Agnostic Reinforcement Learning with Low-Rank MDPs and Rich Observations

Christoph Dann
Google Research
cdann@cdann.net

Yishay Mansour
Google Research & Tel Aviv University
mansour.yishay@gmail.com

Mehryar Mohri
Google & Courant Institute
mohri@google.com

Ayush Sekhari
Cornell University
as3663@cornell.edu

Karthik Sridharan
Cornell University
ks999@cornell.edu

Abstract

There have been many recent advances on provably efficient Reinforcement Learning (RL) in problems with rich observation spaces. However, all these works share a strong realizability assumption about the optimal value function of the true MDP. Such realizability assumptions are often too strong to hold in practice. In this work, we consider the more realistic setting of agnostic RL with rich observation spaces and a fixed class of policies Π that may not contain any near-optimal policy. We provide an algorithm for this setting whose error is bounded in terms of the rank d of the underlying MDP. Specifically, our algorithm enjoys a sample complexity bound of $\widetilde{O}\left((H^{4d}K^{3d}\log|\Pi|)/\varepsilon^2\right)$ where H is the length of episodes, K is the number of actions and $\varepsilon > 0$ is the desired sub-optimality. We also provide a nearly matching lower bound for this agnostic setting that shows that the exponential dependence on rank is unavoidable, without further assumptions.

1 Introduction

Reinforcement Learning (RL) has achieved several remarkable empirical successes in the last decade, which include playing Atari 2600 video games at superhuman levels [Mnih et al., 2015], AlphaGo or AlphaGo Zero surpassing champions in Go [Silver et al., 2018], AlphaStar's victory over top-ranked professional players in StarCraft [Vinyals et al., 2019], or practical self-driving cars. These applications all correspond to the setting of rich observations, where the state space is very large and where observations may be images, text or audio data. In contrast, most provably efficient RL algorithms are still limited to the classical tabular setting where the state space is small [Kearns and Singh, 2002, Brafman and Tennenholtz, 2002, Azar et al., 2017, Dann et al., 2019] and do not scale to the rich observation setting.

To derive guarantees for large state spaces, much of the existing work in RL theory relies on a *realizability* and a *low-rank* assumption [Krishnamurthy et al., 2016, Jiang et al., 2017, Dann et al., 2018, Du et al., 2019a, Misra et al., 2020, Agarwal et al., 2020a]. Different notions of rank have been adopted in the literature, including that of a low-rank transition matrix [Jin and Luo, 2019], a low Bellman rank [Jiang et al., 2017], Witness rank [Sun et al., 2019], Eluder dimension [Osband and Van Roy, 2014], Bellman-Eluder dimension [Jin et al., 2021], or bilinear classes [Du et al., 2021]. These studies also show that learning without any such structural assumptions requires a sample size that grows exponentially in the time horizon of the MDP [Dann and Brunskill, 2015, Krishnamurthy et al., 2016, Du et al., 2019b]. The choice of the most suitable and most general notion of rank is the topic of much active research in RL theory.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

In comparison, the realizability assumption has received much less attention. This is the strong premise that the optimal value function belongs to the class of functions considered, which typically does not hold in practice. In many applications, the optimal value function $Q^\star$ is highly complex and we cannot hope to accurately approximate it in the absence of some strong domain knowledge. Can we relax the realizability assumption in RL?

Value-function realizability can be viewed as the analogue of the PAC-realizability assumption in classical statistical learning theory. That assumption rarely holds, which has motivated the development and analysis of numerous algorithms for the agnostic PAC learnability model. Those algorithms provably learn to predict as well as the best predictor in the given function class, independently of whether the Bayes predictor belongs to the class. The counterparts of such results in reinforcement learning are mostly unavailable, which prompts the following question: Can we derive a theory of agnostic reinforcement learning?

Here, we precisely initiate that study. In this agnostic setting, we adopt common structural assumptions, e.g. small rank of the transition matrix, but seek to learn to perform as well as the best policy in the given policy class, independently of how close this class represents the Bellman-optimal policy. Specifically, we study agnostic Reinforcement Learning (RL) with a fixed policy class Π in the episodic MDPs with rich observations. Provably sample-efficient algorithms for agnostic RL would be highly desirable but it is still unknown to what degree learning is possible in this setting. Our work provides new insights about learnability with structural assumptions in the absence of (approximate) realizability in RL.

Agnostic RL without any additional structural assumptions has been considered in the past. By evaluating each policy in the class individually, one can easily obtain a sample complexity upper bound of $O(|\Pi|/\varepsilon^2)$. Kearns et al. [2000] also showed that an upper bound of $(K^H \log|\Pi|)/\varepsilon^2$ is possible, where K is the number of actions and H is the time horizon. However, as discussed in prior work such as [Krishnamurthy et al., 2016], bounds of this form are rather unsatisfactory as one of them admits a linear dependence on the size of the function class, which is prohibitively large, and the other one admits an exponential dependence on the length of the episodes H, which is typically long. Using existing constructions, one can derive a lower bound on the sample complexity of the form $\min\{|\Pi|, K^H\}/\varepsilon^2$ in the rich observation setting. This further justifies our adoption of rank as a natural structural assumption.

Our Contributions: The following highlights our main technical contributions, where d is the rank of the state transition matrix induced by any policy in the class Π, and is assumed to be small.

- We provide a uniform exploration-based algorithm that can find an ε-sub-optimal policy w.r.t. the policy class Π after collecting $O((HK/d)^{4d}\log(d|\Pi|)/\varepsilon^2)$ samples in the MDP. This bound shows that one can achieve a sample complexity that is polynomial in both H and $\log|\Pi|$, while being exponential in rank d only (which we assume is small).

  In addition to the sample complexity bound obtained here, the algorithmic techniques itself might be of independent interest and useful beyond this work. The algorithm is based on showing that for every policy, the expected rewards follows an autoregressive model of degree d. Thus obtaining samples of $O(d)$-length paths for a policy we show that one can extrapolate expected rewards for the entire episode.
- We complement this upper bound with a sample complexity lower bound of $\Omega\big((H/d)^{d/2}/\varepsilon^2\big)$ (when $K = 2$), thereby showing that the $H^{O(d)}$ term in the upper bound is unavoidable. The lower bound also highlights which structures in the policy class induce the $H^{O(d)}$ terms thus shedding light on what structural assumptions could help alleviate the exponential dependence on the rank.
- Finally, we seek to improve upon the H^d term and provide an adaptive algorithm that admits a sample complexity that depends on the eigenspectrum of the transition matrix of the MDP; while in the worst case that bound matches the above one, it provides a significantly better guarantee when the eigenspectrum is more favorable.

However, we view the main benefit of our work to be the initiation of the study of agnostic reinforcement learning and the presentation of an in-depth analysis of a natural structural assumption within that setting. This can form the basis for future research in this domain with alternative and perhaps more favorable rank-type assumptions.

2 Problem Setup

We consider an episodic Markov decision process with episode length $H \in \mathbb{N}$, observation space $\mathcal{X}$ and action space $\mathcal{A} := \{1, \dots, K\}$. For ease of exposition, we assume that the observation space $\mathcal{X}$ is finite (albeit extremely large), but our results can be readily extended to countably infinite and possibly uncountably infinite observation spaces. Each episode is a sequence $((x_1, a_1, r_1), (x_2, a_2, r_2), \dots, (x_H, a_H, r_H)) \in (\mathcal{X} \times \mathcal{A} \times \mathbb{R})^H$, where the initial observation x_0 is drawn from the initial distribution $\mu_0 \in \Delta(\mathcal{X})$, the actions are generated by the learning agent and all the following observations are sampled from the transition kernel $x_{h+1} \sim T(\cdot|x_h, a_h) \in \Delta(\mathcal{X})$ that depends on the previous observation and action. Finally, the rewards r_h are drawn from a sub-Gaussian distribution with mean $r(x_h, a_h)$ where $r\colon \mathcal{X} \times \mathcal{A} \mapsto [0, 1]$. The learning agent does not know the transition kernel T, the initial distribution μ_0, or the reward function r.

In our setting, the agent is given a policy class $\Pi \subseteq \{\mathcal{X} \mapsto \Delta(\mathcal{A})\}$ consisting of policies that map observations to distributions over the actions $\mathcal{A}$. For any policy $\pi \in \Pi$, we denote by $T^\pi \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$, the transition matrix induced by π, i.e., for any $x, x' \in \mathcal{X}$,

$$[T^\pi]_{(x',x)} = \mathbb{E}_{a \sim \pi(x)} T(x' \mid x, a).$$

Assumption 1 (Low-rank transition)**.** *There exists* $d \in \mathbb{N}$ *such that* $\operatorname{rank}(T^\pi) \leq d$ *for all* $\pi \in \Pi$.

For the main part of the paper, we assume that the learner knows the value of d, but later extend our results to the case where d is unknown. We define $\lambda^\pi = (\lambda_1^\pi, \dots, \lambda_d^\pi)^\top \in \mathbb{C}^d$ to denote the eigenvalues of the transition matrix T^π with rank at most d. Without any loss of generality, assume that $|\lambda_1^\pi| \geq |\lambda_2^\pi| \geq \dots \geq |\lambda_d^\pi|$.

We denote by $\mathbb{P}^\pi$ the distribution over episodes when following policy π and by $\mathbb{E}^\pi$ its expectation. We call the expected rewards obtained at time h by policy π **expected policy rewards**:

$$R_h^\pi := \mathbb{E}^\pi[r(x_h, a_h)]. \tag{1}$$

The value function of π at time h is given by $V_h^\pi(x) = \mathbb{E}^\pi\left[\sum_{h'=h}^H r(x_{h'}, a_{h'}) \mid x_{h'} = x\right]$. Further, when using V^π without a time index and arguments, we mean the value or expected H-step return:

$$V^\pi := \mathbb{E}[V_0^\pi(x_0)] = \mathbb{E}^\pi\left[\sum_{h=1}^H r(x_h, a_h)\right] = \sum_{h=1}^H R_h^\pi, \tag{2}$$

the value function averaged over initial observations.

Learning objective. The goal of the learner is to return a policy $\widetilde{\pi}$, after interacting with the MDP for n episodes of length H, such that the value of the returned policy is as close as possible to the value of the best policy in Π, that is,

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

where the error ε is a small as possible and may depend on n, the policy class Π and the MDP.

3 Related work

We give a brief overview of the most closely related works here, and defer a more detailed discussion to Appendix A.

Recently there has been great interest in designing RL algorithms with general function approximation [Jiang et al., 2017, Dann et al., 2018, Sun et al., 2019, Du et al., 2019a, Wang et al., 2020, Du et al., 2021]. In particular, Jiang et al. [2017] introduced the notion of Bellman rank, a measure of complexity that depends on the underlying environment and the value function class $\mathcal{F}$, and provide statistically efficient algorithms for learning problems for which Bellman rank is bounded. This was later extended to model-based algorithms by Sun et al. [2019]. While these algorithms work across a variety of problem settings, their sample complexity scales with $\log(|\mathcal{F}|)$. Furthermore, these algorithms also require the optimal value function f^* to be realized in $\mathcal{F}$. In our work, we do not

assume that the learner has access to a value function class $\mathcal{F}$. In fact, given a value function class $\mathcal{F}$, we can construct the policy class $\Pi_{\mathcal{F}}$ that corresponds to greedy policies induced by the class $\mathcal{F}$. However, given just a policy class Π, one cannot construct a value function class, without additional knowledge of the underlying dynamics.

Our Assumption 1 implies that for any policy $\pi \in \Pi$, the transition dynamics exhibits a low-rank decomposition with dimension d, that is $T^{\pi}(x'|x) = \langle \phi^{\pi}(x), \psi^{\pi}(x') \rangle$, for some d-dimensional feature maps $\phi^{\pi}, \psi^{\pi} \colon \mathcal{X} \mapsto \mathbb{R}^d$. Low rank MDPs and linear transition models have recently gained a lot of attention in the RL literature [Yang and Wang, 2020, Jin et al., 2020, Modi et al., 2020, Wang et al., 2021a]. The works most closely related to our setup are those of Jin et al. [2020] and Yang and Wang [2020], who give algorithms to find an optimal policy in low rank MDPs with known feature maps ϕ. Similarly, the other algorithms also assume that the learner either observes the feature $\phi(x)$, or the feature $\psi(x)$. Agarwal et al. [2020a] and Modi et al. [2021] learn under weaker assumptions and only assume that the learner has access to a function class that realizes ϕ. However, in our setup, the learner neither observes the features ϕ^{π}, ψ^{π} nor has access to a realizable function class for them, and thus these methods are not applicable.

Several of the works mentioned above recognize the issue of a strict realizability assumption and provide results only when the function class contains a good approximation to the optimal value function of model. However, the goal in our agnostic setting is more ambitious. We would like to find a policy that can compete with the best policy in the given class Π, independent of how close the best return in the class $\max_{\pi \in \Pi} V^{\pi}$ is to the return of the optimal policy $V^{\pi^\star}$ for that MDP.

There have also been several approaches for provably efficient RL with non-parametric function classes [Yang et al., 2020, Long et al., 2021, Shah et al., 2020]. However, these approaches still aim to learn the optimal value function and their regret necessarily scales with the complexity of the optimal value function in the RKHS which can be very high. Instead, in our agnostic setting we would like to be able to quickly identify the best policy from the given policy class with low complexity containing a good but not necessarily optimal policy. Finally, there are a few prior works in agnostic RL that directly compete against a policy class [Abbasi-Yadkori et al., 2013, Azar et al., 2013]. However, they either make strong assumptions on the feedback model, e.g. Abbasi-Yadkori et al. [2013] assumes that the the agent fully observed the current transition kernel and reward instead of just a sample from it, or the provided bound [Azar et al., 2013] scales linearly with the size of the policy class, instead of logarithmically.

4 Upper bound

In this section, we describe our main algorithm for finding a policy that is close to the best-in-class in Π. This algorithm presented in Algorithm 1, is an instance of policy search with uniform exploration. Specifically, we first collect a dataset $\mathcal{D}$ of n episodes by picking actions uniformly at random and subsequently use those episodes to estimate the value of each policy in Π. The algorithm then simply returns the policy $\widetilde{\pi}$ with the highest estimated value.

Our main technical innovation is a new estimation procedure for policy values in Algorithm 2 that leverages the low-rank structure of the transition matrix. A straightforward way to estimate the policy value is to take the sum of the rewards on average across all episodes where all actions are consistent with the policy [Kearns et al., 2000]. Unfortunately, this rejection sampling approach yields an error of $\Omega(\sqrt{K^H})$. Instead, our procedure only estimates the expected policy rewards for the first $3d$ steps. Specifically, when invoked with a given policy π, ValEstimate estimates the expected rewards for that policy by considering the subset of trajectories in $\mathcal{D}$ where π agrees with the chosen action till the first 3d steps, and by averaging the observed rewards in those trajectories. ValEstimate then predicts the future expected rewards for that policy by extrapolating these $3d$ estimated expected rewards. The prediction is computed by recognizing that the expected rewards for any policy π satisfy an autoregressive relation of order d as shown in Lemma 1.

In order to find the coefficients of this autoregression, ValEstimate computes $\widehat{\lambda} \in \mathbb{C}^d$ by solving the optimization problem (4), where the coefficient $\alpha_k(\lambda)$ are the sum of degree k monomials:

$$\alpha_k(\lambda) = \sum_{x \in \{0,1\}^d \text{ s.t. } \|x\|_1 = k} \lambda_1^{x_1} \lambda_2^{x_2} \dots \lambda_d^{x_d}. \tag{3}$$

After estimating $\widehat{\lambda}$, ValEstimate then predicts the expected rewards for all future time steps for the policy π by unfolding the autoregression model whose coefficients are given by $\alpha_k(\widehat{\lambda})$. The estimate for the value of the given policy π, denoted by $\widetilde{V}^\pi$, is then computed as the sum of the predicted expected rewards for H steps.

Finally, Algorithm 1 returns the policy $\widetilde{\pi}$ whose estimated value function is highest amongst all the policies in Π. The following theorem characterizes the performance guarantee for the policy $\widetilde{\pi}$ returned by our algorithm.

Algorithm 1 Policy search algorithm

Input: horizon H, rank d, number of episodes n, finite policy class Π

1: Collect the dataset $\mathcal{D} = \{(x_h^t, a_h^t, r_h^t)_{h=1}^H\}_{t=1}^n$ of n trajectories by drawing actions from Uniform($\mathcal{A}$).
2: **for** policy $\pi \in \Pi$ **do**
3: Estimate $\widetilde{V}^\pi$ by calling ValEstimate$(H, d, \mathcal{D}, \pi)$.
4: **Return:** policy $\widetilde{\pi}$ with best estimated value, i.e. $\widetilde{\pi} \in \operatorname{argmax}_{\pi\in\Pi} \widetilde{V}^\pi$.

Algorithm 2 Value estimation by autoregressive extrapolation

1: **function** VALESTIMATE$(H, d, \mathcal{D}, \pi)$:
2: **for** time step $h = 1, \ldots, 3d$ **do**
3: Estimate expected rewards by importance sampling

$$\widehat{R}_h = \frac{1}{n}\sum_{t=1}^{n} r_h^t \prod_{h'\le h} \left(K\mathbb{1}\{\pi(x_{h'}^t) = a_{h'}^t\}\right)$$

4: Estimate eigenvalues of the autoregression by solving the optimization problem:

$$(\widehat{\lambda}, \widehat{\Delta}) \leftarrow \operatorname*{argmin}_{\lambda\in\mathbb{C}^d, \Delta\in\mathbb{R}} \Delta \qquad \text{s.t. } |\lambda_k| \le 1 \quad \text{for } k = 1, \ldots, d \tag{4}$$

$$\text{and} \quad \left|\sum_{k=1}^{d} (-1)^{k+1}\alpha_k(\lambda)\widehat{R}_{h-k} - \widehat{R}_h\right| \le \Delta \qquad \text{for } h = d+1, \ldots, 3d$$

5: Predict $\widetilde{R}_h$ as:

$$\widetilde{R}_h = \begin{cases} \widehat{R}_h & \text{for } 1 \le h \le d \\ \sum_{k=1}^{d} (-1)^{k+1}\alpha_k(\widehat{\lambda})\widetilde{R}_{h-k} & \text{for } d+1 \le h \le H \end{cases}. \tag{5}$$

6: **return:** Estimate of the value $\widetilde{V} = \sum_{h=1}^{H} \widetilde{R}_h$.

Theorem 1 (Main Theorem). *For a given $\delta \in (0,1)$, d-rank MDP, horizon $H \ge d$ and a finite policy class Π, after collecting n episodes, Algorithm 1 returns a policy $\widetilde{\pi}$ that with probability at least $1-\delta$ admits the following guarantee:*

$$V^{\widetilde{\pi}} \ge \max_{\pi\in\Pi} V^\pi - O\Big(d^3 \cdot \Big(\frac{H}{d}\Big)^{2d} \sqrt{\frac{K^{3d}\log(6\Pi d/\delta)}{n}}\Big).$$

Theorem 1 implies that Algorithm 1 can find an ε-optimal policy with probability $1-\delta$ as long as the number of samples n satisfies

$$n = \Omega\Big(\Big(\frac{H}{d}\Big)^{4d} \frac{K^{3d}\log(6d|\Pi|/\delta)}{\varepsilon^2}\Big).$$

The key idea used in ValEstimate, is that for any policy π for which $\operatorname{rank}(T^\pi) \le d$, the expected rewards satisfy an auto-regression of order d. The following lemma formalize this idea.

Lemma 1 (Autoregression on expected rewards). *Let π be any policy for which the transition matrix T^π has rank at most d. Then, for any time step $h \ge d+1$, the expected reward for policy π at time*

step h, denoted by R_h^π, satisfies the auto-regression

$$R_h^\pi = \sum_{k=1}^{d} (-1)^{k+1} \alpha_k(\lambda^\pi) R_{h-k}^\pi, \tag{6}$$

where $\lambda^\pi \in \mathbb{C}^d$ denotes the set of eigenvalues of the matrix T^π, and $\alpha_k(\lambda^\pi)$ is as defined in (3).

We defer the proof of Lemma 1 to Appendix C.1. The proof uses the fact that for any policy $\pi \in \Pi$, the distribution over observations at time step h is given by $\mu_h^\pi = (T^\pi)^h \mu_0$, where μ_0 denotes the distribution over the observation space at initialization. If $\mathrm{rank}(T^\pi) \leq d$, an application of the Cayley-Hamilton theorem implies that we can write $(T^\pi)^h$ as a linear combination of $((T^\pi)^{h-1}, \ldots, (T^\pi)^{h-d})$. This implies that μ_h^π, and thus the expected rewards R_h^π, satisfy an auto-regression of order d. While the expected rewards R_h^π satisfy an auto-regression for every policy π, note that we cannot hope for a similar relation between the instantaneous rewards $r_h(s_h^\pi, \pi(s_h))$ observed when taking actions according to π.

The following result shows that we can simultaneously estimate the expected rewards for the first $3d$ steps for every policy $\pi \in \Pi$. Let $\mathcal{D}$ be a dataset of n episodes in the MDP collected by drawing actions uniformly at random from $\mathcal{A}$. Then, for any policy π, there are approximately n/K^{3d} episodes in $\mathcal{D}$ where the actions taken during the first $3d$ time steps matches the predictions of π on those observations. We compute $\widehat{R}_h^\pi$ as the empirical average of the hth step reward in the corresponding n/K^{3d} episodes that match with π for the first $3d$ steps.

Lemma 2 (Importance sampling)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$, for any policy $\pi \in \Pi$ and time step $h \in [3d]$, the estimates $\widehat{R}_h^\pi$ computed using importance sampling satisfy the error bound*

$$|\widehat{R}_h^\pi - R_h^\pi| \leq \sqrt{\frac{2K^{3d}\log(6d|\Pi|/\delta)}{n}} + \frac{2K^{3d}\log(6d|\Pi|/\delta)}{n}.$$

For a given policy π, if we had access to the expected rewards $\{R_1^\pi, \ldots, R_d^\pi\}$, we could have solved for the coefficients $\alpha_k(\lambda)$ exactly. However, we only have access to the empirical estimates $\{\widehat{R}_1, \ldots, \widehat{R}_d\}$ of the expected rewards, and thus we compute the coefficients $\alpha_k(\widehat{\lambda})$ by solving the optimization problem in (4). We predict the future expected rewards by extrapolating using $\alpha_k(\widehat{\lambda})$. The following lemma bounds the error propagated due to this mismatch in our estimation.

Lemma 3 (Error propagation bound)**.** *Let $\lambda, \widehat{\lambda} \in \mathbb{C}^d$ be such that $\max\{|\lambda_1|, |\widehat{\lambda}_1|\} \leq 1$. Further, with the initial values $R_1, \ldots, R_d$ and $\widetilde{R}_1, \ldots, \widetilde{R}_d$, let the sequence $\{R_h\}$ and $\{\widetilde{R}_h\}$ be given by*

$$R_h = \sum_{k=1}^{d} (-1)^{k+1} \alpha_k(\lambda) R_{h-k} \qquad \text{and} \qquad \widetilde{R}_h = \sum_{k=1}^{d} (-1)^{k+1} \alpha_k(\widehat{\lambda}) \widetilde{R}_{h-k},$$

where the coefficients $\alpha_k(\lambda)$ and $\alpha_k(\widehat{\lambda})$ are define as in (3). Then, for all $h \geq 3d+1$,

$$|\widetilde{R}_h - R_h| \leq 2d \cdot \left(\frac{16eh}{d}\right)^{2d} \cdot \max_{h' \leq 3d} |R_{h'} - \widetilde{R}_{h'}|.$$

We defer the proof of Lemma 3 to Appendix C.3. The proof of Theorem 1 follows from combining the above three technical results. Lemma 1 suggests that for any policy $\pi \in \Pi$ for which $\mathrm{rank}(T^\pi) \leq d$, the expected per step rewards satisfy an auto-regression of order at most d. The error propagation bound in Lemma 3 and the bound on the estimation of the expected rewards for the first $3d$ steps given in Lemma 3 implies that, for every policy $\pi \in \Pi$, the estimated value $\widetilde{V}^\pi$ is close to the true value V^π. Specifically, the estimation error in the value of every policy in π is bounded by $\widetilde{O}((H/d)^{2d}\sqrt{K^{3d}\log(|\Pi|)/n})$. Thus, when $n = \widetilde{O}((H/d)^{4d}K^{3d}/\varepsilon^2)$, we have that $|\widetilde{V}^\pi - V^\pi| \leq \varepsilon$ for every policy $\pi \in \Pi$ simultaneously. This implies that the returned policy , that maximize the estimated value $\widetilde{V}^\pi$, is 2ε sub-optimal w.r.t. the best policy in Π. We defer full details of the proof of Theorem 1 to Appendix C.5.

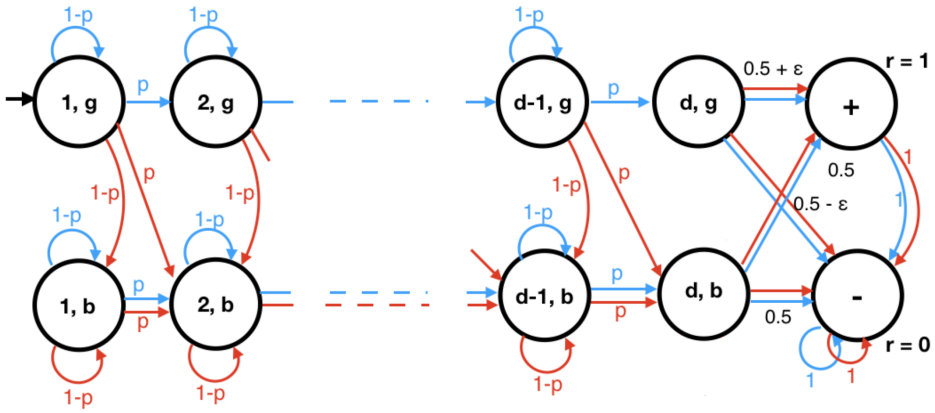

Figure 1: Latent state construction: contextual combination lock. As long as the agent follows actions of π^*(blue arrows), the agent remains in good states (i, g) and receives a Bernoulli$(1/2 + \varepsilon)$ reward but otherwise transits to bad states (i, b) and receives a Bernoulli$(1/2)$ reward.

5 Lower Bound

After presenting an algorithm with sample-complexity bound of $\widetilde{O}((H/d)^{2d}K^{3d}/\varepsilon^2)$, we now show through a lower-bound that the dependency on H and d cannot be improved significantly:

Theorem 2 (Lower bound). *Let $\varepsilon \in (0, 1/26)$, $\delta \in (0, 1/2)$, $d \geq 4$, $K = 2$ and $H \geq 219d$. There exists a policy class of size $(H/d)^d$ and a family of MDPs with rank at most $\Theta(d)$, finite observation space, horizon H and two actions such that the optimal policy for each MDP in the family is contained in the policy class and the following holds: Any algorithm that returns an ε-optimal policy, with probability at least $1 - \delta$, for every MDP in this family has to collect at least*

$$\Omega\Big(\frac{1}{H\varepsilon^2}\Big(\frac{H}{41d}\Big)^{d/2}\log\Big(\frac{1}{2\delta}\Big)\Big)\,.$$

episodes in expectation in some MDP in this family.

The above lower bound shows that an exponential dependency on d in the form of $(H/d)^d$ is unavoidable, even when a realizable policy class with $\pi^* \in \Pi$ and moderate size $\log|\Pi| = d\log(H/d)$ is given to the learner. We now provide a brief description of the problem class used in the proof of our lower bound but defer details of our construction and the proof to Appendix E.

The Markov decision processes in the proof of our lower bound bear some similarity to the so-called *combination lock* constructions used in prior works [Krishnamurthy et al., 2016, Du et al., 2019b], where the algorithm only receives positive feedback after playing a certain sequence of actions. Modelling a combination lock typically requires K^H states in MDPs and $\Theta(H)$ latent states in POMDP. In contrast, our contextual version of a combination lock uses a low-rank MDP with very large observation space but where the transition dynamics are governed by $\Theta(d) \ll H$ hidden states (and thus the rank is $\Theta(d)$). The latent state structure is shown in Figure 1. The agent starts at the top left latent state and always progresses with probability $p = d/H$ to the right. As long as it chooses good actions (blue edges), it progresses in the top chain where it will eventually reach state (d, g) with constant probability and receive a reward of 1 with probability $1/2 + \varepsilon$. If at any time before reaching state (d, g), it chooses a bad action (red edges), then it moves to the lower chain where it eventually has a $1/2$ chance of receiving a reward of 1.

If the latent states $s \in \mathcal{S}$ were directly observable, an ε-optimal policy could be learned with $O(dH/\varepsilon^2)$ samples. However, in the latent state s, the agent only receives an observation drawn uniformly from a large set $\mathcal{X}_s$. The sets $\{\mathcal{X}_s\}_{s\in\mathcal{S}}$ form a partition of the entire observation space $\mathcal{X}$ and there is a mapping $\phi\colon \mathcal{X} \mapsto \mathcal{S}$ that identifies the latent state for each observation. Each MDP $M_{\pi^*,\phi}$ in our problem class is parameterized by the mapping ϕ and a policy $\pi^* \in \Pi$. The class of policies Π can be arbitrary as long as each pair of policies differ on at least a constant fraction of $\mathcal{X}$. In MDP $M_{\pi^*,\phi}$, only the action $\pi^*(x)$ is a good action (blue action) and allows the agent to

stay in the top latent state chain. Thus, finding the $\Theta(\varepsilon)$-best policy in Π for $M_{\pi^*,\phi}$ is equivalent to identifying π^*.

Importantly, our problem class contains MDP $M_{\pi^*,\phi}$ for every possible $\pi^* \in \Pi$ and latent state mapping ϕ. We pick the number of observations large enough so that observations become uninformative and it is virtually impossible for a learner to learn ϕ. Instead it can only hope to learn π^* by identifying the bias ε in the rewards. We can show that this requires number of samples that are not much smaller than collecting $\Theta(1/\varepsilon^2 \ln(1/\delta))$ episodes with each of the $(H/d)^d$ policies in Π.

While the lower bound in Theorem 2 does not have a dependence on $\log(|\Pi|)$. The simple observation that the contextual bandit problem can be seen as an instance of our setup where $d = 1$, implies that some dependence on $\log(|\Pi|)$ is necessary based on standard contextual bandit lower bounds [Lattimore and Szepesvári, 2020]. However, getting a lower bound of the form $\Omega(H^d \log(|\Pi|))$ is an interesting question, which we leave open for future work. Finally, while the provided lower bound is constructed using an MDP with two actions, it can easily be extended to incorporate multiple actions, and when the learner has access to a generative model.

6 Adaptive algorithms

In Section 4, the algorithm introduced benefits from the guarantee provided by Theorem 1, which is near optimal in the worst case as the lower bound construction shows. However, in cases where the transition matrix induced by the policy class all have nicer eigenspectra, one could expect to have an improved sample complexity. Ideally, the algorithm should automatically adapt to more favorable eigenspectra. This is precisely what we describe in this section. We give an adaptive algorithm whose sample complexity improves when the eigenspectrum of transition matrices induced by the policy class admits a more favorable property.

6.1 Adaptivitity to the eigenspectrum

Our adaptive algorithm, presented in Algorithm 3 in Appendix D.3, is a policy search algorithm similar to Algorithm 1 where, instead of invoking the procedure ValEstimate, we compute the value function for every policy π by invoking the procedure AdaValEstimate given in Appendix D.3.

AdaValEstimate follows along the lines of ValEstimate. When invoked for a policy π, it first estimates the expected rewards for the first $3d$ time steps. Then, AdaValEstimate computes the autoregression coefficients $\alpha_k(\widehat{\lambda})$, and uses them to predict the expected rewards for all future time steps by extrapolating. The major difference between ValEstimate and AdaValEstimate is the way the coefficients $\alpha_k(\widehat{\lambda})$ are computed. Specifically, using $\Delta := 2d4^d\sqrt{(8K^{3d}\log(6d|\Pi|)/\delta)/n}$, the procedure AdaValEstimate computes the coefficients $\widehat{\lambda}$ by solving the optimization problem

$$\widehat{\lambda} \leftarrow \operatorname*{argmin}_{\lambda\in\mathbb{C}^d} \prod_{k=2}^{d}\Big(\sum_{h=0}^{H-1}|\lambda_k|^h\Big) \quad \text{s.t. } \lambda_1 = 1, |\lambda_k| \le 1 \qquad \text{for } 2 \le k \le d,$$

$$\text{and } \Big|\sum_{k=1}^{d}(-1)^{k+1}\alpha_k(\lambda)\widehat{R}_{h-k} - \widehat{R}_h\Big| \le \Delta \qquad \text{for } d+1 \le h \le 3d.$$

The above modification to the computation of $\widehat{\lambda}$ allows our error propagation bound to adapt to λ, which defines the coefficients of autoregression for the expected rewards in policy π (given in Lemma 1). The propagated error would be small if the coordinates of λ are bounded away from 1. The policy $\widetilde{\pi}$, returned by Algorithm 3, thus enjoys the following adaptive performance guarantee.

Theorem 3 (Adaptive upper bound)**.** *For a given $\delta \in (0,1)$, d-rank MDP, horizon H and a finite policy class Π, after collecting n episodes, Algorithm 3 returns a policy $\widetilde{\pi}$ that with probability at least $1-\delta$ admits the following guarantee:*

$$V^{\widetilde{\pi}} \ge \max_{\pi\in\Pi} V^{\pi} - O\Bigg(dH^2(16e)^{2d}\cdot\max_{\pi'\in\Pi}\prod_{k=2}^{d}\Big(\sum_{j=0}^{H-1}|\lambda_k^{\pi'}|^j\Big)^2\sqrt{\frac{K^{3d}\log(6\Pi d/\delta)}{n}}\Bigg),$$

Proof of Theorem 3 follows along the lines of the proof of Theorem 1 where we replace the error propagation bound (in Lemma 3) by a similar bound that adapts to the eigenspectrum of the transition matrix T^π. We defer the proof details to appendix Appendix D.3. Note that for any $|\lambda_k| \leq 1$ and thus $\sum_{h=0}^{H-1} |\lambda_k|^h \leq H$. Using this fact in Theorem 3 recovers the result of Theorem 1, albeit upto a multiplicative factor of 2^{2d}. In the following, we provide an example of a low rank MDP problem in which the adaptive bound above could be much better than the worst case upper bound in Theorem 1.

Corollary 1 (Well mixing MDP)**.** *Given $\delta \in (0,1)$, horizon H and a finite policy class Π. Let M be a d-rank MDP such that the second largest eigenvalue of the transition matrix T^π satisfies $|\lambda_2^\pi| \leq 1-\gamma$ for every policy $\pi \in \Pi$. Then, after collecting n episodes, our adaptive algorithm returns a policy $\widetilde{\pi}$ that with probability at least $1-\delta$ admits the following guarantee:*

$$V^{\widetilde{\pi}} \geq \max_{\pi \in \Pi} V^\pi - \widetilde{O}\Big(\Big(\frac{K}{\gamma}\Big)^{2d} \frac{1}{\sqrt{n}}\Big),$$

where the $\widetilde{O}$ hides polynomial factors of $d, H, \log(1/\delta)$ and multiplicative constants.

We next show through a lower bound that the adaptive upper bound in Theorem 3 cannot be improved further. We defer the proof details to Appendix E.3.

Theorem 4 (Adaptive lower bound)**.** *Let $\varepsilon \in (0, 1/16)$, $\delta \in (0, 1/2)$, $d \geq 4$ and $(\lambda_i)_{i\in[d]} \in [0,1]^d$ satisfy*

$$d^{2d} \lesssim \prod_{i=1}^{d} \frac{1}{1-\lambda_i} \lesssim \exp(H) \qquad \text{and} \qquad \sum_{i=1}^{d} \frac{1}{1-\lambda_i} \leq \frac{H}{4\ln(4d)}\,.$$

Then, there is a realizable policy class and a family of MDPs with rank at most $\Theta(d)$, finite observation space, horizon H and two actions such that: For each $i \in [d]$, policy π and MDP M in this class, there is an eigenvalue of the induced transition matrix T_M^π in $[\lambda_i/2, \lambda_i]$. Furthermore, any algorithm that returns, with probability at least $1-\delta$ an ε-optimal policy for any MDP in this family, has to collect at least

$$\Omega\Big(\frac{1}{\varepsilon^2 d^d} \sqrt{\prod_{i=1}^{d} \frac{1}{1-\lambda_i}} \log(1/2\delta)\Big)$$

episodes in expectation in some MDP in this family.

Adaptivity to rank. In Appendix D.4, we also provide an adaptive algorithm that can find the best policy in the class Π without knowing the value of the rank parameter d^*. Our adaptive algorithm, given in Algorithm 5, follows from standard techniques in the model selection literature. For every $d \in [H]$, we compute an optimal policy $\widetilde{\pi}_d$ assuming that the rank $d^* = d$. Then, for each $d \in [H]$, we estimate the value function for the policy $\widetilde{\pi}_d$ by drawing $n/2H$ fresh trajectories using that policy. Finally, we return the policy $\widetilde{\pi}$ from the set $\{\widetilde{\pi}_d\}_{d\in[H]}$ with the highest estimated value. The returned policy $\widetilde{\pi}$ satisfies, with probability at least $1-\delta$,

$$V^{\widetilde{\pi}} \geq \max_{\pi \in \Pi} V^\pi - \widetilde{O}\Big(\Big(\frac{H}{d^*}\Big)^{2d^*} \sqrt{\frac{(8K)^{3d^*} \log(|\Pi|/\delta)}{n}}\Big).$$

We defer full details of the analysis to the Appendix.

7 Conclusion

We presented a new analysis of reinforcement learning with rich observations in the agnostic setting, under the low rank MDP assumption. We gave both a non-adaptive and an adaptive algorithm for learning a quasi-optimal policy in this scenario, which we showed to benefit from guarantees that are only polynomial in the horizon and the number of actions, and only logarithmic in the size of the policy class considered. While our bound is exponential in the MDP rank, we give nearly matching lower bounds proving that that dependency is unavoidable. The agnostic setting is a more realistic setting that has received less attention in the literature. We view this work as initiating the study of this general setting under workable assumptions and believe that many other algorithmic and theoretical aspects of such scenarios need to be studied further.

Acknowledgements

AS was an intern at Google Research, NYC for the duration of the work. KS acknowledges support from NSF CAREER Award 1750575. YM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (grant number 993/17) and the Yandex Initiative for Machine Learning at Tel Aviv University.

Funding Transparency Statement

Funding in direct support of this work: NSF CAREER Award 1750575, NSF-CCF-1815893, a Google Faculty Fellowship, an NSERC Discovery Grant, and a University of Waterloo startup grant.

References

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):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, 2017.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. *International Conference on Machine Learning*, 2019.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient pac rl with rich observations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019a.

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*, pages 6961–6971. PMLR, 2020.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in Neural Information Processing Systems*, 33, 2020a.

Tiancheng Jin and Haipeng Luo. Learning adversarial markov decision processes with bandit feedback and unknown transition. *arXiv preprint arXiv:1912.01192*, 2019.

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*, pages 2898–2933. PMLR, 2019.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pages 1466–1474, 2014.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

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

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

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*, 2019b.

Michael J Kearns, Yishay Mansour, and Andrew Y Ng. Approximate planning in large pomdps via reusable trajectories. In *Advances in Neural Information Processing Systems*, pages 1001–1007, 2000.

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, 2020.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

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

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

Yining Wang, Ruosong Wang, Simon Shaolei Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*, 2021a.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*, 2020.

Jihao Long, Jiequn Han, et al. An l2 analysis of reinforcement learning in high dimensions with kernel and neural network approximation. *arXiv preprint arXiv:2104.07794*, 2021.

Devavrat Shah, Dogyoon Song, Zhi Xu, and Yuzhe Yang. Sample efficient reinforcement learning via low-rank matrix estimation. *arXiv preprint arXiv:2006.06135*, 2020.

Yasin Abbasi-Yadkori, Peter L Bartlett, and Csaba Szepesvári. Online learning in markov decision processes with adversarially chosen transition probability distributions. *arXiv preprint arXiv:1303.3055*, 2013.

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Regret bounds for reinforcement learning with policy advice. In *Joint European Conference on Machine Learning and*

Knowledge Discovery in Databases, pages 97–112. Springer, 2013.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Tor Lattimore and Marcus Hutter. Pac bounds for discounted mdps. In *International Conference on Algorithmic Learning Theory*, pages 320–334. Springer, 2012.

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 *In Proc. 19th International Conference on Machine Learning*. Citeseer, 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 preprint arXiv:1707.06347*, 2017.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *arXiv preprint arXiv:1908.00261*, 2019.

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, 2020.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13399–13412. Curran Associates, Inc., 2020b.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *arXiv preprint arXiv:1810.12429*, 2018.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *arXiv preprint arXiv:1906.04733*, 2019.

Ruosong Wang, Dean Foster, and Sham M. Kakade. What are the statistical limits of offline {rl} with linear function approximation? In *International Conference on Learning Representations*, 2021b.

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

Gellert 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.

J Segercrantz. Improving the cayley-hamilton equation for low-rank transformations. *The American mathematical monthly*, 99(1):42–44, 1992.

DJ Hartfiel. Tracking in matrix systems. *Linear algebra and its applications*, 165:233–250, 1992.

Darald J Hartfiel. Dense sets of diagonalizable matrices. *Proceedings of the American Mathematical Society*, 123(6):1669–1672, 1995.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Pascal Massart. *Concentration inequalities and model selection*. Springer, 2007.

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.