When is Agnostic Reinforcement Learning Statistically Tractable?

Anonymous Authors[1]

Abstract

We study the problem of agnostic PAC reinforcement learning (RL): given a policy class Π, how many rounds of interaction with an unknown MDP (with a potentially large state and action space) are required to learn an ε-suboptimal policy with respect to Π? Towards that end, we introduce a new complexity measure, called the *spanning capacity*, that depends solely on the set Π and is independent of the MDP dynamics. With a generative model, we show that the spanning capacity characterizes PAC learnability for every policy class Π. However, for online RL, the situation is more subtle. We show there exists a policy class Π with a bounded spanning capacity that requires a superpolynomial number of samples to learn. This reveals a surprising separation for agnostic learnability between generative access and online access models (as well as between deterministic/stochastic MDPs under online access). On the positive side, we identify an additional *sunflower* structure which in conjunction with bounded spanning capacity enables statistically efficient online RL via a new algorithm called POPLER, which takes inspiration from classical importance sampling methods as well as recent developments for reachable-state identification and policy evaluation in reward-free exploration.

1. Introduction

Reinforcement Learning (RL) has emerged as a powerful paradigm for solving complex decision-making problems, demonstrating impressive empirical successes in a wide array of challenging tasks, from achieving superhuman performance in the game of Go (Silver et al., 2017) to solving intricate robotic manipulation tasks (Lillicrap et al., 2016; Akkaya et al., 2019; Ji et al., 2023). Many practical domains in RL often involve rich observations such as images, text, or audio (Mnih et al., 2015; Li et al., 2016; Ouyang et al., 2022). Since these state spaces can be vast and complex, traditional tabular RL approaches (Kearns and Singh, 2002; Brafman and Tennenholtz, 2002; Azar et al., 2017; Jin et al., 2018) cannot scale. This has led to a need to develop provable and efficient approaches for RL that utilize *function approximation* to extrapolate rich, high-dimensional observations to unknown states/actions.

The goal of this paper is to study the sample complexity of policy-based RL, which is arguably the simplest setting for RL with function approximation (Kearns et al., 1999; Kakade, 2003). In policy-based RL, an abstract function class Π of *policies* (mappings from states to actions) is given to the learner. For example, Π can be the set of all the policies represented by a certain deep neural network architecture. The objective of the learner is to interact with an unknown MDP to find a policy $\widehat{\pi}$ that competes with the best policy in the class, i.e., for some small ε, the policy $\widehat{\pi}$ satisfies

$$V^{\widehat{\pi}} \geq \max_{\pi \in \Pi} V^{\pi} - \varepsilon, \tag{1}$$

where V^{π} denotes the value of policy π on the underlying MDP. We henceforth call Eq. (1) the "agnostic PAC reinforcement learning" objective. Our paper addresses the following research question:

What structural assumptions on Π enable statistically efficient agnostic PAC RL?

Characterizing (agnostic) learnability for various problem settings is perhaps the most fundamental question in statistical learning theory. For the simpler setting of supervised learning (which is RL with binary actions, horizon 1, and binary rewards), the story is complete: a hypothesis class Π is agnostically learnable iff its VC dimension is bounded (Vapnik and Chervonenkis, 1971; 1974; Blumer et al., 1989; Ehrenfeucht et al., 1989), and the ERM algorithm—which returns the hypothesis with the smallest training loss—is

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML). Do not distribute.

statistically optimal. However, RL (with $H > 1$) is significantly more challenging, and we are still far from a rigorous understanding of when agnostic RL is tractable, or what algorithms to use in large-scale RL problems.

While significant effort has been invested over the past decade in both theory and practice to develop algorithms that utilize function approximation, existing theoretical guarantees require additional assumptions on the MDP. The most commonly adopted assumption is *realizability*: the learner can precisely model the value function or the dynamics of the underlying MDP (see, e.g., Russo and Van Roy, 2013; Jiang et al., 2017; Sun et al., 2019; Wang et al., 2020a; Du et al., 2021; Jin et al., 2021a; Foster et al., 2021a). Unfortunately, realizability is a fragile assumption that rarely holds in practice. Moreover, even mild misspecification can cause catastrophic breakdown of theoretical guarantees (Du et al., 2019a; Lattimore et al., 2020). Furthermore, in various applications, the optimal policy $\pi^\star := \arg\max_{\pi\in\Pi} V^\pi$ may have a succinct representation, but the optimal value function $V^\star$ can be highly complex, rendering accurate approximation of dynamics/value functions infeasible without substantial domain knowledge (Dong et al., 2020). Thus, we desire algorithms for agnostic RL that can work with *no modeling assumptions on the underlying MDP*. On the other hand, it is also well known without any assumptions on Π, when Π is large and the MDP has a large state and action space, agnostic RL may be intractable with sample complexity scaling exponentially in the horizon (Agarwal et al., 2019). Thus, some structural assumption on Π is needed, and towards that end, the goal of our paper is to understand what assumptions are sufficient or necessary for statistically efficient agnostic RL, and to develop provable algorithms for learning. Our main contributions are:

- We introduce a new complexity measure called the *spanning capacity*, which solely depends on the policy class Π and is independent of the underlying MDP. We illustrate the spanning capacity with examples, and show why it is a natural complexity measure for agnostic PAC RL (Section 3).
- We show that the spanning capacity is both necessary and sufficient for agnostic PAC RL with a generative model, with upper and lower bounds matching up to $\log|\Pi|$ and $\mathrm{poly}(H)$ factors (Section 4).
- Moving to the online setting, we first show that the spanning capacity by itself is *insufficient* for agnostic PAC RL by proving a superpolynomial lower bound on the sample complexity required to learn a specific Π, thus demonstrating a separation between generative and online interaction models for PAC RL (Section 5).
- Given the above lower bound, we propose an additional property of the policy class called *sunflower* structure, that allows for efficient exploration, and is satisfied by many policy classes of interest. We provide an agnostic PAC RL algorithm called POPLER that is statistically efficient whenever the given policy class has bounded spanning capacity and has the *sunflower* structure. POPLER unifies the existing approaches of importance sampling and reward-free exploration in tabular RL algorithms, particularly the approach of identifying highly-reachable states (Section 6).

2. Setup and Motivation

2.1. Reinforcement Learning Preliminaries

We formally introduce the setup for reinforcement learning in a finite horizon Markov decision process (MDP). Denote the MDP as $M = (\mathcal{S}, \mathcal{A}, H, P, R, \mu)$, which consists of a state space $\mathcal{S}$, action space $\mathcal{A}$, horizon H, probability transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$, and initial distribution $\mu \in \Delta(\mathcal{S})$. For ease of exposition, we assume that $\mathcal{S}$ and $\mathcal{A}$ are finite (but possibly large) with cardinality S and A respectively. We assume a layered state space, i.e., $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_H$ where $\mathcal{S}_i \cap \mathcal{S}_j = \varnothing$ for all $i \neq j$. Thus, given a state $s \in \mathcal{S}$, it can be inferred which $\mathcal{S}_h$, or time step in the MDP, it belongs to. We denote a trajectory $\tau = (s_1, a_1, r_1, \ldots, s_H, a_H, r_H)$, where at each step $h \in [H]$, an action $a_h \in \mathcal{A}$ is played, a reward r_h is drawn independently from the distribution $R(s_h, a_h)$, and each subsequent state s_{h+1} is drawn from $P(\cdot|s_h, a_h)$. Lastly, we assume that the cumulative reward of any trajectory is bounded by 1.

Policy-based reinforcement learning. In our setting, the learner is given a policy class $\Pi \subseteq \mathcal{A}^{\mathcal{S}}$. For any policy $\pi \in \mathcal{A}^{\mathcal{S}}$, we denote $\pi(s)$ as the action that π takes when presented state s. We use $\mathbb{E}^\pi[\cdot]$ and $\Pr^\pi[\cdot]$ to denote the expectation and probability under the process of a trajectory drawn from the MDP M by policy π. Also, for any $h, h' \leq H$, we say that a partial trajectory $\tau = (s_h, a_h, s_{h+1}, a_{h+1}, \ldots, s_{h'}, a_{h'})$ is consistent with π if for all $h \leq i \leq h'$, we have $\pi(s_i) = a_i$. We use the notation $\pi \rightsquigarrow \tau$ to denote that τ is consistent with π.

We also define the value function and Q-function such that for any π, and s, a,

$$V_h^\pi(s) = \mathbb{E}^\pi\left[\sum_{h'=h}^{H} R(s_{h'}, a_{h'}) \mid s_{h'} = s\right],$$

$$Q_h^\pi(s, a) = \mathbb{E}^\pi\left[\sum_{h'=h}^{H} R(s_{h'}, a_{h'}) \mid s_{h'} = s, a_{h'} = a\right].$$

We often denote $V^\pi := \mathbb{E}_{s_1\sim\mu} V_1^\pi(s_1)$. For any policy $\pi \in \mathcal{A}^{\mathcal{S}}$, we also define the *occupancy measure* as $d_h^\pi(s, a) := \mathbb{P}^\pi[s_h = s, a_h = a]$ and $d_h^\pi(s) := \mathbb{P}^\pi[s_h = s]$.

Models of interaction. We consider two standard models of interaction in RL.

- **Generative model.** The learner can make a query to a simulator at any (s, a), and observe a sample (s', r) drawn as $s' \sim P(\cdot|s, a)$ and $r \sim R(s, a)$.
- **Online interaction model.** The learner can submit a (potentially non-Markovian) policy $\widetilde{\pi}$ and receive back a trajectory sampled by running $\widetilde{\pi}$ on the MDP. Since online saccess can be simulated via generative access, learning under online access is only more challenging than learning under generative access (up to a factor of H). We colloquially refer to this as "online RL".

We define $\mathcal{M}^{\mathsf{sto}}$ as the set of all MDPs of horizon H. Similarly, we define $\mathcal{M}^{\mathsf{detP}} \subset \mathcal{M}^{\mathsf{sto}}$, and $\mathcal{M}^{\mathsf{det}} \subset \mathcal{M}^{\mathsf{detP}}$ to denote the set of all MDPs with deterministic transitions but stochastic rewards, and of all MDPs with both deterministic transitions and deterministic rewards, respectively.

2.2. Agnostic PAC RL

Our goal is to understand the sample complexity of agnostic PAC RL. An algorithm $\mathbb{A}$ is an (ε, δ)-PAC RL algorithm for an MDP M, if after interacting with M (either in the generative model or online RL), $\mathbb{A}$ returns a policy $\widehat{\pi}$ that satisfies the guarantee[1]

$$V^{\widehat{\pi}} \geq \max_{\pi \in \Pi} V^{\pi} - \varepsilon,$$

with probability at least $1 - \delta$. We say that $\mathbb{A}$ has sample complexity $n^{\mathbb{A}}_{\mathsf{on}}(\Pi; \varepsilon, \delta)$ (resp. $n^{\mathbb{A}}_{\mathsf{gen}}(\Pi; \varepsilon, \delta)$) if for every MDP M, $\mathbb{A}$ is an (ε, δ)-PAC RL algorithm and collects at most $n_{\mathsf{on}}(\mathbb{A}, \Pi; \varepsilon, \delta)$ many trajectories in the online interaction model (resp. generative model) in order to return $\widehat{\pi}$.

We define the *minimax sample complexity* for agnostically learning Π as the minimum sample complexity for any (ε, δ) PAC algorithm:

$$n_{\mathsf{on}}(\Pi; \varepsilon, \delta) := \inf_{\mathbb{A}} n^{\mathbb{A}}_{\mathsf{on}}(\Pi; \varepsilon, \delta), \quad \text{and}$$
$$n_{\mathsf{gen}}(\Pi; \varepsilon, \delta) := \inf_{\mathbb{A}} n^{\mathbb{A}}_{\mathsf{gen}}(\Pi; \varepsilon, \delta).$$

Known results in agnostic RL. We first note that following classical result which shows that agnostic PAC RL is statistically intractable, in the worst case.

[1]Our results are agnostic in the sense that we do not make the assumption that the optimal policy for the underlying MDP is in Π, but instead, only wish to complete with the best policy in Π. Additionally, recall that we do not assume that the learner has a value function class or a model class that captures the optimal value functions or dynamics.

Proposition 1 (No Free Lunch for RL (Kakade, 2003; Krishnamurthy et al., 2016))**.** *There exists a policy class Π for which the minimax sample complexity under a generative model is at least $n_{\mathsf{gen}}(\Pi; \varepsilon, \delta) = \Omega(\min\{A^H \log|\Pi|, |\Pi|, SA\}/\varepsilon^2)$.*

Since online RL is only harder than learning with a generative model, the lower bound in Proposition 1 extends to the online RL. Proposition 1 is the analogue of the classical *No Free Lunch* results in statistical learning theory (Shalev-Shwartz and Ben-David, 2014); it indicates that without placing further assumptions on the MDP or the policy class Π (e.g., by constraining the state/action space sizes, policy class size, or the horizon), sample efficient agnostic PAC RL is impossible.

Indeed, an almost matching upper bound of $n_{\mathsf{on}}(\Pi; \varepsilon, \delta) = \widetilde{\mathcal{O}}(\min\{A^H, |\Pi|, HSA\}/\varepsilon^2)$ is quite easy to obtain. The $|\Pi|/\varepsilon^2$ guarantee can simply be obtained by iterating over $\pi \in \Pi$, collecting $\widetilde{\mathcal{O}}(1/\varepsilon^2)$ trajectories per policy, and then picking the one with highest empirical value. The HSA/ε^2 guarantee can be obtained by running known algorithms for tabular RL (Zhang et al., 2021a). Finally, the A^H/ε^2 guarantee is achieved by the classical importance sampling (IS) algorithm (Kearns et al., 1999; Agarwal et al., 2019). Since Importance Sampling will be an important technique that we repeatedly use and build upon in this paper, we give a formal description of the algorithm below:

- Collect $n = \mathcal{O}(A^H \cdot \log|\Pi|/\varepsilon^2)$ trajectories by executing actions $(a_1, \dots, a_H) \sim \mathrm{Unif}(\mathcal{A}^H)$.
- Return $\widehat{\pi} = \arg\max_{\pi \in \Pi} \widehat{v}^{\pi}_{\mathrm{IS}}$, where $\widehat{v}^{\pi}_{\mathrm{IS}} := \frac{A^H}{n} \sum_{i=1}^{n} \mathbb{1}\{\pi \rightsquigarrow \tau^{(i)}\}(\sum_{h=1}^{H} r^{(i)}_h)$.

For every $\pi \in \Pi$, the quantity $\widehat{v}^{\pi}_{\mathrm{IS}}$ is an unbiased estimate of V^{π} with variance A^H; the sample complexity result follows by standard concentration guarantees (see, e.g., Agarwal et al., 2019).

Towards structural assumptions for statistically efficient agnostic PAC RL. Of course, No Free Lunch results do not necessarily spell doom—for example in supervised learning, various structural assumptions have been studied that enable statistically efficient learning. Furthermore, there has been a substantial effort in developing complexity measures like VC dimension, fat-shattering dimension, covering numbers, etc. that characterize agnostic PAC learnability under different scenarios (Shalev-Shwartz and Ben-David, 2014). In this paper, we initiate a similar study for Agnostic learning in RL. The key question that we are interested in understanding is whether there exists a complexity measure $\mathfrak{C}(\Pi)$ which characterizes learnability for every policy class Π. Formally, can we establish that the minimax sample complexity of learning any Π is[2]

$$n_{\mathsf{on}}(\Pi;\varepsilon,\delta) = \widetilde{\Theta}\Big(\mathrm{poly}\Big(\mathfrak{C}(\Pi), H, \log|\Pi|, \varepsilon^{-1}, \log\delta^{-1}\Big)\Big)?$$

Do we even need a new complexity measure? In light of Proposition 1, one obvious candidate is $\widetilde{\mathfrak{C}}(\Pi) = \min\{A^H, |\Pi|, SA\}$. While $\widetilde{\mathfrak{C}}(\Pi)$ is definitely sufficient to upper bound the minimax sample complexity for any policy class Π, it is not clear if it is also necessary. In fact, our next proposition suggests that $\widetilde{\mathfrak{C}}(\Pi)$ is indeed not the right measure of complexity by giving example of a policy class for which $\mathfrak{C}(\Pi) := \min\{A^H, |\Pi|, SA\}$ can be exponentially larger than the minimax sample complexity for agnostic learning w.r.t. that policy class, even when ε is constant.

Proposition 2. *Let $H \in \mathbb{N}$, $\mathcal{S} = [2^H] \times [H]$, and $\mathcal{A} = \{0,1\}$. Consider the singleton policy class: $\Pi_{\mathrm{sing}} := \{\pi_i \mid \pi_i(s) = \mathbb{1}\{s = i\}\}$, where π_i takes the action i on state i, and 0 everywhere else. Then $\min\{A^H, |\Pi_{\mathrm{sing}}|, SA\} = 2^H$ but $n_{\mathsf{on}}(\Pi_{\mathrm{sing}};\varepsilon,\delta) \le \widetilde{\mathcal{O}}(H^3 \cdot \log(1/\delta)/\varepsilon^2)$.*

The upper bound on minimax sample complexity can be obtained as a corollary of our more general upper bound in Section 6. The key intuition for why Π_{sing} can be learned in $\mathrm{poly}(H)$ samples is that even though the policy class and state space are large, the set of possible trajectories obtained by running any $\pi \in \Pi_{\mathsf{sing}}$ has "low complexity". In particular, every trajectory τ has at most one $a_h = 1$. This observation enables us to employ the straightforward modification of the classical IS algorithm: draw $\mathrm{poly}(H) \cdot \log(1/\delta)/\varepsilon^2$ samples from the uniform distribution over $\Pi_{\mathrm{core}} = \{\pi_h \mid h \in [H]\}$ where the policy π_h takes the action 1 on every state at layer h and 0 everywhere else. The variance of the resulting estimator $\widehat{v}^{\pi}_{\mathrm{IS}}$ is $1/H$, so the sample complexity of this modified variant of IS has only $\mathrm{poly}(H)$ dependence by standard concentration bounds.

In the sequel, we present a new complexity measure that formalizes this intuition.

3. Spanning Capacity

The spanning capacity precisely captures the intuition that trajectories obtained by running any $\pi \in \Pi$ have "low complexity." We first define a notion of reachability: in deterministic MDP $M \in \mathcal{M}^{\mathsf{det}}$, we say (s,a) is *reachable* by $\pi \in \Pi$ if (s,a) lies on the trajectory obtained by running π on M. Roughly speaking, the spanning capacity measures "complexity" of Π as the maximum number of state-action pairs which are reachable by some $\pi \in \Pi$ in any *deterministic* MDP.

[2]Throughout the paper, we restrict ourselves to finite (but large) policy classes and assume that the $\log|\Pi|$ factors in our upper bounds are mild. Standard techniques from empirical process theory, e.g. VC dimension and covering numbers, could be imported in our proofs to improve the $\log|\Pi|$ dependence in our upper bounds when Π is continuous, but structured.

Definition 1 (spanning capacity)**.** *Fix a deterministic MDP $M \in \mathcal{M}^{\mathsf{det}}$. We define the* cumulative reachability *at layer $h \in [H]$ as denoted $C^{\mathsf{reach}}_h(\Pi; M) :=$*

$$|\{(s,a) \mid (s,a) \text{ is reachable by } \Pi \text{ at layer } h\}|.$$

We define the spanning capacity *of Π to be*

$$\mathfrak{C}(\Pi) := \max_{h\in[H]} \max_{M\in\mathcal{M}^{\mathsf{det}}} C^{\mathsf{reach}}_h(\Pi; M).$$

To build intuition, we first loogmk at some simple examples for which spanning capacity is well-behaved:

- **Contextual bandits:** Consider the standard formulation of contextual bandits (i.e., RL with $H = 1$). For any policy class Π_{cb}, since $H = 1$, the largest deterministic MDP we can construct has a single state s_1 and at most A actions available on s_1, so $\mathfrak{C}(\Pi_{\mathrm{cb}}) \le A$.
- **Tabular MDPs:** Consider tabular RL with the policy class $\Pi_{\mathrm{tab}} = \mathcal{A}^{\mathcal{S}}$. Depending on the relationship between S, A and H, we have two possible bounds on $\mathfrak{C}(\Pi_{\mathrm{tab}}) \le \min\{A^H, SA\}$. If the state space is exponentially large in H, then it is possible to construct a full A-ary "tree" such that every (s,a) pair at layer H is visited, giving us the A^H bound. However, if the state space is small, then the number of (s,a) pairs available at any layer H is trivally bounded by the total SA.
- **Small policy classes:** If the policy class Π_{small} itself is small in cardinality then we get the bound $\mathfrak{C}(\Pi_{\mathrm{small}}) \le |\Pi_{\mathrm{small}}|$, since in any deterministic MDP, in any layer each $\pi \in \Pi_{\mathrm{small}}$ can visit at most one (s,a) pair.
- **Singletons:** For the singleton class we have $\mathfrak{C}(\Pi_{\mathrm{sing}}) = H + 1$, since once we fix a deterministic MDP, there are at most H states where we can split from the trajectory taken by the policy which always plays $a = 0$, so the maximum number of (s,a) pairs reachable at layer $h \in [H]$ is $h + 1$. Observe that in light of Proposition 2, the spanning capacity is "on the right order" for Π_{sing}.

Before proceeding, we note that for any policy class Π, the spanning capacity is always bounded.

Proposition 3. *For any policy class Π, we have $\mathfrak{C}(\Pi) \le \min\{A^H, |\Pi|, SA\}$.*

Proposition 3 recovers the worst-case upper and lower bound from Section 2.2. However, for many policy classes, spanning capacity is substantially smaller than upper bound of Proposition 3. In addition to the examples we provided above, the following lists other policy classes with small spanning capacity. All proofs are deferred to Appendix B.

Below we provide more examples of policy classes where spanning capacity is substantially smaller than upper bound of Proposition 3. For these policy classes we take $\mathcal{S} = [S] \times [H]$ and $\mathcal{A} = \{0, 1\}$.

- **ℓ-tons**: is a natural generalization of singletons. We define $\Pi_{\ell-\mathrm{ton}} := \{\pi_J \mid J \subset \mathcal{S}, |J| \le \ell\}$, where the policy π_J is defined s.t. $\pi_J(s) = \mathbb{1}\{s \in J\}$ for any $s \in \mathcal{S}$. Here, $\mathfrak{C}(\Pi_{\ell-\mathrm{ton}}) = \Theta(H^\ell)$.
- **1-active policies**: We define $\Pi_{1-\mathrm{act}}$ to be the class of policies which have two possible actions on a single state in each layer, i.e., $\Pi_{1-\mathrm{act}} := \{\pi \mid \pi(s,h) = 0 \text{ if } s \ne 1\}$. Here, $\mathfrak{C}(\Pi_{1-\mathrm{act}}) = \Theta(H)$.
- **All-active policies**: We define $\Pi_{\mathrm{act}} := \bigcup_{j \ge 1} \Pi_{j-\mathrm{act}}$. Here, $\mathfrak{C}(\Pi_{\mathrm{act}}) = \Theta(H^2)$.

A natural interpretation of the spanning capacity is that it represents the largest "needle in a haystack" that can be embedded in a deterministic MDP using the policy class Π. To see this, let $(M^\star, h^\star)$ be the MDP and layer which witnesses $\mathfrak{C}(\Pi)$, and let $\{(s_i, a_i)\}_{i=1}^{\mathfrak{C}(\Pi)}$ be the set of state-action pairs reachable by Π in $M^\star$ at layer $h^\star$. Then one can hide a reward of 1 on one of these state-action pairs; since every trajectory visits a single (s_i, a_i) at layer $h^\star$, we need at least $\mathfrak{C}(\Pi)$ samples in order to discover which state-action pair has the hidden reward. Note that in this agnostic learning setup, we only need to care about the states that are reachable using Π, even though the $h^\star$ layer may have other non-reachable states and actions.

3.1. Connection to Coverability

The spanning capacity has another interpretation as the worst-case *coverability*, a structural parameter defined by (Xie et al., 2022).

Definition 2 (Coverability, Xie et al. (2022)). *For any MDP M and policy class Π, the coverability coefficient C^{cov} is denoted*

$$
\begin{aligned}
C^{\mathsf{cov}}(\Pi; M) &:= \inf_{\mu_1, \dots \mu_H \in \Delta(\mathcal{S} \times \mathcal{A})} \sup_{\pi \in \Pi, h \in [H]} \left\| \frac{d_h^\pi}{\mu_h} \right\|_\infty \\
&= \max_{h \in [H]} \sum_{s,a} \sup_{\pi \in \Pi} d_h^\pi(s,a).
\end{aligned}
$$

Coverage conditions date back to the analysis of the classic Fitted Q-Iteration (FQI) algorithm (Munos, 2007; Munos and Szepesvári, 2008), and have extensively been studied in offline RL. Various models like tabular MDPs, linear MDPs, low-rank MDPs, and exogenous MDPs satisfy the above coverage condition (Antos et al., 2008; Chen and Jiang, 2019; Jin et al., 2021b; Rashidinejad et al., 2021; Zhan et al., 2022; Xie et al., 2022), and recently, Xie et al. showed that *coverability* can be used to prove regret guarantees for online RL, albeit under the additional assumption of value function realizability.

Our notion of spanning capacity is *exactly* worst-case coverability, even taken worst case over any stochastic MDP. Thus, there always exists a deterministic MDP that witnesses worst-case coverability.

Lemma 1. *For any policy class Π, we have $\sup_{M \in \mathcal{M}^{\mathsf{sto}}} C^{\mathsf{cov}}(\Pi; M) = \mathfrak{C}(\Pi)$.*

While spanning capacity is related to the worst-case-coverability, we note that there are important differences. Firstly, coverability was used to characterize when sample efficient learning is possible in value function-based RL, where the learner has access to a realizable value function class. On the other hand, we introduce spanning capacity to characterize sample complexity in the much weaker agnostic RL setting, where learner only has access to a policy class. Note that a realizable value function class can be used to construct a policy class that contains the optimal policy, but the converse is not true. Secondly, the above equivalence holds only in a worst-case sense (over MDPs). In fact, as we show in Appendix C, coverability alone is not enough for sample efficient agnostic PAC RL.

4. Generative Model: Spanning Capacity is Necessary and Sufficient

In this section, we show that spanning capacity characterizes the minimax sample complexity of learning in the generative model.

Theorem 1 (Upper bound for generative model). *For any Π, the minimax sample complexity (ε, δ)-PAC learning Π is at most $n_{\mathsf{gen}}(\Pi; \varepsilon, \delta) \le \mathcal{O}\Big(\frac{H \cdot \mathfrak{C}(\Pi)}{\varepsilon^2} \cdot \log \frac{|\Pi|}{\delta}\Big)$.*

The proof can be found in Appendix D.1, and is a straightforward modification of the classic *trajectory tree method* from (Kearns et al., 1999): using generative access, sample $\mathcal{O}(\log |\Pi| / \varepsilon^2)$ deterministic trajectory trees from the MDP to get unbiased evaluations for every $\pi \in \Pi$; since the size of each deterministic tree is at most $H \cdot \mathfrak{C}(\Pi)$, we have a bound on the number of queries used.

Theorem 2 (Lower bound for generative model). *For any Π, the minimax sample complexity (ε, δ)-PAC learning Π is at least $n_{\mathsf{gen}}(\Pi; \varepsilon, \delta) \ge \Omega\Big(\frac{\mathfrak{C}(\Pi)}{\varepsilon^2} \cdot \log \frac{1}{\delta}\Big)$.*

The proof can be found in Appendix D.2. Intuitively, given an MDP $M^\star$ which witnesses $\mathfrak{C}(\Pi)$, one can embed a bandit instance on the relevant (s,a) pairs spanned by Π in $M^\star$. The lower bound follows by a reduction to the lower bound for (ε, δ)-PAC learning multi-armed bandits.

Together, Theorem 1 and Theorem 2 paint a relatively complete picture for the minimax sample complexity of learning any policy class Π, in the generative model, up to a $H \cdot \log|\Pi|$ factor.

Deterministic MDPs. A similar guarantee holds for online RL over deterministic MDPs.

Corollary 1. *Over the class $\mathcal{M}^{\mathsf{detP}}$ of MDPs with deterministic transitions, the minimax sample complexity of (ε,δ)-PAC learning any Π is*

$$\Omega\Big(\frac{\mathfrak{C}(\Pi)}{\varepsilon^2}\cdot\log\frac{1}{\delta}\Big) \le n_{\mathsf{on}}(\Pi;\varepsilon,\delta) \le \mathcal{O}\Big(\frac{H\cdot\mathfrak{C}(\Pi)}{\varepsilon^2}\cdot\log\frac{|\Pi|}{\delta}\Big).$$

The upper bound follows because the trajectory tree algorithm for deterministic just samples the same tree over and over again (with different stochastic rewards). The lower bound trivially extends because the lower bound of Theorem 2 actually uses an $M \in \mathcal{M}^{\mathsf{detP}}$ whose transitions are known to the learner.

5. Online RL: Spanning Capacity is Not Sufficient

Given that fact that spanning capacity characterizes the minimax sample complexity of Agnostic PAC RL in the generative model, one might be tempted to conjecture that spanning capacity is also the right characterization in online RL. The lower bound is clear since online RL is at least as hard as learning with a generative model, Theorem 2 already shows that spanning capacity is *necessary*.

In this section, we prove a surprising negative result showing that spanning capacity is not sufficient to characterize the minimax sample complexity in online RL. In particular, we provide an example for which we have a *superpolynomial* (in H) lower bound on the numbers of samples needed for learning, that is not captured by any polynomial function of spanning capacity.[3]

Theorem 3 (Lower bound for online RL)**.** *Fix any $H \ge 10^5$. Let $\varepsilon \in (1/H^{100}, 1/(100H))$ and $\ell \in \{2,\dots,\lfloor \log H\rfloor\}$.*[4] *There exists a policy class Π of size $1/(6\varepsilon^{\ell})$ with $\mathfrak{C}(\Pi) \le \mathcal{O}(H^{4\ell+2})$ and a family of MDPs $\mathcal{M}$ with state space $\mathcal{S}$ of size $H\cdot 2^{2H+1}$, binary action space, and horizon H such that: for any $(\varepsilon/16, 1/8)$-PAC algorithm, there exists an $M \in \mathcal{M}$ in which the algorithm must collect at least $\Omega(\min\{\frac{1}{\varepsilon^{\ell}}, 2^{H/3}\})$ online trajectories in expectation.*

Informally speaking, the above lower bound suggests that there exists a policy class Π for which $n_{\mathsf{on}}(\Pi;\varepsilon,\delta) = \Omega(1/\varepsilon^{\log_H \mathfrak{C}(\Pi)})$. In conjunction with the results of Section 4, Theorem 3 shows that (1) online RL is *strictly harder* than RL with generative access, and (2) online RL for stochastic MDPs is *strictly harder* than online RL for MDPs with deterministic transitions. We defer the proof of Theorem 3 to Appendix E. Our lower bound introduces several technical novelties: the family $\mathcal{M}$ utilizes a *contextual* variant of the combination lock, and the policy class Π is constructed via a careful probabilistic argument such that it is hard to explore despite having small spanning capacity.

6. Efficient Agnostic RL under Online Model

The lower bound in Theorem 3 suggests that further structural assumptions on Π are needed for statistically efficient agnostic RL under the online model. Essentially, the lower bound construction in Theorem 3 is hard to learn because any two distinct policies $\pi, \pi' \in \Pi$ can differ substantially on a large subset of states (of size at least $\varepsilon \cdot 2^{2H}$). Thus, we cannot hope to learn "in parallel" via a low variance IS strategy that utilizes extrapolation to evaluate all $\pi \in \Pi$, as we did for the singleton class.

In this sequel, we consider the following sunflower property to rule out such worst-case scenarios, and show how bounded spanning capacity and the sunflower property enable sample-efficient agnostic RL in the online model. The sunflower property only depends on the state space, action space, and policy class, and is independent of the transition dynamics and rewards of the underlying MDP.

Definition 3 (Petals and Sunflowers)**.** *For a policy π, policy set $\bar{\Pi}$, and states $\bar{\mathcal{S}} \subseteq \mathcal{S}$, π is said to be a $\bar{\mathcal{S}}$-petal on $\bar{\Pi}$ if for all $h \le h' \le H$, and partial trajectories $\tau = (s_h, a_h, \cdots, s_{h'}, a_{h'})$ that are consistent with π: either τ is also consistent with some $\pi' \in \bar{\Pi}$, or there exists $i \in (h, h']$ s.t. $s_i \in \bar{\mathcal{S}}$.*

A policy class Π is said to be a (K, D)-sunflower if there exists a set Π_{core} of Markovian policies with $|\Pi_{\mathrm{core}}| \le K$ such that for every policy $\pi \in \Pi$ there exists a set $\mathcal{S}_\pi \subseteq \mathcal{S}$, of size at most D, so that π is S_π-petal on Π_{core}.

Note that a class Π may be a (K, D)-sunflower for many different choices of K and D. Since our sample complexity upper bounds in this section scale with any valid choice of (K, D), we are free to choose K and D to minimize the corresponding sample complexity bound.

Theorem 4. *Let $\varepsilon, \delta > 0$. Suppose the policy class Π satisfies Definition 1 with spanning capacity $\mathfrak{C}(\Pi)$, and is a (K, D)-sunflower. Then, for any MDP M, with probability at least $1-\delta$,* POPLER *(Algorithm 1) succeeds in returning a policy $\widehat{\pi}$ that satisfies $V^{\widehat{\pi}} \ge \max_{\pi\in\Pi} V^{\pi} - \varepsilon$, after collecting*

$$\widetilde{\mathcal{O}}\Big(\Big(\frac{1}{\varepsilon^2} + \frac{HD^6\mathfrak{C}(\Pi)}{\varepsilon^4}\Big)\cdot K^2\log\frac{|\Pi|}{\delta}\Big) \quad \textit{online trajectories in } M.$$

[3] In the lower bound construction, the optimal policy $\pi^\star$ for the underlying MDP belongs to the set Π. This shows that realizability of the optimal policy in the policy class also does not help.

[4] We have made no attempt to optimize range of ε as well as other constants in the statement. In particular, this lower bound can be extended to work for any $\varepsilon = \Theta(1/\mathrm{poly}(H))$.

The proof of Theorem 4, and the hyperparameters needed to obtain the above bound, can be found in Appendix F. In order to get a polynomial sample complexity in Theorem 4, both $\mathfrak{C}(\Pi)$, and (K, D), are required to be $\mathrm{poly}(H, \log|\Pi|)$. All of the policy classes considered in Section 3 are (K, D)-sunflowers, with both $K, D = \mathrm{poly}(H)$, and thus our sample complexity bounds extends for all these classes; moreover for many examples we have $K = \mathrm{poly}(H)$ and $D = 0$, so we also obtain the optimal $\widetilde{\mathcal{O}}(1/\varepsilon^2)$ dependence on ε. See Appendix B for details.

In Theorem 3, we already showed that just bounded $\mathfrak{C}(\Pi)$ alone is not sufficient for polynomial sample complexity. Likewise, bounded (K, D) alone is also not sufficient for polynomial sample complexity (see Appendix F for details), and hence both assumptions are individually necessary.

Why the sunflower structure enables sample-efficient learning. Intuitively, the sunflower condition captures the intuition of simultaneous estimation of all policies $\pi \in \Pi$ via IS, and allows control of both the bias and the variance. Let π is a $\mathcal{S}_\pi$-petal on Π_{core}. Any trajectory $\tau \rightsquigarrow \pi$ that avoids $\mathcal{S}_\pi$ will be covered by the data collected using $\pi' \sim \mathrm{Unif}(\Pi_{\mathrm{core}})$. Thus, using IS with variance scaling with K, one can create a biased estimator for V^π, where the bias is *only due* to trajectories that pass through $\mathcal{S}_\pi$. If the reachability $d^\pi(s) \ll \varepsilon$ for all $s \in \mathcal{S}_\pi$, the IS estimate will have low bias (linear in $|\mathcal{S}_\pi|$). So the only issue arises if $d^\pi(s)$ is large for some $s \in \mathcal{S}_\pi$—since there are at most D of them, it is possible to explicitly control the bias that arises from trajectories passing through them.

6.1. Algorithm and Proof Ideas

POPLER takes as input a policy class Π as well as sets Π_{core} and $\{\mathcal{S}_\pi\}_{\pi\in\Pi}$ which can be computed beforehand by enumeration. The algorithm uses three subroutines, whose pseudocode are deferred to Appendix F: DataCollector, DP_Solver, and Evaluate. POPLER has two phases: a *state identification phase*, where it finds "petal" states $s \in \cup_{\pi\in\Pi}\mathcal{S}_\pi$ that are reachable with decent probability; and an *evaluation phase* where it computes estimates $\widehat{V}^\pi$ for every $\pi \in \Pi$ by constructing a Markov Reward Process (MRP) and using dynamic programming. The structure of the algorithm is reminiscent of reward-free exploration algorithms in tabular RL (e.g., Jin et al., 2020), which first identify states that are highly reachable and build a policy cover for these states, and then uses planning to estimate the values. However, our setting necessitates new technical innovations. We cannot simply enumerate over all petal states and check if they are highly-reachable by some policy $\pi \in \Pi$. Instead, we discover the petal states in a sample-efficient, *sequential* manner that interleaves IS estimates and the construction of specific tabular Markov reward processes (MRPs) to compute reachability (as well as value estimates).

Algorithm 1 **P**olicy **OP**timization by **L**earning ε-**R**eachable States (POPLER)

Require: Policy class Π, Sets Π_{core} and $\{\mathcal{S}_\pi\}_{\pi\in\Pi}$, Parameters $K, D, n_1, n_2, \varepsilon, \delta$.

```
Define start state s_⊤ (at h = 0) and end state s_⊥ (at h = H + 1).
Initialize I = {s_⊤}, T ← {(s_⊤, Null)}, and for every π ∈ Π, define S_π^+ := S_π ∪ {s_⊤, s_⊥}.
D_⊤ ← DataCollector(s_⊤, Null, Π_core, n_1)
/* Identification of Reachable States */
while Terminate = False do
    Set Terminate = True.
    for π ∈ Π do
        Compute reachable states S_π^rch = S_π^+ ∩ I, and remaining states S_π^rem = S_π \ S_π^rch.
        Estimate transition probability P̂^π = {P̂^π_{s→s'} | s ∈ S_π^rch, s' ∈ S_π^+} using (2).
        for s̄ ∈ S_π^rem do
            Estimate probability of reaching s̄ under π as d̂^π(s̄) ← DP_Solver(S_π^+, P̂^π, s̄).
            if d̂^π(s̄) ≥ ε/6D then
                Update I ← I ∪ {s̄}, T ← T ∪ {(s̄, π)}, and set Terminate = False.
                Collect dataset D_s̄ ← DataCollector(s̄, π, Π_core, n_2).
            end if
        end for
    end for
end while
/* Policy Evaluation and Optimization */
for π ∈ Π do
    V̂^π ← Evaluate(Π_core, I, {D_s}, π).
end for
Return π̂ ∈ argmax_π V̂^π.
```

The key challenge is doing all of this "in parallel" for every $\pi \in \Pi$ through extensive sample reuse to avoid a blowup of $|\Pi|$ or S in sample complexity.

State identification phase. In the state identification phase, the algorithm proceeds in a loop. The algorithm maintains a set $\mathcal{T}$, which contains tuples of the form (s, π_s), where $s \in \bigcup_{\pi\in\Pi}\mathcal{S}_\pi$ and π_s denotes a policy that reaches s with probability at least $\Omega(\varepsilon/D)$. Initially $\mathcal{T}$ only contains a dummy start state $s_\top$ and a null policy. In every loop, the algorithm first collects a fresh dataset using the DataCollector: for every $(s, \pi_s) \in \mathcal{T}$, first run π_s to reach state s, and then afterwards restart exploration using the random policy $\mathrm{Unif}(\Pi_{\mathrm{exp}})$. Then, it tries to find a new "petal" state $\bar{s}$ for some $\pi \in \Pi$ that is guaranteed to be $\Omega(\varepsilon/D)$-reachability under π. This is accomplished by constructing an (imaginary) MRP on the state space $\mathcal{S}_\pi$ whose transitions

$P_{s,s'}$ are estimated by using IS from the collected datatset. Specifically, for every $\pi \in \Pi$, POPLER estimates the transition probabilities between states in $\mathcal{S}_\pi$ using the following estimator:

$$\widehat{P}^{\pi}_{s\to s'} = \frac{|\Pi_{\mathrm{core}}|}{|\mathcal{D}_s|} \sum_{\tau \in \mathcal{D}_s} \Bigg(\frac{\mathbb{1}\{\pi \rightsquigarrow \tau_{h:h'}\}}{\sum_{\pi' \in \Pi_{\mathrm{core}}} \mathbb{1}\{\pi' \rightsquigarrow \tau_{h:h'}\}} \times \mathbb{1}\left\{ \substack{\tau_{h:h'} \text{ goes from } s \text{ to } s' \\ \text{without going through any other } \mathcal{S}_\pi} \right\} \Bigg). \tag{2}$$

Evaluation phase. The state identification phase cannot go on forever— each $(s, \pi_s) \in \mathcal{T}$ contributes at least $\Omega(\varepsilon/D)$ to cumulative reachability, but since cumulative reachability is bounded by $\mathfrak{C}(\Pi)$ (Lemma 1), we know that $|\mathcal{T}| \le \mathcal{O}(D\mathfrak{C}(\Pi)/\varepsilon)$. At this point, POPLER moves to the evaluation phase. Using the collected data, it executes the Evaluate subroutine for every $\pi \in \Pi$ to estimate $\widehat{V}^\pi$ (via a similar tabular MRP construction and using DP_Solver). The quantity $\widehat{V}^\pi$ is a biased estimate, but the bias is negligible since it is now only due to the states in $\mathcal{S}_\pi$ that are not $\Omega(\varepsilon/D)$-reachable. Thus we can guarantee that $\widehat{V}^\pi$ is an accurate estimate for every $\pi \in \Pi$, and therefore POPLER returns a near-optimal policy.

7. Conclusion

In this paper, we investigate when agnostic RL is statistically tractable in large state and action spaces, and introduce spanning capacity as a natural measure of complexity that only depends on the policy class and is independent of the MDP rewards and transitions. We show that the spanning capacity is both necessary and sufficient for agnostic PAC RL with a generative model. However, we also provided a negative result that spanning capacity is not sufficient for online RL, thus showing a surprising separation between RL with a generative model and online interaction.

Our results pave the way for several future lines of inquiry. In particular, the most interesting direction is to explore complexity measures that can tightly characterize the minimax sample complexity for online RL (c.f. the fundamental theorem of statistical learning). In our work, we showed that bounded spanning capacity along with an additional sunflower structure is sufficient for online RL (and provided a new algorithm called POPLER that works under these assumptions), but are they also necessary? Is there a single tight complexity measure that captures both of them? Other interesting directions for future research include: sharpening the rate in the upper bound, developing regret minimization algorithms for agnostic RL, and understanding issues of computational efficiency, e.g., via oracle efficient algorithms.

References

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Yandong Ji, Gabriel B Margolis, and Pulkit Agrawal. Dribblebot: Dynamic legged manipulation in the wild. *arXiv preprint arXiv:2304.01159*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49: 209–232, 2002.

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.

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.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Michael Kearns, Yishay Mansour, and Andrew Ng. Approximate planning in large pomdps via reusable trajectories. *Advances in Neural Information Processing Systems*, 12, 1999.

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

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16 (2):264, 1971.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

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.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, 2019.

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020a.

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

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*, 34:13406–13418, 2021a.

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

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*, 2019a.

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.

Kefan Dong, Yuping Luo, Tianhe Yu, Chelsea Finn, and Tengyu Ma. On the expressivity of neural networks for deep reinforcement learning. In *International conference on machine learning*, pages 2627–2637. PMLR, 2020.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4, 2019.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. *Advances in Neural Information Processing Systems*, 29, 2016.

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

Zihan Zhang, Xiangyang Ji, and Simon Du. Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. In *Conference on Learning Theory*, pages 4528–4531. PMLR, 2021a.

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.

Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 2008.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning*, 2021b.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 2021.

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.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

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

Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91:325–349, 2013.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. In *Advances in Neural Information Processing Systems*, 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dud'ik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019b.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, 2020.

Zakaria Mhammedi, Dylan J Foster, and Alexander Rakhlin. Representation learning with multi-step inverse kinematics: An efficient and optimal approach to rich-observation rl. *arXiv preprint arXiv:2304.05889*, 2023.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline RL in low-rank MDPs. *arXiv:2110.04652*, 2021.

Audrey Huang, Jinglin Chen, and Nan Jiang. Reinforcement learning in low-rank mdps with density features. *arXiv preprint arXiv:2302.02252*, 2023.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. In *International Conference on Learning Representations*, 2021.

Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In *Conference on Learning Theory*, pages 5062–5127. PMLR, 2022.

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

Andrea Zanette. Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. In *International Conference on Machine Learning*, pages 12287–12297. PMLR, 2021.

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.

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline reinforcement learning: Fundamental barriers for value function approximation. In *Conference on Learning Theory*, 2021b.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

Omer Gottesman, Yao Liu, Scott Sussex, Emma Brunskill, and Finale Doshi-Velez. Combining parametric and nonparametric models for off-policy evaluation. In *International Conference on Machine Learning*, pages 2366–2375. PMLR, 2019.

Ming Yin and Yu-Xiang Wang. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958. PMLR, 2020.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 2019.

Ayush Sekhari, Christoph Dann, Mehryar Mohri, Yishay Mansour, and Karthik Sridharan. Agnostic reinforcement learning with low-rank MDPs and rich observations. *Advances in Neural Information Processing Systems*, 2021.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702. PMLR, 2019.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. PC-PG: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 2020.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *arXiv preprint arXiv:2105.11066*, 2021.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23 (282):1–36, 2022.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

Sergey Levine and Vladlen Koltun. Guided policy search. In *International conference on machine learning*, pages 1–9. PMLR, 2013.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Nataly Brukhim, Elad Hazan, and Karan Singh. A boosting approach to reinforcement learning. *Advances in Neural Information Processing Systems*, 35:33806–33817, 2022.

Naman Agarwal, Brian Bullins, and Karan Singh. Variance-reduced conservative policy iteration. In *International Conference on Algorithmic Learning Theory*, pages 3–33. PMLR, 2023.

James Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

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.

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

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 7599–7608. PMLR, 2021.

Gen Li, Yuling Yan, Yuxin Chen, and Jianqing Fan. Minimax-optimal reward-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2304.07278*, 2023.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020c.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020.

Weitong Zhang, Dongruo Zhou, and Quanquan Gu. Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems*, 34:1582–1593, 2021b.

Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*, pages 22430–22456. PMLR, 2022.

Wenlong Mou, Zheng Wen, and Xi Chen. On the sample complexity of reinforcement learning with policy space generalization. *arXiv preprint arXiv:2008.07353*, 2020.

Gene Li, Pritish Kamath, Dylan J Foster, and Nati Srebro. Understanding the eluder dimension. *Advances in Neural Information Processing Systems*, 35:23737–23750, 2022.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2): 377–399, 2019.

Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pages 578–598. PMLR, 2021.

Yury Polyanskiy and Yihong Wu. Information theory: From coding to learning, 2022.

A. Detailed Comparison to Related Works

Reinforcement Learning (RL) has seen substantial progress over the past few years, with several different directions of work being pursued for efficiently solving RL problems that occur in practice. The classical approach to solving an RL problem is to model it as a tabular MDP. With this viewpoint, a long line of work (Sutton and Barto, 2018; Agarwal et al., 2019; Kearns and Singh, 2002; Brafman and Tennenholtz, 2002; Auer et al., 2008; Azar et al., 2017; Gheshlaghi Azar et al., 2013; Jin et al., 2018) has studied provably sample-efficient learning algorithms that can find the optimal policy for tabular RL. Unfortunately, the sample complexity of such tabular algorithms scales with the size of the state / action space, and thus they fail to be efficient in practical RL problems with large state / action spaces. On the other hand, the key focus of our work is to develop algorithms for MDPs with large state / action spaces, and towards that end, we take an agnostic viewpoint of RL. In particular, we assume that the learner is given a policy class Π (which the learner believes contains a good policy for the underlying MDP), and the goal of the learner is to find a policy that perform as well as the best policy in the given class.

We now provide a detailed comparison of our setup and assumptions with the existing literature.

RL with Function Approximation. A popular paradigm for developing algorithms for MDPs with large state/action spaces is to use function approximation to either model the MDP dynamics or optimal value functions. Over the last decade, there has been a long line of work (Jiang et al., 2017; Dann et al., 2018; Sun et al., 2019; Du et al., 2019b; Wang et al., 2020a; Du et al., 2021; Foster et al., 2021a; Jin et al., 2021a) in understanding structural conditions on the function class, and the underlying MDP, that allow statistically efficient RL. However, all of these works rely on a crucial realizability assumption, namely that the true model / value function belong to the chosen class. Unfortunately, such an assumption is too strong to hold in practice. Furthermore, the prior works using function approximation make additional assumptions like Bellman Completeness that are difficult to verify for the underlying task.

In our work, we study the problem of agnostic RL to sidestep these challenges. In particular, instead of modeling the value/dynamics, the learner now models "good policies" for the underlying task, and the learning objective is to find a policy that can perform as well as the best in the chosen policy class. We note that while a realizable value class / dynamics class $\mathcal{F}$ can be converted into a realizable policy class $\Pi_{\mathcal{F}}$ by choosing the greedy policies for each value function/dynamics, the converse is not true. Thus, our agnostic RL objective relies on strictly weaker modeling assumption.

RL with Rich Observations. Various RL problem settings have been studied where the dynamics comprise a simple latent state space, but instead of observing the latent states directly, the learner gets rich observations corresponding to the underlying states. These include Block MDP (Krishnamurthy et al., 2016; Du et al., 2019b; Misra et al., 2020; Mhammedi et al., 2023), Low-Rank MDPs (Uehara et al., 2021; Huang et al., 2023), Exogenous Block MDPs (Efroni et al., 2021; Xie et al., 2022), Exogenous MDPs (Efroni et al., 2022), etc. However, the prior works on RL with rich observations assume that the learner is given a realizable decoder class (consisting of functions that map observations to latent states) that contains the true decoder for the underlying MDP. Additionally, they require strong assumptions on the underlying latent state space dynamics, e.g. it is tabular or low-rank, in order to make learning tractable. Thus, their guarantees are not agnostic. In fact, given a realizable decoder class and additional structure on the latent state dynamics, one can construct a policy class that contains the optimal policy for the MDP, but the converse is not true. Thus, our agnostic RL setting is strictly more general.

Relation to Exponential Lower Bounds for RL with Function Approximation Recently, many statistical lower bounds have been developed in RL with function approximation. A line of work including (Wang et al., 2020b; Zanette, 2021; Weisz et al., 2021; Foster et al., 2021b), showed that the sample complexity scales exponentially in the horizon H for learning the optimal policy for RL problems where only the optimal value function $Q^\star$ is linear w.r.t. the given features. Similarly, Du et al. (2019a) showed that one may need exponentially in H even if the optimal policy is linear w.r.t. the true features. These lower bounds can be extended to our agnostic RL setting, giving similar exponential in H lower bounds for agnostic RL, thus supplementing the well-known lower bounds (Krishnamurthy et al., 2016) showing that Agnostic RL is tractable without additional structural assumptions on the policy class. Note that the entire focus of this paper is to try to come up with assumptions, like Definition 1 or 3, that circumvent these lower bounds and allow for sample efficient Agnostic RL.

Importance Sampling for RL. Various important sampling based estimators (Xie et al., 2019; Jiang and Li, 2016; Gottesman et al., 2019; Yin and Wang, 2020; Thomas and Brunskill, 2016; Nachum et al., 2019) have been developed in RL theory literature to provide reliable off-policy evaluation in offline RL. However, these methods also work under realizable value function approximation and rely on additional assumptions on the off-policy / offline data, in particular, that the offline data covers the state / action space that is explored by the comparator policy. We note that this line of work does not directly overlap with our current approach but provides a valuable tool for dealing with off-policy data.

Agnostic RL in Low-Rank MDPs. A recent work of Sekhari et al. (2021) explored agnostic PAC-RL in low-rank MDPs, and showed that one can perform agnostic learning w.r.t. any policy class in MDPs that have a small rank. While their guarantees are similar to ours, i.e., they compete with the best policy in the given class and they also do not assume access to a realizable dynamics / value-function class, we remark that the key objective of the two works is complementary. In particular, Sekhari et al. (2021) explore what assumptions on the underlying MDP dynamics suffice for agnostic learning with any given policy class, whereas we ask what assumptions on the given policy class are sufficient for agnostic learning for any underlying dynamics. Exploring the benefits of structure in both the policy class and the underlying MDP in Agnostic RL is an interesting direction for future research.

Policy Gradient Methods. A significant body of work in RL, in both theory (Agarwal et al., 2021; Abbasi-Yadkori et al., 2019; Bhandari and Russo, 2019; Liu et al., 2020; Agarwal et al., 2020; Zhan et al., 2021; Xiao, 2022) and practice (Kakade, 2001; Kakade and Langford, 2002; Levine and Koltun, 2013; Schulman et al., 2015; 2017), studies policy-gradient based methods that can directly search for the best policy in a given policy class. These approaches often leverage mirror descent-style analysis, and can deliver guarantees that are similar to ours, i.e. the returned policy can compete with any policy in the given class, which can be perceived as an agnostic guarantee. However, they are primarily centered around smooth and parametric policy classes, e.g. tabular and linear policy classes, which limits their applicability for a broader range of problem instances. Furthermore, they require strong additional assumptions to work, for instance that the learner is given a good reset distribution that can cover the occupancy measure of the policy that we wish to compare to, and that the policy class satisfies a certain ”policy completeness assumption”; both of which are difficult to verify in practice. In contrast, our work makes no such assumptions but instead studies what kind of policy classes are learnable with a few samples.

CPI, PSDP, and Other Reductions to Supervised Learning. Various RL methods have been developed that return a policy that performs as well as the best policy in the given policy class, by reducing the RL problem from supervised learning. The key difference from policy gradient based methods (that we discussed earlier) is that these approaches do not require a smoothly parameterized policy class, but instead rely on access to a supervised learning oracle w.r.t. the given policy class. Popular approaches include Conservative Policy Iteration (CPI) (Kakade and Langford, 2002; Kakade, 2003; Brukhim et al., 2022; Agarwal et al., 2023), PSDP (Bagnell et al., 2003), Behavior Cloning (Ross and Bagnell, 2010; Torabi et al., 2018), etc. We note that these algorithms rely on additional assumptions, including “policy completeness assumption” and a good sampling / reset distribution that covers the policies that we wish to compare to; in comparison, we do not make any such assumptions in our work.

Efficient RL via reductions to online regression oracles w.r.t. the given policy class have also been studied, e.g. DAgger (Ross et al., 2011), AggreVaTe (Ross and Bagnell, 2014), etc. However, these algorithms rely on a much stronger feedback. In particular the learner, on the states which it visits, can query an expert policy (that we wish to complete with) for its actions or the value function. On the other hand, in this paper, we restrict ourselves to the standard RL setting where the learner only gets instantenous reward signal.

Reward-Free RL. From a technical viewpoint, our algorithm (Algorithm 1) share similaries to algorithms developed in the reward-free RL literature (Jin et al., 2020). In reward-free RL, the goal of the learner is to output a dataset, or set of policies, after interacting with the underlying MDP, that can be later used for planning (with no further interaction with the MDP) for downstream reward functions. The key ideas in our Algorithm 1, in particular, that the learner first finds states $\mathcal{I}$ that are $O(\varepsilon)$-reachable and corresponding policies that can reach them, and then outputs datasets $\{\mathcal{D}_s\}_{s\in\mathcal{I}}$ that can be later used for evaluating any policy $\pi \in \Pi$, share similarities to algorithmic ideas used in reward-free RL. However, we note that our algorithm strictly generalizes prior works in reward-free RL, and in particular can work with large state-action spaces where the notion of reachability as well as the offline-RL objective, is defined w.r.t. the given policy class. In comparison, prior reward-free RL works compete with the best policy for the underlying MDP, and make structure assumptions on the dynamics, e.g. tabular structure (Jin et al., 2020; Ménard et al., 2021; Li et al., 2023) or linear dynamics (Wang et al., 2020c; Zanette et al., 2020; Zhang et al., 2021b; Wagenmaker et al., 2022), to make the problem tractable.

Other Complexity Measures for RL. A recent work by Mou et al. (2020) proposed a new notion of eluder dimension for the policy class, and provide upper bounds for policy-based RL when the class Π has bounded eluder dimension. However, they make various additional assumptions including that the policy class contains the optimal policy for the MDP, the learner has access to a generative model, and that the optimal value function has a gap. On the other hand, we do not make any such assumption and characterize learnability in terms of spanning capacity or size of the minimal sunflower in Π. Looking forward, however, it is interesting to explore the relationship between the complexity measures that we introduced in this paper, and other well known complexity measures including eluder dimension, star number, threshold dimension, etc (see, e.g., Li et al., 2022).

B. Examples of Policy Classes

In this section, we will prove that examples in Section 3 have bounded spanning capacity, and also have the sunflower property. To facilitate our discussion, we define the following notation: for any policy class Π we let

$$\mathfrak{C}_h(\Pi) \coloneqq \max_{M \in \mathcal{M}^{\mathsf{det}}} C_h^{\mathsf{reach}}(\Pi; M),$$

where $C_h^{\mathsf{reach}}(\Pi; M)$ is defined in Definition 1. That is, $\mathfrak{C}_h(\Pi)$ is the per-layer spanning capacity of Π. Then as defined in Definition 1, we have

$$\mathfrak{C}(\Pi) = \max_{h \in [H]} \mathfrak{C}_h(\Pi).$$

Tabular MDP: Since there are at most $|\mathcal{S}_h|$ states in layer h, it is obvious that $\mathfrak{C}_h(\Pi) \le |\mathcal{S}_h| A$, so therefore $\mathfrak{C}(\Pi) \le SA$. Additionally, if we choose $\Pi_{\mathrm{core}} = \{\pi_a : \pi_a(s) \equiv a, a \in \mathcal{A}\}$ and $\mathcal{S}_\pi = \mathcal{S}$ for every $\pi \in \Pi$, then any partial trajectory which satisfies the condition in Definition 3 is of the form (s_h, a_h), which is consistent with $\pi_{a_h} \in \Pi_{\mathrm{core}}$. Hence Π is a (A, S)-sunflower.

Contextual Bandit: Since there is only one layer, any deterministic MDP has a single state with at most A actions possible, so $\mathfrak{C}(\Pi) \le A$. Additionally, if we choose $\Pi_{\mathrm{core}} = \{\pi_a : \pi_a(s) \equiv a, a \in \mathcal{A}\}$, and $\mathcal{S}_\pi = \varnothing$ for every $\pi \in \Pi$, then any partial trajectory which satisfies the condition in Definition 3 is in the form (s, a), which is consistent with $\pi_a \in \Pi_{\mathrm{core}}$. Hence Π is a $(A, 0)$-sunflower.

H-Layer Contextual Bandit: By induction, it is easy to see that any deterministic MDP has at most A^{h-1} states in layer h, each of which has at most A actions. Hence $\mathfrak{C}(\Pi) \le A^H$. Additionally, if we choose

$$\Pi_{\mathrm{core}} = \{\pi_{a_1, \cdots, a_H} : \pi_{a_1, \cdots, a_H}(s_h) \equiv a_h, a_1, \cdots, a_H \in \mathcal{A}\}$$

and $\mathcal{S}_\pi = \varnothing$ for every $\pi \in \Pi$, then any partial trajectory which satisfies the condition in Definition 3 is in the form $(s_1, a_1, \cdots, s_H, a_H)$, which is consistent with $\pi_{a_1, a_2, \cdots, a_H} \in \Pi_{\mathrm{core}}$. Hence Π is a $(A^H, 0)$-sunflower.

ℓ-tons: In the following, we will denote $\Pi_\ell \coloneqq \Pi_{\ell\text{-ton}}$. We will first prove that $\mathfrak{C}(\Pi_\ell) \le 2H^\ell$. To show this, we will prove that $\mathfrak{C}_h(\Pi_\ell) \le 2h^\ell$ by induction on H. When $H = 1$, the class is a subclass of the above contextual bandit class, hence we have $\mathfrak{C}_1(\Pi_\ell) \le 2$. Next, suppose $\mathfrak{C}_{h-1}(\Pi_\ell) \le 2(h-1)^\ell$. We notice that any deterministic MDP must have the first state s_1, and for policies taking $a = 1$ at s_1 can only take $a = 1$ on $\ell - 1$ states in the following layers. Such policies arrive at $\mathfrak{C}_{h-1}(\Pi_{\ell-1})$ states in layer h. Policies taking $a = 0$ at s_1 can only take $a = 1$ on ℓ states in the following layers. Such policies arrive at $\mathfrak{C}_{h-1}(\Pi_\ell)$ states in layer h. Hence we get

$$\mathfrak{C}_h(\Pi_\ell) \le \mathfrak{C}_{h-1}(\Pi_{\ell-1}) + \mathfrak{C}_{h-1}(\Pi_\ell) \le 2(h-1)^{\ell-1} + 2(h-1)^\ell \le 2h^\ell.$$

This finishes the proof of the induction hypothesis. Based on the induction argument, we get

$$\mathfrak{C}(\Pi_\ell) = \max_{h \in [H]} \mathfrak{C}_h(\Pi_\ell) \le 2H^\ell.$$

Additionally, if we choose

$$\Pi_{\mathrm{core}} = \{\pi_0\} \cup \{\pi_h : 1 \le h \le H\},$$

where $\pi_0(s) \triangleq 0$, and $\pi_h(s) \triangleq \mathbb{1}\{s \in \mathcal{S}_h\}$. For every $\pi \in \Pi_\ell$, we choose $\mathcal{S}_\pi$ to be those states that π chooses 1 (the number of such states is at most ℓ). Then any partial trajectory τ which satisfies $\pi \rightsquigarrow \tau$ and also the condition in Definition 3 is in the form $\tau = (s_h, a_h \cdots, s_{h'}, a_{h'})$ where $\forall h + 1 \le i \le h'$, $s_i \notin \mathcal{S}_\pi$ and we have $a_i = 0$. Hence $\pi_h \rightsquigarrow \tau$ (if $a_h = 1$) or $\pi_0 \rightsquigarrow \tau$ (if $a_h = 0$), and τ is consistent with some policy in Π_{core}. Therefore, Π_ℓ is a $(H + 1, \ell)$-sunflower.

1**-active class:** We will first prove that $\mathfrak{C}(\Pi_{1-\text{act}}) \le 2H$. For any deterministic MDP, we use $\bar{\mathcal{S}}_h$ to denote the set of states reachable by $\Pi_{1-\text{act}}$ at layer h. We will show that $\bar{\mathcal{S}}_h \le h$ by induction on h. For $h=1$, this holds since any deterministic MDP has only one state in the first layer. Suppose it holds at layer h. Then we have

$$|\bar{S}_{h+1}| \le |\{(s,\pi(s)) | s \in \bar{S}_h, \pi \in \Pi\}|.$$

Notice policies in $\Pi_{1-\text{act}}$ must take $a=0$ on every $s \notin \{1(1),\cdots,1(H)\}$. Hence $|\{(s,\pi(s)) \mid s \in \bar{S}_h, \pi \in \Pi\}| \le |\bar{\mathcal{S}}_h| + 1 \le h+1$. Thus, the induction argument is complete. As a consequence we have $\mathfrak{C}_h(\Pi) \le 2h$ for all h, so

$$\mathfrak{C}(\Pi_{1-\text{act}}) = \max_{h\in[H]} \mathfrak{C}_h(\Pi_{1-\text{act}}) \le 2H.$$

Additionally, if we choose $\mathcal{S}_\pi = \{1(1),\cdots,1(H)\}$ for all $\pi \in \Pi$,

$$\Pi_{\text{core}} = \{\pi_0\} \cup \{\pi_h : 1 \le h \le H\},$$

where $\pi_0(s) \triangleq 0$, and $\pi_h(s) \triangleq \mathbb{1}\{s \in \mathcal{S}_h\}$. Then then any partial trajectory which satisfies $\pi \rightsquigarrow \tau$ and also the condition in Definition 3 is in the form $\tau = (s_h, a_h \cdots, s_{h'}, a_{h'})$ where $\forall h+1 \le i \le h'$, $s_i \notin \{1(1),\cdots,1(H)\}$ hence $a_i = 0$. Hence $\pi_h \rightsquigarrow \tau$ (if $a_h = 1$) or $\pi_0 \rightsquigarrow \tau$ (if $a_h = 0$). Hence τ is consistent with some policy in Π_{core}. Therefore, $\Pi_{1-\text{act}}$ is a $(H+1,H)$-sunflower.

All-active class: For any deterministic MDP, there is a single state $j(1)$ in the first layer. Any policy which takes $a=1$ at state $j(1)$ must belong to $\Pi_{j-\text{act}}$. Hence such policies can reach at most $\mathfrak{C}_{h-1}(\Pi_{j-\text{act}})$ states in layer h. For polices which take action 0 at state h, all these policies will transit to a fix state in layer 2. Hence such policies can reach at most $\mathfrak{C}_{h-1}(\Pi_{\text{act}})$ states at layer h. Therefore, we get

$$\mathfrak{C}_h(\Pi_{\text{act}}) \le \mathfrak{C}_{h-1}(\Pi_{\text{act}}) + \max_j \mathfrak{C}_{h-1}(\Pi_{j-\text{act}}) \le \mathfrak{C}_{h-1}(\Pi_{\text{act}}) + 2(h-1).$$

By telescoping, we get

$$\mathfrak{C}_h(\Pi_{\text{act}}) \le h(h-1),$$

which indicates that

$$\mathfrak{C}(\Pi_{\text{act}}) = \max_{h\in[H]} \mathfrak{C}_h(\Pi_{\text{act}}) \le H(H-1).$$

Additionally, if we choose $\mathcal{S}_\pi = \{j(1),\cdots,j(H)\}$ for all $\pi \in \Pi_j$,

$$\Pi_{\text{core}} = \{\pi_0\} \cup \{\pi_h : 1 \le h \le H\},$$

where $\pi_0(s) \triangleq 0$, and $\pi_h(s) \triangleq \mathbb{1}\{s \in \mathcal{S}_h\}$. Then then any partial trajectory which satisfies $\pi \rightsquigarrow \tau$ and also the condition in Definition 3 is in the form $\tau = (s_h, a_h \cdots, s_{h'}, a_{h'})$ where $\forall h+1 \le i \le h'$, $s_i \notin \mathcal{S}_\pi$ hence $a_i = 0$. Hence $\pi_h \rightsquigarrow \tau$ (if $a_h = 1$) or $\pi_0 \rightsquigarrow \tau$ (if $a_h = 0$). Hence τ is consistent with some policy in Π_{core}. Therefore, Π_{act} is a $(H+1,H)$-sunflower.

C. Proofs for Section 3

C.1. Proof of Lemma 1

Fix any $M \in \mathcal{M}^{\mathsf{sto}}$, as well as $h \in [H]$. We claim that

$$\Gamma_h := \sum_{s_h\in\mathcal{S}_h, a_h\in\mathcal{A}_h} \sup_{\pi\in\Pi} d_h^\pi(s_h,a_h;M) \le \max_{M'\in\mathcal{M}^{\mathsf{det}}} C_h^{\mathsf{reach}}(\Pi;M'). \tag{3}$$

Here, $d_h^\pi(s_h,a_h;M)$ is the state-action visitation distribution over M.

We will set up some additional notation. Let us define a *prefix* as any tuple of pairs of the form

$$(s_1,a_1,s_2,a_2,\ldots,s_k,a_k) \quad \text{or} \quad (s_1,a_1,s_2,a_2,\ldots,s_k,a_k,s_{k+1}).$$

We will denote prefix sequences as $(s_{1:k},a_{1:k})$ or $(s_{1:k+1},a_{1:k})$ respectively. For any prefix $(s_{1:k},a_{1:k})$ (similarly prefixes of the type $(s_{1:k+1},a_{1:k})$) we let $d_h^\pi(s_h,a_h \mid (s_{1:k},a_{1:k});M)$ denote the conditional probability of reaching (s_h,a_h) under policy π given one observed prefix $(s_{1:k}, a_{1:k})$ in MDP M, with $d_h^\pi(s_h, a_h \mid (s_{1:k}, a_{1:k}); M) = 0$ if $\pi \not\rightsquigarrow (s_{1:k}, a_{1:k})$ or $\pi \not\rightsquigarrow (s_h, a_h)$.

In the following proof, we assume that the start state s_1 is fixed, but this is without loss of generality, and the proof can easily be adapted to hold for stochastic start states.

Our strategy will be to explicitly compute the quantity Γ_h in terms of the dynamics of M and show that we can upper bound it by a "derandomized" MDP M' which maximizes reachability at layer h. Let us unroll one step of the dynamics:

$$\begin{aligned}
\Gamma_h &:= \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} d_h^\pi(s_h, a_h; M) \\
&\overset{(i)}{=} \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} d_h^\pi(s_h, a_h \mid s_1; M), \\
&\overset{(ii)}{=} \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} \left\{ \sum_{a_1 \in \mathcal{A}} d_h^\pi(s_h, a_h \mid s_1, a_1; M) \right\} \\
&\overset{(iii)}{\le} \sum_{a_1 \in \mathcal{A}} \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} d_h^\pi(s_h, a_h \mid s_1, a_1; M).
\end{aligned}$$

The equality (i) follows from the fact that M always starts at s_1. The equality (ii) follows from the fact that π is deterministic, so there exists exactly one $a' = \pi(s_1)$ for which $d_h^\pi(s_h, a_h \mid s_1, a'; M) = d_h^\pi(s_h, a_h \mid s_1; M)$, with all other $a'' \neq a'$ satisfying $d_h^\pi(s_h, a_h | s_1, a''; M) = 0$. The inequality (iii) follows by taking the supremum inside.

Continuing in this way, we can show that

$$\begin{aligned}
\Gamma_h &\le \sum_{a_1 \in \mathcal{A}} \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} \left\{ \sum_{s_2 \in \mathcal{S}_2} P(s_2 | s_1, a_1) \sum_{a_2 \in \mathcal{A}} d_h^\pi(s_h, a_h \mid (s_{1:2}, a_{1:2}); M) \right\} \\
&\le \sum_{a_1 \in \mathcal{A}} \sum_{s_2 \in \mathcal{S}_2} P(s_2 | s_1, a_1) \sum_{a_2 \in \mathcal{A}} \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} d_h^\pi(s_h, a_h \mid (s_{1:2}, a_{1:2}); M) \\
&\dots \\
&\le \sum_{a_1 \in \mathcal{A}} \sum_{s_2 \in \mathcal{S}_2} P(s_2 | s_1, a_1) \sum_{a_2 \in \mathcal{A}} \cdots \sum_{s_{h-1} \in \mathcal{S}_{h-1}} P(s_{h-1} | s_{h-1}, a_{h-2}) \sum_{a_{h-1} \in \mathcal{A}} \\
&\qquad \sum_{s_h \in \mathcal{S}_h, a_h \in \mathcal{A}} \sup_{\pi \in \Pi} d_h^\pi(s_h, a_h \mid (s_{1:h-1}, a_{1:h-1}); M).
\end{aligned}$$

Now we examine the conditional visitation $d_h^\pi(s_h, a_h \mid (s_{1:h-1}, a_{1:h-1}); M)$. Observe that it can be rewritten as

$$d_h^\pi(s_h, a_h \mid (s_{1:h-1}, a_{1:h-1}); M) = P(s_h | s_{h-1}, a_{h-1}) \cdot \mathbb{1}\{\pi \rightsquigarrow (s_{1:h}, a_{1:h})\}.$$

Plugging this back into the previous display and taking the supremum inside the sum again,

$$\begin{aligned}
\Gamma_h &\le \sum_{a_1 \in \mathcal{A}} \cdots \sum_{s_h \in \mathcal{S}_h} \mathrm{P}(s_{h-1} | s_{h-1}, a_{h-1}) \sum_{a_h \in \mathcal{A}} \sup_{\pi \in \Pi} \mathbb{1}\{\pi \rightsquigarrow (s_{1:h}, a_{1:h})\} \\
&= \sum_{a_1 \in \mathcal{A}} \cdots \sum_{s_h \in \mathcal{S}_h} P(s_{h-1} | s_{h-1}, a_{h-1}) \sum_{a_h \in \mathcal{A}} \mathbb{1}\{\exists \pi \in \Pi : \pi \rightsquigarrow (s_{1:h}, a_{1:h})\}
\end{aligned}$$

Our last step is to apply "derandomization" to the above, simply by taking the sup over transition probabilities:

$$\Gamma_h \le \sum_{a_1 \in \mathcal{A}} \sup_{s_2 \in \mathcal{S}_2} \sum_{a_2 \in \mathcal{A}} \dots \sup_{s_h \in \mathcal{S}_h} \sum_{a_h \in \mathcal{A}} \mathbb{1}\{\exists \pi \in \Pi : \pi \rightsquigarrow (s_{1:h}, a_{1:h})\} = \max_{M' \in \mathcal{M}^{\mathsf{det}}} C_h^{\mathsf{reach}}(\Pi; M').$$

The right hand side of the inequality is exactly the definition of $\max_{M' \in \mathcal{M}^{\mathsf{det}}} C_h^{\mathsf{reach}}(\Pi; M')$, thus proving Eq. (3). In particular, the above process defines the deterministic MDP which maximizes the reachability at level h. Taking the maximum over h concludes the proof of Lemma 1. $\square$

C.2. Coverability is Not Sufficient for Online RL

We now observe that coverability is not sufficient for agnostic PAC RL in the online setting. In fact, we prove a statement of this form: Theorem 3 shows there exists a policy class with bounded spanning capacity that is hard to learn in the online setting. The policy class in question must also have bounded coverability via Lemma 1.

However, we can immediately get a stronger lower bound if we only assume bounded coverability. Specifically, the lower bound construction of (Sekhari et al., 2021) satisfies $C^{\mathsf{cov}}(\Pi; M) = \mathcal{O}(1)$ for every $M \in \mathcal{M}$, yet they show a lower bound of $2^{\Omega(H)}$ on the sample complexity of any $(\Theta(1), \Theta(1))$-PAC learner (by setting the rank of the MDP to $d = \Theta(H)$ in their Theorem 2).

D. Proofs for Section 4

D.1. Proof of Theorem 1

Algorithm 2 Trajectory_Tree (Kearns et al., 1999)

Require: Policy class Π, generative access to M, number of samples n

1: Initialize dataset of trajectory trees $\mathcal{M} = \varnothing$.
2: **for** $t = 1, \dots, n$ **do**
3: Initialize trajectory tree $\widehat{M}_t = \varnothing$
4: Sample initial state $s_1^{(t)} \sim \mu$.
/* Sample transitions and rewards for a trajectory tree */
5: **while** True **do**
6: Find any unsampled (s, a) s.t. s is reachable by some $\pi \in \Pi$ in $\widehat{M}_t$.
7: **if** no such (s, a) exists **then**
8: **break**
9: **end if**
10: Sample $s' \sim P(\cdot | s, a)$ and $r \sim R(s, a)$
11: Add (s, a, r, s') to $\widehat{M}_t$.
12: **end while**
13: $\mathcal{M} \leftarrow \mathcal{M} \cup \widehat{M}_t$.
14: **end for**
/* Policy evaluation */
15: **for** $\pi \in \Pi$ **do**
16: Set $\widehat{V}^\pi \leftarrow \frac{1}{n} \sum_{t=1}^n \widehat{v}_t^\pi$, where $\widehat{v}_t^\pi$ is the cumulative reward of π on $\widehat{M}_t$.
17: **end for**
18: **Return** $\widehat{\pi} \leftarrow \arg\max_{\pi \in \Pi} \widehat{V}^\pi$.

We show that the Trajectory_Tree algorithm of (Kearns et al., 1999) attains the guarantee in Theorem 1. Pseudocode can be found in Algorithm 2. The key modification is line 2: we simply observe that only (s, a) pairs which are reachable by some $\pi \in \Pi$ in the current tree $\widehat{M}_t$ need to be sampled (in the original algorithm, they sample all 2^H transitions).

Fix any $\pi \in \Pi$. For any tree $t \in [n]$, we have collected enough transitions so that $\widehat{v}_t^\pi$ is well-defined, by line 2 of the algorithm. The cumulative reward $\widehat{v}_t^\pi$ is an unbiased estimate of V^π. One can consider an alternative process for the construction of $\widehat{M}_t$ as first constructing the path that π takes and then filling out the rest of the tree. The only difference between this process and the actual one is the *order* in which the transitions are sampled, so all of the transitions and rewards are still sampled from the correct distributions. Also, it is easy to see that the $\widehat{v}_t^\pi$ are independent for different $t \in [n]$.

Therefore, using Hoeffding's inequality for $[0, 1]$-bounded random variables we see that $|V^\pi - \widehat{V}^\pi| \le \sqrt{\frac{\log(2/\delta)}{2n}}$. Applying union bound we see that when the number of trajectory trees exceeds $n \gtrsim \frac{\log(|\Pi|/\delta)}{\varepsilon^2}$, with probability at least $1 - \delta$, for all $\pi \in \Pi$, the estimates satisfy $|V^\pi - \widehat{V}^\pi| \le \varepsilon/2$. Thus the Trajectory Tree algorithm returns an ε-optimal policy. Since each trajectory tree uses at most $H \cdot \mathfrak{C}(\Pi)$ queries to the generative model, we have the claimed sample complexity bound. $\square$

D.2. Proof of Theorem 2

Fix any worst-case deterministic MDP $M^\star$ which witnesses $\mathfrak{C}(\Pi)$ at layer $h^\star$. We can also assume that the algorithm knows $M^\star$ and $h^\star$ (this only makes the lower bound stronger). Observe that we can embed a bandit instance with $\mathfrak{C}(\Pi)$ many arms by putting rewards only on the (s, a) pairs at level $h^\star$ which are reachable by some $\pi \in \Pi$. The proof concludes by using existing PAC lower bounds which show that the sample complexity of PAC learning a K-armed multi-armed bandit is at least $\Omega(\frac{K}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ (see, e.g., Mannor and Tsitsiklis, 2004). $\square$

D.3. Proof of Corollary 1

The upper bound is obtained by a simple modification of the argument in the proof of Theorem 1. In terms of data collection, the trajectory tree collected every time is the same fixed deterministic MDP (with different rewards); furthermore, one can always execute line 2 and line 2 for a deterministic MDP since the algorithm can execute a sequence of actions to get to any new (s, a) pair required by line 2. Thus in every episode of online interaction we are guaranteed to add the new (s, a) pair to the trajectory tree.

The lower bound trivially extends because the proof of Appendix D.2 uses an MDP with deterministic transitions (that are even known to the algorithm beforehand).

E. Proofs for Section 5

In this section, we prove Theorem 3, which shows a superpolynomial lower bound on the sample complexity required to learn bounded spanning capacity classes. The theorem is restated below with explicit constants.

Theorem 5 (Lower bound for online RL)**.** *Fix any $H \geq 10^5$. Let $\varepsilon \in (1/H^{100}, 1/(100H))$ and $\ell \in \{2, \dots, \lfloor \log H \rfloor\}$. There exists a policy class $\Pi^{(\ell)}$ of size $1/(6\varepsilon^{\ell})$ with $\mathfrak{C}(\Pi^{(\ell)}) \leq O(H^{4\ell+2})$ and a family of MDPs $\mathcal{M}$ with state space $\mathcal{S}$ of size $H \cdot 2^{2H+1}$, binary action space, horizon H such that: for any $(\varepsilon/16, 1/8)$-PAC algorithm, there exists an $M \in \mathcal{M}$ in which the algorithm has to collect at least*

$$\min\left\{\frac{1}{120\varepsilon^{\ell}}, 2^{H/3-3}\right\} \quad \textit{online trajectories in expectation.}$$

E.1. Construction of State Space, Action Space, and Policy Class

State and action spaces. We define the state space $\mathcal{S}$. In every layer $h \in [H]$, there will be 2^{2H+1} states. The states will be paired up, and each state will be denoted by either $j[h]$ or $j'[h]$, so $\mathcal{S}_h = \{j[h] : j \in [2^{2H}]\} \cup \{j'[h] : j \in [2^{2H}]\}$. For any state $s \in \mathcal{S}$, we define the *index* of s, denoted $\mathrm{idx}(s)$ as the unique $j \in [2^{2H}]$ such that $s \in \{j[h]\}_{h \in [H]} \cup \{j'[h]\}_{h \in [H]}$. In total there are $H \cdot 2^{2H+1}$ states. The action space is $\mathcal{A} = \{0, 1\}$.

Policy class. For the given ε and $\ell \in \{2, \dots, \lfloor \log H \rfloor\}$, we show via a probabilistic argument the existence of a large policy class $\Pi^{(\ell)}$ which has bounded spanning capacity but is hard to explore. We state several properties in Lemma 2 which will be exploited in the lower bound.

We introduce some additional notation. For any $j \in [2^H]$ we denote

$$\Pi_j^{(\ell)} := \{\pi \in \Pi^{(\ell)} : \exists h \in [H], \pi(j[h]) = 1\},$$

that is, $\Pi_j^{(\ell)}$ are the policies which take an action $a = 1$ on at least one state with index j.

We also define the set of *relevant state indices* for a given policy $\pi \in \Pi^{(\ell)}$ as

$$\mathcal{J}_{\mathrm{rel}}^{\pi} := \{j \in [2^H] : \pi \in \Pi_j^{(\ell)}\}.$$

For any policy π we denote $\pi(j_{1:H}) := (\pi(j[1]), \dots, \pi(j[H])) \in \{0,1\}^H$ to be the vector that represents the actions that π takes on the states in index j. The vector $\pi(j'_{1:H})$ is defined similarly.

Lemma 2. *For the given ε and $\ell \in \{2, \dots, \lfloor \log H \rfloor\}$, there exists a policy class $\Pi^{(\ell)}$ of size $1/(6\varepsilon^{\ell})$ which satisfies the following properties.*

(1) For every $j \in [2^H]$ we have $|\Pi_j^{(\ell)}| \in [\varepsilon N/2, 2\varepsilon N]$.

(2) For every $\pi \in \Pi$ we have $|\mathcal{J}_{\mathrm{rel}}^{\pi}| \geq \varepsilon/2 \cdot 2^{2H}$.

(3) For every $\pi \in \Pi_j^{(\ell)}$, the vector $\pi(j_{1:H})$ is unique and always equal to $\pi(j'_{1:H})$.

(4) Bounded spanning capacity: $\mathfrak{C}(\Pi^{(\ell)}) \leq c \cdot H^{4\ell+2}$ for some universal constant $c > 0$.

E.2. Construction of MDP Family

The family $\mathcal{M} = \{M_{\pi^\star, \phi}\}_{\pi^\star \in \Pi^{(\ell)}, \phi \in \Phi}$ will be a family of MDPs which are indexed by a policy $\pi^\star$ as well as a *decoder* function $\phi : \mathcal{S} \mapsto \{\textsc{good}, \textsc{bad}\}$, which assigns each state to be "good" or "bad" in a sense that will be described later on.

Decoder function class. The decoder function class Φ will be all possible mappings which for every $j \in [2^{2H}]$, $h \geq 2$ assign exactly one of $j[h], j'[h]$ to the label Good and the other one to Bad. There are $(2^{H-1})^{2^{2H}}$ such functions. The label of a state will be used to describe the transition dynamics. Intuitively, a learner who does not know the decoder function ϕ will not be able to tell if a certain state has the label Good or Bad upon visiting a state index j for the first time.

Transition dynamics. The MDP $M_{\pi^\star,\phi}$ will be a uniform distribution over 2^{2H} combination locks $\{\mathsf{CL}_j\}_{j\in[2^{2H}]}$ with disjoint states. More formally, $s_1 \sim \mathrm{Unif}(\{j[1]\}_{j\in[2^{2H}]})$. From each start state $j[1]$, only the $2H-2$ states corresponding to index j at layers $h \geq 2$ will be reachable in combination lock CL_j.

Now we will describe each combination lock CL_j, which forms the basic building block of the MDP construction.

- **Good/bad set.** At every layer $h \in [H]$, for each $j[h]$ and $j'[h]$, the decoder function ϕ assigns one of them to be GOOD and one of them to be BAD. We will henceforth denote $j_g[h]$ to be the good state and $j_b[h]$ to be the bad state. Observe that by construction in Eq. (6), for every $\pi \in \Pi^{(\ell)}$ and $h \in [H]$ we have $\pi(j_g[h]) = \pi(j_b[h])$.
- **Dynamics of CL_j, if $j \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}$.** Here, the transition dynamics of the combination locks are deterministic. We let $T(s,a)$ denote the state that (s,a) transitions to, i.e., $T(s,a) = s'$ if and only if $P(s'|s,a) = 1$. We also use $T(s,\pi) := T(s,\pi(s))$ as shorthand. For every $h \in [H]$,
  - On good states $j_g[h]$ we transit to the next good state iff the action is $\pi^\star$:
  $$T(j_g[h], \pi^\star) = j_g[h+1], \quad \text{and} \quad T(j_g[h], 1-\pi^\star) = j_b[h+1].$$
  - On bad states $j_b[h]$ we always transit to the next bad state:
  $$T(j_b[h], a) = j_b[h+1], \quad \text{for all } a \in \mathcal{A}.$$
- **Dynamics of CL_j, if $j \notin \mathcal{J}_{\mathrm{rel}}^{\pi^\star}$.** If j is not a relevant index for $\pi^\star$, then the transitions are uniformly random regardless of the current state/action. For every $h \in [H]$,
  $$T(j_g[h], a) = T(j_b[h], a) = \mathrm{Unif}(\{j_g[h+1], j_b[h+1]\}, \quad \text{for all } a \in \mathcal{A}.$$
- **Reward structure.** The reward function is nonzero only at layer H, and is defined as
  $$R(s,a) = \mathrm{Ber}\left(\frac{1}{2} + \frac{1}{4} \cdot \mathbb{1}\{\pi^\star \in \Pi_j^{(\ell)}\} \cdot \mathbb{1}\{s = j_g[H], a = \pi^\star(j_g[H])\}\right)$$
  That is, we get $3/4$ whenever we reach the H-th good state for an index j which is relevant for $\pi^\star$, and $1/2$ reward otherwise.

Reference MDP M_0. We also define a *reference MDP* M_0. In the reference MDP M_0, all the combination locks behave the same and have uniform transitions to the next state. The distribution over all 2^{2H} combination locks is again taken to be the uniform distribution. The rewards for M_0 will be $\mathrm{Ber}(1/2)$ for every $(s,a) \in \mathcal{S}_H \times \mathcal{A}$.

E.3. Proof of Theorem 5

Now we are ready to prove the lower bound using the construction $\mathcal{M}$.

Value calculation. Consider any $M_{\pi^\star,\phi} \in \mathcal{M}$. For any policy $\pi \in \mathcal{A}^{\mathcal{S}}$ we use $V_{\pi^\star,\phi}(\pi)$ to denote the value of running π in MDP $M_{\pi^\star,\phi}$. By construction we can see that

$$V_{\pi^\star,\phi}(\pi) = \frac{1}{2} + \frac{1}{4} \cdot \mathrm{Pr}_{\pi^\star,\phi}\left[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \pi(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\right]. \tag{4}$$

In words, the second term counts the additional reward that π gets for solving a combination lock rooted at a relevant state index $\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}$. By Property (2) and (3) of Lemma 2, we additionally have $V_{\pi^\star,\phi}(\pi^\star) \geq 1/2 + \varepsilon/8$, as well as $V_{\pi^\star,\phi}(\pi) = 1/2$ for all other $\pi \neq \pi^\star \in \Pi^{(\ell)}$.

By Eq. (4), if π is an $\varepsilon/16$-optimal policy on $M_{\pi^\star,\phi}$ it must satisfy

$$\mathrm{Pr}_{\pi^\star,\phi}\left[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \pi(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\right] \geq \frac{\varepsilon}{4}.$$

Averaged measures. We define the following measures which will be used in the analysis. First, let us define $\Pr_{\pi^\star}[\cdot] = \frac{1}{|\Phi|}\sum_{\phi\in\Phi}\Pr_{\pi^\star,\phi}[\cdot]$ to be the averaged measure where we first pick ϕ uniformly among all decoders and then consider the distribution induced by $M_{\pi^\star,\phi}$. Also, let the MDP $M_{0,\pi^\star,\phi}$ have the same transitions as $M_{\pi^\star,\phi}$ but with all rewards at the last layer to be $\mathrm{Ber}(1/2)$, the same as the rewards for M_0. Then we can define the averaged measure $\Pr_{0,\pi^\star}[\cdot] = \frac{1}{|\Phi|}\sum_{\phi\in\Phi}\Pr_{0,\pi^\star,\phi}[\cdot]$ where we pick ϕ uniformly and then consider the distribution induced by $M_{0,\pi^\star,\phi}$. For both averaged measures the expectations $\mathbb{E}_{\pi^\star}$ and $\mathbb{E}_{0,\pi^\star}$ are defined analogously.

Algorithm and stopping time. Recall that an algorithm $\mathbb{A}$ is comprised of two phases. In the first phase, it collects some number of trajectories by interacting with the MDP in episodes. We use η to denote the (random) number of episodes after which $\mathbb{A}$ terminates. We also use $\mathbb{A}_t$ to denote the intermediate policy that the algorithm runs in round t for $t\in[\eta]$. In the second phase, $\mathbb{A}$ outputs a policy $\widehat{\pi}$. We use the notation $\mathbb{A}_f : \{\tau^{(t)}\}_{t\in[\eta]} \mapsto \mathcal{A}^{\mathcal{S}}$ to denote the second phase of $\mathbb{A}$ which outputs the $\widehat{\pi}$ as a measurable function of collected data.

For any policy $\pi^\star$, decoder ϕ, and dataset $\mathcal{D}$ we define the event

$$\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D})) := \Big\{\Pr_{\pi^\star,\phi}\Big[\mathrm{idx}(s_1)\in\mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\Big] \ge \frac{\varepsilon}{4}\Big\}.$$

The randomness in $\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))$ is due to randomness in $\mathcal{D}$, which is the data collection process of $\mathbb{A}$. Note that the event $\mathcal{E}$ is well defined for $\mathcal{D}$ that is collected on *any* MDP, not just $M_{\pi^\star,\phi}$.

Under this notation, the PAC learning guarantee on $\mathbb{A}$ implies that for every $\pi^\star\in\Pi^{(\ell)}$, $\phi\in\Phi$ we have

$$\Pr_{\pi^\star,\phi}\big[\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big] \ge 7/8.$$

Moreover via an averaging argument we also have

$$\Pr_{\pi^\star}\big[\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big] \ge 7/8. \tag{5}$$

Lower bound argument. We apply a truncation to the stopping time η. Define $T_{\max} := 2^{H/3}$. Observe that if $\Pr_{\pi^\star}[\eta > T_{\max}] > 1/8$ for some $\pi^\star\in\Pi^{(\ell)}$ then the lower bound immediately follows, since

$$\max_{\phi\in\Phi}\mathbb{E}_{\pi^\star,\phi}[\eta] > \mathbb{E}_{\pi^\star}[\eta] \ge \Pr_{\pi^\star}[\eta > T_{\max}]\cdot T_{\max} \ge T_{\max}/8,$$

so there must exist an MDP $M_{\pi^\star,\phi}$ for which $\mathbb{A}$ collects at least $T_{\max}/8 = 2^{H/3-3}$ samples in expectation.

Otherwise we have $\Pr_{\pi^\star}[\eta > T_{\max}] \le 1/8$ for all $\pi^\star\in\Pi^{(\ell)}$. This further implies that for all $\pi^\star\in\Pi^{(\ell)}$,

$$\begin{aligned}&\Pr_{\pi^\star}\big[\eta < T_{\max} \text{ and } \mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big]\\ &= \Pr_{\pi^\star}\big[\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big] - \Pr_{\pi^\star}\big[\eta > T_{\max} \text{ and } \mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big] \ge 3/4.\end{aligned}$$

In this second case, we will show that $\mathbb{A}$ requires a lot of samples on M_0. This is formalized in the following lemma.

Lemma 3 (Stopping time lemma). *Let $\delta\in(0,1/8]$. Let $\mathbb{A}$ be an $(\varepsilon/16,\delta)$-PAC algorithm. Let $T_{\max}\in\mathbb{N}$. Suppose that $\Pr_{\pi^\star}\big[\eta < T_{\max}$ and $\mathcal{E}(\pi^\star,\phi,\mathbb{A}_f(\mathcal{D}))\big] \ge 1-2\delta$ for all $\pi^\star\in\Pi^{(\ell)}$. The expected stopping time for $\mathbb{A}$ on M_0 is at least*

$$\mathbb{E}_0[\eta] \ge \left(\frac{|\Pi^{(\ell)}|}{2} - \frac{4}{\varepsilon}\right)\cdot\frac{1}{7}\log\left(\frac{1}{2\delta}\right) - |\Pi^{(\ell)}|\cdot\frac{{T_{\max}}^2}{2^{H+3}}\left(T_{\max} + \frac{1}{7}\log\left(\frac{1}{2\delta}\right)\right).$$

Using Lemma 3 with $\delta = 1/8$ and plugging in the value of $|\Pi^{(\ell)}|$ and $T_{\max}$, we see that

$$\mathbb{E}_0[\eta] \ge \left(\frac{|\Pi^{(\ell)}|}{2} - \frac{4}{\varepsilon}\right)\cdot\frac{1}{7}\log\left(\frac{1}{2\delta}\right) - |\Pi^{(\ell)}|\cdot\frac{{T_{\max}}^2}{2^{H+3}}\left(T_{\max} + \frac{1}{7}\log\left(\frac{1}{2\delta}\right)\right) \ge \frac{|\Pi^{(\ell)}|}{20}.$$

For the second inequality, we used the fact that $\ell\ge 2$, $H\ge 10^5$, and $\varepsilon < 1/10^7$. So therefore the lower bound on the sample complexity is at least

$$\min\left\{\frac{1}{120\varepsilon^{\ell}}, 2^{H/3}\right\}.$$

This concludes the proof of Theorem 3. $\square$

E.4. Proof of Lemma 2

To prove Lemma 2, we first use a probabilistic argument to construct a certain binary matrix B which satisfies several properties, and then construct $\Pi^{(\ell)}$ using B and verify it satisfies Properties (1)-(4).

Binary matrix construction. First we define a block-free property of binary matrices.

Definition 4. *Fix parameters $k, \ell \in \mathbb{N}$. We say a binary matrix $B \in \{0,1\}^{N \times d}$ is (k, ℓ) **block-free** if the following holds: for every $I \subset [N]$ with $|I| = k$, and $J \subset [d]$ with $|J| = \ell$ there exists some $(i,j) \in I \times J$ with $B_{ij} = 0$.*

In words, matrices which are (k, ℓ) block-free do not contain a $k \times \ell$ "block" of all 1s.

Lemma 4. *Fix any $\varepsilon \in (0, 1/10)$ and $\ell \in \mathbb{N}$. For any*

$$d \in \Big[\frac{16\ell \cdot \log(1/\varepsilon)}{\varepsilon}, \frac{1}{20} \cdot \exp\Big(\frac{1}{48\varepsilon^{\ell-1}}\Big)\Big],$$

there exists a binary matrix $B \in \{0,1\}^{N \times d}$ with $N = 1/(6 \cdot \varepsilon^{\ell})$ such that:

1. *(Row sum): for every row $i \in [N]$, we have $\sum_j B_{ij} \ge \varepsilon d/2$.*
2. *(Column sum): for every column $j \in [d]$, we have $\sum_i B_{ij} \in [\varepsilon N/2, 2\varepsilon N]$.*
3. *The matrix B is $(\ell \log d, \ell)$ block-free.*

Proof of Lemma 4. The existence of B is proven by a probabilistic argument. Let $\widetilde{B} \in \{0,1\}^{N \times d}$ be a random matrix where each entry is i.i.d. chosen to be 1 with probability ε.

By Chernoff bounds, for every row $i \in [N]$, we have $\mathrm{P}[\sum_j B_{ij} \le \frac{\varepsilon d}{2}] \le \exp(-\varepsilon d/8)$; likewise for every column $j \in [d]$ we have $\mathrm{P}[\sum_j B_{ij} \notin [\frac{\varepsilon N}{2}, 2\varepsilon N]] \le 2\exp(-\varepsilon N/8)$. By union bound, the matrix $\widetilde{B}$ satisfies the first two properties with probability at least 0.8 as long as

$$d \ge (8 \log 10N)/\varepsilon, \quad \text{and} \quad N \ge (8 \log 20d)/\varepsilon.$$

One can check that under the choice of $N = 1/(6 \cdot \varepsilon^{\ell})$ and the assumption on d, both constraints are met.

Now we examine the probability of $\widetilde{B}$ satisfies the block-free property with parameters $(k = \ell \log d, \ell)$. Let X be the random variable which denotes the number of submatrices which violate to the block-free property in $\widetilde{B}$, i.e.,

$$X = |\{I \times J : I \subset [N], |I| = k, J \subset [d], |J| = \ell, \widetilde{B}_{ij} = 1 \ \forall \ (i,j) \in I \times J\}|.$$

By linearity of expectation we have

$$\mathbb{E}[X] \le N^k d^\ell \varepsilon^{k\ell}.$$

We now plug in the choice $k = \ell \log d$ and observe that as long as $N \le 1/(2e \cdot \varepsilon^{\ell})$ we have $\mathbb{E}[X] \le 1/2$. By Markov's inequality, $\mathrm{P}[X = 0] \ge 1/2$.

Therefore with positive probability, $\widetilde{B}$ satisfies all 3 properties (otherwise we would have a contradiction via inclusion-exlusion principle). We can conclude the existence of B which satisfies all 3 properties, proving the result of Lemma 4. □

Policy class construction. For the given ε and $\ell \in \{2, \dots, \lfloor \log H \rfloor\}$ we will use Lemma 4 to construct a policy class $\Pi^{(\ell)}$ which has bounded spanning capacity but is hard to explore. We instantiate Lemma 4 with the given ℓ and $d = 2^{2H}$, and use the resulting matrix B to construct $\Pi^{(\ell)} = \{\pi_i\}_{i \in [N]}$ with $|\Pi^{(\ell)}| = N = 1/(6\varepsilon^{\ell})$. One can check that whenever $H \ge 10^5$ and $\varepsilon \in \big[\frac{1}{H^{100}}, \frac{1}{100H}\big]$, the requirement of Lemma 4 is met:

$$d = 2^{2H} \in \Bigg[\frac{16\ell \cdot \log(1/\varepsilon)}{\varepsilon}, \frac{1}{20} \cdot \exp\Bigg(\frac{1}{48\varepsilon^{\ell-1}}\Bigg)\Bigg].$$

Moreover we see that $2\varepsilon N < 2^H$ (i.e., the column sum in B does not exceed 2^H).

We define the policies as follows: for every $\pi_i \in \Pi^{(\ell)}$ we set

$$\text{for every } j \in [2^H]: \quad \pi_i(j[h]) = \pi_i(j'[h]) = \begin{cases} \mathrm{bit}_h(\sum_{a \le i} B_{aj}) & \text{if } B_{ij} = 1, \\ 0 & \text{if } B_{ij} = 0. \end{cases} \tag{6}$$

The function $\mathrm{bit}_h : [2^H - 1] \mapsto \{0, 1\}$ selects the h-th bit in the binary representation of the input.

Verifying Properties (1) - (4). Properties (1) - (3) are straightforward from the construction of B and $\Pi^{(\ell)}$, since $\pi_i \in \Pi_j^{(\ell)}$ if and only if $B_{ij} = 1$. We require that $2\varepsilon N < 2^H$ in order for Property (3) to hold, since otherwise we cannot assign the behaviors of the policies according to Eq. (6).

We now prove Property (4): that $\Pi^{(\ell)}$ has bounded spanning capacity. To prove this we will use the block-free property of the underlying binary matrix B.

Fix any deterministic MDP $M^\star$ which witnesses $\mathfrak{C}(\Pi^{(\ell)})$ at layer $h^\star$. To bound $\mathfrak{C}(\Pi^{(\ell)})$, we need to count the contribution to $C_{h^\star}^{\mathsf{reach}}(\Pi; M^\star)$ from trajectories τ which are produced by some $\pi \in \Pi^{(\ell)}$ on M. We first define a *layer decomposition* for a trajectory $\tau = (s_1, a_1, s_2, a_2, \dots, s_H, a_H)$ as the unique tuple of indices $(h_1, h_2, \dots h_m)$, where each $h_k \in [H]$. The layer decomposition satisfies the following properties:

- The layers satisfy $h_1 < h_2 < \dots < h_m$.
- The layer h_1 represents the first layer where $a_{h_1} = 1$.
- The layer h_2 represents the first layer where $a_{h_2} = 1$ on some state s_{h_2} such that
$$\mathrm{idx}(s_{h_2}) \notin \{\mathrm{idx}(s_{h_1})\}.$$
- The layer h_3 represents the first layer where $a_{h_3} = 1$ on some state s_{h_3} such that
$$\mathrm{idx}(s_{h_3}) \notin \{\mathrm{idx}(s_{h_1}), \mathrm{idx}(s_{h_2})\}.$$
- More generally the layer h_k, $k \in [m]$ represents the first layer where $a_{h_k} = 1$ on some state s_{h_k} such that
$$\mathrm{idx}(s_{h_k}) \notin \{\mathrm{idx}(s_{h_1}), \dots, \mathrm{idx}(s_{h_{k-1}})\}.$$
In other words, the layer h_k represents the k-th layer for where action is $a = 1$ on a new state index which τ has never played $a = 1$ on before.

We will count the contribution to $C_{h^\star}^{\mathsf{reach}}(\Pi; M^\star)$ by doing casework on the length of the layer decomposition for any τ. That is, for every length $m \in \{0, \dots, H\}$, we will bound $C_{h^\star}(m)$, which is defined to be the total number of (s, a) at layer $h^\star$ which, for some $\pi \in \Pi^{(\ell)}$, a trajectory $\tau \rightsquigarrow \pi$ that has a m-length layer decomposition visits. Then we apply the bound

$$C_{h^\star}^{\mathsf{reach}}(\Pi; M^\star) \le \sum_{m=0}^{H} C_{h^\star}(m). \tag{7}$$

Note that this will overcount, since the same (s, a) pair can belong to multiple different trajectories with different length layer decompositions.

We have the following lemma.

Lemma 5. *The following bounds hold.*

- *For any $m \le \ell$, $C_{h^\star}(m) \le H^m \cdot \prod_{k=1}^{m} (2kH) = \mathcal{O}(H^{4m})$.*
- *We have $\sum_{m \ge \ell+1} C_{h^\star}(m) \le \mathcal{O}(\ell \cdot H^{4\ell+1})$.*

Therefore, applying Lemma 5 to Eq. (7), we have the bound that

$$\mathfrak{C}(\Pi^{(\ell)}) \le \left(\sum_{m \le \ell} O(H^{4m})\right) + O(\ell \cdot H^{4\ell+1}) \le O(H^{4\ell+2}).$$

This concludes the proof of Lemma 2. □

Proof of Lemma 5. All of our upper bounds will be monotone in the value of $h^\star$, so we will prove the bounds for $C_H(m)$.

First we start with the case where $m = 0$. The trajectory τ must play $a = 0$ at all times; since there is only one such τ, we have $C_H(0) = 1$.

Now we will bound $C_H(m)$, for any $m \in \{1, \dots, \ell\}$. Observe that there are $\binom{H}{m} \le H^m$ ways to pick the tuple $(h_1, \dots, h_m)$. Now we will fix $(h_1, \dots, h_m)$ and count the contributions to $C_H(m)$ for trajectories τ which have this fixed layer decomposition, and then sum up over all possible choices of $(h_1, \dots, h_m)$.

In the MDP M, there is a unique state s_{h_1} which τ must visit. In the layers between h_1 and h_2, all trajectories are only allowed take 1 on states with index $\mathrm{idx}(s_{h_1})$, but they are not required to. Thus we can compute that the contribution to $C_{h_2}(m)$ from trajectories with the fixed layer decomposition to be at most $2H$. The reasoning is as follows. At h_1, there is exactly one (s, a) pair which is reachable by trajectories with this fixed layer decomposition, since any τ must take $a = 1$ at s_{h_1}. Subsequently we can add at most two reachable pairs in every layer $h \in \{h_1 + 1, \dots, h_2 - 1\}$ due to encountering a state $j[h]$ or $j'[h]$ where $j = \mathrm{idx}(s_{h_1})$, and at layer h_2 we must play $a = 1$, for a total of $1 + 2(h_2 - h_1 - 1) \le 2H$. Using similar reasoning the contribution to $C_{h_3}(m)$ from trajectories with this fixed layer decomposition is at most $(2H) \cdot (4H)$, and so on. Continuing in this way, we have the final bound of $\prod_{k=1}^m (2kH)$. Since this holds for a fixed choice of $(h_1, \dots, h_m)$ in total we have $C_H(m) \le H^m \cdot \prod_{k=1}^m (2kH) = \mathcal{O}(H^{4m})$.

When $m \ge \ell + 1$, observe that the block-free property on B implies that for any $J \subseteq [2^H]$ with $|J| = \ell$ we have $|\cap_{j \in J} \Pi_j| \le \ell \log 2^{2H}$. So for any trajectory τ with layer decomposition such that $m \ge \ell$ we can redo the previous analysis and argue that there is at most $\ell \log 2^{2H}$ multiplicative factor contribution to the value $C_H(m)$ due to *all* trajectories which have layer decompositions longer than ℓ. Thus we arrive at the bound $\sum_{m \ge \ell+1} C_H(m) \le \mathcal{O}(H^{4\ell}) \cdot \ell \log 2^{2H} \le \mathcal{O}(H^{4\ell+1})$.

This concludes the proof of Lemma 5. □

E.5. Proof of Lemma 3

The proof of this stopping time lemma follows standard machinery for PAC lower bounds (Garivier et al., 2019; Domingues et al., 2021; Sekhari et al., 2021). In the following we use $\mathrm{KL}(P \| Q)$ to denote the Kullback-Leibler divergence between two distributions P and Q and $\mathrm{kl}(p \| q)$ to denote the Kullback-Leibler divergence between two Bernoulli distributions with parameters $p, q \in [0, 1]$.

For any $\pi^\star \in \Pi^{(\ell)}$ we denote the random variable

$$N^{\pi^\star} = \sum_{t=1}^{\eta \wedge T_{\max}} \mathbb{1}\left\{\mathbb{A}_t(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H}) \text{ and } \mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}\right\},$$

the number of episodes for which the algorithm's policy at round $t \in [\eta \wedge T_{\max}]$ matches that of $\pi^\star$ on a certain relevant state of $\pi^\star$.

In the sequel we will prove upper and lower bounds on the intermediate quantity $\sum_{\pi^\star \in \Pi} \mathbb{E}_0\left[N^{\pi^\star}\right]$ and relate these quantities to $\mathbb{E}_0[\eta]$.

Step 1: upper bound. First we prove an upper bound. We can compute that

$$\sum_{\pi^\star \in \Pi} \mathbb{E}_0\left[N^{\pi^\star}\right]$$

$$= \sum_{t=1}^{T_{\max}} \sum_{\pi^\star \in \Pi} \mathbb{E}_0\left[\mathbb{1}\{\eta > t - 1\}\mathbb{1}\left\{\mathbb{A}_t(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H}) \text{ and } \mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}\right\}\right]$$

$$= \sum_{t=1}^{T_{\max}} \mathbb{E}_0\left[\mathbb{1}\{\eta > t-1\} \sum_{\pi^\star \in \Pi} \mathbb{1}\left\{\mathbb{A}_t(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H}) \text{ and } \mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}\right\}\right]$$

$$\overset{(i)}{\le} \sum_{t=1}^{T_{\max}} \mathbb{E}_0[\mathbb{1}\{\eta > t-1\}] \le \mathbb{E}_0[\eta \wedge T_{\max}] \le \mathbb{E}_0[\eta]. \tag{8}$$

Here, the first inequality follows because for every index j and every $\pi^\star \in \Pi_j^{(\ell)}$, each $\pi^\star$ admits a unique sequence of actions (by Property (3) of Lemma 2), so any policy $\mathbb{A}_t$ can completely match with at most one of the $\pi^\star$.

Step 2: lower bound. Now we turn to the lower bound. We use a change of measure argument.

$$\begin{aligned}
\mathbb{E}_0\left[N^{\pi^\star}\right] &\overset{(i)}{\ge} \mathbb{E}_{0,\pi^\star}\left[N^{\pi^\star}\right] - T_{\max}\Delta(T_{\max}) \\
&= \frac{1}{|\Phi|}\sum_{\phi\in\Phi} \mathbb{E}_{0,\pi^\star,\phi}\left[N^{\pi^\star}\right] - T_{\max}\Delta(T_{\max}) \\
&\overset{(ii)}{\ge} \frac{1}{7}\cdot\frac{1}{|\Phi|}\sum_{\phi\in\Phi} \mathrm{KL}\left(\mathrm{Pr}_{0,\pi^\star,\phi}^{\mathcal{F}_{\eta\wedge T_{\max}}} \,\|\, \mathrm{Pr}_{\pi^\star,\phi}^{\mathcal{F}_{\eta\wedge T_{\max}}}\right) - T_{\max}\Delta(T_{\max}) \\
&\overset{(iii)}{\ge} \frac{1}{7}\cdot \mathrm{KL}\left(\mathrm{Pr}_{0,\pi^\star}^{\mathcal{F}_{\eta\wedge T_{\max}}} \,\|\, \mathrm{Pr}_{\pi^\star}^{\mathcal{F}_{\eta\wedge T_{\max}}}\right) - T_{\max}\Delta(T_{\max})
\end{aligned}$$

The inequality (i) follows from a change of measure argument using Lemma 6, with $\Delta(T_{\max}) := {T_{\max}}^2/2^{H+3}$. Here, $\mathcal{F}_{\eta\wedge T_{\max}}$ denotes the natural filtration generated by the first $\eta \wedge T_{\max}$ episodes. The inequality (ii) follows from Lemma 7, using the fact that $M_{0,\pi^\star,\phi}$ and $M_{\pi^\star,\phi}$ have identical transitions and only differ in rewards at layer H for the trajectories which reach the end of a relevant combination lock. The number of times this occurs is exactly $N^{\pi^\star}$. The factor $1/7$ is a lower bound on $\mathrm{kl}(1/2\|3/4)$. The inequality (iii) follows by the convexity of KL divergence.

Now we apply Lemma 8 to lower bound the expectation for any $\mathcal{F}_{\eta\wedge T_{\max}}$-measurable random variable $Z \in [0,1]$ as

$$\begin{aligned}
\mathbb{E}_0\left[N^{\pi^\star}\right] &\ge \frac{1}{7}\cdot \mathrm{kl}\left(\mathbb{E}_{0,\pi^\star}[Z] \| \mathbb{E}_{\pi^\star}[Z]\right) - T_{\max}\Delta(T_{\max}) \\
&\ge \frac{1}{7}\cdot(1-\mathbb{E}_{0,\pi^\star}[Z])\log\left(\frac{1}{1-\mathbb{E}_{\pi^\star}[Z]}\right) - \frac{\log(2)}{7} - T_{\max}\Delta(T_{\max}),
\end{aligned}$$

where the second inequality follows from the bound $\mathrm{kl}(p\|q) \ge (1-p)\log(1/(1-q)) - \log(2)$ (see, e.g., Domingues et al., 2021, Lemma 15).

Now we pick $Z = Z_{\pi^\star} := \mathbb{1}\{\eta < T_{\max} \text{ and } \mathcal{E}(\pi^\star, \phi, \mathbb{A}_f(\mathcal{D}))\}$ and note that $\mathbb{E}_{\pi^\star}[Z_{\pi^\star}] \ge 1-2\delta$ by assumption. This implies that

$$\mathbb{E}_0\left[N^{\pi^\star}\right] \ge (1-\mathbb{E}_{0,\pi^\star}[Z_{\pi^\star}])\cdot\frac{1}{7}\log\left(\frac{1}{2\delta}\right) - \frac{\log(2)}{7} - T_{\max}\Delta(T_{\max}).$$

Another application of Lemma 6 gives

$$\mathbb{E}_0\left[N^{\pi^\star}\right] \ge (1-\mathbb{E}_0[Z_{\pi^\star}])\cdot\frac{1}{7}\log\left(\frac{1}{2\delta}\right) - \frac{\log(2)}{7} - \Delta(T_{\max})\left(T_{\max} + \frac{1}{7}\log\left(\frac{1}{2\delta}\right)\right).$$

Summing the above over $\pi^\star \in \Pi^{(\ell)}$, we get

$$\sum_{\pi^\star}\mathbb{E}_0\left[N^{\pi^\star}\right] \ge \left(|\Pi^{(\ell)}| - \sum_{\pi^\star}\mathbb{E}_0[Z_{\pi^\star}]\right)\cdot\frac{1}{7}\log\left(\frac{1}{2\delta}\right) - |\Pi^{(\ell)}|\cdot\frac{\log(2)}{7} - |\Pi^{(\ell)}|\cdot\Delta(T_{\max})\left(T_{\max} + \frac{1}{7}\log\left(\frac{1}{2\delta}\right)\right). \tag{9}$$

It remains to prove an upper bound on $\sum_{\pi^\star}\mathbb{E}_0[Z_{\pi^\star}]$. We calculate that

$$\sum_{\pi^\star}\mathbb{E}_0[Z_{\pi^\star}] = \sum_{\pi^\star}\mathbb{E}_0\left[\mathbb{1}\{\eta < T_{\max} \text{ and } \mathcal{E}(\pi^\star, \phi, \mathbb{A}_f(\mathcal{D}))\}\right]$$

$$\leq \sum_{\pi^\star} \mathbb{E}_0\Big[\mathbb{1}\Big\{\mathrm{Pr}_{\pi^\star}\Big[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\Big] \geq \frac{\varepsilon}{4}\Big\}\Big]$$

$$\leq \frac{4}{\varepsilon} \cdot \mathbb{E}_0\Bigg[\sum_{\pi^\star} \mathrm{Pr}_{\pi^\star}\Big[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\Big]\Bigg] \tag{10}$$

The last inequality is an application of Markov's inequality.

Now we carefully investigate the sum. For any $\phi \in \Phi$, the sum can be rewritten as

$$\begin{aligned}
&\sum_{\pi^\star} \mathrm{Pr}_{\pi^\star,\phi}\Big[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\Big] \\
=\ &\sum_{\pi^\star} \sum_{s_1 \in \mathcal{S}_1} \mathrm{Pr}_{\pi^\star,\phi}[s_1]\, \mathrm{Pr}_{\pi^\star,\phi}\Big[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H}) \ \Big|\ s_1\Big] \\
\overset{(i)}{=}\ &\frac{1}{|\mathcal{S}_1|} \sum_{s_1 \in \mathcal{S}_1} \sum_{\pi^\star} \mathrm{Pr}_{\pi^\star,\phi}\Big[\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H}) \ \Big|\ s_1\Big] \\
\overset{(ii)}{=}\ &\frac{1}{|\mathcal{S}_1|} \sum_{s_1 \in \mathcal{S}_1} \sum_{\pi^\star} \mathbb{1}\Big\{\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\Big\}.
\end{aligned} \tag{11}$$

The equality (i) follows because regardless of which MDP $M_{\pi^\star}$ we are in, the first state is distributed uniformly over $\mathcal{S}_1$. The equality (ii) follows because once we condition on the first state s_1, the probability is either 0 or 1.

Fix any start state s_1. We can write

$$\begin{aligned}
&\sum_{\pi^\star} \mathbb{1}\Big\{\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } \mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H})\pi^\star(\mathrm{idx}(s_1)_{1:H})\Big\} \\
&= \sum_{\pi^\star \in \Pi^{(\ell)}_{\mathrm{idx}(s_1)}} \mathbb{1}\{\mathbb{A}_f(\mathcal{D})(\mathrm{idx}(s_1)_{1:H}) = \pi^\star(\mathrm{idx}(s_1)_{1:H})\} = 1,
\end{aligned}$$

where the second equality uses the fact that on any index j, each $\pi^\star \in \Pi^{(\ell)}_j$ behaves differently (Property (3) of Lemma 2), so $\mathbb{A}_f(\mathcal{D})$ can match at most one of these behaviors. Plugging this back into Eq. (11), averaging over $\phi \in \Phi$, and combining with Eq. (10), we arrive at the bound

$$\sum_{\pi^\star} \mathbb{E}_0[Z_{\pi^\star}] \leq \frac{4}{\varepsilon}.$$

We now use this in conjunction with Eq. (9) to arrive at the final lower bound

$$\sum_{\pi^\star} \mathbb{E}_0\Big[N^{\pi^\star}\Big] \geq \Big(|\Pi^{(\ell)}| - \frac{4}{\varepsilon}\Big) \cdot \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big) - |\Pi^{(\ell)}| \cdot \frac{\log(2)}{7} - |\Pi^{(\ell)}| \cdot \Delta(T_{\max})\Big(T_{\max} + \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big)\Big). \tag{12}$$

Step 3: putting it all together. Combining Eqs. (8) and (12), plugging in our choice of $\Delta(T_{\max})$, and simplifying we get

$$\begin{aligned}
\mathbb{E}_0[\eta] &\geq \Big(|\Pi^{(\ell)}| - \frac{4}{\varepsilon}\Big) \cdot \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big) - |\Pi^{(\ell)}| \cdot \frac{\log(2)}{7} - |\Pi^{(\ell)}| \cdot \Delta(T_{\max})\Big(T_{\max} + \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big)\Big). \\
&\geq \Big(\frac{|\Pi^{(\ell)}|}{2} - \frac{4}{\varepsilon}\Big) \cdot \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big) - |\Pi^{(\ell)}| \cdot \frac{{T_{\max}}^2}{2^{H+3}}\Big(T_{\max} + \frac{1}{7}\log\Big(\frac{1}{2\delta}\Big)\Big).
\end{aligned}$$

The last inequality follows since $\delta \leq 1/8$ implies $\log(1/(2\delta)) \geq 2\log(2)$.

This concludes the proof of Lemma 3. □

E.6. Change of Measure Lemma

Lemma 6. *Let $Z \in [0,1]$ be a $\mathcal{F}_{T_{\max}}$-measurable random variable. Then, for every $\pi^\star \in \Pi^{(\ell)}$,*

$$|\mathbb{E}_0[Z] - \mathbb{E}_{0,\pi^\star}[Z]| \leq \Delta(T_{\max}) := \frac{{T_{\max}}^2}{2^{H+3}}$$

Proof. First we note that

$$|\mathbb{E}_0[Z] - \mathbb{E}_{0,\pi^\star}[Z]| \le \mathrm{TV}\Big(\mathrm{Pr}_0^{\mathcal{F}_{T_{\max}}}, \mathrm{Pr}_{0,\pi^\star}^{\mathcal{F}_{T_{\max}}}\Big) \le \sum_{t=1}^{T_{\max}} \mathbb{E}_0\big[\mathrm{TV}\big(\mathrm{Pr}_0[\cdot|\mathcal{F}_{t-1}], \mathrm{Pr}_{0,\pi^\star}[\cdot|\mathcal{F}_{t-1}]\big)\big].$$

Here $\mathrm{Pr}_0[\cdot|\mathcal{F}_t]$ denotes the conditional distribution of the t-th trajectory given the first $t-1$ trajectories. Similarly $\mathrm{Pr}_{0,\pi^\star}[\cdot|\mathcal{F}_t]$ is the averaged over decoders condition distribution of the t-th trajectory given the first $t-1$ trajectories. The second inequality follows by chain rule of TV distance (see, e.g., Polyanskiy and Wu, 2022, pg. 152).

Now we examine each term $\mathrm{TV}\big(\mathrm{Pr}_0[\cdot|\mathcal{F}_{t-1}], \mathrm{Pr}_{0,\pi^\star}[\cdot|\mathcal{F}_{t-1}]\big)$.

Fix a history $\mathcal{F}_{t-1}$ and sequence $s_{1:H}$ where all s_i have the same index. We want to bound the quantity

$$\Big|\mathrm{Pr}_{0,\pi^\star}\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] - \mathrm{Pr}_0\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big]\Big|,$$

where it is understood that the random variable $S_{1:H}^{(t)}$ is drawn according to the MDP dynamics and algorithm's policy $\mathbb{A}_t$ (which is in turn a measurable function of $\mathcal{F}_{t-1}$).

We observe that the second term can exactly calculated to be

$$\mathrm{Pr}_0\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] = \frac{1}{|\mathcal{S}_1|} \cdot \frac{1}{2^{H-1}},$$

since the state s_1 appears with probability $1/|\mathcal{S}_1|$ and the transitions in M_0 are uniform to the next state in the combination lock, so each sequence is equally as likely.

For the first term, again the state s_1 appears with probability $1/|\mathcal{S}_1|$. Suppose that $\mathrm{idx}(s_1) \notin \mathcal{J}_{\mathrm{rel}}^{\pi^\star}$. Then the dynamics of $\mathrm{Pr}_{0,\pi^\star,\phi}$ for all $\phi \in \Phi$ are exactly the same as M_0, so again the probability in this case is $1/(|\mathcal{S}_1| 2^{H-1})$. Now consider when $\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}$. At some point $\widehat{h} \in [H+1]$, the policy $\mathbb{A}_t$ will deviate from $\pi^\star$ for the first time (if $\mathbb{A}_t$ never deviates from $\pi^\star$ we set $\widehat{h} = H+1$). The layer $\widehat{h}$ is only a function of s_1 and $\mathbb{A}_t$ and doesn't depend on the MDP dynamics. The correct decoder must assign $\phi(s_{1:\widehat{h}-1}) = \textsc{Good}$ and $\phi(s_{\widehat{h}:H}) = \textsc{Bad}$, so therefore we have

$$\begin{aligned} &\mathrm{Pr}_{0,\pi^\star}\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] \\ &= \mathrm{Pr}_{0,\pi^\star}[\phi(s_{1:\widehat{h}-1}) = \textsc{Good} \text{ and } \phi(s_{\widehat{h}:H}) = \textsc{Bad} \mid \mathcal{F}_{t-1}] \end{aligned}$$

If $s_1 \notin \mathcal{F}_{t-1}$, i.e., we are seeing s_1 for the first time, then the conditional distribution over the labels given by ϕ is the same as the unconditioned distribution:

$$\mathrm{Pr}_{0,\pi^\star}[\phi(s_{1:\widehat{h}-1}) = \textsc{Good} \text{ and } \phi(s_{\widehat{h}:H}) = \textsc{Bad} \mid \mathcal{F}_{t-1}] = \frac{1}{|\mathcal{S}_1|} \cdot \frac{1}{2^{H-1}}.$$

Otherwise, if $s_1 \in \mathcal{F}_{t-1}$ then we bound the conditional probability by 1.

$$\mathrm{Pr}_{0,\pi^\star}\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] \le \frac{1}{|\mathcal{S}_1|}.$$

Putting this all together we can compute

$$\mathrm{Pr}_{0,\pi^\star}\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] \begin{cases} = \frac{1}{|\mathcal{S}_1|} \cdot \frac{1}{2^{H-1}} & \text{if} \quad \mathrm{idx}(s_1) \notin \mathcal{J}_{\mathrm{rel}}^{\pi^\star}, \\ = \frac{1}{|\mathcal{S}_1|} \cdot \frac{1}{2^{H-1}} & \text{if} \quad \mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } s_1 \notin \mathcal{F}_{t-1}, \\ \le \frac{1}{|\mathcal{S}_1|} & \text{if} \quad \mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star} \text{ and } s_1 \in \mathcal{F}_{t-1}, \\ = 0 & \text{otherwise.} \end{cases}$$

Therefore we have the bound

$$\Big|\mathrm{Pr}_{0,\pi^\star}\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big] - \mathrm{Pr}_0\Big[S_{1:H}^{(t)} = s_{1:H} \mid \mathcal{F}_{t-1}\Big]\Big|,$$

$$\le \frac{1}{|\mathcal{S}_1|}\mathbb{1}\Big\{\mathrm{idx}(s_1) \in \mathcal{J}_{\mathrm{rel}}^{\pi^\star}, s_1 \in \mathcal{F}_{t-1}\Big\}.$$

Summing over all possible sequences $s_{1:H}$ we have

$$\mathrm{TV}\big(\mathrm{Pr}_0[\cdot|\mathcal{F}_{t-1}], \mathrm{Pr}_{0,\pi^\star}[\cdot|\mathcal{F}_{t-1}]\big) \le \frac{1}{2}\cdot\frac{(t-1)\cdot 2^{H-1}}{|\mathcal{S}_1|},$$

since the only sequences $s_{1:H}$ for which the difference in the two measures are nonzero are the ones for which $s_1 \in \mathcal{F}_{t-1}$, of which there are $(t-1)\cdot 2^{H-1}$ of them.

Lastly, taking expectations and summing over $t = 1$ to $T_{\max}$ and plugging in the value of $|\mathcal{S}_1| = 2^{2H}$ we have the final bound. □

The next lemma is a straightforward modification of (Domingues et al., 2021, Lemma 5), with varying rewards instead of varying transitions.

Lemma 7. *Let M and M' be two MDPs that are identical in transition and differ in the reward distributions, denote $r_h(s,a)$ and $r'_h(s,a)$. Assume that for all (s,a) we have $r_h(s,a) \ll r'_h(s,a)$. Then for any stopping time η with respect to $(\mathcal{F}^t)_{t\ge 1}$ that satisfies $\mathrm{Pr}_M[\eta < \infty] = 1$,*

$$\mathrm{KL}\Big(\mathrm{Pr}_M^{I_\eta} \,\|\, \mathrm{Pr}_{M'}^{I_\eta}\Big) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{h\in[H]} \mathbb{E}_M[N_{s,a,h}^\eta]\cdot \mathrm{KL}\big(r_h(s,a)\|r'_h(s,a)\big),$$

where $N_{s,a,h}^\eta := \sum_{t=1}^{\eta}\mathbb{1}\Big\{(S_h^{(t)}, A_h^{(t)}) = (s,a)\Big\}$ and $I_\eta : \Omega \mapsto \bigcup_{t\ge 1}\mathcal{I}_t : \omega \mapsto I_{\eta(\omega)}(\omega)$ is the random vector representing the history up to episode η.

Lemma 8 (Lemma 1, (Garivier et al., 2019))**.** *Consider a measurable space $(\Omega, \mathcal{F})$ equipped with two distributions $\mathbb{P}_1$ and $\mathbb{P}_2$. For any $\mathcal{F}$-measurable function $Z : \Omega \mapsto [0,1]$ we have*

$$\mathrm{KL}(\mathbb{P}_1\|\mathbb{P}_2) \ge \mathrm{kl}(\mathbb{E}_1[Z]\|\mathbb{E}_2[Z])$$

F. Proofs for Section 6

F.1. Algorithmic Details and Preliminaries

In this subsection, we provide the details of the subroutines that do not appear in the main body, in Algorithm 3, Algorithm 4 and Algorithm 5. The reward function in line 5 in Algorithm 5 is computed using (17), which is specified below, after introducing additional notation.

Algorithm 3 DataCollector

Require: State: s, Reacher policy: π_s, Exploration policy: Π_{core}, Number of samples: n.

/* Uniform sampling for start state $s_\top$ */

if $s = s_\top$ **then**
    for $t = 1, \dots, n$ **do**
        Sample $\pi' \sim \mathrm{Uniform}(\Pi_{\mathrm{core}})$, and run to collect $\tau = \{s_1, a_1, \dots, s_H, a_H\}$.
        $\mathcal{D}_s \leftarrow \mathcal{D}_s \cup \{\tau\}$.
    end for
else
    /* π_s based sampling for all other states $s \neq s_\top$ */
    Identify the layer h such that $s \in \mathcal{S}_h$.
    for $t = 1, \dots, n$ **do**
        Run π_s for the first $h - 1$ time steps, and collect trajectory $\{s_1, a_1, \dots, s_{h-1}, a_{h-1}, s_h\}$.
        if $s_h = s$ **then**
            Sample $\pi' \sim \mathrm{Uniform}(\Pi_{\mathrm{core}})$, and run to collect remaining $\{s_h, a_h, \dots, s_H, a_H\}$.
            $\mathcal{D}_s \leftarrow \mathcal{D}_s \cup \{\tau = \{s_1, a_1, \dots, s_H, a_H\}\}$.
        end if
    end for
end if
Return dataset $\mathcal{D}_s$.

Algorithm 4 DP_Solver

Require: State space S^{tab}, Transition P, State $\bar{s} \in \mathcal{S}^{\mathrm{tab}}$.

Initialize $V(s) = \mathbb{1}\{s = \bar{s}\}$ for all $s \in S^{\mathrm{tab}}$.
Repeat $H + 1$ times:
    For all $s \in S^{\mathrm{tab}}$, calculate $V(s) \leftarrow \sum_{s' \in \mathcal{S}^{\mathrm{tab}}} P_{s \to s'} \cdot V(s')$. // Dynamic Programming
Return $V(s_\top)$.

We recall the definition of Petals and Sunflowers in Definition 3. In the rest of this section, we assume that Π is a (K, D)-sunflower with Π_{core} and $\mathcal{S}_\pi$ for any $\pi \in \Pi$.

Definition 5 (Petals and Sunflowers). *For a policy π, policy set $\bar{\Pi}$, and states $\bar{\mathcal{S}} \subseteq \mathcal{S}$, π is said to be a $\bar{\mathcal{S}}$-petal on $\bar{\Pi}$ if for all $h \le h' \le H$, and partial trajectories $\tau = (s_h, a_h, \cdots, s_{h'}, a_{h'})$ that are consistent with π: either τ is also consistent with some $\pi' \in \bar{\Pi}$, or there exists $i \in (h, h']$ s.t. $s_i \in \bar{\mathcal{S}}$.*

A policy class Π is said to be a (K, D)-sunflower if there exists a set Π_{core} of Markovian policies with $|\Pi_{\mathrm{core}}| \le K$ such that for every policy $\pi \in \Pi$ there exists a set $\mathcal{S}_\pi \subseteq \mathcal{S}$, of size at most D, so that π is S_π-petal on Π_{core}.

Additional notation. Recall that we assumed that the state space is layered. Thus, given a state s, we can infer the layer h such that $s \in \mathcal{S}_h$. In the following, we define additional notation:

(a) *Sets* $\mathfrak{T}_\pi(s \to s')$: For any policy π, and states $s, s' \in \mathcal{S}$, we define $\mathfrak{T}_\pi(s \to s')$ as the set of all the trajectories that are consistent with π, and that go from s to s' without passing through any state in $\mathcal{S}_\pi$ in between.

More formally, let $\pi \in \Pi$, state s be at layer h, and s' be at layer h'. Then, $\mathfrak{T}_\pi(s \to s')$ denotes the set of all the trajectories $\tau = (s_1, a_1, \dots, s_H, a_H)$ that satisfy all of the following:

- τ is consistent with π, i.e. $\pi \rightsquigarrow \tau$.

Algorithm 5 Evaluate

Require: Policy set Π_{core}, Reachable states $\mathcal{I}$, Datasets $\{\mathcal{D}_s\}_{s \in \mathcal{I}}$, Policy π to be evaluated.

1: Compute $\mathcal{S}^{\text{rch}}_{\pi} \leftarrow \mathcal{S}^{+}_{\pi} \cap \mathcal{I}$ and $S^{\text{tab}} = \mathcal{S}^{\text{rch}}_{\pi} \cup \{s_\perp\}$.
2: **for** s, s' in S^{tab} **do**
3:     `/* Compute transitions and rewards on` S^{tab} `*/`
4:     Let h, h' be such that $s \in \mathcal{S}_h$ and $s' \in \mathcal{S}_{h'}$
5:     **if** $h < h'$ **then**
6:         Calculate $\widehat{P}^{\pi}_{s \to s'}, \widehat{r}^{\pi}_{s \to s'}$ according to (16) and (17);
7:     **else**
8:         Set $\widehat{P}^{\pi}_{s \to s'} \leftarrow 0$, $\widehat{r}^{\pi}_{s \to s'} \leftarrow 0$.
9:     **end if**
10: **end for**
11: Set $\widehat{V}(s) = 0$ for all $s \in S^{\text{tab}}$.
12: **Repeat** for $H + 1$ times: `// Evaluate` π `by dynamic programming`
13:     For all $s \in S^{\text{tab}}$, calculate $\widehat{V}(s) \leftarrow \sum_{S^{\text{tab}}} \widehat{P}^{\pi}_{s \to s'} \cdot \left(\widehat{r}^{\pi}_{s \to s'} + \widehat{V}(s')\right)$.
14: **Return** $\widehat{V}(s_\top)$.

- $s_h = s$, where s_h is the state at timestep h in τ.
- $s_{h'} = s'$, where $s_{h'}$ is the state at timestep h' in τ.
- For all $h < \tilde{h} < h'$, the state $s_{\tilde{h}}$, at time step $\tilde{h}$ in τ, does not lie in the set $\mathcal{S}_\pi$.

Note that when $h' \le h$, we define $\mathfrak{T}_\pi(s \to s') = \emptyset$. Additionally, we define $\mathfrak{T}_\pi(s_\top \to s')$ as the set of all trajectories consistent with π that go to s' (from a start state) without going through any state in $\mathcal{S}_\pi$ in between. Finally, we define $\mathfrak{T}_\pi(s \to s_\perp)$ as the set of all the trajectories that are consistent with π and go from s at time step h to the end of the episode without passing through any state in $\mathcal{S}_\pi$ in between.

(b) *Sets* $\mathsf{T}(s \to s'; \neg\bar{\mathcal{S}})$: For any set $\bar{\mathcal{S}}$, and states $s, s' \in \mathcal{S}$, we define $\mathsf{T}(s \to s'; \neg\bar{\mathcal{S}})$ as the set of all the trajectories that go from s to s' without passing through any state in $\bar{\mathcal{S}}$ in between.

More formally, let state s be at layer h, and s' be at layer h'. Then, $\mathsf{T}(s \to s'; \neg\bar{\mathcal{S}})$ denotes the set of all the trajectories $\tau = (s_1, a_1, \dots, s_H, a_H)$ that satisfy all of the following:

- $s_h = s$, where s_h is the state at timestep h in τ.
- $s_{h'} = s'$, where $s_{h'}$ is the state at timestep h' in τ.
- For all $h < \tilde{h} < h'$, the state $s_{\tilde{h}}$, at time step $\tilde{h}$ in τ, does not lie in the set $\bar{\mathcal{S}}$.

Note that when $h' \le h$, we define $\mathsf{T}(s \to s'; \neg\bar{\mathcal{S}}) = \emptyset$. Additionally, we define $\mathsf{T}(s_\top \to s; \neg\bar{\mathcal{S}})$ as the set of all trajectories that go to s' (from a start state) without going through any state in $\bar{\mathcal{S}}$ in between. Finally, we define $\mathsf{T}(s \to s_\perp; \neg\bar{\mathcal{S}})$ as the set of all the trajectories that go from s at time step h to the end of the episode without passing through any state in $\bar{\mathcal{S}}$ in between.

(c) Using the above notation, for any $s \in \mathcal{S}$ and set $\bar{\mathcal{S}} \subseteq \mathcal{S}$, we define $\bar{d}^{\pi}(s; \bar{\mathcal{S}})$ as the probability of reaching s (from a start state) without passing through any state in $\bar{\mathcal{S}}$ in between, i.e.

$$\begin{aligned} \bar{d}^{\pi}(s; \bar{\mathcal{S}}) &= \Pr^{\pi, M}\left(\tau \text{ reaches } s \text{ without passing through any state in } \bar{\mathcal{S}} \text{ before reaching } s\right) \\ &= \Pr^{\pi, M}\left(\tau \ \in \mathsf{T}(\top \to s; \neg\bar{\mathcal{S}})\right) \end{aligned} \tag{13}$$

We next define the empirical rewards that are calculated in line 5 in Algorithm 5.

Markov Reward Process (MRP). A Markov Reward Process $\mathfrak{M} = \mathrm{MRP}(\mathcal{S}, P, r, H, s_\top, s_\perp)$ is defined over the state space $\mathcal{S}$, with the transition kernel P, reward kernel r, start state $s_\top$, end state $s_\perp$ and trajectory length $H + 2$. Without loss of generality, we assume $\{s_\top, s_\perp\} \in \mathcal{S}$.

A trajectory in $\mathfrak{M}$ is of the form $\tau = (s_\top, s_1, \dots, s_H, s_\perp)$, where $s_h \in \mathcal{S}$ for all $h \in [H]$. From any state $s \in \mathcal{S}$, the MRP transitions[5] to another state $s' \in \mathcal{S}$ with probability $P_{s \to s'}$, and obtains the rewards $r_{s \to s'}$. Thus,

$$\Pr^{\mathfrak{M}}(\tau) = P_{s_\top \to s_1} \cdot \prod_{h=1}^{H-1} P_{s_h \to s_{h+1}} \cdot P_{s_H \to s_\perp},$$

and the rewards

$$R^{\mathfrak{M}}(\tau) = r_{s_\top \to s_1} + \sum_{h=1}^{H} r_{s_h \to s_{h+1}} + r_{s_H \to s_\perp}.$$

Furthermore, in MRP, we have $P_{s_\perp \to s_\perp} = 1$ and $r_{s_\perp \to s_\perp} = 0$.

Policy-Specific Markov Reward Processes. For the rest of the proofs, we will be defining various policy-specific Markov Reward Processes corresponding to different sets $\mathcal{I}$. Given a set $\mathcal{I}$ such that $s_\top \in \mathcal{I}$ but $s_\perp \notin \mathcal{I}$, recall that for any policy π, we have $\mathcal{S}_\pi^+ = \mathcal{S}_\pi \cup \{s_\top, s_\perp\}$, $\mathcal{S}_\pi^{\mathrm{rch}} = \mathcal{S}_\pi^+ \cap \mathcal{I}$ and $\mathcal{S}_\pi^{\mathrm{rem}} = \mathcal{S}_\pi^+ \setminus \mathcal{S}_\pi^{\mathrm{rch}}$.

We define the Markov Reward Process $\mathfrak{M}_{\mathcal{I}}^\pi = \mathrm{MRP}(\mathcal{S}_\pi^+, P^\pi, r^\pi, H, s_\top, s_\perp)$ where

- *Transition Kernel P^π:* For any $s \in \mathcal{S}_\pi^{\mathrm{rch}}$ and $s' \in \mathcal{S}_\pi^+$, we have

$$P_{s \to s'}^\pi = \mathbb{E}_{\tau \sim \pi}\left[\mathbb{1}\left\{\tau \in \mathfrak{T}_\pi(s \to s')\right\} \middle| s_h = s\right], \tag{14}$$

  where the expectation above is w.r.t. the trajectories drawn using π in the underlying MDP, and h denotes the time step such that $s_h \in \mathcal{S}_h$ (again, in the underying MDP). This transition $P_{s \to s'}^\pi$ denotes the probability of taking policy π from s and directly transiting to s' without touching any other states in $\mathcal{S}_\pi$. Furthermore, $P_{s \to s'}^\pi = \mathbb{1}\left\{s' = s_\perp\right\}$ for all $s \in \mathcal{S}_\pi^{\mathrm{rem}} \cup \{s_\perp\}$.

- *Reward Kernel r^π:* For any $s \in \mathcal{S}_\pi^{\mathrm{rch}}$ and $s' \in \mathcal{S}_\pi^+$, we have

$$r_{s \to s'}^\pi \triangleq \mathbb{E}_{\tau \sim \pi}\left[R(\tau_{h:h'})\mathbb{1}\left\{\tau \in \mathfrak{T}_\pi(s \to s')\right\} \middle| s_h = s\right] \tag{15}$$

  where the expectation above is w.r.t. the trajectories drawn using π in the underlying MDP, $R(\tau_{h:h'})$ denotes the reward for the partial trajectory $\tau_{h:h'}$ in the underlying MDP, and h denotes the time step such that $s_h \in \mathcal{S}_h$ (again, in the underying MDP). The reward $r_{s \to s'}^\pi$ denotes the expectation of rewards collected by taking policy π from s and directly transiting to s' without touching any other states in $\mathcal{S}_\pi$. Furthermore, $r_{s \to s'}^\pi = 0$ for all $s \in \mathcal{S}_\pi^{\mathrm{rem}} \cup \{s_\perp\}$.

Since the learner only has sampling access to the underlying MDP, it can not directly construct the MRP $\mathfrak{M}_{\mathcal{I}}^\pi$. Instead, in Algorithm 1, the learner constructs the following empirical MRP.

Empirical Versions of Policy Specific MRPs Given a set $\mathcal{I}$ such that $s_\top \in \mathcal{I}$ but $s_\perp \notin \mathcal{I}$, recall that, for any policy π, $\mathcal{S}_\pi^+ = \mathcal{S}_\pi \cup \{s_\top, s_\perp\}$, $\mathcal{S}_\pi^{\mathrm{rch}} = \mathcal{S}_\pi^+ \cap \mathcal{I}$ and $\mathcal{S}_\pi^{\mathrm{rem}} = \mathcal{S}_\pi^+ \setminus \mathcal{S}_\pi^{\mathrm{rch}}$.

In Algorithm 1, we define an empirical Markov Reward Process $\widehat{\mathfrak{M}}_{\mathcal{I}}^\pi = \mathrm{MRP}(\mathcal{S}_\pi^+, \widehat{P}^\pi, \widehat{r}^\pi, H, s_\top, s_\perp)$ where

- *Transition Kernel $\widehat{P}^\pi$:* For any $s \in \mathcal{S}_\pi^{\mathrm{rch}}$ and $s' \in \mathcal{S}_\pi^+$, we have

$$\widehat{P}_{s \to s'}^\pi = \frac{|\Pi_{\mathrm{core}}|}{|\mathcal{D}_s|} \sum_{\tau \in \mathcal{D}_s} \frac{\mathbb{1}\{\pi \rightsquigarrow \tau_{h:h'}\}}{\sum_{\pi' \in \Pi_{\mathrm{core}}} \mathbb{1}\{\pi_e \rightsquigarrow \tau_{h:h'}\}} \mathbb{1}\left\{\tau \in \mathfrak{T}_\pi(s \to s')\right\} \tag{16}$$

  where Π_{core} denotes the core of the sunflower corresponding to Π and $\mathcal{D}_s$ denotes a dataset of trajectories collected via $\mathsf{DataCollector}(s, \pi_s, \Pi_{\mathrm{core}}, n_2)$. Furthermore, $\widehat{P}_{s \to s'}^\pi = \mathbb{1}\left\{s' = s_\perp\right\}$ for all $s \in \mathcal{S}_\pi^{\mathrm{rem}} \cup \{s_\perp\}$.

[5]Our definition of Markov Reward Processes (MRP) deviates from MDPs that we considered in the paper, in the sense that we do not assume that the state space $\mathcal{S}$ is layered in an MRP. This variation is only adapted to simplify the proofs and the notation in the rest of the paper.

- *Reward Kernel* $\widehat{r}^{\pi}$*:* For any $s \in \mathcal{S}_\pi^{\mathrm{rch}}$ and $s' \in \mathcal{S}_\pi^+$, we have

$$\widehat{r}^{\pi}_{s\to s'} = \frac{|\Pi_{\mathrm{core}}|}{|\mathcal{D}_s|} \sum_{\tau \in \mathcal{D}_s} \frac{\mathbb{1}\{\pi \rightsquigarrow \tau_{h:h'}\}}{\sum_{\pi' \in \Pi_{\mathrm{core}}} \mathbb{1}\{\pi_e \rightsquigarrow \tau_{h:h'}\}} \mathbb{1}\Big\{\tau \in \mathfrak{T}_\pi(s \to s')\Big\} R(\tau_{h:h'}), \tag{17}$$

where Π_{core} denotes the core of the sunflower corresponding to Π, $\mathcal{D}_s$ denotes a dataset of trajectories collected via $\mathsf{DataCollector}(s, \pi_s, \Pi_{\mathrm{core}}, n_2)$, and $R(\tau_{h:h'}) = \sum_{i=h}^{h'-1} r_i$. Furthermore, $\widehat{r}^{\pi}_{s\to s'} = 0$ for all $s \in \mathcal{S}_\pi^{\mathrm{rem}}$.

Parameters Used in Algorithm 1. Here, we list all the parameters that are used in Algorithm 1, and its subroutines:

$$\begin{aligned} N_1 &= \frac{C_1 (D+1)^4 K^2 \log(|\Pi|(D+1)/\delta)}{\varepsilon^2}, \\ N_2 &= \frac{C_2 D^3 (D+1)^2 K^2 \log(|\Pi|(D+1)^2/\delta)}{\varepsilon^3}, \end{aligned} \tag{18}$$

F.2. Supporting Technical Results

We start by looking at the following variant of the classical simulation lemma (Kearns and Singh, 2002; Agarwal et al., 2019; Foster et al., 2021a).

Lemma 9 (Simulation lemma (Foster et al., 2021a, Lemma F.3))**.** *Let $M = (\mathcal{S}, P, r, H, s_\top, s_\perp)$ and $\widehat{M} = (\mathcal{S}, \widehat{P}, \widehat{r}, H, s_\top, s_\perp)$ be two Markov Reward Processes. Then we have*

$$|V - \widehat{V}| \le \sum_{s \in \mathcal{S}} d_M(s) \cdot \left(\sum_{s' \in \mathcal{S}} |P_{s\to s'} - \widehat{P}_{s\to s'}| + |r_{s\to s'} - \widehat{r}_{s\to s'}| \right),$$

where $d_M(s)$ is the probability of reaching s under M, and V and $\widehat{V}$ denotes the value of $s_\top$ under M and $\widehat{M}$ respectively.

Lemma 10. *Let Algorithm 1 be run with the parameters given in (18), and consider any iteration of the while loop in line 1 with the instantaneous set $\mathcal{I}$. Further, suppose that $|\mathcal{D}_s| \ge \varepsilon N_2 / 24D$ for all $s \in \mathcal{I}$. Then, with probability at least $1-\delta$, the following hold:*

(a) *For all $\pi \in \Pi$, $s \in \mathcal{S}_\pi^{\mathrm{rch}}$ and $s' \in \mathcal{S}_\pi \cup \{s_\perp\}$,*

$$\max\Big\{|P^{\pi}_{s\to s'} - \widehat{P}^{\pi}_{s\to s'}|, |r^{\pi}_{s\to s'} - \widehat{r}^{\pi}_{s\to s'}|\Big\} \le \frac{\varepsilon}{12D(D+1)}.$$

(b) *For all $\pi \in \Pi$ and $s' \in \mathcal{S}_\pi \cup \{s_\perp\}$,*

$$\max\{|P^{\pi}_{s_\top\to s'} - \widehat{P}^{\pi}_{s_\top\to s'}|, |r^{\pi}_{s_\top\to s'} - \widehat{r}^{\pi}_{s_\top\to s'}|\} \le \frac{\varepsilon}{12(D+1)^2}.$$

Proof. We first prove the bound for $s \in \mathcal{S}_\pi^{\mathrm{rch}}$. Let s be at layer h. Fix any policy $\pi \in \Pi$, and consider any state $s' \in \mathcal{S}_\pi \cup \{s_\perp\}$, where s' is at layer h'. Note that since Π is a (K, D)-sunflower, with its core Π_{core} and petals $\{\mathcal{S}_\pi\}_{\pi\in\Pi}$, we must have that any trajectory $\tau \in \mathfrak{T}_\pi(s \to s')$ is also consistent with at least one $\pi_e \in \Pi_{\mathrm{core}}$. Furthermore, for any such π_e, we have

$$\begin{aligned} \Pr^{\pi_e}(\tau_{h:h'} \mid s_h = s) &= \prod_{i=h}^{h'-1} \Pr(s_{i+1} \mid s_i, \pi_e(s_i), s_h = s) \\ &= \prod_{i=h}^{h'-1} \Pr(s_{i+1} \mid s_i, \pi(s_i), s_h = s) = \Pr^{\pi}(\tau_{h:h'} \mid s_h = s), \end{aligned} \tag{19}$$

where the second line holds because both $\pi \rightsquigarrow \tau_{h:h'}$ and $\pi_e \rightsquigarrow \tau_{h:h'}$. Next, recall from (14), that

$$P^{\pi}_{s\to s'} = \mathbb{E}^{\pi}\Big[\mathbb{1}\Big\{\tau \in \mathfrak{T}_\pi(s \to s')\Big\} \mid s_h = s\Big]. \tag{20}$$

Furthermore, from (16), recall that the empirical estimate $\widehat{P}^{\pi}_{s\to s'}$ of $P^{\pi}_{s\to s'}$ is given by :

$$\widehat{P}^{\pi}_{s\to s'} = \frac{1}{|\mathcal{D}_s|}\sum_{\tau\in\mathcal{D}_s}\frac{\mathbb{1}\left\{\tau\in\mathfrak{T}_\pi(s\to s')\right\}}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}}, \tag{21}$$

where the dataset $\mathcal{D}_s$ consists of i.i.d. samples, and is collected in lines 3-3 in Algorithm 3 (DataCollector), by first running the policy π_s for h timesteps and if the trajectory reaches s, then executing $\pi_e\sim\mathrm{Unif}(\Pi_{\mathrm{core}})$ for the remaining time steps (otherwise this trajectory is rejected). Let the law of this process be q. We thus note that,

$$\begin{aligned}
\mathbb{E}\Big[\widehat{P}^{\pi}_{s\to s'}\Big] &= \mathbb{E}_{\tau\sim q}\left[\frac{\mathbb{1}\left\{\tau\in\mathfrak{T}_\pi(s\to s')\right\}}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}}\mid s_h=s\right] \\
&= \sum_{\tau_{h:h'}\in\mathfrak{T}_\pi(s\to s')}\Pr_q(\tau_{h:h'}\mid s_h=s)\cdot\frac{1}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}} \\
&\overset{(i)}{=} \sum_{\tau_{h:h'}\in\mathfrak{T}_\pi(s\to s')}\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\Pr^{\pi_e}(\tau_{h:h'}\mid s_h=s)\cdot\frac{1}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}} \\
&\overset{(ii)}{=} \sum_{\tau_{h:h'}\in\mathfrak{T}_\pi(s\to s')}\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\Pr^{\pi}(\tau_{h:h'}\mid s_h=s)\cdot\frac{\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}} \\
&= \sum_{\tau_{h:h'}\in\mathfrak{T}_\pi(s\to s')}\Pr^{\pi}(\tau_{h:h'}\mid s_h=s) \\
&\overset{(iii)}{=} \mathbb{E}^{\pi}\Big[\mathbb{1}\left\{\tau\in\mathfrak{T}_\pi(s\to s')\right\}\mid s_h=s\Big] = P^{\pi}_{s\to s'},
\end{aligned}$$

where (i) follows from the sampling strategy in Algorithm 3 after observing $s_h=s$, and (ii) simply uses the relation (19). Finally, in (iii), we use the relation in (20).

The above implies that $\widehat{P}^{\pi}_{s\to s'}$ is an unbiased estimate of $P^{\pi}_{s\to s'}$ for any π and $s,s'\in\mathcal{S}^+_\pi$. Thus, using Hoeffding's inequality, followed by a union bound, we get that with probability at least $1-\delta/4$, for all $\pi\in\Pi$, $s\in\mathcal{S}^{\mathrm{rch}}_\pi$, and $s'\in\mathcal{S}_\pi\cup\{s_\perp\}$,

$$|\widehat{P}^{\pi}_{s\to s'}-P^{\pi}_{s\to s'}|\le|\Pi_{\mathrm{core}}|\sqrt{\frac{2\log(4|\Pi|D(D+1)/\delta)}{|\mathcal{D}_s|}},$$

where the additional factor of $|\Pi_{\mathrm{core}}|$ in the above appears because for any $\tau\in\mathfrak{T}_\pi(s\to s')$, there must exist some $\pi_e\in\Pi_{\mathrm{core}}$ that is also consistent with τ (as we showed above), which implies that each of the terms in (21) satisfies the bound:

$$\left|\frac{\mathbb{1}\left\{\tau\in\mathfrak{T}_\pi(s\to s')\right\}}{\frac{1}{|\Pi_{\mathrm{core}}|}\sum_{\pi_e\in\Pi_{\mathrm{core}}}\mathbb{1}\{\pi_e\rightsquigarrow\tau_{h:h'}\}}\right|\le|\Pi_{\mathrm{core}}|\le K.$$

Since $|\mathcal{D}_s|\ge{}^{\varepsilon N_2}/_{24D}$, the above implies that

$$|\widehat{P}^{\pi}_{s\to s'}-P^{\pi}_{s\to s'}|\le K\sqrt{\frac{48D\log(4|\Pi|D(D+1)/\delta)}{\varepsilon N_2}}.$$

Repeating the above for the empirical reward estimation in (16), we get that with probability at least $1-\delta/4$, for all $\pi\in\Pi$, and $s\in\mathcal{S}^{\mathrm{rch}}_\pi$ and $s'\in\mathcal{S}_\pi\cup\{s_\perp\}$, we have that

$$|\widehat{r}^{\pi}_{s\to s'}-r^{\pi}_{s\to s'}|\le K\sqrt{\frac{48D\log(4|\Pi|D(D+1)/\delta)}{\varepsilon N_2}}.$$

Similarly, we can also get for any $\pi \in \Pi$ and $s' \in \mathcal{S}_\pi \cup \{s_\perp\}$, with probability at least $1 - \delta/2$,

$$\max\left\{|\widehat{r}^\pi_{s_\top \to s'} - r^\pi_{s_\top \to s'}|, |\widehat{P}^\pi_{s_\top \to s'} - P^\pi_{s_\top \to s'}|\right\} \le K\sqrt{\frac{2\log(4|\Pi|(D+1)/\delta)}{|\mathcal{D}_{s_\top}|}}$$
$$= K\sqrt{\frac{2\log(4|\Pi|(D+1)/\delta)}{N_1}},$$

where the last line simply uses the fact that $|\mathcal{D}_{s_\top}| = N_1$. The final statement is due to a union bound on the above results.

□

Lemma 11. *Fix a policy π, for any $s \in \mathcal{S}^{\mathrm{rem}}_\pi$, if we use $d^{\mathfrak{M}_1}(s)$ to denote the occupancy of state s in $\mathfrak{M}_1$, and $\bar{d}^\pi(s; \mathcal{S}^{\mathrm{rem}}_\pi)$ is defined in (13), then we have*

$$d^{\mathfrak{M}_1}(s) = \bar{d}^\pi(s; \mathcal{S}^{\mathrm{rem}}_\pi).$$

Proof. We use $\bar{\tau}$ to denote a trajectory in $\mathfrak{M}_1$ and τ to denote a trajectory in the original MDP M, then we have

$$\begin{aligned}
d^{\mathfrak{M}_1}(s) &= \sum_{\bar{\tau} \text{ s.t. } s \in \bar{\tau}} P^{\mathfrak{M}_1}(\bar{\tau}) \\
&= \sum_{s_{h_1} = s_\top, s_{h_2}, \cdots, s_{h_t} \in \mathcal{S}^{\mathrm{rch}}_\pi} P^{\mathfrak{M}_1}(\bar{\tau} \triangleq (s_\top, s_{h_2}, \cdots, s_{h_t}, s)) \\
&= \sum_{s_{h_1} = s_\top, s_{h_2}, \cdots, s_{h_t} \in \mathcal{S}^{\mathrm{rch}}_\pi} P^{\pi, M}(\tau : \tau \cap \mathcal{S}_\pi = \{s_{h_2}, \cdots, s_{h_t}, s\}) \\
&= \sum_{s_{h_1} = s_\top, s_{h_2}, \cdots, s_{h_t} \in \mathcal{S}^{\mathrm{rch}}_\pi} \prod_{i=1}^{t} P^{\pi, M}(\tau : \tau \in \mathsf{T}(s_{h_i} \to s_{h_{i+1}}; \neg \mathcal{S}_\pi) | \tau[h_i] = s_{h_i}) \qquad (s_{h_{t+1}} \triangleq s) \\
&= P^{\pi, M}(\tau : \tau[h_i] = s_{h_i}, \forall 1 \le i \le t+1, \tau[h] \notin \mathcal{S}_\pi, \forall h \ne h_1, \cdots, h_t) \\
&= P^{\pi, M}(\tau : s \in \tau, s_h \notin \mathcal{S}^{\mathrm{rem}}_\pi, \forall 1 \le h \le h_{t+1}) = P^{\pi, M}[\tau : \tau \in \mathsf{T}(s_\top \to s; \neg \bar{S})] \\
&= \bar{d}^\pi(s; \mathcal{S}^{\mathrm{rem}}_\pi).
\end{aligned}$$

□

Lemma 12. *With probability at least $1 - 2\delta$, any (s, π) that is added into $\mathcal{T}$ (in line 1 in Algorithm 1) satisfies $d^\pi(s) \ge \varepsilon/12D$.*

Proof. Note that, for any $(s, \pi) \in \mathcal{T}$, when we collect $\mathcal{D}_s$ in Algorithm 3, the probability that a trajectory will be accepted, i.e. the trajectory would satisfy the "if" condition in line 3, is exactly $d^\pi(s)$. Thus, using Hoeffding inequality, we get that with probability at least $1 - \frac{\delta}{D|\Pi|}$,

$$\left|\frac{|\mathcal{D}_s|}{N_2} - d^\pi(s)\right| \le \sqrt{\frac{2\log(D|\Pi|/\delta)}{N_2}}.$$

Since $|\mathcal{T}| \le D|\Pi|$, taking the union bound over all $(s, \pi) \in \mathcal{T}$, we get that the above holds for all $\pi \in \Pi$, $s \in \mathcal{S}_\pi$, and $\mathcal{D}_s$ with probability at least $1 - \delta$. The above implies that for any s, for which $d^\pi(s) \ge \varepsilon/12D$, we must have that

$$|\mathcal{D}_s| \ge N_2 d^\pi(s) - \sqrt{2N_2 \log(D|\Pi|/\delta)} \ge \frac{\varepsilon N_2}{12D} - \frac{\varepsilon N_2}{24D} = \frac{\varepsilon N_2}{24D}, \tag{22}$$

where the second inequality follows by the bound on $d^\pi(s)$, and our choice of parameter N_2 in (18).

In the following, we prove by induction that every (s, π) that is added into $\mathcal{T}$ in the while loop from lines 1-1 satisfies $d^\pi(s) \ge \varepsilon/12D$. This is trivially true at initialization $\mathcal{T} = \{(s_\top, \mathrm{Null})\}$, since every trajectory starts from $s_\top$ which implies that $d^{\mathrm{Null}}(s_\top) = 1$.

We now proceed to the induction hypothesis. Suppose that in some iteration of the while loop, all tuples (s, π) that are already in $\mathcal{T}$ satisfy $d^\pi(s) \ge \varepsilon/12D$, and that $(\bar{s}, \bar{\pi})$ is a new tuple that will be added to $\mathcal{T}$. We will show that $(\bar{s}, \bar{\pi})$ will also satisfy $d^{\bar{\pi}}(\bar{s}) \ge \varepsilon/12D$.

Recall that $\mathcal{S}_{\bar\pi}^{+} = \mathcal{S}_{\bar\pi} \cup \{s_\top, s_\perp\}$, $\mathcal{S}_{\bar\pi}^{\mathrm{rch}} = \mathcal{S}_{\bar\pi}^{+} \cap \mathcal{I}$, and $\mathcal{S}_{\bar\pi}^{\mathrm{rem}} = \mathcal{S}_{\bar\pi}^{+} \setminus \mathcal{S}_{\bar\pi}^{\mathrm{rch}}$. Let $\mathfrak{M}_1 = \mathrm{MRP}(\mathcal{S}_{\bar\pi}^{+}, P^{\bar\pi}, r^{\bar\pi}, H, s_\top, s_\perp)$ be a tabular Markov Reward Process, where $P^{\bar\pi}$ and $r^{\bar\pi}$ are defined in (14) and (15) respectively, for the policy $\bar\pi$. Note that for any state $s \in \mathcal{S}_{\bar\pi}^{\mathrm{rch}}$, the bound in (22) holds, using (a) in Lemma 10, we get that

$$|P^{\bar\pi}_{s\to s'} - \widehat{P}^{\bar\pi}_{s\to s'}| \le \frac{\varepsilon}{12D(D+1)}, \qquad \text{for all} \qquad s' \in \mathcal{S}_{\bar\pi} \cup \{s_\perp\}. \tag{23}$$

Therefore, noticing that $\widehat{d}^{\bar\pi}(\bar s) \leftarrow \mathsf{DP\_Solver}(\mathcal{S}_{\bar\pi}^{+}, \widehat{P}^{\bar\pi}, \bar s)$, and also $P^{\bar\pi}_{s\to s'}$ is the transition function of MRP $M^{\bar\pi}_{\mathrm{tab}}$, according to to Lemma 9, we have

$$\begin{aligned} |\widehat{d}^{\bar\pi}(\bar s) - d^{\mathfrak{M}_1}(\bar s)| &\le \sup_{s \in \mathcal{S}_{\bar\pi}^{\mathrm{rch}}} (D+1) \cdot \sup_{s' \in \mathcal{S}_{\bar\pi} \cup \{s_\perp\}} |\widehat{P}^{\bar\pi}_{s\to s'} - P^{\bar\pi}_{s\to s'}| \\ &\le \frac{\varepsilon}{12D(D+1)} \cdot (D+1) = \frac{\varepsilon}{12D} \end{aligned} \tag{24}$$

where the second inequality follows from (23). Additionally, Lemma 11 indicates that $d^{\mathfrak{M}_1}(\bar s) = \bar d^{\bar\pi}(\bar s; \mathcal{S}_{\bar\pi}^{\mathrm{rem}})$. Therefore we obtain

$$|\bar d^{\bar\pi}(\bar s; \mathcal{S}_{\bar\pi}^{\mathrm{rem}}) - \widehat{d}^{\bar\pi}(\bar s)| \le \frac{\varepsilon}{12D}.$$

Hence if a new state-policy pair $(\bar s, \bar\pi)$ is added into $\mathcal{T}$, we will have

$$\bar d^{\pi}(\bar s; \mathcal{S}_{\pi}^{\mathrm{rem}}) \ge \frac{\varepsilon}{6D} - \frac{\varepsilon}{12D} = \frac{\varepsilon}{12D}.$$

Noticing that

$$\bar d^{\bar\pi}(\bar s; \mathcal{S}_{\pi}^{\mathrm{rem}}) = P^{\bar\pi, M}\left[\tau : \tau \in \mathsf{T}(s_\top \to \bar s; \neg \bar S)\right] \le P^{\bar\pi, M}[\tau : \bar s \in \tau] = d^{\bar\pi}(\bar s),$$

we have proved the induction hypothesis $d^{\pi}(s) \ge \frac{\varepsilon}{12D}$ for the next round.

$\square$

Lemma 13. *With probability at least* $1 - 2\delta$*,*

(a) *The while loop in line 1 in Algorithm 1 will terminate after at most* $\frac{12HD\mathfrak{C}(\Pi)}{\varepsilon}$ *rounds.*

(b) *After the termination of the above while loop, for any* $\pi \in \Pi$*, the remaining states* $s \in \mathcal{S}_{\pi}^{\mathrm{rem}}$ *that are not added in* $\mathcal{I}$ *(or* $\mathcal{T}$*) satisfy* $\bar d^{\pi}(s; \mathcal{S}_{\pi}^{\mathrm{rem}}) \le \varepsilon/4D$*.*

Notice that according to our algorithm, the same state cannot be added twice into $\mathcal{I}$*. Therefore,* $|\mathcal{I}| \le D|\Pi_{\mathrm{core}}|$ *and the maximum number of rounds in the while loop is* $D|\Pi_{\mathrm{core}}|$*.*

Proof. According to Lemma 10 and Lemma 12, (24) holds with probability at least $1 - 2\delta$.

(a) First note that from Lemma 1 and the definition of coverage in (3), we have

$$\sum_{s \in \mathcal{S}} \sup_{\pi \in \Pi} d^{\pi}(s) \le H C^{\mathsf{cov}}(\Pi; M) \le H\mathfrak{C}(\Pi).$$

Furthermore, (24) implies that any $(s, \pi_s) \in \mathcal{T}$ satisfies $d^{\pi_s}(s) \ge \varepsilon/12D$. Thus,

$$\sum_{s \in \mathcal{I}} \sup_{\pi \in \Pi} d^{\pi}(s) \ge \sum_{s \in \mathcal{I}} d^{\pi_s}(s) \ge |\mathcal{T}| \cdot \frac{\varepsilon}{12D},$$

where we used the fact that $\mathcal{I}$ denotes the set of states in $\mathcal{T}$, and $|\mathcal{I}| = |\mathcal{T}|$. Since, $\mathcal{I} \subseteq \mathcal{S}$, the two bound above taken together indicate that

$$|\mathcal{T}| \le \frac{12HD\mathfrak{C}(\Pi)}{\varepsilon}.$$

Since, every iteration of the while loop adds at least one new (s, π_s) to $\mathcal{T}$, the while loop from lines 1-1 will terminate after at most $\frac{12HD\mathfrak{C}(\Pi)}{\varepsilon}$ many rounds.

(b) Additionally, we know that after the while loop terminated, for every $\pi \in \Pi$ and $s \in \mathcal{S}_\pi^{\text{rem}}$, we must have that $\widehat{d}^\pi(s) \le \frac{\varepsilon}{6D}$, or else the condition in line 1 in Algorithm 1 will fail.

Hence according to (24), we get

$$\bar{d}^\pi(s; \mathcal{S}_\pi^{\text{rem}}) \le \frac{\varepsilon}{6D} + \frac{\varepsilon}{12D} = \frac{\varepsilon}{4D}.$$

□

Lemma 14. *Suppose (a) and (b) in Lemma 10 holds. Fix $\pi \in \Pi$, suppose for any $s \in \mathcal{S}_\pi^{\text{rem}}$ we have*

$$\bar{d}^\pi(s; \mathcal{S}_\pi^{\text{rem}}) \le \frac{\varepsilon}{4D},$$

the output of $\widehat{V}^\pi$ in Algorithm 5 satisfies

$$|\widehat{V}^\pi - V^\pi| \le \varepsilon$$

Proof. We first notice that the output $\widehat{V}^\pi$ of Algorithm 5 is exact the value function of MRP $\widehat{\mathfrak{M}}_\mathcal{I}^\pi$ defined by (16) and (17). We further let V_{tab}^π to be the value function of $\mathfrak{M}_\mathcal{I}^\pi$ defined by (14) and (15). Then when $D = 0$, according to (b) in Lemma 10, we obtain

$$|\widehat{V}^\pi - V_{\text{tab}}^\pi| = |\widehat{r}^\pi_{s_\top \to s_\perp} - \bar{r}^\pi_{s_\top \to s_\perp}| \le \frac{\varepsilon}{12(D+1)^2} \le \frac{\varepsilon}{2}.$$

When $D \ge 1$, we have $\frac{\varepsilon}{12D(D+1)} \le \frac{\varepsilon}{8(D+2)}$. Additionally, according to Lemma 10, we have

$$|r^\pi_{s_\top \to s'} - \widehat{r}^\pi_{s_\top \to s'}| \le \frac{\varepsilon}{12D(D+1)}, \quad |P^\pi_{s_\top \to s'} - \widehat{P}^\pi_{s_\top \to s'}| \le \frac{\varepsilon}{12(D+1)^2}, \quad \forall s' \in \mathcal{S}_\pi^+$$
$$|r^\pi_{s \to s'} - \widehat{r}^\pi_{s \to s'}| \le \frac{\varepsilon}{12D(D+1)}, \quad |P^\pi_{s \to s'} - \widehat{P}^\pi_{s \to s'}| \le \frac{\varepsilon}{12D(D+1)}, \quad \forall s \in \mathcal{S}_\pi^{\text{rch}} \backslash \{s_\top\}, s' \in \mathcal{S}_\pi^+,$$

Hence according to the simulation lemma (Lemma 9), we get

$$\begin{aligned}
|\widehat{V}^\pi - V_{\text{tab}}^\pi| &\le 2(D+2) \max_{s,s' \in \mathcal{S}_\pi^+} \left(\left| P^\pi_{s \to s'} - \widehat{P}^\pi_{s \to s'} \right| + \left| r^\pi_{s \to s'} - \widehat{r}^\pi_{s \to s'} \right| \right) \\
&\le 2(D+2) \left(\frac{\varepsilon}{8(D+2)} + \frac{\varepsilon}{8(D+2)} \right) \le \frac{\varepsilon}{2}.
\end{aligned}$$

Additionally for any $s_{h_1} = s_\top, s_{h_2}, \cdots, s_{h_{t-1}}, s_{h_t} = s_\perp \in \{s_\perp\} \cup \mathcal{S}_\pi^{\text{rch}}$, the probability of seeing trajectory $\bar{\tau} = (s_{h_1}, s_{h_2}, \cdots, s_{h_t})$ in $\mathfrak{M}_\mathcal{I}^\pi$ is

$$\begin{aligned}
P^{\mathfrak{M}_\mathcal{I}^\pi}(\bar{\tau} = (s_{h_1}, s_{h_2}, \cdots, s_{h_t})) &= \prod_{i=1}^{t-1} P^\pi_{s_{h_i} \to s_{h_{i+1}}} \\
&= \prod_{i=1}^{t-1} P^{\pi,M}(\tau : \tau \in \mathsf{T}(s_{h_i} \to s_{h_{i+1}}; \neg \mathcal{S}_\pi) | \tau[h_i] = s_{h_i}) \\
&= P^{\pi,M}(\tau : \tau[h_i] = s_{h_i}, \forall 1 \le i \le t, \tau[h] \notin \mathcal{S}_\pi, \forall h \ne h_1, \cdots, h_t).
\end{aligned}$$

Similarly, the expectation of rewards we collected in $\mathfrak{M}_\mathcal{I}^\pi$ with trajectory $\bar{\tau}$ is

$$\begin{aligned}
&\mathbb{E}^{\mathfrak{M}_\mathcal{I}^\pi}\left[R[\bar{\tau}] \mathbb{1}\{\bar{\tau} = (s_{h_1}, s_{h_2}, \cdots, s_{h_t})\} \right] \\
&\quad = \mathbb{E}^{\pi,M}\left[R[\tau] \mathbb{1}\{\tau[h_i] = s_{h_i}, \forall 1 \le i \le t, \tau[h] \notin \mathcal{S}_\pi, \forall h \ne h_1, \cdots, h_t\} \right].
\end{aligned}$$

Summing over all possible $s_{h_1}, \cdots, s_{h_t}$, we will get

$$V_{\text{tab}}^\pi = \mathbb{E}^{\pi,M}\left[R[\tau] \mathbb{1}\{\tau \cap \mathcal{S}_\pi^{\text{rem}} = \varnothing\} \right].$$

Hence since $\forall s \in \mathcal{S}_\pi^{\mathrm{rem}}, \bar{d}^\pi(s; \mathcal{S}_\pi^{\mathrm{rem}}) \le \frac{\varepsilon}{4D}$, we get

$$\begin{aligned}
|V^\pi - V^\pi_{\mathrm{tab}}| &= \mathbb{E}^{\pi,M}\left[R[\tau]\right] - \mathbb{E}^{\pi,M}\left[R[\tau]\mathbb{1}\{\tau \cap \mathcal{S}_\pi^{\mathrm{rem}} = \varnothing\}\right] \\
&= \mathbb{E}^{\pi,M}\left[R[\tau]\mathbb{1}\{\tau \cap \mathcal{S}_\pi^{\mathrm{rem}} \neq \varnothing\}\right] \\
&= \sum_{s \in \mathcal{S}_\pi^{\mathrm{rem}}} \mathbb{E}^{\pi,M}\left[R[\tau]\mathbb{1}\{s \in \tau, \tau[0:s] \cap \mathcal{S}_\pi^{\mathrm{rem}} = \varnothing\}\right] \\
&\le \sum_{s \in \mathcal{S}_\pi^{\mathrm{rem}}} \bar{d}^\pi(s; \mathcal{S}_\pi^{\mathrm{rem}}) \le D \cdot \frac{\varepsilon}{4D} = \frac{\varepsilon}{4},
\end{aligned}$$

which indicates that

$$|\widehat{V}^\pi - V^\pi| \le |V^\pi - V^\pi_{\mathrm{tab}}| + |\widehat{V}^\pi - V^\pi_{\mathrm{tab}}| \le \frac{\varepsilon}{2} + \frac{\varepsilon}{4} < \varepsilon.$$

□

F.3. Proof of Theorem 4

In the following proof, we assume the event defined in Lemma 12 holds (which happens with probability at least $1 - \delta$). With our choices of N_1, N_2:

$$N_1 = \frac{C_1(D+1)^4 K^2 \log(|\Pi|(D+1)/\delta)}{\varepsilon^2}, N_2 = \frac{C_2 D^3 (D+1)^2 K^2 \log(|\Pi|(D+1)^2/\delta)}{\varepsilon^3},$$

if further noticing that the while loop runs at most $\frac{12HD\mathfrak{C}(\Pi)}{\varepsilon}$ rounds (Lemma 13), the total number of samples used in our algorithm is upper bounded by

$$N_1 + N_2 \cdot \frac{12HD\mathfrak{C}(\Pi)}{\varepsilon} = \widetilde{\mathcal{O}}\Big(\Big(\tfrac{1}{\varepsilon^2} + \tfrac{HD^6\mathfrak{C}(\Pi)}{\varepsilon^4}\Big) \cdot K^2 \log \tfrac{|\Pi|}{\delta}\Big).$$

Additionally, after the termination of while loop, Lemma 12 indicates that for any policy $\pi \in \Pi$, and $s \in \mathcal{S}_\pi^{\mathrm{rem}}$ we have

$$\bar{d}^\pi(s; \mathcal{S}_\pi^{\mathrm{rem}}) \le \frac{\varepsilon}{4D}.$$

Therefore, Lemma 14 indicates that for any $\pi \in \Pi$, $|\widehat{V}^\pi - V^\pi| \le \varepsilon$. Hence the output policy $\widehat{\pi} \in \arg\max_\pi \widehat{V}^\pi$ satisfies that

$$\max_{\pi \in \Pi} V^\pi - V^{\widehat{\pi}} \le 2\varepsilon + \widehat{V}^\pi - \widehat{V}^{\widehat{\pi}} \le 2\varepsilon.$$

Rescaling ε by 2ε, and δ by 2δ, the proof of Theorem 4 is complete.