

Improving Reinforcement Learning Sample-Efficiency using Local Approximation

Mohit Prashant^a and Arvind Easwaran^b

Abstract. In this study, we derive Probably Approximately Correct (PAC) bounds on the asymptotic sample-complexity for RL within the infinite-horizon Markov Decision Process (MDP) setting that are sharper than those in existing literature. The premise of our study is twofold: firstly, the further two states are from each other, transition-wise, the less relevant the value of the first state is when learning the ϵ -optimal value of the second; secondly, the amount of 'effort', sample-complexity-wise, expended in learning the ϵ -optimal value of a state is independent of the number of samples required to learn the ϵ -optimal value of a second state that is a sufficient number of transitions away from the first. Inversely, states within each other's vicinity have values that are dependent on each other and will require a similar number of samples to learn. By approximating the original MDP using smaller MDPs constructed using subsets of the original's state-space, we are able to reduce the sample-complexity by a logarithmic factor to $O(SA \log A)$ timesteps, where S and A are the state and action space sizes. We are able to extend these results to an infinite-horizon, model-free setting by constructing a PAC-MDP algorithm with the aforementioned sample-complexity. We conclude with showing how significant the improvement is by comparing our algorithm against prior work in an experimental setting.

1 Introduction

The field of reinforcement learning (RL) has made rapid progress in recent years, with studies demonstrating its application in complex decision making tasks [14]. As a consequence, several frameworks have been proposed to build complex decision making systems utilizing RL in domains like robotics, healthcare and autonomous transport [4]. However, to ensure trust in these systems, it is necessary to evaluate how successful a learner is during the learning process. Noting this, our aim is to derive the number of timesteps required to learn an ϵ -optimal policy, where $\epsilon \in (0, 1)$ is the distance of the learnt policy from the best possible policy and a timestep is an interaction of the learner with the environment.

A body of work that addresses this question is on Probably Approximately Correct Markov Decision Processes (PAC-MDP). They are a class of algorithms that are applicable to discrete state-action MDPs and are known to provide PAC guarantees on the sample-complexity; that is, they guarantee ϵ -optimal solutions with a specific confidence, $1 - \delta$, while having a sample-complexity that is polynomial in the environment size. As it is infeasible to obtain transition and reward models in many domains [8], we focus on computing sample-complexity bounds for model-free approaches to RL problems.

The current SOTA sample-complexity bound proposed for RL algorithms is $O(SA \log(SA))$ timesteps, where S and A are the state and action space sizes. This sample-complexity is demonstrated by algorithms like Delayed Q-Learning [17], Variance Reduced Q-Learning [23] and Q-Learning with UCB Exploration [3] amongst other studies [9, 5]. The difficulty in deriving sample-complexity bounds in model-free RL is that the lack of knowledge regarding the underlying MDP results in statistical bounds placed on the learning convergence to be conservative in nature. Citing Delayed Q-Learning and Variance Reduced Q-Learning as examples, the state-action space is over-sampled in trying to bound the variance of updates [3].

The aim of this study is to present sharper PAC-bounds on the sample-complexity of reinforcement learning within environments endowed with a distance metric. Intuitively, the premise of our approach is two-fold. **(1)** Firstly, the further two states are from each other, the less relevant the value of the first state is in learning the ϵ -optimal value of the second. Consequently, there exists a distance, dependent on the value of ϵ , beyond which the values of the two states are effectively independent, with high confidence. **(2)** Secondly, the amount of 'effort' expended to learn the ϵ -optimal value of a state, in terms of interactions the learner has with it, is independent of the effort required to learn the ϵ -optimal value of a sufficiently distant state. The inverse of this is that the amount of effort expended in learning the value of a state is similar to the effort required to learn the values of its transition neighbors.

1.1 Our Contributions

We make fundamental improvements on the sample-complexity bounds for model-free PAC-MDPs by approximating a larger MDP using several smaller MDPs constructed using subsets of the original state-space. With this, we implement a new model-free PAC-MDP algorithm, Probabilistic Delayed Q-Learning (PDQL), and prove its asymptotic sample-complexity bound is $O\left(\frac{SA}{\epsilon^3(1-\gamma)^3} \log\left(\frac{A}{\delta(1-\gamma)}\right) \log \frac{1}{\epsilon} \log \frac{1}{\delta}\right)$. The value of our work lies in the following contributions.

1. **We eliminate a logarithmic dependency on state-space size**, $\log S$, from prior SOTA model-free PAC-MDP convergence bounds, significantly increasing the learning rate when generalizing ϵ -optimal value functions over larger state-action spaces or sparse reward settings.
2. We experimentally compare PDQL with prior work in terms of convergence rates using benchmark environments and show it generalizes ϵ -optimal solutions faster.

Table 1: A comparison of model-free RL algorithms in terms of sample-complexity for infinite-horizon Markov Decision Processes

Algorithm	Sample-Complexity
Delayed Q-Learning [17]	$O\left(\frac{SA}{\epsilon^4(1-\gamma)^8} \log\left(\frac{SA}{\delta\epsilon(1-\gamma)}\right) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\epsilon(1-\gamma)}\right)\right)$
Speedy Q-Learning [1]	$O\left(\frac{SA}{\epsilon^2(1-\gamma)^4} \log\left(\frac{SA}{\delta}\right)\right)$
Variance Reduced Q-Learning [23]	$O\left(\frac{SA}{\epsilon^2(1-\gamma)^3} \log\left(\frac{SA}{\delta(1-\gamma)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$
Q-Learning with UCB [3]	$O\left(\frac{SA}{\epsilon^2(1-\gamma)^7} \log(SA) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{1-\gamma}\right)\right)$
UCB-Multistage-Advantage [26]	$O\left(\frac{SA}{\epsilon^2(1-\gamma)^{5.5}} \log(SA) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\epsilon(1-\gamma)}\right)\right)$
Phased Q-Learning [12]	$O\left(\frac{SA}{\epsilon^2} \log\left(\frac{SA}{\delta}\right) \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{\delta}\right)\right)$
Probabilistic Delayed Q-Learning (this work)	$O\left(\frac{SA}{\epsilon^3(1-\gamma)^3} \log\left(\frac{A}{\delta(1-\gamma)}\right) \log\frac{1}{\epsilon} \log\frac{1}{\delta}\right)$

2 Related Works

The literature on model-free PAC-MDP algorithms is centered around Q-Learning [24]. It should be noted that an extensive amount of analysis has been conducted on Q-Learning and various studies have improved on the base algorithm to improve sample efficiency in experimental settings [9, 22, 18]. Though, with regard to works that have established upper bounds on the sample-complexity of learning ϵ -optimal value functions, one of the earliest studies to derive PAC bounds for model-free RL is Delayed Q-Learning (DQL) [17]. The principle behind this work is to stabilize training using a sampling procedure for Q-value update. This presents conditions for optimality of each update for which the corresponding mathematical analysis evaluates the probability of violation. The sample-complexity bound derived for this algorithm is $O\left(\frac{SA}{\epsilon^4(1-\gamma)^8} \log\left(\frac{SA}{\delta\epsilon(1-\gamma)}\right) \log\left(\frac{1}{\delta}\right) \log\left(\frac{1}{\epsilon(1-\gamma)}\right)\right)$, where γ is the discount factor. This bound is noted to be quite conservative as it depends on an infinite-length execution of the algorithm, where the transitions from each state-action are not necessarily observed with equal frequency [3, 12].

Under the assumption of access to a generative model/simulator, the Variance Reduced Q-Learning (VRQL) algorithm presents a sample-complexity of $O\left(\frac{SA}{\epsilon^2(1-\gamma)^3} \log\left(\frac{SA}{\delta(1-\gamma)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$ [23]. The underlying principle of the algorithm is to bound the variance of each Q-value update using a batched process, similar to [17]. However, the assumption of access to a generative model in this work is able to reduce the sample complexity by a factor of $\epsilon^2(1-\gamma)^5$ as the implication of this assumption is that the current interaction with the environment can be made independently of the previous interaction, similar to an oracle. This leads to an unbiased sampling of the Q-values in the state-action space and a more stable update than in Delayed Q-Learning. Other notable works that posit similar bounds are Q-Learning with UCB Exploration [3] and Speedy Q-Learning [1]. To the best of our knowledge, all prior work in the derivation of PAC guarantees for model-free RL show that the sample-complexity of learning an ϵ -optimal value function is asymptotically bounded by $O(SA \log(SA))$ timesteps. This matches the known information theoretic bounds for infinite-horizon MDP problems presented in [9, 2, 23]. We present a table of known relevant sample-complexity bounds for PAC-MDPs in Table 1 for comparison.

Our study builds upon the results presented by DQL and VRQL by making the assumption that learning takes place within a discrete environment endowed with a distance metric between states. By utilizing episodes of a finite length as opposed to an infinite-length execution and directing the agent to states that require value updates, we are able to guarantee faster convergence. Furthermore, in doing so, we are also able to reduce the number of samples required to meet the optimality conditions posited in [23] by a factor of $\log S$.

3 Problem Formalization and Preliminaries

3.1 Notation

In this study, the learning guarantee is denoted using error and confidence parameters, $\epsilon, \delta \in (0, 1)$, where the learner is able to generalize a policy with at most ϵ error with $1 - \delta$ confidence. The environment that the learning problem is defined over is a discrete, finite-state MDP, M , that is characterized by the tuple, $(S, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$; where the state-space, S , of M has a size of S ; the action-space of M , \mathcal{A} , has a size of A , i.e. the maximum number of actions available to a single state in M is A ; the underlying transition function, $\mathcal{T} : S \times S \times \mathcal{A} \rightarrow [0, 1]$, is a function that determines the transition probability to a state given a state-action in M ; $\mathcal{R} : S \times \mathcal{A} \rightarrow [0, 1]$ is a function specifying the reward for executing an action from a state; lastly $\gamma \in (0, 1)$ is the discount factor for future rewards. As the learning problem is model-free, note that \mathcal{T} and \mathcal{R} are unknown to the learner. Furthermore, note that the transition from executing an action on a state is stochastic and, therefore, the state-action reward is also stochastic. Hence, for some $(s, a) \in S \times \mathcal{A}$, $\mathcal{R}(s, a)$ is the expected reward of the state-action. For the purpose of clarity, assume there also exists a reward function, $R : S \rightarrow [0, 1]$, that denotes the true reward for visiting a state, unlike \mathcal{R} .

The value of a state, $s \in S$, is equal to the expected cumulative discounted rewards experienced over a transition walk from s for an arbitrary policy that determines the walk. Assuming the rewards experienced by a walk from s over an infinite-horizon are described by the set $\{r_1, r_2, \dots\}$, the value of s is the expectation over all possible walks of the discounted sum of the set, i.e. $V(s) := \mathbb{E} \left[\sum_{i=0}^{\infty} \gamma^i r_i \right]$. Similarly, the value of a state evaluated over a fixed interval of T -steps, for some positive integer T , is $V^T(s) := \mathbb{E} \left[\sum_{i=0}^T \gamma^i r_i \right]$. This study assumes a greedy policy is implemented for the learner to follow, recursively defining the value function as follows, where s' is the subsequent state following the application of a on s . The Q-value of a state-action is defined similarly. Note that the value function is a maximization of the Q-value over \mathcal{A} , i.e. $V(s) := \max_{a \in \mathcal{A}} (Q(s, a))$. We denote the optimal value and Q-value functions over M as $V_*(s)$ and $Q_*(s, a)$, respectively.

3.2 Problem Formalization

Assume there exists an optimal value function defined over M 's state-space, $V_*(s)$, and a corresponding optimal Q-Value function defined over M 's state-action space, $Q_*(s, a)$, such that the application of a greedy policy from any state or state-action maximizes the expected cumulative discounted reward of transition walks enacted by the greedy policy. Letting t be a positive integer, let $V_t(s)$ and $Q_t(s, a)$ be the value and Q-value function approximations learnt by an RL algorithm at timestep t . As the class of RL algorithms that

we consider in this study are model-free PAC-MDP algorithms, the following definitions are used to provide constraints over the design of a model-free PAC-MDP algorithm [10, 9, 6].

Definition 1. (PAC-MDP Sample-Complexity) [10] For $\epsilon, \delta \in (0, 1)$ and $t \in \mathbb{Z}^+$, a PAC-MDP algorithm applied to MDP M is an algorithm that is able to generalize an ϵ -optimal value function, $V_t(s)$, such that $\mathbb{P}(|V_t(s) - V_*(s)| < \epsilon) > 1 - \delta, \forall s \in \mathcal{S}$ following t timesteps; where t is bounded by a polynomial in S and A , the state-space and action-space sizes of M .

Definition 2. (Model-Free Space-Complexity) [9] Any model-free learning algorithm applied to MDP M must have a space-complexity bounded by $\tilde{O}(SA)$, where S and A are the state-space and action-space sizes of M .

The second definition has been utilized in previous studies to restrict algorithms from storing excessive amounts of information regarding the MDP dynamics within memory, distinguishing model-based RL [8, 19]. Using these definitions, we present our model-free PAC-MDP algorithm, PDQL, in Section 6. From this, the problem being addressed in our study is formalized in the statement below.

Problem 3. Let PDQL be a PAC-MDP algorithm. For the application of PDQL over M , for values $\epsilon, \delta \in (0, 1)$, derive the greatest lower-bound on $t \in \mathbb{Z}^+$ such that $\mathbb{P}(|V_t(s) - V_*(s)| < \epsilon) > 1 - \delta, \forall s \in \mathcal{S}$.

3.3 Setting Formalization

Our work takes place in settings where the environment is endowed with a distance metric (e.g. Euclidean, Manhattan etc.). Our objective is to utilize properties of locality within these settings to make learning more efficient. Though our results strictly apply to these settings, we observe that many real-world problems are set in these environments (e.g. $\mathbb{R}^2, \mathbb{R}^3$, lattice etc.) in domains like navigation, control and healthcare [20, 25]. Further, there is extensive work on the application of RL within similar metric-based settings [13, 16, 15, 11].

Proposition 4. (Distance Metric) The state-space, \mathcal{S} , of M is endowed with a distance metric, $\mathcal{D} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$.

We use this proposition to further comment on locality and outline transition kernels within these environments.

Proposition 5. (Locality) The transition kernel over any state $s \in \mathcal{S}$, is defined over the set of states that are within one unit distance of s . Denoting this set as $\mathcal{S}'_s := \{s' \in \mathcal{S} \mid \mathcal{D}(s, s') \leq 1\}$, with $|\mathcal{S}'_s| \leq A$.

We upper-bound the number of unique, possible transitions for any state in M to establish a measure of locality using the size of the action-space. With Proposition 5, we are able to construct sub-MDPs using a subset of states within M and still preserve the overall dynamics of M within the region of the sub-MDP.

4 Local Approximation of Value Functions

Within this section, we establish the notion of sub-MDP and, through it, local approximations of value functions. Prior to defining a sub-MDP, we present the following Lemma.

Lemma 6. For $\epsilon, \gamma \in (0, 1)$, where γ is the discount factor, if $T \geq \log_\gamma(\epsilon(1 - \gamma))$, then $V_t(s) - V_t^T(s) \leq \epsilon, \forall t \in \mathbb{Z}^+, \forall s \in \mathcal{S}$.

Proof. Letting an infinite-horizon transition walk from state s at timestep t result in the maximum possible reward sequence, i.e. $\{1, 1, 1, \dots\}$, the cumulative discounted reward is the geometric series $V_t(s) = \sum_0^\infty \gamma$. Noting that $V_t^T(s) = \sum_0^T \gamma$, for the condition $V_t(s) - V_t^T(s) \leq \epsilon$ to hold, the following must be true and solving this expression for T yields the bound presented in the Lemma statement.

$$\sum_0^\infty \gamma - \sum_0^T \gamma = \gamma^{T+1} + \gamma^{T+2} \dots = \frac{\gamma^{T+1}}{1 - \gamma} \leq \epsilon$$

□

Using Lemma 6, noting MDP M is characterized by tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, we define a sub-MDP M_1 within M as follows.

Definition 7. (Sub-MDP) A sub-MDP M_1 is constructed using parameters $s_1 \in \mathcal{S}$ and $\epsilon \in (0, 1)$. M_1 is centered on s_1 and is characterized by tuple $(\mathcal{S}_1, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. The state-space of sub-MDP M_1 is $\mathcal{S}_1 := \{s \in \mathcal{S} \mid \mathcal{D}(s_1, s) < \log_\gamma(\epsilon(1 - \gamma))\}$. As $\mathcal{S}_1 \subset \mathcal{S}$, a transition defined in M from a state within \mathcal{S}_1 to a state outside of \mathcal{S}_1 results in a self-loop within M_1 .

Per Proposition 5, the sub-MDP is centered on a state within M and constructed using a breadth-wise expansion, denoted radius, of $T = \lceil \log_\gamma(\epsilon(1 - \gamma)) \rceil$ transitions from the center state. Intuitively, the size of the sub-MDP M_1 determines how close the optimal policy trained over it is to the optimal policy trained over the MDP M for state s_1 .

Remark 8. (Sub-MDP Size) Let a sub-MDP be constructed using parameters $s_1 \in \mathcal{S}$ and $\epsilon \in (0, 1)$. It will have a transition radius of $\lceil \log_\gamma(\epsilon(1 - \gamma)) \rceil$ and its state-space size is bounded by $\lceil \log_\gamma^A(\epsilon(1 - \gamma)) \rceil$.

The premise of our study is as follows, the further away a state $s_2 \in \mathcal{S}$ is from $s_1 \in \mathcal{S}$, the less bearing $V_*(s_2)$ has on $V_*(s_1)$. This is formalized in Lemma 6, which determines the length of rollout required for finite value function approximation $V_*^T(s_1)$ to approximate $V_*(s_1)$. As such, an optimal value function, $V_{*1}(s)$, generalized over a sub-MDP M_1 and state-space $\mathcal{S}_1 \subset \mathcal{S}$, parametrized by the center state s_1 and radius $T = \lceil \log_\gamma(\epsilon(1 - \gamma)) \rceil$, is ϵ -optimal with respect to M at s_1 . Furthermore, $V_{*1}(s)$ is relatively less optimal with respect to M further away from the center. This is formalized in Lemma 9.

Lemma 9. Let sub-MDP M_1 be constructed from MDP M using a subset of states $\mathcal{S}_1 \subset \mathcal{S}$, centered on state $s_1 \in \mathcal{S}_1$, as in Definition 7. Letting $V_*(s)$ be the optimal value function over M and $V_{*1}(s)$ be the optimal value function over M_1 , $|V_{*1}(s) - V_*(s)| \leq \epsilon/\gamma^{\mathcal{D}(s_1, s)}$ for all $s \in \mathcal{S}_1$.

Proof. Per Definition 7, sub-MDP M_1 is constructed with a transition radius of $T = \lceil \log_\gamma(\epsilon(1 - \gamma)) \rceil$. By Lemma 6, the theoretical optimal value function that can be generalized over M_1 will have at most ϵ error at s_1 . Noting that the transition radius determines the closeness of the value function over the sub-MDP to $V_*(s)$, the largest sub-MDP that can be constructed within \mathcal{S}_1 and centered on a state $s \in \mathcal{S}_1$ will have a transition radius of $T - \mathcal{D}(s_1, s)$.

Let $\epsilon_1 \in (0, 1)$ represent the error of the optimal value function $V_{*1}(s)$ over M_1 at a state $s \in \mathcal{S}_1$ when compared to the optimal value function $V_*(s)$ over M . Extending Lemma 6, the following holds.

$$\log_\gamma(\epsilon(1-\gamma)) - \mathcal{D}(s_1, s) \geq \log_\gamma(\epsilon_1(1-\gamma))$$

$$\frac{\epsilon(1-\gamma)}{\gamma^{\mathcal{D}(s_1, s)}} \geq \epsilon_1(1-\gamma)$$

□

With Lemma 9, we establish the relation between the optimal value function over M , $V_*(s)$, and the optimal value function over its sub-MDP M_1 , $V_{*1}(s)$, in terms of ϵ with respect to the state-space. Noting Remark 8 and the size of each sub-MDP, we posit the sample-complexity of learning for each sub-MDP in Lemma 10.

Lemma 10. *Let sub-MDP M_1 be constructed from MDP M , centered on state $s_1 \in \mathcal{S}$, as in Definition 7. The sample-complexity of learning an ϵ -optimal value function, $V_{*1}(s)$, with high confidence within M_1 is asymptotically bounded by $\tilde{O}(A \log_\gamma^A(\epsilon(1-\gamma)) \log(A \log_\gamma^A(\epsilon(1-\gamma))))$.*

Proof. Let M be a discrete MDP with a state-space size of S and an action-space size of A . Further, let $\delta \in (0, 1)$ be the confidence parameter and $\gamma \in (0, 1)$ be the discount factor. The sample-complexity bound, proposed in [23], for model-free reinforcement learning over M , without Propositions 4 and 5, is $O\left(\frac{SA}{\epsilon^2(1-\gamma)^3} \log\left(\frac{SA}{\delta(1-\gamma)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$. Noting that the state-space size of a sub-MDP within M is $\log_\gamma^A(\epsilon(1-\gamma))$, extending the previous sample-complexity bound to the sub-MDP results in a bound of

$$O\left(\frac{A \log_\gamma^A(\epsilon(1-\gamma))}{\epsilon^2(1-\gamma)^3} \log\left(\frac{A \log_\gamma^A(\epsilon(1-\gamma))}{\delta(1-\gamma)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$$

which is simplified to the expression in the lemma statement. □

Lemma 10 provides a sample-complexity bound for learning ϵ -optimal policies within sub-MDPs. Let M_1 be a sub-MDP of M , parametrized by central state $s_1 \in \mathcal{S}$ and error $\epsilon \in (0, 1)$, with a generalized ϵ -optimal value function $V_{*1}(s)$. Noting Lemma 6, i.e. $|V_{*1}(s_1) - V_*(s_1)| \leq \epsilon$, we get $|V_{*1}(s_1) - V_*(s_1)| \leq 2\epsilon$. Lemma 10 further implies that, assuming a naive division of M into S sub-MDPs by constructing a sub-MDP at each state, it is possible to generalize a 2ϵ -optimal policy over M within a fixed number of timesteps. This is formally stated in the following remark.

Remark 11. (Naive Sample-Complexity Bound) *Given the size of \mathcal{S} is S , assuming a sub-MDP is constructed centered on each state within \mathcal{S} , the sample-complexity of learning a 2ϵ -optimal function over M is naively bounded by $\tilde{O}(SA \log_\gamma^A(\epsilon(1-\gamma)) \log(A \log_\gamma^A(\epsilon(1-\gamma))))$.*

5 Sharper Asymptotic Bound on Sample-Complexity

In this section, we derive a sample-complexity bound for reinforcement learning using local approximations that is sharper than the naive bound proposed in Remark 11. Noting the bound on sample-complexity presented in Lemma 10, intuitively, the aim of this section is to determine the minimum number of sub-MDPs of M that are required to generalize a near-optimal value function over M .

Through Lemma 9, we note that there is a correlation between the approximation error of the sub-MDP's optimal value function at a state to $V_*(s)$ and the distance from the state to the center of the sub-MDP. Subsequently, we note that multiple sub-MDPs within M may overlap in state-space, and thus, we may obtain multiple sub-optimal

estimates of the value function for states that are not sub-MDP centers. Using Lemma 9, though, we are aware of the degree of sub-optimality and can make use of multiple estimates of the value function at any state in M to obtain a better approximation. In deriving sharper sample-complexity bounds, we utilize the following lemma.

Lemma 12. *Let $s_0 \in \mathcal{S}$ be a state in MDP M . Let $V_*(s_0)$ be the theoretical optimal value of s_0 with respect to M . Let $V_{*1}(s_0), V_{*2}(s_0) \dots V_{*N}(s_0)$ be $\epsilon_1, \epsilon_2 \dots \epsilon_N \in (0, 1)$ approximations of $V_*(s_0)$, respectively. Then,*

$$\mathbb{P}\left(\left|\frac{1}{N} \sum_{i=1}^N V_{*i}(s_0) - V_*(s_0)\right| \geq \epsilon\right) \leq 2e^{\frac{-2N^2\epsilon^2}{\sum_{i=1}^N 2\epsilon_i^2}}.$$

Proof. Sketch. This result is achieved through a direct application of Hoeffding's Inequality [7]. □

We use Lemma 12 in determining the minimum overlap required by sub-MDPs in order to generalize an ϵ -optimal value function over M with high confidence. This allows us to determine the minimum number of sub-MDPs required. Let $M_1, M_2 \dots M_N$ represent sub-MDPs of M that are constructed using arbitrary centers and error parameter $\epsilon \in (0, 1)$ over state-spaces that are subsets of \mathcal{S} , i.e. $\mathcal{S}_1, \mathcal{S}_2 \dots \mathcal{S}_N \subset \mathcal{S}$, respectively. Let $V_{*1}(s), V_{*2}(s) \dots V_{*N}(s)$ be the optimal value functions over the respective sub-MDPs. Let $s_0 \in \mathcal{S}_1 \cap \mathcal{S}_2 \cap \dots \cap \mathcal{S}_N$ be a common state within all sub-MDPs and $\epsilon_1, \epsilon_2 \dots \epsilon_N$ be the respective errors between $V_*(s_0)$ and $V_{*1}(s_0), V_{*2}(s_0) \dots V_{*N}(s_0)$. A direct application of Lemma 12 allows us to determine the confidence with which the value function generalized through local approximation is ϵ -optimal. Within Lemma 13, we derive the amount of overlap in state-spaces required by sub-MDPs to achieve the ϵ -optimal result with a confidence given by $\delta \in (0, 1)$.

Lemma 13. *Let $\epsilon, \delta \in (0, 1)$ represent the error and confidence parameters. For all states $s \in \mathcal{S}$, if there are at least $\frac{2}{\epsilon} \log \frac{2S}{\delta}$ sub-MDP centers within $\lceil \log_\gamma 0.5 \rceil$ transitions of s , an ϵ -optimal value function over M can be generalized with $1 - \delta$ confidence using the sub-MDP value function estimates.*

Proof. Let $V_*(s)$ be the optimal value function over M . Let $M_1, M_2 \dots M_N$ be sub-MDPs of M parametrized by centers $s_1, s_2 \dots s_N \in \mathcal{S}$ and approximation error $\epsilon \in (0, 1)$. Let $V_{*1}(s), V_{*2}(s) \dots V_{*N}(s)$ represent the optimal value functions over the respective sub-MDPs. For a state $s_0 \in \mathcal{S}$, if $|V_{*i}(s_0) - V_*(s_0)| \leq 2\epsilon$, then $\mathcal{D}(s_0, s_i) \leq \log_\gamma 0.5$ by Lemma 6 for all $i \in \{1, 2 \dots N\}$, as the transition radius from s_0 to the boundary of M_i permits the learning of a 2ϵ -optimal value function.

Noting that the size of \mathcal{S} is S , we use Lemma 12 to bound the probability that the difference between $\frac{1}{N} \sum_{i=1}^N V_{*i}(s_0)$ and $V_*(s_0)$ is greater than ϵ in the following expression by substituting 2ϵ for all ϵ_i .

$$\mathbb{P}\left(\left|\frac{1}{N} \sum_{i=1}^N V_{*i}(s_0) - V_*(s_0)\right| \geq \epsilon\right) \leq 2e^{\frac{-N^2\epsilon^2}{\sum_{i=1}^N 2\epsilon_i^2}} \leq \frac{\delta}{S}$$

Resolving this, we obtain the following result.

$$N \geq \frac{2}{\epsilon} \log \frac{2S}{\delta}$$

□

The implication of Lemma 13 is that the number of sub-MDP overlaps for any state scales logarithmically with the state-space size.

Furthermore, for a fixed value of ϵ , the confidence of the generalized value function over M increases with the sub-MDP overlap within the state-space. However, we still lack a correlation between the size of the state-action space and the number of sub-MDPs required. Intuitively, the number of sub-MDPs required increases with the size of the state-space as there are more states to be covered. However, for a fixed state-space size, an increase in the action-space size increases the connectivity of MDP M , thereby decreasing the transition distance between states. Consequently, an increase in the action-space increases the state-space size of a sub-MDP for the same transition radius and decreases the overall number of sub-MDPs required within M to provide a value function estimate. This relation is formalized in Lemma 14.

Lemma 14. *Let $\epsilon \in (0, 1)$ represent the construction error for sub-MDPs $M_1, M_2 \dots M_L$ constructed in M . Let $S_1, S_2 \dots S_L$ be the respective state-spaces of the sub-MDPs. Further, let $S_1 \cup S_2 \cup \dots \cup S_L = S$. The value of L is lower-bounded by $\frac{2S}{\epsilon \log_{\gamma}^A \epsilon(1-\gamma)} \log \frac{2}{\delta}$.*

Proof. As a consequence of Propositions 4 and 5, the coverage of states in M by the sub-MDPs is maximized when the sub-MDP centers are distributed uniformly over S . Letting the action-space size of M and, subsequently, $M_1, M_2 \dots M_L$, be A , noting Remark 8, the state-space size of each individual sub-MDP is upper bounded by $\lceil \log_{\gamma}^A \epsilon(1-\gamma) \rceil$. Noting that the overall number of value estimates made is L multiplied by the sub-MDP size, we establish the following bound relating the number of estimates with N , the number of per-state estimates.

$$L \log_{\gamma}^A \epsilon(1-\gamma) > SN$$

Equivalently,

$$L \log_{\gamma}^A \epsilon(1-\gamma) > S \frac{2}{\epsilon} \log \frac{2S}{\delta}$$

Noting that there are L sub-MDP centers, i.e. states that do not require repeat estimation via overlap, the following adjustment to the bound is made.

$$L \log_{\gamma}^A \epsilon(1-\gamma) > S \frac{2}{\epsilon} \log \frac{2S}{\delta} - L$$

We relax the inequality by letting $\log S$ be a lower-bound on L in the RHS. Solving for L yields the bound in the lemma statement. \square

Using Lemma 13, which indicates the number of times a state's value needs to be independently determined within M , we establish Lemma 14, which indicates the number of sub-MDPs required in an MDP with state-space size S and action-space size A to achieve this coverage. We further posit that the overall sample-complexity required to learn an 2ϵ -optimal value function over M with $1-\delta$ confidence is the sub-MDP sample-complexity, established in Lemma 10, multiplied by $\frac{2S}{\epsilon \log_{\gamma}^A \epsilon(1-\gamma)} \log \frac{2}{\delta}$. This result is intuitive as the number of sub-MDPs multiplied by the sample-complexity required for each sub-MDP yields the overall sample-complexity. We formalize this in the following theorem.

Theorem 15. (Main Result) *Let $\epsilon, \delta, \gamma \in (0, 1)$ represent error, confidence and discount factor. For an MDP M with state-space size S and action-space size A , by locally approximating M using sub-MDPs, an ϵ -optimal value function will be generalized within $O\left(\frac{SA}{\epsilon^3(1-\gamma)^3} \log\left(\frac{A}{\delta(1-\gamma)}\right) \log \frac{1}{\epsilon} \log \frac{1}{\delta}\right)$ timesteps with $1-\delta$ confidence.*

Proof. The proof of this theorem follows from Lemmas 10 and 14. We determine the sample-complexity of learning a 2ϵ -optimal value function as an asymptotic bound in $O(LC)$, where L is the number of sub-MDPs required to be constructed and C is the sample-complexity of learning for each sub-MDP. We have determined the number of sub-MDPs required to approximate the larger MDP M to be bounded by $\frac{2S}{\epsilon \log_{\gamma}^A \epsilon(1-\gamma)} \log \frac{2}{\delta}$. By Lemma 10, we have determined the sample-complexity of each sub-MDP to be $O\left(\frac{A \log_{\gamma}^A (\epsilon(1-\gamma))}{\epsilon^2(1-\gamma)^3} \log\left(\frac{A \log_{\gamma}^A (\epsilon(1-\gamma))}{\delta(1-\gamma)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$. The multiplication of the two terms results in the asymptotic bound presented in the theorem statement.

Note that the sample-complexity required to generalize an ϵ -optimal value function, as opposed to it being 2ϵ -optimal, requires substituting $\epsilon/2$ into the lemma statement. This introduces a constant factor that is disregarded when taking the asymptotic bound. \square

Theorem 15 presents a theoretical bound on the sample-complexity of learning. We construct an algorithm in Section 6 that makes use of this result to efficiently learn a value function.

6 The PDQL Algorithm

Within this section, we describe a model-free RL algorithm, Probabilistic Delayed Q-Learning (PDQL), that utilizes local approximation in generalizing a 2ϵ -optimal value function. We extend results from existing algorithms, DQL [17] and VRQL [23] using the results in the prior sections. Intuitively, there are two characteristics of PDQL that distinguish it from Q-Learning: the *update condition* and the *lock condition*. PDQL is detailed in Algorithm 1.

6.1 The Update Condition

The algorithm begins by initializing all state-action Q-values to $1/(1-\gamma)$, where $\gamma \in (0, 1)$ is the discount factor. For $\epsilon, \delta \in (0, 1)$, the update condition for PDQL is as follows, where $\mathbb{E}_T[Q_t(s, a)]$ is the expectation of the T -step Q-value computed over the distribution of possible transition walks from (s, a) after t timesteps. Letting T correspond with sub-MDP radius $\lceil \log_{\gamma} \epsilon(1-\gamma) \rceil$ per Lemma 6, if the following inequality is satisfied, the stored Q-value is updated to the new expected Q-value.

$$Q_t(s, a) - \mathbb{E}_T[Q_t(s, a)] \geq 2\epsilon \quad (1)$$

The rationale for this condition follows from [17]. When generalizing an ϵ -optimal policy, to prevent instability, it is sufficient to update a Q-value if the difference between the current value and the expected value is significant enough. In order to estimate the expected Q-value, we sample the Q-value of a state q times and average it when making the update. The update step in line 7 of Algorithm 1 is formalized as follows, where $V_t^T(s)$ is the T -step value function at timestep t and s'_i is the state transitioned to on the i -th execution of (s, a) .

$$Q_{t+1}(s, a) := \frac{1}{q} \sum_{i=1}^q \left(r_i(s, a) + \gamma V_t^T(s'_i) \right) + \epsilon, \quad (2)$$

Note that we add an ϵ to the update to ensure that $Q_t(s, a) > Q_*(s, a)$ for all $t \geq 0$. Assuming q is large enough, then with δ confidence, the stored value of $Q_{t+1}(s, a)$ is within ϵ of the expected value. The following lemma indicates the lower-bound of q for this to occur.

Algorithm 1 Probabilistic Delayed Q-Learning

Inputs:

- 1: State-Space: \mathcal{S}
- 2: Action-Space: \mathcal{A}
- 3: Transition Function: \mathcal{T}
- 4: Reward Function: \mathcal{R}

Parameters:

- 1: Discount Factor: γ
- 2: Sampling Parametre: q
- 3: Error Bound: ϵ
- 4: Transition Radius: T

Initialize Values:

- 1: **for** all values $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
- 2: $Q(s, a) \leftarrow \frac{1}{1-\gamma}$ %current Q-Value
- 3: $U(s, a) \leftarrow 0$ %update attempt
- 4: $C(s, a) \leftarrow 0$ %visit counter
- 5: $UNLOCK(s, a) \leftarrow True$ %learning lock
- 6: **end for**

Note: Let R be the current state reward and s' be the subsequent state after executing a on s under \mathcal{T}

Output:

- 1: **while** there exists $(s, a) \mid UNLOCK(s, a) = True$ **do**
 - 2: Select state-action
 - 3: $(s, a) \mid a = \text{argmax}_{a \in \mathcal{A}} Q(s, a)$
 - 4: $U(s, a) \leftarrow U(s, a) + R + \gamma Q(s', a')$
 - 5: $C(s, a) \leftarrow C(s, a) + 1$,
 - 6: **if** $C(s, a) = q$ **then**
 - 7: $C(s, a) \leftarrow 0$,
 - 8: **if** $Q(s, a) - U(s, a)/q \geq 2\epsilon$ **then**
 - 9: $Q(s, a) \leftarrow U(s, a)/q + \epsilon$
 - 10: **for all** $s' \mid \mathcal{D}(s, s') < \lceil \log_\gamma(\epsilon(1-\gamma)) \rceil$ **do**
 - 11: **for all** $a' \in \mathcal{A}$ **do**
 - 12: $UNLOCK(s', a') \leftarrow True$
 - 13: **end for**
 - 14: **end for**
 - 15: **else**
 - 16: $UNLOCK(s, a) \leftarrow False$
 - 17: **end if**
 - 18: **end if**
 - 19: **end while**
-

Lemma 16. For $\epsilon, \delta \in (0, 1)$, if the following bound on q holds, then $Q_t(s, a) - \mathbb{E}_T[Q_t(s, a)] < \epsilon$ holds for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and all timesteps $t \geq 0$ with $1 - \delta$ confidence.

$$q \geq \frac{\log\left(\frac{2SA}{\epsilon} \log \frac{2}{\delta} \left(\frac{1}{\log_\gamma \epsilon(1-\gamma)} + \frac{A}{\epsilon(1-\gamma)}\right)\right)}{2\epsilon^2(1-\gamma)^2}$$

Proof. Note that each state-action's Q-value can be updated at most $\frac{1}{\epsilon(1-\gamma)}$ times due to the update condition requiring that there is a difference of ϵ between each update and the initialization of each Q-value to $\frac{1}{1-\gamma}$. As such, there can be at most $\frac{A \log_\gamma \epsilon(1-\gamma)}{\epsilon(1-\gamma)}$ successful Q-value updates made within a sub-MDP.

The remainder of this proof utilizes Subsection 6.2. Further noting the locking condition and Lemma 17, in the worst case there can be at most $AL \left(1 + \frac{A \log_\gamma \epsilon(1-\gamma)}{\epsilon(1-\gamma)}\right)$ attempted updates made, where L is the lower-bound on the number of sub-MDPs posited by Lemma 14. This scenario occurs when an update is attempted, only for the condition in (1) to not be met, resulting in an unsuccessful update attempt and the state being *locked*. If further updates are possible, regardless of the lock being placed, there may be up to AL updates made before the state and its associated sub-MDP are unlocked.

As such, we use an application of Hoeffding's Inequality to yield the following inequalities, bounding the difference between the expected Q-value and mean of encountered Q-values with high confidence. Solving for q yields the result in the lemma statement.

$$\mathbb{P}(Q_t(s, a) - \mathbb{E}_T[Q_t(s, a)] > \epsilon) < \exp(-2q\epsilon^2(1-\gamma)^2)$$

$$\frac{\delta}{AL \left(1 + \frac{A \log_\gamma \epsilon(1-\gamma)}{\epsilon(1-\gamma)}\right)} < \exp(-2q\epsilon^2(1-\gamma)^2)$$

□

This lower-bound on sampling size is corroborated by VRQL and is needed to be held in order for the sample-complexity of learning for each sub-MDP to be equivalent to the asymptotic bound presented in Lemma 10, derived from [23]. If a state-action (s, a) is

sampled q times and the update condition in (1) is not met, the state s is *locked* and further updates are not to take place. We elaborate on this in the following subsection.

6.2 The Lock Condition

The primary method by which we apply local approximation in the formulation of PDQL is through the *locking/unlocking mechanism* of PDQL. For each state, PDQL assigns a boolean variable that indicates whether further updates are possible to its value. Following an attempted update, if the condition in (1) is not met, the update is not carried out and the state is *locked* - barring it from future updates. The locking mechanism has two purposes: firstly, it is used to indicate convergence, i.e. when all states are locked, the algorithm has converged; secondly, it allows the algorithm to converge faster by focusing effort toward the *unlocked* states.

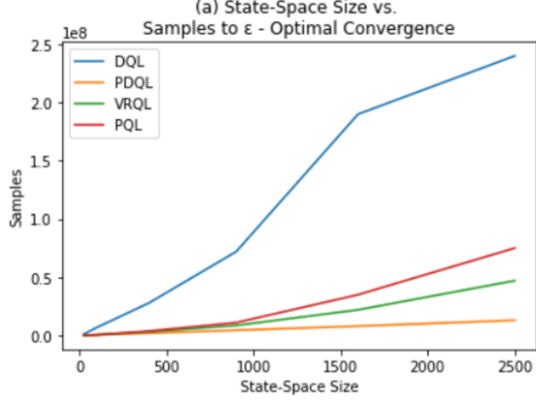
Per Lemma 6, for values $\epsilon, \gamma \in (0, 1)$, an update to a Q-value of a state $s_1 \in \mathcal{S}$ can affect the Q-values of the states within $\lceil \log_\gamma \epsilon(1-\gamma) \rceil$ transitions of s_1 . This corresponds with Definition 7, which defines sub-MDPs. We formulate the following lemma regarding this.

Lemma 17. (Unlocking Radius) If a state's value is affected by more than ϵ from an update to a state $s_1 \in \mathcal{S}$, it must lie within the state-space of a sub-MDP centered on s_1 with a radius of $\lceil \log_\gamma \epsilon(1-\gamma) \rceil$ transitions.

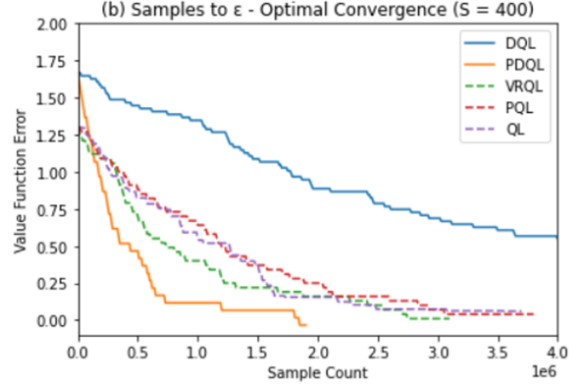
Proof. Sketch. This lemma is a consequence of Lemma 6. □

With regard to Lemma 17, noting that an update on a state-action may result in updates to state-action Q-values in its vicinity, all states in the vicinity of s_1 must be subsequently *unlocked* to generalize an ϵ -optimal value function. That is, for the sub-MDP constructed within the state-space \mathcal{S}_1 , where $\mathcal{S}_1 := \{s \in \mathcal{S} \mid \mathcal{D}(s_1, s) < \log_\gamma(\epsilon(1-\gamma))\}$, all $s \in \mathcal{S}_1$ are *unlocked* within PDQL. This is conducted in line 9 of Algorithm 1.

Intuitively, the *unlocking* procedure is a method of constructing sub-MDPs within M based on the states that probably require up-



(a) Average growth of convergence rate with respect to the state-space size, where the algorithm is said to have converged upon reaching ϵ -optimality.



(b) Comparison of convergence rates between algorithms (Q-Learning, DQL, PQL, VRQL and PDQL) with respect to sample count, t , where mean error is measured as the expected difference between the current value function and the optimal value function, i.e. $\mathbb{E}_s[V_t(s) - V_*(s)]$.

dates. With minimal overlap, Lemma 14 indicates how many sub-MDPs can be unlocked simultaneously. Similarly, the *locking* procedure diverts the algorithm's efforts to areas of the state-space that are more likely to require attention, where Lemmas 13 and 16 indicate precisely how much effort is required to be provided. With this, PDQL demonstrates the asymptotic sample-complexity derived in Theorem 15.

The locking procedure in our work is in direct contrast with DQL [17], where a *global unlocking* procedure is implemented instead of a *local unlocking* resulting in greater sample-efficiency.

6.3 Space-Complexity Analysis of PDQL

Referring to Definition 2 for the space-complexity bounds on model-free RL algorithms, we briefly show PDQL is within the bound of $\tilde{O}(SA)$. Algorithm 1 stores the following: a record of up to q Q-values for each state-action $(s, a) \in \mathcal{S} \times \mathcal{A}$, $O(SA)$; and a lock variable for each state, $O(S)$. It is realistic to assume the existence of an oracle that can be queried to yield the minimum number of transitions between two states due to the distance metric placed by Proposition 4. However, in its absence, a minor adjustment to the algorithm can be made such that the sub-MDP records are stored to facilitate the unlocking procedure, requiring the following space-complexity $O(L \log_{\gamma}^A \epsilon(1 - \gamma))$. Simplifying this, the asymptotic space-complexity of PDQL is $O(qSA + S + \frac{2S}{\epsilon} \log \frac{2}{\delta})$, which is equivalent to $\tilde{O}(SA)$.

7 Experiments

In order to demonstrate the results of this study, we have conducted experiments to observe the convergence rate of PDQL in various environments against similar tabular PAC-MDP algorithms. As our primary result is the removal of the logarithmic dependency on the state-space size, we deploy our learning algorithm in environments of different sizes to show the growth in the convergence rate of PDQL. We demonstrate our results in discretized versions of the *Lunar Lander* environment from *Gymnasium* [21].

The parameters of our experiment are as follows: $\gamma = 0.9$; $\delta = 0.001$; $\epsilon = 0.01$; $S = \{50, 200, 500, 1000, 1500, 2000\}$; with the state-wise reward scaled to the range $[0, 1]$. Considering

that this algorithm is based on Q-Learning, this paper shows its comparison with Q-Learning, Delayed Q-Learning (DQL)[17], Phased Q-Learning (PQL) [12] and Variance Reduced Q-Learning (VRQL) [23].

There are two characteristics of the data that we comment on. The first is the rate of convergence displayed in Figure b, which shows that PDQL has a faster rate of convergence than the other forms of Q-Learning. This also confirms the proposition that DQL and VRQL are conservative in the number of samples required prior to an update, noting its relatively slow rate of convergence when compared with the other algorithms. This was observed during all the trials recorded. The second is the rate of growth in the number of samples to convergence with respect to the size of the state-space, which scales more efficiently in $SA \log A$ as the bound suggests for PDQL than $SA \log(SA)$ for the remaining algorithms (Figure a).

8 Conclusion

The objective of our study has been to sharpen sample-complexity bounds of model-free RL under Propositions 4 and 5. We do so by constructing local approximations of the larger MDP and learning the value function locally. We then combine the learnt values alongside the associated error to approximate the value function across the larger MDP with high confidence. The main result that we present is that an ϵ -optimal value function can be generalized within $O(SA \log A)$ timesteps with $1 - \delta$ confidence. As such, the algorithm that we present in this paper, PDQL, demonstrates faster convergence than previous works.

Acknowledgements

To be added...

References

- [1] M. G. Azar, R. Munos, M. Ghavamzadeh, and H. J. Kappen. Speedy q-learning. In *NIPS*, 2011.
- [2] C. Dann, T. Lattimore, and E. Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning, 2018.
- [3] K. Dong, Y. Wang, X. Chen, and L. Wang. Q-learning with ucb exploration is sample efficient for infinite-horizon mdp, 2019.
- [4] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driv-ing. *STAT*, 1050:8, 2017.

- [5] C.-N. Fiechter. Efficient reinforcement learning. In *Proceedings of the Seventh Annual Conference on Computational Learning Theory, COLT '94*, page 88–97, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916557. doi: 10.1145/180139.181019.
- [6] J. He, D. Zhou, and Q. Gu. Nearly minimax optimal reinforcement learning for discounted mdps, 2022.
- [7] W. Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.
- [8] Q. Huang. Model-based or model-free, a review of approaches in reinforcement learning. In *2020 International Conference on Computing and Data Science (CDS)*, pages 219–221, 2020. doi: 10.1109/CDS49703.2020.00051.
- [9] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [10] S. Kakade. On the sample complexity of reinforcement learning. 01 2003.
- [11] S. Kakade, M. J. Kearns, and J. Langford. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 306–312, 2003.
- [12] M. Kearns and S. Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems*, 11, 1998.
- [13] L. Li, R. Yang, and D. Luo. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*, 2020.
- [14] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.
- [15] S. R. Sinclair, S. Banerjee, and C. L. Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019.
- [16] Z. Song and W. Sun. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.
- [17] A. Strehl, L. Li, E. Wiewiora, J. Langford, and M. Littman. Pac model-free reinforcement learning. volume 2006, 01 2006. doi: 10.1145/1143844.1143955.
- [18] C. Szepesvári. The asymptotic convergence-rate of q-learning. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997.
- [19] I. Szita and C. Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. pages 1031–1038, 08 2010.
- [20] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone. Deep reinforcement learning for robotics: A survey of real-world successes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28694–28698, 2025.
- [21] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- [22] J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16:185–202, 1994.
- [23] M. J. Wainwright. Variance-reduced q-learning is minimax optimal, 2019.
- [24] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
- [25] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion planning and control for mobile robot navigation using machine learning: a survey. *Autonomous Robots*, 46(5):569–597, 2022.
- [26] Z. Zhang, Y. Zhou, and X. Ji. Model-free reinforcement learning: from clipped pseudo-regret to sample complexity, 2020.