oracle for $\pi^\circ(S_{t,m,h})$
        $\tilde{\Delta}_{S_{t,m,h},a_t^\circ} \leftarrow \texttt{measureTD}(S_{t,m,h}, a_t^\circ, 4E_d n_{\texttt{eval}})$ ▷ Refined data
        $\Theta_t \leftarrow \Theta_{t-1} \cap \{\theta \mid |\langle \tilde{\Delta}_{S_{t,m,h},a_t^\circ}, 1 \oplus \theta \rangle| \le \varepsilon_{\texttt{tol}}\}$ ▷ New admissible θs
        Exit current iteration, $t \leftarrow t + 1$, Goto Line 5.
      end if
      $A_{t,m,h} \leftarrow \operatorname{argmin}_{a \in [A]} \left|\langle \hat{\Delta}_{S_{t,m,h},a}, 1 \oplus \theta_t \rangle\right|$ ▷ Else consistent, keep playing
      Play $A_{t,m,h}$, get $R_{t,m,h}, S_{t,m,h+1} \sim \texttt{MDP}$ ▷ Roll forward
    end for
  end for
  if `consistent == true` **then**
    return θ_t ▷ No inconsistency for m rollouts $\implies$ success
  end if
end for
return θ_{E_d+1}

$\mathcal{T}v(s)$ globally, so that $v = v^\star$. The above constraint is a maximum over actions (recall Equation (4)), which we wish to avoid. We instead say that a value function is consistent at a state s if *there exists* any action such that $v(s) = r(s,a) + \mathbb{E}_{s' \sim \mathcal{P}(s,a)}[v(s')]$. The version space Θ_t is thus all consistent parameters on past transitions. We again proceed by optimism over this new version space, and roll out the policy defined by (**??**). As before, if no inconsistencies are observed, we can derive that $v_\theta \approx v^{\pi_\theta}$ and thus $v^{\pi_\theta} \ge v^\star$ by optimism. The last argument is to bound the number of iterations required. Following steps similar to Equation (16), it holds that for the action which was consistent we have $0 = \langle \Delta_{s,a}, 1 \oplus \theta \rangle$. The *existence* of a consistent action is thus equivalent to the product of these constraints: $0 = \prod_{a \in A} \langle \Delta_{s,a}, 1 \oplus \theta \rangle$. We can write this in the *tensor space* as $\langle \otimes_a \Delta_{s,a}, \otimes_a (1 \oplus \theta) \rangle$. This corresponds to an orthogonality constraint in a $(d+1)^A$-dimensional space, and thus the iteration complexity is bounded by $(d+1)^A$ by the same argument as before.

The extension of TENSORPLAN to the new setting naturally incorporates the expert

demonstrations, while simultaneously (1) having low oracle requirements, (2) addressing the exponential sample complexity of TensorPlan, and (3) rendering the algorithm computationally efficient.

Related works Our proposed setting can be solved by traditional (interactive) Imitation Learning (IL) methods, although with worse rates. As in our setting, interactive IL considers the case where the learner has access to an expert oracle that can be queried adaptively. It differs from our setting, however, since traditionally in IL the learner does not observe reward information. We further differ from the IL setting since IL generally studies function approximation with policy classes and assume realizability of policies, whereas we consider value function approximation. We further assume access to a RESET function. Despite that many demonstrations of interactive IL occur in simulated domains [42, 43, 44], the benefits of this feature have not previously been studied. Our assumption of v° linearity entails that many IL methods are not directly applicable. Indeed, the policy π° itself does not need to be linear (despite that v° is), so it is unclear which policy class to use for those algorithms. Assuming for the sake of comparison that linear policies can be used, IL methods would still obtain worse oracle rates. Indeed, using results from Agarwal et al. [34], Behaviour Cloning (for the passive case) or AggreVaTe [44, 45] (for the interactive case) have worst-case oracle complexities of $N = \mathcal{O}(dH^4/\varepsilon^2)$.[8] This is in sharp contrast to our $\mathcal{O}(d\ln(1/\varepsilon))$ oracle calls, which in independent of H and logarithmic in $1/\varepsilon$, and demonstrates the improvement due to exploration with the help of value-function approximation. Beyond these approaches, another intuitive method would be to perform regression by doing a Monte Carlo estimation for the value of $v^\circ(s)$ for each s along a certain "good" set of features which would be suitable for extrapolation. This would require collecting rollouts from those states, which will again introduce a factor of H in the number of oracle queries. Our algorithm instead finds a set of state-action pairs where the Temporal difference (TD) errors (which we represent as vectors) span orthogonal directions. These can be estimated with a single transition, and this "local fitting" approach is novel to the IL literature and avoids the factors of H and $1/\varepsilon$ from previous works. In terms of linear structure in IL, most relevant is the recent work of Rajaraman et al. [46], which, in the reward-free case, assumes that the expert policy is linear. A sample complexity of $\tilde{\mathcal{O}}(dH/\varepsilon)$ is shown for Behaviour Cloning in this case, again suffering from dependence on H and $1/\varepsilon$, and no lower bound is given.

Part III
Beyond the standard objectives

In the papers presented in the previous part (Part II), we have focused on finding the minimal sufficient conditions which would enable tractable RL with linear function approximation. The main takeaway is that, without further structure, weak realizability assumptions alone are insufficient for sample-efficient learning in the linear setting. Thinking beyond

[8] Those results hold for the discounted setting, so we applied the standard conversion $H \mapsto (1-\gamma)^{-1}$.

the linear setting to more challenging (and practically relevant settings), this implies that either

1. weaker guarantees need to be studied, and/or
2. more structure will be required.

In the sequel, we intend to study both of these points, through a variety of problem settings where open questions remain. We begin by presenting completed work on the effects of misspecification for the policy evaluation problem in RL (Section 6). We then propose two interesting avenues for future research. Firstly, an investigation into redefining regret and PAC objectives so as to make them meaningful in settings where one cannot recover the optimal policy (Section 7). Secondly, a study into whether conditions which have been shown to be necessary in offline RL can be leveraged as structural conditions for online RL (Section 8).

6 Beyond Realizability: Optimal Misspecified Policy Evaluation [Completed, in submission]

Our first paper will begin by studying what happens when one removes the realizability assumption, namely the general *misspecified* setting. In practice, realizability assumptions rarely hold, and the degree to which they are violated is largely unknown. Thus, we need algorithms that do not rely on the realizability assumption and whose guarantees *automatically* scale with the degree of misspecification. When the ground truth solution is not representable by the function class, a natural relaxed objective is to instead recover the *best-in-class solution*, i.e. the function in the function class which is closest to the true solution as measured by some norm. The "minimal" error incurred by the best-in-class function is called the *misspecification* error. The ratio between the error of the attained solution and that of the best-in-class solution is called the *approximation ratio*.

Existing error bounds for misspecified RL problems often suffer large multiplicative factors of this misspecification error in addition to other statistical errors [47], and it is rarely the case that attention is brought to whether these blowup factors are necessary, or if the ratios attained are optimal. In a myriad of easier settings (such as linear regression or empirical risk minimization), it is indeed possible to recover an approximation factor of 1 (or arbitrarily close to 1) [48, 49]. Whether or not similar guarantees are possible in RL problems, or what the optimal ratios would be, has been largely unstudied. Towards studying this question, we formulate a simple offline RL problem with linear features, and examine the optimal approximation ratio achieved by any estimator (even *asymptotic* ones).

Concretely, our learning problem is that of linear off-policy value function estimation in infinite-horizon discounted MDPs. In this problem, the learner is given access to a feature-map $\varphi : \mathcal{S} \mapsto \mathbb{R}^d$ and an offline dataset $\{(s_i, r_i, s_i')\}_{i=1}^n$ of tuples collected using a fixed policy in the MDP. The goal is to evaluate the value function of this policy. Since the policy is fixed, we simply denote the value function as $v_{\mathcal{M}} \in \mathbb{R}^S$ and the transition matrix as $P \in \mathbb{R}^{S \times S}$. The states s_i are sampled i.i.d. from an arbitrary ("off-policy") distribution μ. We also study the *aliased* setting where the dataset takes the form $\{\varphi(s_i), r_i, \varphi(s_i')\}_{i=1}^n$,

i.e. the states can only be observed through their feature mapping. We do not assume anything about the off-policy distribution beyond that it yields a non-degenerate second moment matrix. We also do not assume that the value function to be evaluated is linear in the given feature mapping, and thus we formulate the task of the learner as simply outputting the *best possible linear approximation* of the true value function (as measured by some norm). Under misspecification, one often bounds the error of an estimator $\hat{v}$ by an *oracle inequality* of the form:

$$\|\hat{v} - v_{\mathcal{M}}\| \leq \underbrace{\alpha_n(\mathcal{M}, \mu, \varphi)}_{\text{approximation factor}} \underbrace{\inf_{\theta} \|\Phi\theta - v_{\mathcal{M}}\|}_{\text{misspecification error}} + \underbrace{\varepsilon_n(\mathcal{M}, \mu, \varphi)}_{\text{statistical error}}, \tag{17}$$

which holds either with high probability or in expectation. We will consider the limit of infinite sample sizes, where the statistical error is zero, and can think of this as the case when the learner is given the data-generating distribution (call it $\mathbb{Q}_{\mathcal{M},\mu,\varphi}$). A deterministic asymptotic estimator is a map from distributions of the above form to linear functions over the features. The approximation ratio exhibited by such an estimator is:

$$\alpha^{\hat{v}}_{\|\cdot\|}(\mathcal{M}, \mu, \varphi) = \frac{\|\hat{v}(\mathbb{Q}_{\mathcal{M},\mu,\varphi}) - v_{\mathcal{M}}\|}{\inf_{\theta} \|\Phi\theta - v_{\mathcal{M}}\|}, \tag{18}$$

with the convention that $\frac{0}{0} = 1$ and $\frac{x}{0} = \infty$ whenever $x > 0$.[9] We consider two natural choices for the norms, the weighted $L_2(\mu)$ norm and the L_∞ norm (recall that μ is the offline state distribution). For want of space, here we only the results for the $L_2(\mu)$ norm. In summary, our question is:

Question 6.1. *What is the optimal asymptotic approximation factor for linear off-policy value function estimation under misspecification?*

Overview of contributions We provide both upper and lower bound results, with the goal of pinning down the optimal approximation ratio for this problem. For upper bounds, we analyze the well-known (off-policy) Least Squares Temporal Difference (LSTD) algorithm [50], and provide exact characterizations of its error compared to the optimal linear projection. Let us write D for the diagonal matrix with the entries of μ along its diagonal (i.e. $D_{s,s} = \mu(s)$, for $s \in S$, and 0 otherwise), $\Sigma \coloneqq \Phi^\top D \Phi = \mathbb{E}_\mu[\varphi(s)\varphi(s)^\top]$ for the second-moment matrix, and $\Pi_\mu = \Phi\Sigma^{-1}\Phi^\top D$ for the $L_2(\mu)$ projection operator. At the population level, the LSTD estimator θ_{LSTD} is defined as:

$$A \coloneqq \Phi^\top D(I - \gamma P)\Phi = \mathbb{E}_{s,s'}\left[\varphi(s)(\varphi(s) - \gamma\varphi(s'))^\top\right] \tag{19}$$

$$b \coloneqq \Phi^\top D r = \mathbb{E}_{s\sim\mu}\left[\varphi(s) r(s)\right] \tag{20}$$

$$\theta_{\text{LSTD}} \coloneqq A^{-1} b\,, \qquad v_{\text{LSTD}} = \Phi\theta_{\text{LSTD}}\,, \tag{21}$$

whenever A is invertible. Our result is a tight instance-dependent approximation factor for LSTD.

[9] We do not need to consider random asymptotic estimators since Jensen's inequality tells us that deterministic estimators are optimal.

Theorem 6.2. *Assume that the A matrix from Equation (19) is invertible. Then the population LSTD estimator of Equation (21) has an approximation factor upper bound of*

$$\alpha_\mu^{LSTD} \leq \sqrt{1 + \left(\gamma \|\Phi A^{-1} \Phi^\top D P\|_\mu\right)^2} \leq \sqrt{1 + \left(\gamma \frac{\|\Pi_\mu P\|_\mu}{\sigma_{\min}(\Sigma^{-1/2} A \Sigma^{-1/2})}\right)^2} \tag{22}$$

The bound involves two problem-dependent terms ($\|\Pi_\mu P\|_\mu$ and $\sigma_{\min}(\Sigma^{-1/2} A \Sigma^{-1/2})$), which gives two "failure modes" where this estimator can have large approximation ratios. In the aliased case, we can give an instance-dependent lower bound to show that the approximation factor of LSTD is roughly optimal.

Theorem 6.3. *In the aliased setting, $\forall x \in [1, \infty], \forall y \in (0, \frac{1}{2})$, there exists a collection of two instances $\mathbb{M} = \{(\mathcal{M}_1, \mu_1, \varphi_1), (\mathcal{M}_2, \mu_2, \varphi_2)\}$ which both satisfy $\|\Pi_\mu P\|_\mu = x$ and $\sigma_{\min}(\Sigma^{-1/2} A \Sigma^{-1/2}) = y$ and generate the same data distribution $\mathbb{Q}$, yet any estimator $\hat{v}$ will satisfy*

$$\sup_{(\mathcal{M}, \mu, \varphi) \in \mathbb{M}} \alpha_\mu^{\hat{v}}(\mathcal{M}, \mu, \varphi) \geq \sqrt{1 + \gamma^2 \frac{\|\Pi_\mu P\|_\mu^2 - 1}{\sigma_{\min}^2(\Sigma^{-1/2} A \Sigma^{-1/2})}} \tag{23}$$

When $x > \sqrt{2}$, then the upper bound (Eq. (22)) and the lower bound (Eq. (23)) differ by at most a multiplicative factor of 2. Thus, in this regime of the instance-dependent parameters, LSTD attains the asymptotically optimal approximation ratio up to constant factors. Our domain restrictions on x and y in the lower bound statement also do not preclude the interesting asymptotics of the problem, i.e. the cases where $\|\Pi_\mu P\|_\mu$ is large ($\to \infty$) or $\sigma_{\min}(\Sigma^{-1/2} A \Sigma^{-1/2})$ is small ($\to 0$).

The previous result heavily relies on the aliased nature of the problem. In the non-aliased case, the learner can still use the LSTD algorithm, so the upper bound of Theorem 6.2 still holds. For the lower bounds, the class of learners that we are competing against now have more information. We conjecture that the bound in Equation (22) remains optimal, but this remains open. We instead show the weaker results that both of our instance-dependent factors appearing in Equation (22) are independently necessary, meaning that the finiteness of one alone does not guarantee a finite approximation ratio.

Lemma 6.4 ($\|\Pi_\mu P\|_\mu$ is necessary)**.** *In the non-aliased setting, there exists a family of instances $\mathbb{M} = \{(\mathcal{M}, \mu, \varphi)\}$ which all have an $L_2(\mu)$-misspecification of 0, $\sigma_{\min}(\Sigma^{-1/2} A \Sigma^{-1/2}) > 0$, and $\|\Pi P\|_\mu = \infty$, yet any estimator $\hat{v}$ will satisfy*

$$\sup_{(\mathcal{M}, \mu, \varphi) \in \mathbb{M}} \alpha_\mu^{\hat{v}}(\mathcal{M}, \mu, \varphi) = \infty$$

This example illustrates the interpretation that $\|\Pi_\mu P\|_\mu$ intuitively captures the main source hardness in value function estimation. Namely, it is large (or infinite) when there is a lack of "pushforward" coverage [51], meaning that a state $s \in \text{supp}(\mu)$ may transition to a state $s' \notin \text{supp}(\mu)$. Since the value at s depends on the value at s', we may not be able to predict $v_\mathcal{M}(s)$ even under realizability. Our next result shows that, surprisingly, this is not the only source of hardness in the off-policy value estimation problem.

Lemma 6.5 ($\sigma_{\min}(\Sigma^{-1/2}A\Sigma^{-1/2})$ is necessary)**.** *In the non-aliased setting, there exists a family of instances $\{(M, \mu, \varphi)\}$ which all have an $L_2(\mu)$-misspecification of 0, $\|\Pi_\mu P\|_\mu < \infty$, and $\sigma_{\min}(\Sigma^{-1/2}A\Sigma^{-1/2}) = 0$, yet any estimator $\hat{v}$ will satisfy*

$$\sup_{(\mathcal{M},\mu,\varphi)\in\mathbb{M}} \alpha_\mu^{\hat{v}}(\mathcal{M}, \mu, \varphi) = \infty$$

While it may appear surprising that the invertibility of some LSTD-specific quantity (the A matrix) can dictate the hardness of value function estimation for *all* estimators, the intuition is that $A = 0$ implies that the linear subspace $\{v_\theta = \Phi\theta\}_\theta$ can live completely inside of the space of plausible value functions that can be chosen by the environment. In the general case where A is nonzero, its minimum singular value dictates the "angle" between these subspaces, and a small angle indicates a large approximation error (see Figure 2). In conclusion, we have shown that $\|\Pi_\mu P\| < \infty$ and $\sigma_{\min}(\Sigma^{-1/2}A\Sigma^{-1/2}) > 0$ are *both* independently necessary for finite approximation factors in value function estimation under the $L_2(\mu)$ norm. The paper also derives analogous results for optimal approximation factors in the L_∞ norm.

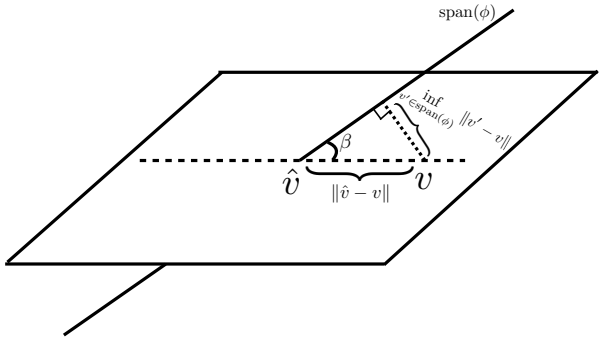

Figure 2: Illustration of lemma 6.5. Plane: possible value functions that can be chosen without giving information to the learner. Line: possible linear predictors ($d = 1$). True value function: v. Best estimator: $\hat{v}$. The angle β controls the approximation ratio, which is $\alpha = \|\hat{v} - v\|/\|v' - v\|$. In our construction, β is controlled by the magnitude of A, and $\beta = 0$ (line in the plane) implies $\alpha = \infty$.

Related works The only prior work studying the optimality approximation factors for this problem showed that LSTD was optimal in the setting where μ is the stationary distribution of P (the *on-policy* setting) and for sample sizes smaller than the size of the state space [52].[10] In particular, no prior work exists on characterizing the necessary blowup of the misspecification error in the off-policy case, even that which is asymptotically achievable. Recent works [53, 54] have provided some negative results for this problem in the realizable setting which demonstrate that the optimal linear approximation may not be recovered in the worst-case (in our language, this corresponds to approximation factors of $+\infty$). Our results explain these hardness results as well as providing new ones.

[10] In the on-policy setting, it is possible to asymptotically a ratio of 1 under the $L_2(\mu)$ norm.

In terms of upper bounds to the approximation ratio, Tsitsiklis and Van Roy [55] give the classical approximation ratio bound of $1/\sqrt{1-\gamma^2}$ for the *on-policy* case, which uses the fact that P (and thus $\Pi_\mu P$) are contractive in the $L_2(\mu)$ norm when μ is the stationary distribution. The bound of $1/\sqrt{1-\gamma^2}$ is sharpened in an instance-dependent fashion by Yu and Bertsekas [56] and Mou et al. [52], which both consider the more general problem of solving projected fixed point equations. The above approximation bounds are similar to our Theorem 6.2, although our proof relies on a simpler and exact error decomposition. Our proof also enables us to obtain L_∞ bounds, whereas all prior bounds are in the $L_2(\mu)$ norm.

7 Beyond Optimality: Optimally Suboptimal Bandits [Proposed]

Our first proposed work will study provable methods in settings where optimality is prohibitive. There are no shortage of such settings (we have already seen some in Part II). With nonlinear function approximation schemes, such as neural networks, the situation is even more dire. In particular, a relatively simple construction by Dong et al. shows that even for bandits which are realizable by a single ReLU function, any algorithm will require $\Omega(\min\{A, \exp(d)\})$ samples to return a good action. This hardness is intrinsically tied to the fact that in bandits and in RL we are tasked with recovering the optimal arm/policy. However, realizability within even just a single ReLU node allows the features to be essentially uninformative for detecting the location of the optimal arm, thus requiring the learner to essentially try all of the arms (A) or try all of the functions ($\exp(d)$, after discretization).

We want to develop a theory for such cases which does not require "assuming the problem away". We will do so by studying if it is possible to meaningfully re-define the standard objectives. The guiding intuition is that the quality of our exploration algorithm should be measured against that of the best other exploration algorithm which could've been used. To formalize this, it is more practical to think in terms of regret. Traditionally, the T-step regret of an algorithm in MDP $\mathcal{M}$ is defined as

$$\texttt{Reg}(\mathcal{M}, T) = Tv^\star(\mu_0) - \mathbb{E}[\sum_{t=1}^{T} v^{\pi_t}(\mu_0)], \tag{24}$$

where π_t is the sequence of policies played by the algorithm. However, in many practical scenarios, optimality is not achievable with a realistic number of samples, which means that the regret will essentially be linear in T (the worst possible) unless T exceeds some unrealistic range (e.g. greater than all state-action pairs). Despite that all algorithms experience linear regret, we should not consider them equal.

Our proposal is to view the first term above ($Tv^\star(\mu_0)$) as the total reward obtained by a "competing" algorithm which has more information about $\mathcal{M}$ than we do, namely the identity of the optimal policy. The optimal competing algorithm which knows the identity of the optimal policy is clearly to play this policy at every round. With this perspective, we can relax the definition of regret by restricting the amount of information given to the

competing optimal algorithm. Loosely, this weaker notion can be defined as

$$\underline{\texttt{Reg}}(\mathcal{M}, T) = \sup_{\mathbb{A} \in \Pi^\star(\mathcal{M})} \mathbb{E}\left[\sum_{t=1}^{T} \left(v^{\pi_t^{\mathbb{A}}}(\mu_0) - v^{\pi_t}(\mu_0)\right)\right], \tag{25}$$

where $\Pi^\star(\mathcal{M})$ is the set of algorithms $\mathbb{A}$ (the "adversaries") which are given certain additional information about $\mathcal{M}$, and $\pi_t^{\mathbb{A}}$ is the set of policies played by this algorithm. This definition is easily seen to generalize the previous one.

We instantiate this idea by examining the simple case of (tabular) stochastic k-armed bandits. Evidently, no algorithm can achieve sublinear regret (as traditionally defined) on a bandit unless T is larger than k. We propose to study the case where the number of arms is much larger than the available number of pulls, even taking the limit of $k \to \infty$. Existing results for many-armed bandits measure regret against the optimal arm, and thus are forced to make structural assumptions to achieve sub-linear regret [58, 59, 60]. We seek to make *no assumptions* on the bandits, but instead to find adversaries which one can reasonably compete against. A natural proposal for the case of the tabular k-armed bandit is to consider the adversaries which know the distributions of all the arms but are given an unknown permutation of the arms. In the limit of infinite arms, this is equivalent to assuming that each new arms arm is sampled i.i.d. from some *reservoir distribution of arms*. We exploited this perspective to write (25) as an *optimal stopping problem* with unknown distributions. This problem is similar yet notably distinct from other stopping time problems such as the prophet inequality [61]. Our results so far have established that this adversary may be too strong still, namely we have shown that the worst-case regret can still be linear. We then showed that this can be solved by considering additional instance-dependent terms related to the reservoir distribution, but the tightness of these terms remains to be seen. We are also investigating possibilities for an even weaker class of adversaries.

Overall, we hope that this perspective opens the way to thinking about several other exciting settings, such as the ReLU bandit model mentioned earlier, or even the simpler case of *misspecified* linear bandits where negative results still exist [35].

8 Leveraging Connections Between Online RL and Offline RL [Proposed]

In this section, we will be considering the much more general setting of *arbitrary* function classes $\mathcal{F}$. When the function class lack useful structure such as linearity, it is easy to construct instances that exhibit a lower bound of $\min\{|\mathcal{F}|, A^H\}$ [34]. In recent years, a line of work [29, 30, 27, 31] has established several *complexity measures* for the hardness of RL with arbitrary function classes. The complexity measures can be viewed as instance-dependent terms (depending on the MDP and the function class) which capture the hardness of that problem. These works provide sample complexity bounds which scale in the canonical parameters ($\log|\mathcal{F}|$, H, and $1/\varepsilon$) as well as in the complexity measure.

In a parallel line of work from offline RL, it has been recognized that a certain no-

tion of *coverage* is required from the offline data distribution. In its basic form, this condition roughly asserts that the data distribution covers all states which could be visited by policies in the MDP. More formally, the *concentrability* coefficient is defined as $C_{\text{conc}}(\mu) \coloneqq \sup_\pi \left\| \frac{d^\pi}{\mu} \right\|_\infty$, where d^π is the state visitation distribution under π. This quantity plays a fundamental role for sample complexity analyses in offline RL [47, 62].

The recent work of Xie et al. [63] merged these two lines of work by establishing that the mere *existence* of a data distribution with good concentrability is in itself a structural condition that enables sample-efficient learning in online RL. Their works assumes realizability as well as a much stronger *completeness* condition, which assumes that $\mathcal{T} f \in \mathcal{F}$ for all $f \in \mathcal{F}$. Their sample complexity scales with the best concentrability coefficient amongst all data distributions, $C_{\text{cov}} \coloneqq \inf_\mu C_{\text{conc}}(\mu)$. Interestingly, the learner does not know or directly attempt to find this data distribution.

We seek to improve the results of this paper, by asking whether a similar result holds without completeness, namely only with realizability and the existence of a covering distribution. Our preliminary results so far indicate that a positive result can be obtained if we assume stronger realizability conditions which are also common in the offline RL literature, namely that of value function realizability AND weight function realizability [64]. We are still investigating whether the weaker value function realizability on its own is sufficient (we suspect that it is not).

References

[1] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[2] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836): 77–82, 2020.

[3] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[4] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[5] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019.

[6] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[7] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

[8] Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 578–598. PMLR, 16–19 Mar 2021. URL https://proceedings.mlr.press/v132/domingues21a.html.

[9] John Tromp and Gunnar Farnebäck. Combinatorics of go. In H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers, editors, *Computers and Games*, pages 84–99, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-75538-8.

[10] 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.

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

[12] 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.

[13] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

[14] Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

[15] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

[16] Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019.

[17] 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. PMLR, 2020.

[18] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020.

[19] 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.

[20] 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, 2019.

[21] Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6995–7004. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yang19b.html.

[22] 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.

[23] 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.

[24] 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*, 2019.

[25] Roshan Shariff and Csaba Szepesvári. Efficient planning in large MDPs with weak linear function approximation. *Neural Information Processing Systems*, 2020.

[26] Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Limiting extrapolation in linear approximate value iteration. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. *arXiv preprint arXiv:2103.10897*, 2021.

[28] Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

[29] 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.

[30] Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 34, 2021.

[31] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

[32] Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021.

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

[34] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 2019.

[35] 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.

[36] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

[37] Yuanhao Wang, Ruosong Wang, and Sham Kakade. An exponential lower bound for linearly realizable mdp with constant suboptimality gap. *Advances in Neural Information Processing Systems*, 34, 2021.

[38] Gellért Weisz, Csaba Szepesvári, and András György. Tensorplan and the few actions lower bound for planning in mdps under linear realizability of optimal value functions. *arXiv preprint arXiv:2110.02195*, 2021.

[39] Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvári. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Conference on Learning Theory*. PMLR, 2021.

[40] Gellért Weisz, Csaba Szepesvári, and András György. Tensorplan and the few actions lower bound for planning in mdps under linear realizability of optimal value functions. In *International Conference on Algorithmic Learning Theory*, pages 1097–1137. PMLR, 2022.

[41] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 2256–2264, 2013.

[42] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[43] Stephane Ross. *Interactive learning for sequential decisions and predictions*. PhD thesis, Carnegie Mellon University, 2013.

[44] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[45] Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pages 3309–3318. PMLR, 2017.

[46] Nived Rajaraman, Yanjun Han, Lin Yang, Jingbo Liu, Jiantao Jiao, and Kannan Ramchandran. On the value of interaction and function approximation in imitation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[47] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

[48] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[49] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[50] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

[51] Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021.

[52] Wenlong Mou, Ashwin Pananjady, and Martin J. Wainwright. Optimal oracle inequalities for solving projected fixed-point equations. *arXiv:2012.05299 [cs, math, stat]*, December 2020. URL http://arxiv.org/abs/2012.05299. arXiv: 2012.05299.

[53] Philip Amortila, Nan Jiang, and Tengyang Xie. A variant of the wang-foster-kakade lower bound for the discounted setting. *arXiv preprint arXiv:2011.01075*, 2020.

[54] Juan C. Perdomo, Akshay Krishnamurthy, Peter Bartlett, and Sham Kakade. A Sharp Characterization of Linear Estimators for Offline Policy Evaluation. *arXiv:2203.04236 [cs, stat]*, March 2022. URL http://arxiv.org/abs/2203.04236. arXiv: 2203.04236.

[55] J.N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, May 1997. ISSN 00189286. doi: 10.1109/9.580874. URL http://ieeexplore.ieee.org/document/580874/.

[56] Huizhen Yu and Dimitri P. Bertsekas. Error Bounds for Approximations from Projected Linear Equations. *Mathematics of Operations Research*, 35(2):306–329, May 2010. ISSN 0364-765X, 1526-5471. doi: 10.1287/moor.1100.0441. URL http://pubsonline.informs.org/doi/abs/10.1287/moor.1100.0441.

[57] Kefan Dong, Jiaqi Yang, and Tengyu Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *Advances in Neural Information Processing Systems*, 34:26168–26182, 2021.

[58] Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, 25(5):2103–2116, 1997.

[59] Yizao Wang, Jean-yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/file/49ae49a23f67c759bf4fc791ba842aa2-Paper.pdf.

[60] Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *International Conference on Machine Learning*, pages 1133–1141. PMLR, 2015.

[61] Brendan Lucier. An economic view of prophet inequalities. *SIGecom Exch.*, 16(1):24–47, sep 2017. doi: 10.1145/3144722.3144725. URL https://doi.org/10.1145/3144722.3144725.

[62] Tengyang Xie and Nan Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559. PMLR, 2020.

[63] Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. *arXiv preprint arXiv:2210.04157*, 2022.

[64] Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.