



Adaptive Discretization for Episodic Reinforcement Learning in Metric Spaces

SEAN R. SINCLAIR, Cornell University, USA

SIDDHARTHA BANERJEE, Cornell University, USA

CHRISTINA LEE YU, Cornell University, USA

We present an efficient algorithm for model-free episodic reinforcement learning on large (potentially continuous) state-action spaces. Our algorithm is based on a novel Q -learning policy with adaptive data-driven discretization. The central idea is to maintain a finer partition of the state-action space in regions which are frequently visited in historical trajectories, and have higher payoff estimates. We demonstrate how our adaptive partitions take advantage of the shape of the optimal Q -function and the joint space, without sacrificing the worst-case performance. In particular, we recover the regret guarantees of prior algorithms for continuous state-action spaces, which additionally require either an optimal discretization as input, and/or access to a simulation oracle. Moreover, experiments demonstrate how our algorithm automatically adapts to the underlying structure of the problem, resulting in much better performance compared both to heuristics and Q -learning with uniform discretization.

CCS Concepts: • **Computing methodologies** → **Q -learning**; *Sequential decision making*; Continuous space search; • **Theory of computation** → *Markov decision processes*;

Keywords: reinforcement learning, metric spaces, adaptive discretization, Q -learning, model-free

ACM Reference Format:

Sean R. Sinclair, Siddhartha Banerjee, and Christina Lee Yu. 2019. Adaptive Discretization for Episodic Reinforcement Learning in Metric Spaces. In *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 3, 3, Article 55 (December 2019). ACM, New York, NY. 44 pages. <https://doi.org/10.1145/3366703>

1 INTRODUCTION

Reinforcement learning (RL) is a natural model for systems involving real-time sequential decision making [26]. An agent interacts with a system having stochastic transitions and rewards, and aims to learn to control the system by exploring available actions and using real-time feedback. This requires the agent to navigate the *exploration exploitation trade-off*, between exploring unseen parts of the environment and exploiting historical high-reward actions. In addition, many RL problems involve large state-action spaces, which makes learning and storing the entire transition kernel infeasible (for example, in memory-constrained devices). This motivates the use of *model-free RL algorithms*, which eschew learning transitions and focus only on learning good state-action mappings. The most popular of these algorithms is Q -learning [2, 10, 29], which forms the focus of our work.

The code for the experiments is available at <https://github.com/seanrsinclair/AdaptiveQLearning>.

Authors' addresses: Sean R. Sinclair, srs429@cornell.edu, Cornell University, USA; Siddhartha Banerjee, sbanerjee@cornell.edu, Cornell University, USA; Christina Lee Yu, cleeyu@cornell.edu, Cornell University, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2476-1249/2019/12-ART55 \$15.00

<https://doi.org/10.1145/3366703>

In even higher-dimensional state-spaces, in particular, continuous spaces, RL algorithms require embedding the setting in some metric space, and then using an appropriate discretization of the space. A major challenge here is in learning an “optimal” discretization, trading-off memory requirements and algorithm performance. Moreover, unlike optimal quantization problems in ‘offline’ settings (i.e., where the full problem is specified), there is an additional challenge of learning a good discretization and control policy when the process of learning itself must also be constrained to the available memory. This motivates our central question:

Can we modify Q -learning to learn a near-optimal policy while limiting the size of the discretization?

Current approaches to this problem consider uniform discretization policies, which are either fixed based on problem primitives, or updated via a fixed schedule (for example, via a ‘doubling trick’). However, a more natural approach is to adapt the discretization over space and time in a data-driven manner. This allows the algorithm to learn policies which are not uniformly smooth, but adapt to the geometry of the underlying space. Moreover, the agent would then be able to explore more efficiently by only sampling important regions.

Adaptive discretization has been proposed and studied in the simpler multi-armed bandit settings [12, 23]. The key idea here is to develop a non-uniform partitioning of the space, whose coarseness depends on the density of past observations. These techniques, however, do not immediately extend to RL, with the major challenge being in dealing with error propagation over periods. In more detail, in bandit settings, an algorithm’s *regret* (i.e., additive loss from the optimal policy) can be decomposed in straightforward ways, so as to isolate errors and control their propagation. Since errors can propagate in complex ways over sample paths, naive discretization could result in over-partitioning suboptimal regions of the space (leading to over exploration), or not discretizing enough in the optimal region due to noisy samples (leading to loss in exploitation). Our work takes an important step towards tackling these issues.

Adaptive partitioning for reinforcement learning makes progress towards addressing the challenge of limited memory for real-time control problems. In particular, we are motivated by considering the use of RL for computing systems problems such as memory management, resource allocation, and load balancing [6, 15]. These are applications in which the process of learning the optimal control policy must be implemented directly on-chip due to latency constraints, leading to restrictive memory constraints. Adaptive partitioning finds a more “efficient” discretization of the space for the problem instance at hand, reducing the memory requirements. This could have useful applications to many control problems, even with discrete state spaces, as long as the model exhibits “smoothness” structure between nearby state-action pairs.

1.1 Our Contributions

As the main contribution of this paper, we design and analyze a *Q -learning algorithm based on data-driven adaptive discretization of the state-action space*. Our algorithm only requires that the underlying state-action space can be embedded in a compact metric space, and that the optimal Q -function is Lipschitz continuous with respect to the metric. This setting is general, encompassing discrete and continuous state-action spaces, deterministic systems with natural metric structures, and stochastic settings with mild regularity assumptions on the transition kernels. Notably, our algorithm only requires access to the metric, unlike other algorithms which require access to simulation oracles. In addition, our algorithm is *model-free*, requiring less space and computational complexity to learn the optimal policy.

From a theoretical perspective, we show that our adaptive discretization policy achieves near-optimal dependence of the regret on the covering dimension of the metric space. In particular, we

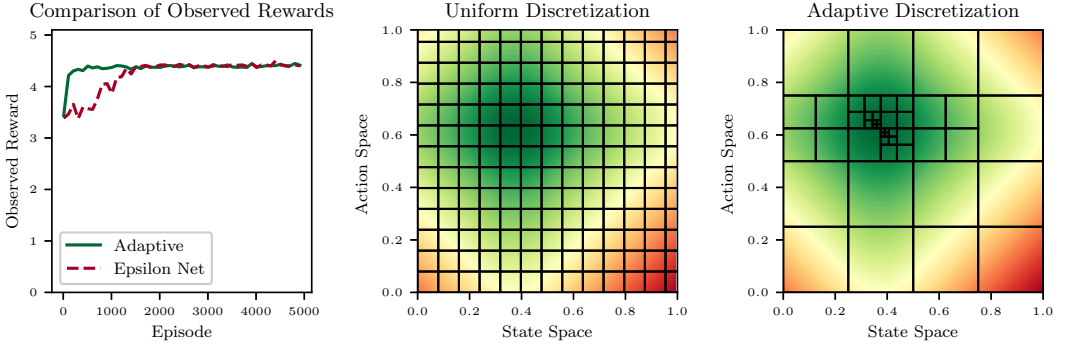


Fig. 1. Comparison of the observed rewards and state-action space discretization under the uniform mesh (ϵ -Net) algorithm [25] and our adaptive discretization algorithm. Both algorithms were applied to the ambulance routing problem with shifting uniform arrival distributions (see Section 7.1). The colors correspond to the relative Q_h^* value of the given state-action pair, where green corresponds to a higher value for the expected future rewards. The adaptive algorithm converges faster to the optimal policy by keeping a coarser discretization of unimportant parts of the space and a fine discretization in important parts.

prove that over K episodes, our algorithm achieves a regret bound

$$R(K) = \tilde{O}\left(H^{5/2}K^{(d+1)/(d+2)}\right)$$

where d is the covering dimension and H is the number of steps in each episode. Moreover, for non-uniform metric spaces where the covering dimension is not tight, we show improved bounds which adapt to the geometry of the space.

Our algorithm manages the trade-off between exploration and exploitation by careful use of event counters and upper-confidence bounds. It then creates finer partitions of regions which have high empirical rewards and/or are visited frequently, to obtain better Q -value estimates. This reduces the memory requirement of our RL algorithm, as it adapts the discretization of the space to learn the shape of the optimal policy. In addition, it requires less instance dependent tuning, as it only needs to tune the scaling of the confidence bounds. Implementing uniform mesh algorithms, in contrast, also requires tuning the mesh size.

We complement our theoretical guarantees with experiments, where we compare our adaptive Q -learning algorithm to the net based Q -learning algorithm from [25] on two canonical problems with continuous spaces. Our algorithm achieves order-wise better empirical rewards compared to the uniform mesh algorithm, while maintaining a much smaller partition. Moreover, the performance gap of our algorithm to the uniform mesh algorithm grow larger with increasing non-uniformity in the underlying Q -function. As an example, in Figure 1 we demonstrate the performance of our algorithm and net based Q -learning for a canonical ambulance routing problem (cf. Section 7). We see that the adaptive discretization maintains different levels of coarseness across the space, resulting in a faster convergence rate to the optimal policy as compared to the uniform mesh algorithm.

1.2 Related Work

Our work sits at the intersection of two lines of work – on model-free Q -learning, and on adaptive zooming for multi-armed bandits. We highlight some of the closest work below; for a more extensive list of references, refer to [26] (for RL) and [4, 24] (for bandits). There are two popular approaches to RL algorithms: model-free and model-based.

Model-based. Model-based algorithms are based on learning a model for the environment, and use this to learn an optimal policy [2, 11, 13, 17–19]. These methods converge in fewer iterations, but have much higher computation and space complexity. As an example, the UCBVI algorithm [2] requires storing an estimate of the transition kernel for the MDP, leading to a space complexity of $O(S^2AH)$ (where S is the number of states, A is the number of actions, and H the number of steps per episode). Another algorithm for discrete spaces, UCRL [1], and its state-aggregation followup [17, 18], maintain estimates of the transition kernel and use this for learning the optimal policy. Other model-based approaches assume the optimal Q -function lies in a function class and hence can be found via kernel methods [30, 32], or that the algorithm has access to an oracle which calculates distributional shifts [8].

There has been some work on developing model-based algorithms for reinforcement learning on metric spaces [13, 17, 19]. The Posterior Sampling for Reinforcement Learning algorithm [19] uses an adaptation of Thompson sampling, showing regret scaling in terms of the Kolmogorov and eluder dimension. Other algorithms like UCCRL [17] and UCCRL-Kernel Density [13] extend UCRL [1] to continuous spaces by picking a uniform discretization of the state space and running a discrete algorithm on the discretization with a finite number of actions. The regret bounds scale in terms of $K^{(2d+1)/(2d+2)}$. Our algorithm achieves better regret, scaling via $K^{(d+1)/(d+2)}$ and works for continuous action spaces.

Model-free. Our work follows the model-free paradigm of learning the optimal policy directly from the historical rewards and state trajectories without fitting the model parameters; these typically have space complexity of $O(SAH)$, which is more amenable for high-dimensional settings or on memory constrained devices. The approach most relevant for us is the work on Q -learning first started by Watkins [29] and later extended to the discrete model-free setting using upper confidence bounds by Jin et al. [10]. They show a regret bound scaling via $\tilde{O}(H^{5/2}\sqrt{SAK})$ where S is the number of states and A is the number of actions.

These works have since led to numerous extensions, including for infinite horizon time discounted MDPs [7], continuous spaces via uniform ϵ -Nets [25], and deterministic systems on metric spaces using a function approximator [31]. The work by Song et al. assumes the algorithm has access to an optimal ϵ -Net as input, where ϵ is chosen as a function of the number of episodes and the dimension of the metric space [25]. Our work differs by adaptively partitioning the environment over the course of learning, only requiring access to a covering oracle as described in Section 2.3. While we recover the same worst-case guarantees, we show an improved covering-type regret bound (Section 4.2). The experimental results presented in Section 7 compare our adaptive algorithm to their net based Q -learning algorithm.

The work by Yang et al. for deterministic systems on metric spaces shows regret scaling via $\tilde{O}(HK^{d/(d+1)})$ where d is the doubling dimension [31]. As the doubling dimension is at most the covering dimension, they achieve better regret specialized to deterministic MDPs. Our work achieves sub-linear regret for stochastic systems as well.

Lastly, there has been work on using Q -learning with nearest neighbors [21]. Their setting considers continuous state spaces but finitely many actions, and analyzes the infinite horizon time-discounted case. While the regret bounds are generally incomparable (as we consider the finite horizon non-discounted case), we believe that nearest-neighbor approaches can also be used in our setting. Some preliminary analysis in this regards is in Section 6.

Adaptive Partitioning. The other line of work most relevant to this paper is the literature on adaptive zooming algorithms for multi-armed bandits. For a general overview on the line of work on regret-minimization for multi-armed bandits we refer the readers to [4, 14, 24]. Most relevant to us is the work on bandits with continuous action spaces where there have been numerous

algorithms which adaptively partition the space [5, 12]. Slivkins [23] similarly uses data-driven discretization to adaptively discretize the space. Our analysis supersedes theirs by generalizing it to reinforcement learning. Recently, Wang et al. [27] gave general conditions for a partitioning algorithm to achieve regret bounds in contextual bandits. Our partitioning can also be generalized in a similar way, and the conditions are presented in Section 6.

1.3 Outline of the paper

Section 2 presents preliminaries for the model. The adaptive Q -learning algorithm is explained in Section 3 and the regret bound is given in Section 4. Section 5 presents a proof sketch of the regret bound. Section 7 presents numerical experiments of the algorithm. Proofs are deferred to the appendix.

2 PRELIMINARIES

In this paper, we consider an agent interacting with an underlying finite-horizon Markov Decision Processes (MDP) over K sequential episodes, denoted $[K] = \{1, \dots, K\}$.

The underlying MDP is given by a five-tuple $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ where \mathcal{S} denotes the set of states, \mathcal{A} the set of actions, and horizon H is the number of steps in each episode. We allow the state-space \mathcal{S} and action-space \mathcal{A} to be large (potentially infinite). Transitions are governed by a collection $\mathbb{P} = \{\mathbb{P}_h(\cdot \mid x, a) \mid h \in [H], x \in \mathcal{S}, a \in \mathcal{A}\}$ of transition kernels, where $\mathbb{P}_h(\cdot \mid x, a) \in \Delta(\mathcal{S})$ gives the distribution of states if action a is taken in state x at step h , and $\Delta(\mathcal{S})$ denotes the set of probability distributions on \mathcal{S} . Finally, the rewards are given by $r = \{r_h \mid h \in [H]\}$, where we assume each $r_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is a deterministic reward function.¹

A policy π is a sequence of functions $\{\pi_h \mid h \in [H]\}$ where each $\pi_h : \mathcal{S} \rightarrow \mathcal{A}$ is a mapping from a given state $x \in \mathcal{S}$ to an action $a \in \mathcal{A}$. At the beginning of each episode k , the agent decides on a policy π^k , and is given an initial state $x_1^k \in \mathcal{S}$ (which can be arbitrary). In each step $h \in [H]$ in the episode, the agent picks the action $\pi_h^k(x_h^k)$, receives reward $r_h(x_h^k, \pi_h^k(x_h^k))$, and transitions to a random state x_{h+1}^k determined by $\mathbb{P}_h(\cdot \mid x_h^k, \pi_h^k(x_h^k))$. This continues until the final transition to state x_{H+1}^k , whereupon the agent chooses the policy π^{k+1} for the next episode, and the process is repeated.

2.1 Bellman Equations

For any policy π , we use $V_h^\pi : \mathcal{S} \rightarrow \mathbb{R}$ to denote the value function at step h under policy π , i.e., $V_h^\pi(x)$ gives the expected sum of future rewards under policy π starting from $x_h = x$ in step h until the end of the episode,

$$V_h^\pi(x) := \mathbb{E} \left[\sum_{h'=h}^H r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) \mid x_h = x \right].$$

We refer to $Q_h^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as the Q -value function at step h , where $Q_h^\pi(x, a)$ is equal to the sum of $r_h(x, a)$ and the expected future rewards received for playing policy π in all subsequent steps of the episode after taking action $a_h = a$ at step h from state $x_h = x$,

$$Q_h^\pi(x, a) := r_h(x, a) + \mathbb{E} \left[\sum_{h'=h+1}^H r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) \mid x_h = x, a_h = a \right].$$

¹This assumption is made for ease of presentation, and can be relaxed by incorporating additional UCB terms for the rewards.

Under suitable conditions on $\mathcal{S} \times \mathcal{A}$ and the reward function, there exists an optimal policy π^* which gives the optimal value $V_h^*(x) = \sup_{\pi} V_h^{\pi}(x)$ for all $x \in \mathcal{S}$ and $h \in [H]$. For simplicity and ease of notation we denote $\mathbb{E}[V_{h+1}(\hat{x})|x, a] := \mathbb{E}_{\hat{x} \sim \mathbb{P}_h(\cdot|x, a)}[V_{h+1}(\hat{x})]$ and set $Q^* = Q^{\pi^*}$. We recall the Bellman equations which state that [20]:

$$\begin{cases} V_h^{\pi}(x) = Q_h^{\pi}(x, \pi_h(x)) \\ Q_h^{\pi}(x, a) = r_h(x, a) + \mathbb{E}[V_{h+1}^{\pi}(\hat{x}) | x, a] \\ V_{H+1}^{\pi}(x) = 0 \quad \forall x \in \mathcal{S}. \end{cases} \quad (1)$$

The optimality conditions are similar, where in addition we have $V_h^*(x) = \max_{a \in \mathcal{A}} Q_h^*(x, a)$.

The agent plays the game for K episodes $k = 1, \dots, K$. For each episode k the agent selects a policy π^k and the adversary picks the starting state x_1^k . The goal of the agent is to maximize her total expected reward $\sum_{k=1}^K V_1^{\pi^k}(x_1^k)$. Similar to the benchmarks used in conventional multi-armed bandits, the agent instead attempts to minimize her *regret*, the expected loss the agent experiences by exercising her policy π^k instead of an optimal policy π^* in every episode. This is defined via:

$$R(K) = \sum_{k=1}^K \left(V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k) \right). \quad (2)$$

Our goal is to show that $R(K) \in o(K)$. The regret bounds presented in the paper scale in terms of $K^{(d+1)/(d+2)}$ where d is a type of dimension of the metric space. We begin the next section by outlining the relevant metric space properties.

2.2 Packing and Covering

Covering dimensions and other notions of dimension of a metric space will be a crucial aspect of the regret bound for the algorithm. The ability to adaptively cover the space while minimizing the number of balls required will be a tenant in the adaptive Q -learning algorithm. Following the notation by Kleinberg, Slivkins, and Upfal [12], let (X, \mathcal{D}) be a metric space, and $r > 0$ be arbitrary (in the regret bounds r will be taken as the radius of a ball). We first note that the diameter of a set B is $\text{diam}(B) = \sup_{x, y \in B} \mathcal{D}(x, y)$ and a ball with center x and radius r is denoted by $B(x, r) = \{y \in X : \mathcal{D}(x, y) < r\}$. We denote by $d_{\max} = \text{diam}(X)$ to be the diameter of the entire space.

Definition 2.1. An r -covering of X is a collection of subsets of X , which cover X , and each of which has diameter strictly less than r . The minimal number of subsets in an r -covering is called the r -covering number of \mathcal{P} and is denoted by N_r .

Definition 2.2. A set of points \mathcal{P} is an r -packing if the distance between any points in \mathcal{P} is at least r . An r -Net of the metric space is an r -packing where $\bigcup_{x \in \mathcal{P}} B(x, r)$ covers the entire space X .

As an aside, the Net based Q -learning algorithm requires an ϵ -Net of the state action space $\mathcal{S} \times \mathcal{A}$ given as input to the algorithm [25].

The last definition will be used for a more interpretable regret bound. It is also used to bound the size of the adaptive partition generated by the adaptive Q -learning algorithm. This is used as a dimension of general metric spaces.

Definition 2.3. The *covering dimension* with parameter c induced by the packing numbers N_r is defined as

$$d_c = \inf\{d \geq 0 \mid N_r \leq cr^{-d} \quad \forall r \in (0, d_{\max}]\}.$$

For any set of finite diameter, the covering dimension is at most the doubling dimension, which is at most d for any set in (\mathbb{R}^d, ℓ_p) . However, there are some spaces and metrics where the covering dimensions can be much smaller than the dimension of the entire space [12].

All of these notions of covering are highly related, in fact there are even more definitions of dimensions (including the doubling dimension) through which the regret bound can be formulated (see Section 3 in [12]).

2.3 Assumptions

In this section we state and explain the assumptions used throughout the rest of the paper. We assume that there exists a metric $\mathcal{D} : (\mathcal{S} \times \mathcal{A})^2 \rightarrow \mathbb{R}_+$ so that $\mathcal{S} \times \mathcal{A}$ is a metric space². To make the problem tractable we consider several assumptions which are common throughout the literature.

ASSUMPTION 1. $\mathcal{S} \times \mathcal{A}$ has finite diameter with respect to the metric \mathcal{D} , namely that

$$\text{diam}(\mathcal{S} \times \mathcal{A}) \leq d_{\max}.$$

This assumption allows us to maintain a partition of $\mathcal{S} \times \mathcal{A}$. Indeed, for any point (x, a) in $\mathcal{S} \times \mathcal{A}$, the ball centered at (x, a) with radius d_{\max} covers all of $\mathcal{S} \times \mathcal{A}$. This is also assumed in other research on reinforcement learning in metric spaces where they set $d_{\max} = 1$ by re-scaling the metric.

ASSUMPTION 2. For every $h \in [H]$, Q_h^* is L -Lipschitz continuous with respect to \mathcal{D} , i.e. for all $(x, a), (x', a') \in \mathcal{S} \times \mathcal{A}$,

$$|Q_h^*(x, a) - Q_h^*(x', a')| \leq L\mathcal{D}((x, a), (x', a')).$$

Assumption 2 implies that the Q_h^* value of nearby state action pairs are close. This motivates the discretization technique as points nearby will have similar Q_h^* values and hence can be estimated together. Requiring the Q_h^* function to be Lipschitz may seem less interpretable compared to making assumptions on the problem primitives; however, we demonstrate below that natural continuity assumptions on the MDP translate into this condition (cf. Appendix C for details).

PROPOSITION 2.4. Suppose that the transition kernel is Lipschitz with respect to total variation distance and the reward function is Lipschitz continuous, i.e.

$$\begin{aligned} \|\mathbb{P}_h(\cdot \mid x, a) - \mathbb{P}_h(\cdot \mid x', a')\|_{TV} &\leq L_1\mathcal{D}((x, a), (x', a')) \text{ and} \\ |r_h(x, a) - r_h(x', a')| &\leq L_2\mathcal{D}((x, a), (x', a')) \end{aligned}$$

for all $(x, a), (x', a') \in \mathcal{S} \times \mathcal{A}$ and h . Then it follows that Q_h^* is also $(2L_1H + L_2)$ Lipschitz continuous.

This gives conditions without additional assumptions on the space $\mathcal{S} \times \mathcal{A}$. One downside, however, is for deterministic systems. Indeed, if the transitions in the MDP were deterministic then the transition kernels \mathbb{P}_h would be point masses. Thus, their total variation distance will be either 0 or 1 and will not necessarily be Lipschitz.

Another setting which gives rise to a Lipschitz Q_h^* function (including deterministic transitions) is seen when \mathcal{S} is in addition a compact separable metric space with metric d_S and the metric on $\mathcal{S} \times \mathcal{A}$ satisfies $\mathcal{D}((x, a), (x', a)) \leq Cd_S(x, x')$ for some constant C and for all $a \in \mathcal{A}$ and $x, x' \in \mathcal{S}$. This holds for several common metrics on product spaces. If \mathcal{A} is also a metric space with metric

² $\mathcal{S} \times \mathcal{A}$ can also be a product metric space, where \mathcal{S} and \mathcal{A} are metric spaces individually and the metric on $\mathcal{S} \times \mathcal{A}$ is a product metric.

$d_{\mathcal{A}}$ then common choices for the product metric,

$$\begin{aligned}\mathcal{D}((x, a), (x', a')) &= d_{\mathcal{S}}(x, x') + d_{\mathcal{A}}(a, a') \\ \mathcal{D}((x, a), (x', a')) &= \max\{d_{\mathcal{S}}(x, x'), d_{\mathcal{A}}(a, a')\} \\ \mathcal{D}((x, a), (x', a')) &= \|(d_{\mathcal{S}}(x, x'), d_{\mathcal{A}}(a, a'))\|_p\end{aligned}$$

all satisfy the property with constant $C = 1$. With this we can show the following.

PROPOSITION 2.5. *Suppose that the transition kernel is Lipschitz with respect to the Wasserstein metric and the reward function is Lipschitz continuous, i.e.*

$$\begin{aligned}|r_h(x, a) - r_h(x', a')| &\leq L_1 D((x, a), (x', a')) \\ d_W(\mathbb{P}_h(\cdot \mid x, a), \mathbb{P}_h(\cdot \mid x', a')) &\leq L_2 D((x, a), (x', a'))\end{aligned}$$

for all $(x, a), (x', a') \in \mathcal{S} \times \mathcal{A}$ and h and where d_W is the Wasserstein metric. Then Q_h^\star and V_h^\star are both $(\sum_{i=0}^{H-h} L_1 L_2^i)$ Lipschitz continuous.

Because \mathcal{S} is assumed to be a metric space as well, it follows that V_h^\star is Lipschitz continuous in addition to Q_h^\star . We also note that the Wasserstein metric is always upper-bounded by the total variation distance, and so Lipschitz with respect to total variation implies Lipschitz with respect to the Wasserstein metric. Moreover, this allows Assumption 2 to hold for deterministic MDPs with Lipschitz transitions. Indeed, if $g_h(x, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ denotes the deterministic transition from taking action a in state x at step h (so that $\mathbb{P}_h(x' \mid x, a) = \mathbb{1}_{[x'=x]}$) then using properties of the Wasserstein metric we see the following [9].

$$\begin{aligned}d_W(\mathbb{P}_h(\cdot \mid x, a), \mathbb{P}_h(\cdot \mid x', a')) &= \sup \left\{ \left| \int f d\mathbb{P}_h(\cdot \mid x, a) - \int f d\mathbb{P}_h(\cdot \mid x', a') \right| : \|f\|_L \leq 1 \right\} \\ &= \sup \{ |f(g_h(x, a)) - f(g_h(x', a'))| : \|f\|_L \leq 1 \} \\ &\leq d_{\mathcal{S}}(g_h(x, a), g_h(x', a')) \leq L D((x, a), (x', a'))\end{aligned}$$

where $\|f\|_L$ is the smallest Lipschitz condition number of the function f and we used the Lipschitz assumption of the deterministic transitions $g_h(x, a)$.

The next assumption is similar to that expressed in the literature on adaptive zooming algorithms for multi-armed bandits [12, 23]. This assumes unrestricted access to the similarity metric \mathcal{D} . The question of representing the metric, learning the metric, and picking a metric are important in practice, but beyond the scope of this paper [28]. We will assume *oracle access* to the similarity metric via specific queries.

ASSUMPTION 3. *The agent has oracle access to the similarity metric \mathcal{D} via several queries that are used by the algorithm.*

The Adaptive Q -Learning algorithm presented (Algorithm 1) requires only a *covering oracle* which takes a finite collection of balls and a set X and either declares that they cover X or outputs an uncovered point. The algorithm then poses at most one oracle call in each round. An alternative assumption is to assume the *covering oracle* is able to take a set X and value r and output an r packing of X . In practice, this can be implemented in several metric spaces (e.g. Euclidean spaces). Alternative approaches using arbitrary partitioning schemes (e.g. decision trees, etc) are presented in Section 6. Implementation details of the algorithm in this setting will be explained in Section 7.

Algorithm 1 Adaptive Q -Learning

```

1: procedure ADAPTIVE  $Q$ -LEARNING( $\mathcal{S}, \mathcal{A}, \mathcal{D}, H, K, \delta$ )
2:   Initiate  $H$  partitions  $\mathcal{P}_h^1$  for  $h = 1, \dots, H$  each containing a single ball with radius  $d_{max}$  and
    $Q_h^1$  estimate  $H$ 
3:   for each episode  $k \leftarrow 1, \dots, K$  do
4:     Receive initial state  $x_1^k$ 
5:     for each step  $h \leftarrow 1, \dots, H$  do
6:       Select the ball  $B_{sel}$  by the selection rule  $B_{sel} = \operatorname{argmax}_{B \in \text{RELEVANT}_h^k(x_h^k)} Q_h^k(B)$ 
7:       Select action  $a_h^k = a$  for some  $(x_h^k, a) \in \operatorname{dom}(B_{sel})$ 
8:       Play action  $a_h^k$ , receive reward  $r_h^k$  and transition to new state  $x_{h+1}^k$ 
9:       Update Parameters:  $t = n_h^{k+1}(B_{sel}) \leftarrow n_h^k(B_{sel}) + 1$ 
10:
11:        $Q_h^{k+1}(B_{sel}) \leftarrow (1 - \alpha_t)Q_h^k(B_{sel}) + \alpha_t(r_h^k + b(t) + V_{h+1}^k(x_{h+1}^k))$  where
12:
13:        $V_{h+1}^k(x_{h+1}^k) = \min(H, \max_{B \in \text{RELEVANT}_{h+1}^k(x_{h+1}^k)} Q_{h+1}^k(B))$  (see Section 3)
14:       if  $n_h^{k+1}(B_{sel}) \geq \left(\frac{d_{max}}{r(B_{sel})}\right)^2$  then SPLIT BALL( $B_{sel}, h, k$ )
15: procedure SPLIT BALL( $B, h, k$ )
16:   Set  $B_1, \dots, B_n$  to be an  $\frac{1}{2}r(B)$ -packing of  $\operatorname{dom}(B)$ , and add each ball to the partition  $\mathcal{P}_h^{k+1}$ 
   (see Definition 2.2)
17:   Initialize parameters  $Q_h^{k+1}(B_i)$  and  $n_h^{k+1}(B_i)$  for each new ball  $B_i$  to inherent values from
   the parent ball  $B$ 

```

3 ALGORITHM

Our algorithm is parameterized by the number of episodes K and a value $\delta \in (0, 1)$ related to the high-probability regret bound.³ This algorithm falls under “Upper Confidence Bound” algorithms popular in multi-armed bandits [4, 14] as the selection rule is greedy with respect to estimates of the Q_h^* function. For each step $h = 1, \dots, H$ it maintains a collection of balls \mathcal{P}_h^k of $\mathcal{S} \times \mathcal{A}$ which is refined over the course learning for each episode $k \in [K]$. Each element $B \in \mathcal{P}_h^k$ is a ball with radius $r(B)$. Initially, when $k = 1$, there is only one ball in each partition \mathcal{P}_h^1 which has radius d_{max} and contains the entire state-action space by Assumption 1. A sample of the partition resulting from our adaptive discretization of the space $\mathcal{S} = [0, 1]$, $\mathcal{A} = [0, 1]$ can be seen in Figure 1 with the metric $\mathcal{D}((x, a), (x', a')) = \max\{|x - x'|, |a - a'|\}$.

In comparison to the previous literature where the algorithm takes an optimal ϵ -Net as input, our algorithm refines the partition \mathcal{P}_h^k in a data-driven manner. In each iteration, our algorithm performs three steps: select a ball via the *selection rule*, *update parameters*, and *re-partition* the space.

For every episode k and step h the algorithm maintains two tables with size linear with respect to the number of balls in the partition \mathcal{P}_h^k . For any ball $B \in \mathcal{P}_h^k$ we maintain an *upper confidence value* $Q_h^k(B)$ for the true Q_h^* value for points in B and $n_h^k(B)$ for the number of times B or its ancestors have been selected by the algorithm at step h in episodes up to k . This is incremented every time B is played. The counter will be used to construct the bonus term in updating the Q estimates and

³Knowledge of the number of episodes K can be relaxed by allowing the algorithm to proceed in phases via the doubling trick.

also for determining when to split a ball. We set the learning rate as follows:

$$\alpha_t = \frac{H+1}{H+t} \quad (3)$$

These learning rates are based on the the algorithm in [10], and are chosen to satisfy certain conditions, captured via the following lemma (Lemma 4.1 from [10]).

LEMMA 3.1. *Let $\alpha_t^i \triangleq a_i \prod_{j=i+1}^t (1 - \alpha_j)$. Then $\{\alpha_t\}_{i \leq t}$ satisfy:*

- (1) $\sum_{i=1}^t \alpha_t^i = 1$, $\max_{i \in [t]} \alpha_t^i \leq \frac{2H}{t}$ and $\sum_{i=1}^t (\alpha_t^i)^2 \leq \frac{2H}{t}$ for every $t \geq 1$
- (2) $\frac{1}{\sqrt{t}} \leq \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{t}} \leq \frac{2}{\sqrt{t}}$ for every $t \geq 1$
- (3) $\sum_{t=i}^\infty \alpha_t^i = 1 + \frac{1}{H}$ for every $i \geq 1$.

These properties will be important in the proof and we will highlight them as they come up in the proof sketch.

At a high level the algorithm proceeds as follows. In each episode k and step h , a state x_h^k is observed. The algorithm selects an action according to the *selection rule* by picking a relevant ball B in \mathcal{P}_h^k which has maximum upper confidence value $Q_h^k(B)$ and taking an action in that ball. Next, the algorithm updates the estimates for $Q_h^k(B)$ by *updating parameters* and lastly *re-partitions* the state-action space.

In order to define the three steps (selection rule, updating parameters, and re-partitioning) we need to introduce some definitions and notation. Fix an episode k , step h , and ball $B \in \mathcal{P}_h^k$. Let $t = n_h^k(B)$ be the number of times B or its ancestors have been selected by the algorithm at step h in episodes up to the current episode k . The *confidence radius* or bonus of ball B is

$$b(t) = 2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + \frac{4Ld_{\max}}{\sqrt{t}}. \quad (4)$$

The first term in Equation 4 corresponds to the uncertainty in the current estimate of the Q value due to the stochastic nature of the transitions. The second term corresponds to the discretization error by expanding the estimate to all points in the same ball. If in addition the rewards were stochastic, there would be a third term to include the confidence in reward estimates.

The *domain* of a ball B is a subset of B which excludes all balls $B' \in \mathcal{P}_h^k$ of a strictly smaller radius

$$\text{dom}(B) = B \setminus \left(\bigcup_{B' \in \mathcal{P}_h^k: r(B') < r(B)} B' \right).$$

The domain of the balls in \mathcal{P}_h^k will cover the entire space $\mathcal{S} \times \mathcal{A}$ and be used in the algorithm as a partition of the space. A ball B is then *relevant* in episode k step h for a point $x \in \mathcal{S}$ if $(x, a) \in \text{dom}(B)$ for some $a \in \mathcal{A}$. The set of all relevant balls for a point is denoted by $\text{RELEVANT}_h^k(x)$. In each round the algorithm selects one relevant ball B for the current state x_h^k and plays an action a in that ball. After subsequently observing the reward $r_h = r(x_h^k, a)$ we increment $t = n_h^{k+1}(B) = n_h^k(B) + 1$, and perform the Q -learning update according to

$$Q_h^{k+1}(B) = (1 - \alpha_t)Q_h^k(B) + \alpha_t(r_h^k + V_{h+1}^k(x_{\text{new}}) + b(t)) \quad (5)$$

where r_h^k is the observed reward, x_{new} is the state the agent transitions to, and

$$V_{h+1}^k(x) = \min(H, \max_{B \in \text{RELEVANT}_{h+1}^k(x)} Q_{h+1}^k(B)) \quad (6)$$

is our estimate of the expected future reward for being in a given state. Let (x_h^k, a_h^k) be the state action pair observed in episode k step h by the algorithm. Then the three rules are defined as follows

- **selection rule:** Select a relevant ball B for x_h^k with maximal value of $Q_h^k(B)$ (breaking ties arbitrarily). Select any action a to play such that $(x_h^k, a) \in \text{dom}(B)$. This is similar to the greedy “upper confidence algorithms” for multi-armed bandits.
- **update parameters:** Increment $n_h^k(B)$ by 1, and update the $Q_h^k(B)$ value for the selected ball given the observed reward according to Equation 5.
- **re-partition the space:** Let B denote the selected ball and $r(B)$ denote its radius. We split when $n_h^{k+1}(B) \geq (d_{\max}/r(B))^2$. We then cover $\text{dom}(B)$ with new balls B_1, \dots, B_n which form an $\frac{1}{2}r(B)$ -Net of $\text{dom}(B)$. We call B the *parent* of these new balls and each child ball inherits all values from its parent. We then add the new balls B_1, \dots, B_n to \mathcal{P}_h^k to form the partition for the next episode \mathcal{P}_h^{k+1} .⁴

See Algorithm 1 for the pseudocode. The full version of the pseudocode is in Appendix B. As a reference, see Table 1 in the appendix for a list of notation used.

4 PERFORMANCE GUARANTEES

We provide three main forms of performance guarantees: uniform (i.e., *worst-case*) regret bounds with arbitrary starting states, refined metric-specific regret bounds, and sample-complexity guarantees for learning a policy. We close the section with a lower-bound analysis of these results, showing that our results are optimal up to logarithmic factors and a factor of H^2 .

4.1 Worst-Case Regret Guarantees

We provide regret guarantees for Algorithm 1. First recall the definition of the covering number with parameter c as

$$d_c = \inf\{d \geq 0 : N_r \leq cr^{-d} \ \forall r \in (0, d_{\max}]\}. \quad (7)$$

We show the regret scales as follows:

THEOREM 4.1. *For any any sequence of initial states $\{x_1^k \mid k \in [K]\}$, and any $\delta \in (0, 1)$ with probability at least $1 - \delta$ Adaptive Q-learning (Alg 1) achieves regret guarantee:*

$$\begin{aligned} R(K) &\leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} \\ &\quad + \gamma HK^{(d_c+1)/(d_c+2)} \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \\ &= \tilde{O}\left(H^{5/2} K^{(d_c+1)/(d_c+2)}\right) \end{aligned}$$

where d_c is the covering number of $\mathcal{S} \times \mathcal{A}$ with parameter c and the problem dependent constant

$$\gamma = 192c^{1/(d_c+2)} d_{\max}^{-d_c/(d_c+2)}.$$

The regret bound in Theorem 4.1 has three main components. The first term $3H^2$ corresponds to the regret due to actions selected in the first episode and its subsequent impact on future episodes, where we loosely initialized the upper confidence value Q of each state action pair as H . The second term accounts for the stochastic transitions in the MDP from concentration inequalities. The third term is the discretization error, and comes from the error in discretizing the state action space by

⁴Instead of covering the parent ball each time it is “split”, we can instead introduce children balls as needed in a greedy fashion until the parent ball is covered. When B is first split, we create a single new ball with center (x, a) and radius $\frac{1}{2}r(B)$ where x and a are the current state and action performed. At every subsequent time the parent ball B is selected where the current state and action $(\hat{x}, \hat{a}) \in \text{dom}(B)$, then we create a new ball again with center (\hat{x}, \hat{a}) and radius $\frac{1}{2}r(B)$, which removes this set from $\text{dom}(B)$.

the adaptive partition. As the partition is adaptive, this term scales in terms of the covering number of the entire space. Setting $\delta = K^{-1/(d_c+2)}$, we get the regret of Algorithm 1 as

$$\tilde{O}\left(H^{5/2}K^{(d_c+1)/(d_c+2)}\right).$$

This matches the regret bound from prior work when the metric space $\mathcal{S} \times \mathcal{A}$ is taken to be a subset of $[0, 1]^d$, such that the covering dimension is simply d , the dimension of the metric space [25]. For the case of a discrete state-action space with the discrete metric $\mathcal{D}((s, a), (s', a')) = \mathbb{1}_{[s=s', a=a']}$, which has a covering dimension $d_c = 0$, we recover the $\tilde{O}(\sqrt{H^5 K})$ bound from discrete episodic RL [10].

Our experiments in Section 7 shows that the adaptive partitioning saves on time and space complexity in comparison to the fixed ϵ -Net algorithm [25]. Heuristically, our algorithm achieves better regret while reducing the size of the partition. We also see from experiments that the regret seems to scale in terms of the covering properties of the shape of the optimal Q_h^* function instead of the entire space similar to the results on contextual bandits in metric spaces [23].

Previous work on reinforcement learning in metric spaces give lower bounds on episodic reinforcement learning on metric spaces and show that any algorithm must achieve regret where H scales as $H^{3/2}$ and K in terms of $K^{(d_c+1)/(d_c+2)}$. Because our algorithm achieves worst case regret $\tilde{O}(K^{(d_c+1)/(d_c+2)}H^{5/2})$ this matches the lower bound up to polylogarithmic factors and a factor of H [23, 25]. More information on the lower bounds is in Section 4.4.

4.2 Metric-Specific Regret Guarantees

The regret bound formulated in Theorem 4.1 is a covering guarantee similar to prior work on bandits in metric spaces [12, 23]. This bound suffices for metric spaces where the inequality in the definition of the covering dimensions $N_r \leq cr^{-d}$ is tight; a canonical example is when $\mathcal{S} \times \mathcal{A} = [0, 1]^d$ under the Euclidean metric, where the covering dimension scales as $N_r = \frac{1}{r^d}$.

More generally, the guarantee in Theorem 4.1 arises from a more refined bound, wherein we replace the $\gamma K^{(d_c+1)/(d_c+2)}$ factor in the third term of the regret with

$$\inf_{r_0 \in (0, d_{max}]} \left(\frac{Kr_0}{d_{max}} + \sum_{\substack{r=d_{max}2^{-i} \\ r \geq r_0}} N_r \frac{d_{max}}{r} \right).$$

The bound in Theorem 4.1 is obtained by taking $r_0 = \Theta\left(K^{\frac{-1}{d_c+2}}\right)$ inside of the infimum.

This regret bound gives a packing N_r type guarantee. Discussion on the scaling is deferred to Section 4.4.

4.3 Policy-Identification Guarantees

We can also adapt our algorithm to give sample-complexity guarantees on learning a policy of desired quality. For such a guarantee, assuming that the starting states are adversarially chosen is somewhat pessimistic. A more natural framework here is that of probably approximately correct (PAC) guarantees for learning RL policies [29]. Here, we assume that in each episode $k \in [K]$, we have a random initial state $X_1^k \in \mathcal{S}$ drawn from some fixed distribution F_1 , and try to find the minimum number of episodes needed to find an ϵ -optimal policy π with probability at least $1 - \delta$.

Following similar arguments as [10], we can show that

THEOREM 4.2. *For $K = \tilde{O}\left((H^{5/2}/\delta\epsilon)^{d_c+2}\right)$ (where d_c is the covering dimension with parameter c), consider a policy π chosen uniformly at random from π_1, \dots, π_K . Then, for initial state $X \sim F_1$, with*

probability at least $1 - \delta$, the policy π obeys

$$V_1^\star(X) - V_1^\pi(X) \leq \epsilon.$$

Note that in the above guarantee, both X and π are random. The proof is deferred to Appendix D

4.4 Lower Bounds

Existing lower bounds for this problem have been established previously in the discrete tabular setting [10] and in the contextual bandit literature [23].

Jin et al. [10] show the following for discrete spaces.

THEOREM 4.3 (PARAPHRASE OF THEOREM 3 FROM [10]). *For any algorithm, there exists an H -episodic discrete MDP with S states and A actions such that for any K , the algorithm's regret is $\Omega(H^{3/2}\sqrt{SAK})$.*

This shows that our scaling in terms of H is off by a linear factor. As analyzed in [10], we believe that using Bernstein's inequality instead of Hoeffding's inequality to better manage the variance of the $Q_h^k(B)$ estimates of a ball will allow us to recover H^2 instead of $H^{5/2}$.

Existing lower bounds for learning in continuous spaces have been established in the contextual bandit literature. A contextual bandit instance is characterized by a context space \mathcal{S} , action space \mathcal{A} , and reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. The agent interacts in rounds, where in each round the agent observes an arbitrary initial context x , either drawn from a distribution or specified by an adversary, and the agent subsequently picks an action $a \in \mathcal{A}$, and receives reward $r(x, a)$. This is clearly a simplification of an episodic MDP where the number of steps $H = 1$ and the transition kernel $\mathbb{P}_h(\cdot \mid x, a)$ is independent of x and a . Lower bounds presented in [12, 23] show that the scaling in terms of N_r in this general regret bound is optimal.

THEOREM 4.4 (PARAPHRASE OF THEOREM 5.1 FROM [23]). *Let $(\mathcal{S} \times \mathcal{A}, \mathcal{D})$ be an arbitrary metric space satisfying the assumptions in Section 2.3 with $d_{\max} = 1$. Fix an arbitrary number of episodes K and a positive number R such that*

$$R \leq C_0 \inf_{r \in (0, 1)} \left(r_0 K + \sum_{\substack{r=2^{-i} \\ r \geq r_0}} \frac{N_r}{r} \log(K) \right).$$

Then there exists a distribution \mathcal{I} over problem instances such that for any algorithm, it holds that $\mathbb{E}_{\mathcal{I}}(R(K)) \geq \Omega(R/\log(K))$.

This shows that the scaling in terms of K in Section 4.2 is optimal up to logarithmic factors. Plugging in $r_0 = \Theta(K^{-1/(d_c+2)})$ and exhibiting the dependence on c from the definition of the covering dimension (Equation 7) gives that the regret of any algorithm over the distribution of problem instances is at least $\Omega(K^{(d_c+1)/(d_c+2)}c^{1/(d_c+2)})$. This matches the dependence on K and c from Theorem 4.1. We can port this lower bound construction over to reinforcement learning by constructing a problem instance with H bandit problems in sequence. An interesting direction for future work is determining which reinforcement learning problem instances have more suitable structure where we can develop tighter regret bounds.

5 PROOF SKETCH

In this section we give a proof sketch for Theorem 4.1; details are deferred to Appendix E. We start with a map of the proof before giving some details.

Recall that the algorithm proceeds over K episodes, with each episode comprising of H steps. We start by showing that our algorithm is *optimistic* [22], which means that with high probability, the

estimates maintained by the algorithm are an upper bound on their true values. This allows us to write the regret in terms of the error in approximating the value function to the true value function for the policy employed on that episode. Next, using induction we relate the error from a given step in terms of the error from the next step. Unraveling the relationship and using the fact that the value function for the last step is always zero, we write the regret as the sum of the confidence bound terms from Equation 4. We finish by bounding these quantities using properties of the splitting rule from Algorithm 1. Together, this shows the regret bound established in Theorem 4.1.

Before giving some details of the proof, we start with some notation. Let B_h^k and (x_h^k, a_h^k) denote the ball and state-action pair selected by the algorithm in episode k , step h . We also denote $n_h^k = n_h^k(B_h^k)$ as the number of times the ball B_h^k or its ancestors has been previously played by the algorithm. The overall regret is then given by (Equation 2):

$$R(K) = \sum_{k=1}^K \left(V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k) \right).$$

To simplify presentation, we denote $(V_h^*(x_h^k) - V_h^{\pi^k}(x_h^k))$ by $(V_h^* - V_h^{\pi^k})(x_h^k)$.

We start by relating the error in the estimates from step h in terms of the $(h + 1)$ step estimates. The following Lemma establishes this relationship, which shows that with high probability Q_h^k is both an upper bound on Q_h^* , and exceeds Q_h^* by an amount which is bounded as a function of the step $(h + 1)$ estimates. This also shows that our algorithm is *optimistic*.

LEMMA 5.1. *For any ball B , step h and episode k , let $t = n_h^k(B)$, and $k_1 < \dots < k_t$ to be the episodes where B or its ancestors were encountered previously by the algorithm in step h . Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta/2$ the following holds simultaneously for all $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and ball B such that $(x, a) \in \text{dom}(B)$:*

1. $Q_h^k(B) \geq Q_h^*(x, a)$ and $V_h^k(x) \geq V_h^*(x)$
2. $Q_h^k(B) - Q_h^*(x, a) \leq \mathbb{1}_{[t=0]}H + \beta_t + \sum_{i=1}^t \alpha_t^i \left(V_{h+1}^{k_i} - V_{h+1}^* \right)(x_{h+1}^{k_i})$

where, for any $t \in [K]$ and $i \leq t$, we define $\alpha_t^i = \alpha_i \prod_{j=i+1}^t (1 - \alpha_j)$ and $\beta_t = 2 \sum_{i=1}^t \alpha_t^i b(i)$ (where α_j are the learning rates, and $b(\cdot)$ the confidence radius).

In Appendix E, we provide an expanded version of Lemma 5.1 with a detailed proof (Lemma E.7). Below we provide a proof sketch. The main claim in the Lemma follows from first expanding the recursive update for $Q_h^k(B)$ from (5), the property that $\sum_{i=1}^t \alpha_t^i = 1$ from Lemma 3.1, and applying definitions from the Bellman Equation (Equation 1) to show that

$$\begin{aligned} Q_h^k(B) - Q_h^*(x, a) &= \mathbb{1}_{[t=0]}(H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i \left((V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + b(i) \right) \\ &+ \sum_{i=1}^t \alpha_t^i \left(V_{h+1}^*(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^*(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] \right) + \sum_{i=1}^t \alpha_t^i \left(Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a) \right). \end{aligned}$$

The last term $\left| \sum_{i=1}^t \alpha_t^i (Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a)) \right|$ is the error due to the bias induced by discretization, which we bound by $4Ld_{\max}/\sqrt{t}$ using Lipschitzness of the Q -function. The second term is the error due to approximating the future value of a state. By using Azuma-Hoeffding's inequality, we show that with probability at least $1 - \delta/2$, the error due to approximating the future value by the next

state as opposed to computing the expectation is bounded above by

$$\left| \sum_{i=1}^t \alpha_i^i \left(V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] \right) \right| \leq 2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}}.$$

The final inequalities in Lemma 5.1 follow by the definition of $b(i)$, β_t , and substituting the above inequalities to the expression for $Q_h^k(B) - Q_h^\star(x, a)$.

By Lemma 5.1, with high probability, $V_h^k(x) \geq V_h^\star(x)$, such that the terms within the summation of the regret $(V_1^\star(x_1^k) - V_1^{\pi^k}(x_1^k))$ can be upper bounded by $(V_1^k - V_1^{\pi^k})(x_1^k)$ to show that

$$R(K) \leq \sum_{k=1}^K \left((V_1^k - V_1^{\pi^k})(x_1^k) \right).$$

The main inductive relationship is stated in Lemma 5.2 which writes the errors of these step h estimates in terms of the $(h+1)$ step estimates.

LEMMA 5.2. *For any $\delta \in (0, 1)$ if $\beta_t = 2 \sum_{i=1}^t \alpha_i^i b(i)$ then with probability at least $1 - \delta/2$, for all $h \in [H]$,*

$$\sum_{k=1}^K (V_h^k - V_h^{\pi^k})(x_h^k) \leq \sum_{k=1}^K \left(H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k \right) + \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (V_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k)$$

where $\xi_{h+1}^k = \mathbb{E} \left[V_{h+1}^\star(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^\star - V_{h+1}^{\pi^k})(x_{h+1}^k)$.

Expanding this inductive relationship, and using the base case that the value functions at step $H+1$, V_{H+1}^k and $V_{H+1}^{\pi^k}$, are always zero, it follows that

$$\begin{aligned} R(K) &\leq \sum_{h=1}^H \left(1 + \frac{1}{H} \right)^{h-1} \sum_{k=1}^K \left(H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k \right) \\ &\leq 3 \sum_{h=1}^H \sum_{k=1}^K \left(H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k \right). \end{aligned}$$

Clearly, $\sum_{k=1}^K H \mathbb{1}_{[n_h^k=0]} = H$ as $n_h^k = 0$ only for the first episode $k = 1$. For the terms with ξ_{h+1}^k we use a standard martingale analysis with Azuma-Hoeffdings inequality to show in Lemma E.9 of the Appendix that

$$\sum_{h=1}^H \sum_{k=1}^K \xi_{h+1}^k \leq 6\sqrt{2H^3 K \log(4HK/\delta)}.$$

The dominating term in the regret is $\sum_{h=1}^H \sum_{k=1}^K \beta_{n_h^k}$, which captures critical terms in the approximation error of Q . By construction of $b(t)$, it follows that $\beta_t = \Theta(\frac{1}{\sqrt{t}})$ from the second condition in Lemma 3.1. Using the following two lemmas we are able to bound this sum. The first lemma states several invariants maintained by the partitioning scheme.

LEMMA 5.3. *For every $(h, k) \in [H] \times [K]$ the following invariants are maintained:*

- (Covering) The domains of each ball in \mathcal{P}_h^k cover $\mathcal{S} \times \mathcal{A}$.
- (Separation) For any two balls of radius r , their centers are at distance at least r .

The next lemma states that the number of samples in a given ball grows in terms of its radius. This is needed to show a notion of “progress”, where as we get more samples in a ball we further partition it in order to get a more refined estimate for the Q_h^\star value for points in that ball.

LEMMA 5.4. For any $h \in [H]$ and child ball $B \in \mathcal{P}_h^K$ (the partition at the end of the last episode K) the number of episodes $k \leq K$ such that B is selected by the algorithm is less than $\frac{3}{4}(d_{\max}/r)^2$ where $r = r(B)$. I.e. denoting by B_h^k the ball selected by the algorithm in step h episode k ,

$$|\{k : B_h^k = B\}| \leq \frac{3}{4} \left(\frac{d_{\max}}{r} \right)^2.$$

Moreover, the number of times that ball B and it's ancestors have been played is at least $\frac{1}{4} \left(\frac{d_{\max}}{r} \right)^2$.

Using these two lemmas we show that $|\mathcal{P}_h^K| \simeq K^{\frac{d_c}{d_c+2}}$. Hence we get using Jensen's and Cauchy's inequality that:

$$\begin{aligned} \sum_{k=1}^K \beta_{n_h^k} &\simeq \sum_{k=1}^K \frac{1}{\sqrt{n_h^k}} \simeq \sum_{B \in \mathcal{P}_h^K} \sum_{k: B_h^k = B} \frac{1}{\sqrt{n_h^k}} \\ &\simeq \sum_{B \in \mathcal{P}_h^K} \sqrt{|k : B_h^k = B|} \simeq \sqrt{|\mathcal{P}_h^K| K} \\ &\simeq K^{(d_c+1)/(d_c+2)}. \end{aligned}$$

Combining these terms gives the regret bound in Theorem 4.1.

Finally we would like to give some intuition for how the induction relationship presented in Lemma 5.2 follows from the key Lemma 5.1. Recall that the selection rule in the algorithm enforces that the selected ball B_h^k is the one that maximizes the upper confidence Q-values amongst relevant balls. By definition, it follows that for any h and k ,

$$\begin{aligned} (V_h^k - V_h^{\pi^k})(x_h^k) &\leq \max_{B \in \text{RELEVANT}_h^k(x_h^k)} Q_h^k(B) - Q_h^{\pi^k}(x_h^k, a_h^k) \\ &= Q_h^k(B_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k) \\ &= Q_h^k(B_h^k) - Q_h^*(x_h^k, a_h^k) + Q_h^*(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k). \end{aligned}$$

By the definition of the Q-function in Equation 1,

$$Q_h^*(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k) = \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right].$$

We bound $Q_h^k(B_h^k) - Q_h^*(x_h^k, a_h^k)$ by Lemma 5.1, as $(x_h^k, a_h^k) \in \text{dom}(B_h^k)$. Putting these bounds together, for $t = n_h^k(B_h^k)$ and for $k_1 < \dots < k_t$ denoting episodes where B_h^k or its ancestors were previously encountered, it follows that

$$(V_h^k - V_h^{\pi^k})(x_h^k) \leq \mathbb{1}_{[t=0]}H + \beta_t + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*(x_{h+1}^{k_i})) + (V_{h+1}^* - V_{h+1}^{\pi^k})(x_{h+1}^k) + \xi_{h+1}^k,$$

where ξ_{h+1}^k is defined in Lemma 5.2. Let us denote $n_h^k = n_h^k(B_h^k)$, and let the respective episodes $k_i(B_h^k)$ denote the time B_h^k or its ancestors were selected for the i -th time. By comparing the above inequality to the final inductive relationship, the last inequality we need show is that upon summing

over all episodes k ,

$$\begin{aligned} & \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i (\mathbf{V}_{h+1}^{k_i(B_h^k)} - V_{h+1}^\star)(x_{h+1}^{k_i(B_h^k)}) + \sum_{k=1}^K (V_{h+1}^\star - V_{h+1}^{\pi^k})(x_{h+1}^k) \\ & \leq \left(1 + \frac{1}{H}\right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k). \end{aligned}$$

For every $k' \in [K]$ the term $(\mathbf{V}_{h+1}^{k'} - V_{h+1}^\star)(x_{h+1}^{k'})$ appears in the summand when $k = n_h^{k'}$. The next time it appears when $k = n_h^{k'} + 1$ and so on. By rearranging the order of the summation,

$$\begin{aligned} & \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i (\mathbf{V}_{h+1}^{k_i(B_h^k)} - V_{h+1}^\star)(x_{h+1}^{k_i(B_h^k)}) \\ & \leq \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) \sum_{t=n_h^k}^{\infty} \alpha_t^{n_h^k}. \end{aligned}$$

The final recursive relationship results from the property that by construction $\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H}$ for all i from Lemma 3.1, and the inequality $V_{h+1}^{\pi^k}(x_{h+1}^k) \leq V_{h+1}^\star(x_{h+1}^k) \leq \mathbf{V}_{h+1}^k(x_{h+1}^k)$.

6 DISCUSSION AND EXTENSIONS

The partitioning method used in Algorithm 1 was chosen due to its implementability. However, our analysis can be extended to provide a framework for adaptive Q -learning algorithms that flexibly learns partitions of the space in a data-driven manner. This allows practitioners to use their favourite partitioning algorithm (e.g. decision trees, kernel methods, etc) to adaptively partition the space. This begs the question: *Amongst different partitioning algorithms, which ones still guarantee the same regret scaling as Theorem 4.1?*

In particular, we consider black box partitioning schemes that incrementally build a nested partition, determining *when* and *how* to split regions as a function of the observed data. This black box partitioning algorithm is then plugged into the “repartitioning” step of Algorithm 1. Let \mathcal{P}_h^k denote the partition kept by the algorithm for step h in episode k . The algorithm stores estimates $Q_h^k(P)$ for the optimal Q_h^\star value for points in a given region $P \in \mathcal{P}_h^k$ and $n_h^k(P)$ for the number of times P or its ancestors has been selected in step h up to the current episode k . In every episode k , step h of the algorithm, the procedure proceeds identically to that in Algorithm 1 by selecting a region P_h^k which is relevant for the current state with maximal $Q_h^k(P)$ value. The update rules are the same as those defined in (5) where instead of balls we consider regions in the partition. When a region is subpartitioned, all of its children must inherit the values of $Q_h^k(P)$ and $n_h^k(P)$ from its parent. The black box partitioning algorithm only decides *when* and *how* to subpartition the selected region P_h^k .

In Theorem 6.1, we extend the regret guarantees to modifications of Algorithm 1 that maintain the conditions below.

THEOREM 6.1. *For any modification of Algorithm 1 with a black box partitioning scheme that satisfies $\forall h \in [H], \mathcal{P}_h^k$:*

- (1) $\{\mathcal{P}_h^k\}_{k \geq 1}$ is a sequence of nested partitions.

(2) There exists constants c_1 and c_2 such that for every h, k and $P \in \mathcal{P}_h^k$

$$\frac{c_1^2}{\text{diam}(P)^2} \leq n_h^k(P) \leq \frac{c_2^2}{\text{diam}(P)^2}.$$

(3) $|\mathcal{P}_h^K| \leq K^{d_c/(d_c+2)}$.

the achieved regret is bounded by $\tilde{O}(H^{5/2}K^{(d_c+1)/(d_c+2)})$.

Our algorithm clearly satisfies the first condition. The second condition is verified in Lemma E.1, and the third condition follows from a tighter analysis in Lemma E.10. The only dependence on the partitioning method used when proving Theorem 4.1 is through these sets of assumptions, and so the proof follows from a straightforward generalization of the results.

Similar generalizations were provided in the contextual bandit setting by [27], although they only show sublinear regret. The first condition requires that the partition evolves in a hierarchical manner, as could be represented by a tree, where each region of the partition has an associated parent. Due to the fact that a child inherits its Q_h^k estimates from its parent, the recursive update expansion in Lemma 5.1 still holds. The second condition establishes that the number of times a given region is selected grows in terms of the square of its diameter. This can be thought of as a bias variance trade-off, as when the number of samples in a given region is large, the variance term dominates the bias term and it is advantageous to split the partition to obtain more refined estimates. This assumption is necessary for showing Lemma 5.4. It also enforces that the coarseness of the partition depends on the density of the observations. The third condition controls the size of the partition and is used to compute the sum $\sum_{k=1}^K \beta_{n_h^k}$ from the proof sketch in Section 5.

This theorem provides a practical framework for developing adaptive Q -learning algorithms. The first condition is easily satisfied by picking the partition in an online fashion and splitting the selected region into sub-regions. The second condition can be checked at every iteration and determines *when* to sub-partition a region. The third condition limits the practitioner from creating unusually shaped regions. Using these conditions, a practitioner can use any decision tree or partitioning algorithm that satisfies these properties to create an adaptive Q -learning algorithm. An interesting future direction is to understand whether different partitioning methods leads to provable instance-specific gains.

7 EXPERIMENTS

7.1 Experimental Set up

We compare the performance of Q -learning with Adaptive Discretization (Algorithm 1) to Net Based Q -Learning [25] on two canonical problems to illustrate the advantage of adaptive discretization compared to uniform mesh. On both problems the state and action space are taken to be $\mathcal{S} = \mathcal{A} = [0, 1]$ and the metric the product metric, i.e. $\mathcal{D}((x, a), (x', a')) = \max\{|x - x'|, |a - a'|\}$. This choice of metric results in a rectangular partition of $\mathcal{S} \times \mathcal{A} = [0, 1]^2$ allowing for simpler implementation and easy comparison of the partitions used by the algorithms. We focus on problems which have a two-dimensional state-action space to provide a proof of concept of the adaptive discretizations constructed by the algorithm. These examples have more structure than the worst-case bound, but provide intuition on the partition “zooming” into important parts of the space. More simulation results are available with the code on github at <https://github.com/seansrsinclair/AdaptiveQLearning>.

Oil Discovery. This problem, adapted from [16], is a continuous variant of the popular ‘Grid World’ game. It comprises of an agent (or agents) surveying a 1D map in search of hidden “oil deposits”. The world is endowed with an unknown survey function $f(x)$ which encodes the oil deposit at each location x . For agents to move to a new location they must pay a cost proportional

to the distance moved; moreover, surveying the land produces noisy estimates of the true value of that location.

We consider an MDP where $\mathcal{S} = \mathcal{A} = [0, 1]$ represent the current and future locations of the agent. The time-invariant transition kernel (i.e., homogeneous for all $h \in [H]$) is defined via $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$, signifying that the agent moves to their chosen location. Rewards are given by $r_h(x, a) = \max\{0, f(a) + \epsilon - |x - a|\}$, where ϵ is independent sub-Gaussian noise and $f(a) \in [0, 1]$ is the survey value of the location a (the max ensures rewards are in $[0, 1]$).

We choose the survey function $f(x)$ to be either $f(x) = e^{-\lambda|x-c|}$ or $f(x) = 1 - \lambda(x - c)^2$ where $c \in [0, 1]$ is the location of the oil well and λ is a smoothing parameter, which can be tuned to adjust the Lipschitz constant.

Ambulance Relocation. This is a widely-studied stochastic variant of the above problem [3]. Consider an ambulance navigating an environment and trying to position itself in a region with high predicted service requests. The agent interacts with the environment by first choosing a location to station the ambulance, paying a cost to travel to that location. Next, an ambulance request is realized, drawn from a fixed distribution, after which the ambulance must travel to meet the demand at the random location.

Formally, we consider an MDP with $\mathcal{S} = \mathcal{A} = [0, 1]$ encoding the current and future locations of the ambulance. The transition kernels are defined via $\mathbb{P}_h(x'|x, a) \sim \mathcal{F}_h$, where \mathcal{F}_h denotes the request distribution for time step h . The reward is $r_h(x'|x, a) = 1 - [c|x - a| + (1 - c)|x' - a|]$; here, $c \in [0, 1]$ models the trade-offs between the cost of relocation (often is less expensive) and cost of traveling to meet the demand.

For the arrival distribution \mathcal{F}_h we consider Beta(5, 2) and Uniform(0, 1) to illustrate dispersed and concentrated request distributions. We also analyzed the effect of changing the arrival distribution over time (e.g. Figure 1). We compare the RL methods to two heuristics: “No Movement”, where the ambulance pays the cost of traveling to the request, but does not relocate after that, and “Median”, where the ambulance always relocates to the median of the observed requests. Each heuristic is near-optimal respectively at the two extreme values of c .

7.2 Adaptive Tree Implementation

We used a tree data structure to implement the partition \mathcal{P}_h^k of Algorithm 1. We maintain a tree for every step $h \in [H]$ to signify our partition of $\mathcal{S} \times \mathcal{A} = [0, 1]^2$. Each node in the tree corresponds to a rectangle of the partition, and contains algorithmic information such as the estimate of the Q_h^k value at the node. Each node has an associated center (x, a) and radius r . We store a list of (possibly four) children for covering the region which arises when *splitting* a ball.

To implement the *selection rule* we used a recursive algorithm which traverses through all the nodes in the tree, checks each node if the given state x_h^k is contained in the node, and if so recursively checks the children to obtain the maximum Q_h^k value. This speeds up computation by following the tree structure instead of linear traversal. For additional savings we implemented this using a max-heap.

For Net Based Q -Learning, based on the recommendation in [25], we used a fixed ϵ -Net of the state-action space, with $\epsilon = (KH)^{-1/4}$ (since $d = 2$). An example of the discretization can be seen in Figure 1 where each point in the discretization is the center of a rectangle.

7.3 Experimental Results

Oil Discovery. First we consider the oil problem with survey function $f(a) = e^{-\lambda|a-c|}$ where $\lambda = 1$. This results in a sharp reward function. Heuristically, the optimal policy can be seen to take the first step to travel to the maximum reward at c and then for each subsequent step stay

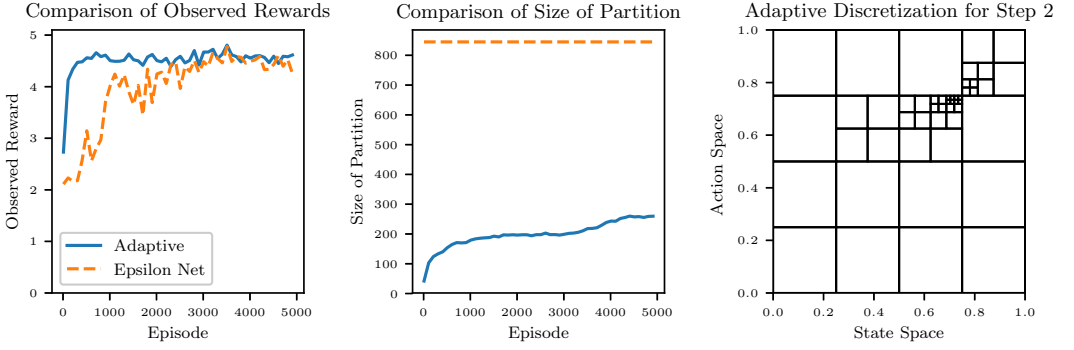


Fig. 2. Comparison of the observed rewards, size of partition, and discretization for the uniform mesh [25] and our adaptive discretization algorithms, on the oil discovery problem with survey function $f(x) = e^{-|x-0.75|}$. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{I}_{[x'=a]}$ and reward function is $r(x, a) = (1 - e^{-|a-0.75|} - |x - a|)_+$ (see Section 7.1). The adaptive algorithm quickly learns the location of the optimal point 0.75 and creates a fine partition of the space around the optimal.

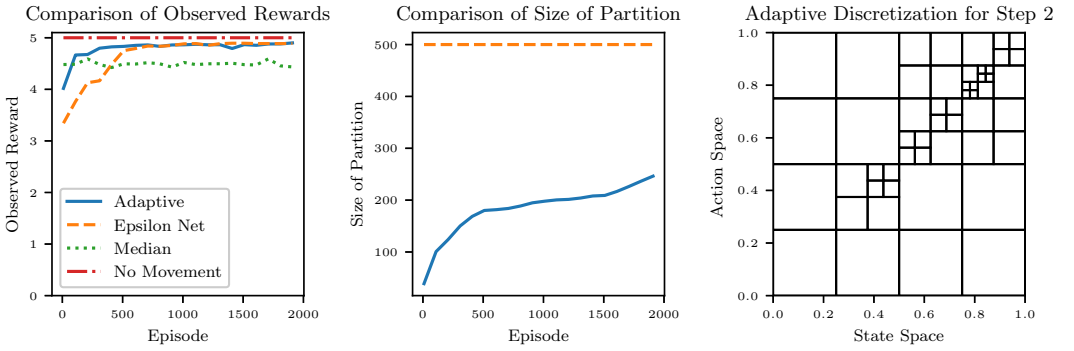


Fig. 3. Comparison of the algorithms on the ambulance problem with Beta(5, 2) arrival distribution and reward function $r(x, a) = 1 - |x - a|$ (see Section 7.1). Clearly, the no movement heuristic is the optimal policy but the adaptive algorithm learns a fine partition across the diagonal of the space where the optimal policy lies.

at the current location. For the experiments we took the number of steps H to be five and tuned the scaling of the confidence bounds for the two algorithms separately. In Figure 2 we see that the adaptive Q -learning algorithm is able to ascertain the optimal policy in fewer iterations than the epsilon net algorithm. Moreover, due to adaptively partitioning the space instead of fixing a discretization before running the algorithm, the size of the partition in the adaptive algorithm is drastically reduced in comparison to the epsilon net. Running the experiment on the survey function $f(a) = 1 - \lambda(x - c)^2$ gave similar results and the graph can be seen in Appendix F. We note that for both experiments as we increase λ , causing the survey distribution to become peaky and have a higher Lipschitz constant, the Net based Q -learning algorithm suffered from a larger discretization error. This is due to that fact that the tuning of the ϵ parameter ignores the Lipschitz constant. However, the adaptive Q -learning algorithm is able to take advantage and narrow in on the point with large reward.

Ambulance Routing. We consider the ambulance routing problem with arrival distribution $\mathcal{F} = \text{Beta}(5, 2)$. The time horizon H is again taken to be five. We implemented two additional heuristics for the problem to serve as benchmarks. The first is the “No Movement” heuristic. This algorithm takes the action to never move, and pays the entire cost of traveling to service the arrival. The second, “Median” heuristic takes the action to travel to the estimated median of the distribution based on all past arrivals. For the case when $c = 1$ no movement is the optimal algorithm. For $c = 0$ the optimal will be to travel to the median.

After running the algorithms with $c = 1$ the adaptive Q -learning algorithm is better able to learn the stationary policy of not moving from the current location, as there is only a cost of moving to the action. The rewards observed by each algorithm is seen in Figure 3. In this case, the discretization (for step $h = 2$) shows that the adaptive Q -learning algorithm maintains a finer partition across the diagonal where the optimal policy lives. Running the algorithm for $c < 1$ shows that the adaptive Q -learning algorithm has a finer discretization around $(x, a) = (0.7, 0.7)$, where 0.7 is the approximate median of a $\text{Beta}(5, 2)$ distribution. The algorithm keeps a fine discretization both around where the algorithm frequently visits, but also places of high reward. For $c < 1$ and the arrival distribution $\mathcal{F} = \text{Uniform}[0, 1]$ the results were similar and the graphs are in Appendix F.

The last experiment was to analyze the algorithms when the arrival distribution changes over time (e.g. over steps h). In Figure 1 we took $c = 0$ and shifting arrival distributions \mathcal{F}_h where $\mathcal{F}_1 = \text{Uniform}(0, 1/4)$, $\mathcal{F}_2 = \text{Uniform}(1/4, 1/2)$, $\mathcal{F}_3 = \text{Uniform}(1/2, 3/4)$, $\mathcal{F}_4 = \text{Uniform}(3/4, 1)$, and $\mathcal{F}_5 = \text{Uniform}(1/2 - 0.05, 1/2 + 0.05)$. In the figure, the color corresponds to the Q_h^* value of that specific state-action pair, where green corresponds to a larger value for the expected future rewards. The adaptive algorithm was able to converge faster to the optimal policy than the uniform mesh algorithm. Moreover, the discretization observed from the Adaptive Q -Learning algorithm follows the contours of the Q -function over the space. This shows the intuition behind the algorithm of storing a fine partition across near-optimal parts of the space, and a coarse partition across sub-optimal parts.

8 CONCLUSION

We presented an algorithm for model-free episodic reinforcement learning on continuous state action spaces that uses data-driven discretization to adapt to the shape of the optimal policy. Under the assumption that the optimal Q^* function is Lipschitz, the algorithm achieves regret bounds scaling in terms of the covering dimension of the metric space.

Future directions include relaxing the requirements on the metric, such as considering weaker versions of the Lipschitz condition. In settings where the metric may not be known a priori, it would be meaningful to be able to estimate the distance metric from data over the course of executing of the algorithm. Lastly, we hope to characterize problems where adaptive discretization outperforms uniform mesh.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from the NSF under grants ECCS-1847393 and DMS-1839346, and the ARL under grant W911NF-17-1-0094.

REFERENCES

- [1] Peter Auer, Thomas Jaksch, and Ronald Ortner. 2009. Near-optimal Regret Bounds for Reinforcement Learning. In *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.). Curran Associates, Inc., 89–96. <http://papers.nips.cc/paper/3401-near-optimal-regret-bounds-for-reinforcement-learning.pdf>
- [2] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. 2017. Minimax Regret Bounds for Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML '17)*. JMLR.org, 263–272. <http://dl.acm.org/citation.cfm?id=3305381.3305409>
- [3] Luce Brotcorne, Gilbert Laporte, and Frederic Semet. 2003. Ambulance location and relocation models. *European journal of operational research* 147, 3 (2003), 451–463.
- [4] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122.
- [5] Sébastien Bubeck, Gilles Stoltz, Csaba Szepesvári, and Rémi Munos. 2009. Online optimization in X-armed bandits. In *Advances in Neural Information Processing Systems*. 201–208.
- [6] Joshua Comden, Sijie Yao, Niangjun Chen, Haipeng Xing, and Zhenhua Liu. 2019. Online Optimization in Cloud Resource Provisioning: Predictions, Regrets, and Algorithms. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 1, Article 16 (March 2019), 30 pages. <https://doi.org/10.1145/3322205.3311087>
- [7] Kefan Dong, Yuanhao Wang, Xiaoyu Chen, and Liwei Wang. 2019. Q-learning with UCB Exploration is Sample Efficient for Infinite-Horizon MDP. *arXiv preprint arXiv:1901.09311* (2019).
- [8] Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. 2019. Provably Efficient Q-learning with Function Approximation via Distribution Shift Error Checking Oracle. *arXiv preprint arXiv:1906.06321* (2019).
- [9] Alison L Gibbs and Francis Edward Su. 2002. On choosing and bounding probability metrics. *International statistical review* 70, 3 (2002), 419–435.
- [10] Jin C, Jordan M.I, Allen-Zhu Z, Bubeck S, and NeurIPS 2018 32nd Conference on Neural Information Processing Systems. 2018. Is Q-learning provably efficient? *Adv. neural inf. proces. syst. Advances in Neural Information Processing Systems* 2018-December (2018), 4863–4873. OCLC: 8096900528.
- [11] Sham Kakade, Michael Kearns, and John Langford. 2003. Exploration in Metric State Spaces. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML '03)*. AAAI Press, 306–312. <http://dl.acm.org/citation.cfm?id=3041838.3041877>
- [12] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. 2019. Bandits and Experts in Metric Spaces. *J. ACM* 66, 4, Article 30 (May 2019), 77 pages. <https://doi.org/10.1145/3299873>
- [13] K. Lakshmanan, Ronald Ortner, and Daniil Ryabko. 2015. Improved Regret Bounds for Undiscounted Continuous Reinforcement Learning. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 524–532. <http://proceedings.mlr.press/v37/lakshmanan15.html>
- [14] Tor Lattimore and Csaba Szepesvári. 2018. Bandit algorithms. *preprint* (2018).
- [15] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. 2016. Resource Management with Deep Reinforcement Learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets '16)*. ACM, New York, NY, USA, 50–56. <https://doi.org/10.1145/3005745.3005750>
- [16] Winter Mason and Duncan J Watts. 2012. Collaborative learning in networks. *Proceedings of the National Academy of Sciences* 109, 3 (2012), 764–769.
- [17] Ronald Ortner. 2013. Adaptive aggregation for reinforcement learning in average reward Markov decision processes. *Annals of Operations Research* 208, 1 (01 Sep 2013), 321–336. <https://doi.org/10.1007/s10479-012-1064-y>
- [18] Ronald Ortner and Daniil Ryabko. 2012. Online Regret Bounds for Undiscounted Continuous Reinforcement Learning. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1763–1771. <http://papers.nips.cc/paper/4666-online-regret-bounds-for-undiscounted-continuous-reinforcement-learning.pdf>
- [19] Ian Osband and Benjamin Van Roy. 2014. Model-based Reinforcement Learning and the Eluder Dimension. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1466–1474. <http://papers.nips.cc/paper/5245-model-based-reinforcement-learning-and-the-eluder-dimension.pdf>
- [20] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., New York, NY, USA.
- [21] Devavrat Shah and Qiaomin Xie. 2018. Q-learning with Nearest Neighbors. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 3111–3121. <http://papers.nips.cc/paper/7574-q-learning-with-nearest-neighbors.pdf>
- [22] Max Simchowitz and Kevin Jamieson. 2019. Non-Asymptotic Gap-Dependent Regret Bounds for Tabular MDPs. [arXiv:cs.LG/1905.03814](https://arxiv.org/abs/1905.03814)

- [23] Aleksandrs Slivkins. 2015. Contextual Bandits with Similarity Information. *Journal of machine learning research : JMLR*. 15, 2 (2015), 2533–2568. OCLC: 5973068319.
- [24] Aleksandrs Slivkins. 2019. Introduction to Multi-Armed Bandits. arXiv:cs.LG/1904.07272
- [25] Zhao Song and Wen Sun. 2019. Efficient Model-free Reinforcement Learning in Metric Spaces. *arXiv:1905.00475 [cs, stat]* (May 2019). <http://arxiv.org/abs/1905.00475> arXiv: 1905.00475.
- [26] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [27] Tianyu Wang, Weicheng Ye, Dawei Geng, and Cynthia Rudin. 2019. Towards Practical Lipschitz Stochastic Bandits. *arXiv e-prints*, Article arXiv:1901.09277 (Jan 2019), arXiv:1901.09277 pages. arXiv:stat.ML/1901.09277
- [28] Nirandika Wanigasekara and Christina Lee Yu. 2019. Nonparametric Contextual Bandits in an Unknown Metric Space. arXiv:cs.LG/1908.01228
- [29] Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards. (1989).
- [30] Lin Yang and Mengdi Wang. 2019. Sample-Optimal Parametric Q-Learning Using Linearly Additive Features. In *International Conference on Machine Learning*. 6995–7004.
- [31] Lin F Yang, Chengzhuo Ni, and Mengdi Wang. 2019. Learning to Control in Metric Space with Optimal Regret. *arXiv preprint arXiv:1905.01576* (2019).
- [32] Lin F. Yang and Mengdi Wang. 2019. Reinforcement Learning in Feature Space: Matrix Bandit, Kernels, and Regret Bound. *arXiv:1905.10389 [cs, stat]* (May 2019). <http://arxiv.org/abs/1905.10389> arXiv: 1905.10389.

A NOTATION TABLE

Symbol	Definition
$\mathcal{S}, \mathcal{A}, H$	MDP specifications (state space, action space, steps per episode)
$r_h(x, a), \mathbb{P}_h(\cdot \mid x, a)$	Reward/transition kernel for taking action a in state x at step h
K	Number of episodes
π_h^\star	Optimal policy in step h
(x_h^k, a_h^k)	State and action executed by the algorithm at step h in episode k
$\Delta(\mathcal{S})$	Set of probability measures on \mathcal{S}
$V_h^\pi(\cdot), Q_h^\pi(\cdot, \cdot)$	Value/Q-function at step h under policy π
$V_h^\star(\cdot), Q_h^\star(\cdot, \cdot)$	Value/Q-function for step h under the optimal policy
L	Lipschitz constant for Q^\star
\mathcal{D}	Metric on $\mathcal{S} \times \mathcal{A}$
d_{max}	Bound on $\mathcal{S} \times \mathcal{A}$ using the metric \mathcal{D}
N_r	r covering number of $\mathcal{S} \times \mathcal{A}$
d_c	The covering dimension of $\mathcal{S} \times \mathcal{A}$ with parameter c
\mathcal{P}_h^k	Partition of $\mathcal{S} \times \mathcal{A}$ for step h in episode k
B_h^k	Ball selected by algorithm in step h episode k
$Q_h^k(B)$	Estimate of the Q value for points in B in step h episode k
$V_h^k(x)$	Estimate of the value function for x in step h episode k
$n_h^k(B)$	Number of times B or its ancestors has been chosen before episode k in step h
α_t	The adaptive learning rate $\frac{H+1}{H+t}$ (cf. Alg 1)
α_t^i	$\alpha_i \prod_{j=i+1}^t (1 - \alpha_j)$
$\text{diam}(B)$	The diameter of a set B
$r(B)$	The radius of a ball B
$b(t)$	The upper confidence term, $2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + \frac{4LD_{max}}{\sqrt{t}}$
$R(K)$	The regret up to episode K
$\text{dom}(B)$	The domain of a ball B , excludes points in a ball with a smaller radius
$\text{RELEVANT}_h^k(x)$	The set of all balls which are relevant for x in step h episode k
$\mathbb{E}[V_{h+1}(\hat{x}) \mid x, a]$	$\mathbb{E}_{\hat{x} \sim \mathbb{P}_h(\cdot \mid x, a)}[V_{h+1}(\hat{x})]$

Table 1. List of common notation

B COMPLETE PSEUDOCODE FOR ADAPTIVE Q -LEARNING ALGORITHM

Algorithm 2 Adaptive Q -Learning

```

1: procedure ADAPTIVE  $Q$ -LEARNING( $\mathcal{S}, \mathcal{A}, \mathcal{D}, H, K, \delta$ )
2:   for  $h \leftarrow 1, \dots, H$  do
3:      $\mathcal{P}_h^1$  contains a single ball  $B$  with radius  $d_{max}$ 
4:      $Q_h^1(B) \leftarrow H$ 
5:     for each episode  $k \leftarrow 1, \dots, K$  do
6:       Receive initial state  $x_1^k$ 
7:       for each step  $h \leftarrow 1, \dots, H$  do
8:         Selection Rule:
9:         Set  $\text{RELEVANT}_h^k(x_h^k) = \{B \in \mathcal{P}_h^k \mid \exists a \in \mathcal{A} \text{ with } (x_h^k, a) \in \text{dom}(B)\}$ 
10:        Select the ball  $B_{sel}$  by the selection rule  $B_{sel} = \arg\max_{B \in \text{RELEVANT}_h^k(x_h^k)} Q_h^k(B)$ 
11:        Select action  $a_h^k = a$  for some  $(x_h^k, a) \in \text{dom}(B_{sel})$ 
12:        Play action  $a_h^k$ , receive reward  $r_h^k$ , and new state  $x_{h+1}^k$ 
13:        Update Parameters:
14:         $n_h^{k+1}(B_{sel}) \leftarrow n_h^k(B_{sel}) + 1$ 
15:         $t \leftarrow n_h^{k+1}(B_{sel})$ 
16:         $b(t) \leftarrow 4\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + \frac{2Ld_{max}}{\sqrt{t}}$ 
17:         $Q_h^{k+1}(B_{sel}) \leftarrow (1 - \alpha_t)Q_h^k(B_{sel}) + \alpha_t(r_h^k + V_{h+1}^k(x_{h+1}^k) + b(t))$  where
18:         $V_h^k(x_h^k) = \min(H, \max_{B \in \text{RELEVANT}_h^k(x_h^k)} Q_h^k(B))$ 
19:        if  $t \geq \left(\frac{d_{max}}{r(B_{sel})}\right)^2$  then
20:          SPLIT BALL( $B_{sel}, h, k$ )
21:           $\mathcal{P}_h^{k+1} \leftarrow \mathcal{P}_h^k$ 
22: procedure SPLIT BALL( $B, h, k$ )
23:   Set  $B_1, \dots, B_n$  to be a  $\frac{1}{2}r(B)$ -Net of  $\text{dom}(B)$ 
24:   for each ball  $B_i$  do
25:      $Q_h^{k+1}(B_i) \leftarrow Q_h^{k+1}(B)$ 
26:
27:    $n_h^{k+1}(B_i) \leftarrow n_h^{k+1}(B)$ 
28:
29:   Add  $B_i$  to  $\mathcal{P}_h^{k+1}$ 

```

C PROOFS FOR LIPSCHITZ Q^* FUNCTION

PROPOSITION C.1. *Suppose that the transition kernel and reward function are Lipschitz continuous with respect to \mathcal{D} , i.e.*

$$\begin{aligned} \|\mathbb{P}_h(\cdot \mid x, a) - \mathbb{P}_h(\cdot \mid x', a')\|_{TV} &\leq L_1 \mathcal{D}((x, a), (x', a')) \text{ and} \\ |r_h(x, a) - r_h(x', a')| &\leq L_2 \mathcal{D}((x, a), (x', a')) \end{aligned}$$

for all $(x, a), (x', a') \in \mathcal{S} \times \mathcal{A}$ and h . Then Q_h^ is also $(2L_1H + L_2)$ Lipschitz continuous.*

PROOF. We use the definition of Q_h^\star . Indeed, for any $h \in [H]$,

$$|Q_h^\star(x, a) - Q_h^\star(x', a')| = |r_h(x, a) - r_h(x', a') + \mathbb{E}[V_{h+1}^\star(\hat{x}) \mid x, a] - \mathbb{E}[V_{h+1}^\star(\hat{x}) \mid x', a']|.$$

However, noting that both $\mathbb{P}_h(\cdot \mid x, a)$ and $\mathbb{P}_h(\cdot \mid x', a')$ are absolutely continuous with respect to the base measure $\lambda = \frac{1}{2}(\mathbb{P}_h(\cdot \mid x, a) + \mathbb{P}_h(\cdot \mid x', a'))$ we are able to deconstruct the difference in expectations with respect to the Radon-Nikodym derivatives as the measures are σ -finite. Hence,

$$\begin{aligned} &\leq |r_h(x, a) - r_h(x', a')| + \left| \int V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} \mid x, a) - \int V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} \mid x', a') \right| \\ &\leq L_2 \mathcal{D}((x, a), (x', a')) + \left| \int V_{h+1}^\star(\hat{x}) \frac{d\mathbb{P}_h(\hat{x} \mid x, a)}{d\lambda(\hat{x})} d\lambda(\hat{x}) - \int V_{h+1}^\star(\hat{x}) \frac{d\mathbb{P}_h(\hat{x} \mid x', a')}{d\lambda(\hat{x})} d\lambda(\hat{x}) \right| \\ &\leq L_2 \mathcal{D}((x, a), (x', a')) + \int |V_{h+1}^\star(\hat{x})| \left| \frac{d\mathbb{P}_h(\hat{x} \mid x, a)}{d\lambda(\hat{x})} - \frac{d\mathbb{P}_h(\hat{x} \mid x', a')}{d\lambda(\hat{x})} \right| d\lambda(\hat{x}) \\ &\leq L_2 \mathcal{D}((x, a), (x', a')) + \|V_{h+1}^\star\|_\infty \int \left| \frac{d\mathbb{P}_h(\hat{x} \mid x, a)}{d\lambda(\hat{x})} - \frac{d\mathbb{P}_h(\hat{x} \mid x', a')}{d\lambda(\hat{x})} \right| d\lambda(\hat{x}) \\ &\leq L_2 \mathcal{D}((x, a), (x', a')) + H \left\| \frac{d\mathbb{P}_h(\cdot \mid x, a)}{d\lambda} - \frac{d\mathbb{P}_h(\cdot \mid x', a')}{d\lambda} \right\|_1 \\ &= L_2 \mathcal{D}((x, a), (x', a')) + 2H \|\mathbb{P}_h(\cdot \mid x, a) - \mathbb{P}_h(\cdot \mid x', a')\|_{TV} \\ &\leq L_2 \mathcal{D}((x, a), (x', a')) + 2HL_1 \mathcal{D}((x, a), (x', a')) \end{aligned}$$

where we have used the fact that the total variation distance is twice the \mathcal{L}_1 distance of the Radon-Nikodym derivatives. \square

Now we further assume that \mathcal{S} is a separable and compact metric space with metric d_S . We also assume that $\mathcal{D}((x, a), (x', a')) \leq Cd_S(x, x')$ where we assume $C = 1$ for simplicity.

As a preliminary we begin with the following lemma.

LEMMA C.2. *Suppose that $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is L Lipschitz and uniformly bounded. Then $g(x) = \sup_{a \in \mathcal{A}} f(x, a)$ is also L Lipschitz.*

PROOF. Fix any x_1 and $x_2 \in \mathcal{S}$. First notice that $|f(x_1, a) - f(x_2, a)| \leq L\mathcal{D}((x_1, a), (x_2, a)) \leq Ld_S(x_1, x_2)$ by choice of product metric.

Thus, for any $a \in \mathcal{A}$ we have that

$$f(x_1, a) \leq f(x_2, a) + Ld_S(x_1, x_2) \leq g(x_2) + Ld_S(x_1, x_2).$$

However, as this is true for any $a \in \mathcal{A}$ we see that $g(x_1) \leq g(x_2) + Ld_S(x_1, x_2)$. Swapping the role of x_1 and x_2 in the inequality shows $|g(x_1) - g(x_2)| \leq Ld_S(x_1, x_2)$ as needed. \square

We will use this and properties of the Wasserstein metric from [9] to give conditions on the Q^\star function to be Lipschitz. First notice the definition of the Wasserstein metric on a separable metric space via:

$$d_W(\mu, \nu) = \sup \left\{ \left| \int f d\mu - \int f d\nu \right| : \|f\|_L \leq 1 \right\}$$

where $\|f\|_L$ is the smallest Lipschitz constant of the function f and μ and ν are measures on the space \mathcal{S} .

PROPOSITION C.3. *Suppose that the transition kernel is Lipschitz with respect to the Wasserstein metric and the reward function is Lipschitz continuous, i.e.*

$$\begin{aligned} |r_h(x, a) - r_h(x', a')| &\leq L_1 \mathcal{D}((x, a), (x', a')) \\ d_W(\mathbb{P}_h(\cdot | x, a), \mathbb{P}_h(\cdot | x', a')) &\leq L_2 \mathcal{D}((x, a), (x', a')) \end{aligned}$$

for all $(x, a), (x', a') \in \mathcal{S} \times \mathcal{A}$ and h where d_W is the Wasserstein metric. Then Q_h^\star and V_h^\star are both $(\sum_{i=0}^{H-h} L_1 L_2^i)$ Lipschitz continuous.

PROOF. We show this by induction on h . For the base case when $h = H + 1$ then $V_{H+1}^\star = Q_{H+1}^\star = 0$ and so the result trivially follows.

Similarly when $h = H$ then by Equation 1 we have that $Q_H^\star(x, a) = r_H(x, a)$. Thus, $|Q_H^\star(x, a) - Q_H^\star(x', a')| = |r_H(x, a) - r_H(x', a')| \leq L_1 \mathcal{D}((x, a), (x', a'))$ by assumption. Moreover, $V_H^\star(x) = \max_{a \in \mathcal{A}} Q_H^\star(x, a)$ is L_1 Lipschitz by Equation 1 and Lemma C.2.

For the step case we assume that Q_{h+1}^\star and V_{h+1}^\star are both $\sum_{i=0}^{H-h-1} L_1 L_2^i$ Lipschitz and show the result for Q_h^\star and V_h^\star . Indeed,

$$\begin{aligned} |Q_h^\star(x, a) - Q_h^\star(x', a')| &= |r_h(x, a) - r_h(x', a') + \mathbb{E}[V_{h+1}^\star(\hat{x}) | x, a] - \mathbb{E}[V_{h+1}^\star(\hat{x}) | x', a']| \\ &\leq |r_h(x, a) - r_h(x', a')| + |\mathbb{E}[V_{h+1}^\star(\hat{x}) | x, a] - \mathbb{E}[V_{h+1}^\star(\hat{x}) | x', a']| \\ &\leq L_1 \mathcal{D}((x, a), (x', a')) + \left| \int V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} | x, a) - \int V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} | x', a') \right| \end{aligned}$$

Now denoting by $K = \sum_{i=0}^{H-h-1} L_1 L_2^i$ by the induction hypothesis and the properties of Wasserstein metric (Equation 3 from [9]) we have that

$$\begin{aligned} |Q_h^\star(x, a) - Q_h^\star(x', a')| &\leq L_1 \mathcal{D}((x, a), (x', a')) \\ &\quad + K \left| \int \frac{1}{K} V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} | x, a) - \int \frac{1}{K} V_{h+1}^\star(\hat{x}) d\mathbb{P}_h(\hat{x} | x', a') \right| \\ &\leq L_1 \mathcal{D}((x, a), (x', a')) + K d_W(\mathbb{P}_h(\cdot | x, a), \mathbb{P}_h(\cdot | x', a')) \\ &\leq L_1 \mathcal{D}((x, a), (x', a')) + K L_2 \mathcal{D}((x, a), (x', a')) \\ &= (L_1 + K L_2) \mathcal{D}((x, a), (x', a')). \end{aligned}$$

Noting that by definition of K we have $L_1 + K L_2 = \sum_{i=0}^{H-h-1} L_1 L_2^i$. To show that V_h^\star is also Lipschitz we simply use Lemma C.2. \square

D POLICY-IDENTIFICATION GUARANTEES PROOF

THEOREM D.1. *For $K = \tilde{O}\left((H^{5/2}/\delta\epsilon)^{d_c+2}\right)$ (where d_c is the covering dimension with parameter c), consider a policy π chosen uniformly at random from π_1, \dots, π_K . Then, for an initial state $X \sim F_1$, with probability at least $1 - \delta$, the policy π obeys*

$$V_1^\star(X) - V_1^\pi(X) \leq \epsilon.$$

Note that in the above guarantee, both X and π are random.

PROOF. Let $X \sim F_1$ be a random starting state sampled from F_1 . Then from Markov's inequality and the law of total expectation, we have that:

$$\begin{aligned} \mathbb{P}(V_1^\star(X) - V_1^\pi(X) > c) &\leq \frac{1}{Kc} \sum_{k=1}^K \mathbb{E} \left[V_1^\star(X) - V_1^{\pi^k}(X) \right] \\ &\leq \frac{1}{Kc} \tilde{O} \left(H^{5/2} K^{(d_c+1)/(d_c+2)} \right) = \tilde{O} \left(\frac{H^{5/2} K^{-\frac{1}{d_c+2}}}{c} \right), \end{aligned}$$

where the last inequality follows from the guarantee in Theorem 4.1. Setting $c = \epsilon$ and $K = \tilde{O} \left((H^{5/2}/\delta\epsilon)^{d_c+2} \right)$ in the above expression gives that $V_1^\star(x) - V_1^\pi(x) \leq \epsilon$ with probability at least $1 - \delta$. \square

E PROOF DETAILS

E.1 Required Lemmas

We start with a collection of lemmas which are required to show Theorem 4.1.

The first two lemmas consider invariants established by the partitioning of the algorithm. These state that the algorithm maintains a partition of the state-action space at every iteration of the algorithm, and that the balls of similar radius are sufficiently apart.

LEMMA E.1 (LEMMA 5.3 FROM THE MAIN PAPER). *For every $(h, k) \in [H] \times [K]$ the following invariants are maintained:*

- (Covering) *The domains of each ball in \mathcal{P}_h^k cover $\mathcal{S} \times \mathcal{A}$.*
- (Separation) *For any two balls of radius r , their centers are at distance at least r .*

PROOF. Let $(h, k) \in [H] \times [K]$ be arbitrary.

For the covering invariant notice that $\cup_{B \in \mathcal{P}_h^k} \text{dom}(B) = \cup_{B \in \mathcal{P}_h^k} B$ where we are implicitly taking the domain with respect to the partition \mathcal{P}_h^k . The covering invariant follows then since \mathcal{P}_h^k contains a ball which covers the entire space $\mathcal{S} \times \mathcal{A}$ from the initialization in the algorithm.

To show the separation invariant, suppose that B_1 and B_2 are two balls of radius r . If B_1 and B_2 share a parent, we note by the splitting rule the algorithm maintains the invariant that the centers are at a distance at least r from each other. Otherwise, suppose without loss of generality that B_1 has parent B^{par} and that B_2 was activated before B_1 . The center of B_1 is some point $(x, a) \in \text{dom}(B^{\text{par}})$. It follows then that $(x, a) \notin \text{dom}(B_2)$ by definition of the domain as $r(B^{\text{par}}) > r(B_2)$. Thus, their centers are at a distance at least r from each other. \square

The second property is useful as it maintains that the centers of the balls of radius r form an r -packing of $\mathcal{S} \times \mathcal{A}$ and hence there are at most $N_r^{\text{pack}} \leq N_r$ balls activated of radius r .

The next theorem gives an analysis on the number of times that a ball of a given radius will be selected by the algorithm. This Lemma also shows the second condition required for a partitioning algorithm to achieve the regret bound as described in Section 6.

LEMMA E.2 (LEMMA 5.4 FROM THE MAIN PAPER). *For any $h \in [H]$ and child ball $B \in \mathcal{P}_h^K$ (the partition at the end of the last episode K) the number of episodes $k \leq K$ such that B is selected by the algorithm is less than $\frac{3}{4}(d_{\max}/r)^2$ where $r = r(B)$. I.e. denoting by B_h^k the ball selected by the algorithm in step h episode k ,*

$$|\{k : B_h^k = B\}| \leq \frac{3}{4} \left(\frac{d_{\max}}{r} \right)^2.$$

Moreover, the number of times that ball B and its ancestors have been played is at least $\frac{1}{4} \left(\frac{d_{\max}}{r} \right)^2$.

For the case when B is the initial ball which covers the entire space then the number of episodes that B is selected is only one.

PROOF. Consider an arbitrary $h \in [H]$ and child ball $B \in \mathcal{P}_h^K$ such that $r(B) = r$. Furthermore, let k be the episode for which ball B was activated. Then B_h^k , the ball selected by the algorithm at step h in episode k is the parent of B . Moreover, if $t = n_h^{k+1}(B_h^k)$ is the number of times that B_h^k or it's ancestors have been played then $t = \left(\frac{d_{\max}}{r(B_h^k)} \right)^2$ by the activation rule. Also, $r(B_h^k) = 2r(B)$ by the re-partitioning scheme. Hence, the number of times that B and its ancestors have been played is at least

$$t = \left(\frac{d_{\max}}{r(B_h^k)} \right)^2 = \left(\frac{d_{\max}}{2r(B)} \right)^2 = \frac{1}{4} \left(\frac{d_{\max}}{r(B)} \right)^2.$$

The number of episodes that B can be selected (i.e. $|\{k : B_h^k = B\}|$) by the algorithm is bounded above by $\left(\frac{d_{\max}}{r(B)} \right)^2 - \left(\frac{d_{\max}}{r(B_h^k)} \right)^2$ as this is the number of samples of B required to split the ball by the partitioning scheme. However, plugging in $r(B_h^k) = 2r(B)$ gives

$$\left(\frac{d_{\max}}{r(B)} \right)^2 - \left(\frac{d_{\max}}{2r(B)} \right)^2 = \frac{3}{4} \left(\frac{d_{\max}}{r(B)} \right)^2$$

as claimed.

Lastly if B is the initial ball which covers the entire space then $r(B) = d_{\max}$ initially and so the ball is split after it is selected only once. \square

Next we provide a recursive relationship for the update in Q estimates.

LEMMA E.3. For any $h, k \in [H] \times [K]$ and ball $B \in \mathcal{P}_h^K$ let $t = n_h^k(B)$ be the number of times that B or its ancestors were encountered during the algorithm before episode k . Further suppose that B and its ancestors were encountered at step h of episodes $k_1 < k_2 < \dots < k_t < k$. By the update rule of Q we have that:

$$Q_h^k(B) = \mathbb{1}_{[t=0]}H + \sum_{i=1}^t \alpha_i^i \left(r_h(x_h^{k_i}, a_h^{k_i}) + \mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) + b(i) \right).$$

PROOF. We show the claim by induction on $t = n_h^k(B)$.

First suppose that $t = 0$, i.e. that the ball B has not been encountered before by the algorithm. Then initially $Q_h^1(B) = H = \mathbb{1}_{[t=0]}H$.

Now for the step case we notice that $Q_h^k(B)$ was last updated at episode k_t . k_t is either the most recent episode when ball B was encountered, or the most recent episode when it's parent was encountered if B was activated and not yet played. In either case by the update rule (Equation 5)

we have

$$\begin{aligned}
Q_h^k(B) &= (1 - \alpha_t)Q_h^{k_t}(B) + \alpha_t \left(r_h(x_h^{k_t}, a_h^{k_t}) + V_{h+1}^{k_t}(x_{h+1}^{k_t}) + b(t) \right) \\
&= (1 - \alpha_t)\alpha_{t-1}^0 H + (1 - \alpha_t) \sum_{i=1}^{t-1} \alpha_t^i \left(r_h(x_h^{k_i}, a_h^{k_i}) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b(i) \right) \\
&\quad + \alpha_t \left(r_h(x_h^{k_t}, a_h^{k_t}) + V_{h+1}^{k_t}(x_{h+1}^{k_t}) + b(t) \right) \text{ by the induction hypothesis} \\
&= \mathbb{1}_{[t=0]}H + \sum_{i=1}^t \alpha_t^i \left(r_h(x_h^{k_i}, a_h^{k_i}) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b(i) \right)
\end{aligned}$$

by definition of α_t^i . □

The next lemma extends the relationship between the optimal Q value $Q_h^*(x, a)$ for any $(x, a) \in \mathcal{S} \times \mathcal{A}$ to the estimate of the Q value for any ball B containing (x, a) . For ease of notation we denote $(V_h^k(x) - V_h^*(x)) = (V_h^k - V_h^*)(x)$.

LEMMA E.4. *For any $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and ball $B \in \mathcal{P}_h^k$ such that $(x, a) \in \text{dom}(B)$ then if $t = n_h^k(B)$ and B and its ancestors were previously encountered at step h in episodes $k_1 < k_2 < \dots < k_t < k$ then*

$$\begin{aligned}
Q_h^k(B) - Q_h^*(x, a) &= \mathbb{1}_{[t=0]}(H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i \left((V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + V_{h+1}^*(x_{h+1}^{k_i}) \right. \\
&\quad \left. - \mathbb{E} \left[V_{h+1}^*(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] + b(i) + Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a) \right)
\end{aligned}$$

PROOF. Consider any $(x, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$ arbitrary and an episode $k \in [K]$. Furthermore, let B be any ball such that $(x, a) \in B$. First notice by Lemma 3.1 that $\mathbb{1}_{[t=0]} + \sum_{i=1}^t \alpha_t^i = 1$ for any $t \geq 0$.

We show the claim by induction on $t = n_h^k(B)$.

For the base case that $t = 0$ then $Q_h^k(B) = H$ as it has not been encountered before by the algorithm. Then

$$Q_h^k(B) - Q_h^*(x, a) = H - Q_h^*(x, a) = \mathbb{1}_{[t=0]}(H - Q_h^*(x, a)).$$

For the step case we use the fact that $\sum_{i=1}^t \alpha_t^i + \mathbb{1}_{[t=0]} = 1$. Thus, $Q_h^*(x, a) = \sum_{i=1}^t \alpha_t^i Q_h^*(x, a) + \mathbb{1}_{[t=0]} Q_h^*(x, a)$. Subtracting this from $Q_h^k(B)$ and using Lemma E.3 yields

$$\begin{aligned}
Q_h^k(B) - Q_h^*(x, a) &= \mathbb{1}_{[t=0]}H + \sum_{i=1}^t \alpha_t^i \left(r_h(x_h^{k_i}, a_h^{k_i}) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b(i) \right) \\
&\quad - \mathbb{1}_{[t=0]}Q_h^*(x, a) - \sum_{i=1}^t \alpha_t^i Q_h^*(x, a) \\
&= \mathbb{1}_{[t=0]}(H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i \left(r_h(x_h^{k_i}, a_h^{k_i}) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b(i) \right. \\
&\quad \left. + Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a) \right).
\end{aligned}$$

However, using the Bellman Equations (1) we know that

$$Q_h^*(x_h^{k_i}, a_h^{k_i}) = r_h(x_h^{k_i}, a_h^{k_i}) + \mathbb{E} \left[V_{h+1}^*(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right].$$

Using this above we get that

$$\begin{aligned}
Q_h^k(B) - Q_h^\star(x, a) &= \mathbb{1}_{[t=0]}(H - Q_h^\star(x, a)) + \sum_{i=1}^t \alpha_t^i \left(V_{h+1}^{k_i}(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] \right. \\
&\quad \left. + b(i) + Q_h^\star(x_h^{k_i}, a_h^{k_i}) - Q_h^\star(x, a) \right) \\
&= \mathbb{1}_{[t=0]}(H - Q_h^\star(x, a)) + \sum_{i=1}^t \alpha_t^i \left((V_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) \right. \\
&\quad \left. + V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] + b(i) + Q_h^\star(x_h^{k_i}, a_h^{k_i}) - Q_h^\star(x, a) \right).
\end{aligned}$$

□

Consider now $V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right]$. We notice that as the next state $x_{h+1}^{k_i}$ is drawn from $\mathbb{P}_h(\cdot \mid x_h^{k_i}, a_h^{k_i})$ then $\mathbb{E} \left[V_{h+1}^\star(x_{h+1}^{k_i}) \right] = \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right]$. Thus, this sequence forms a martingale difference sequence and so by Azuma-Hoeffding's inequality we are able to show the following.

LEMMA E.5. *For all $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and for all $\delta \in (0, 1)$ we have with probability at least $1 - \delta/2$ if $(x, a) \in B$ for some $B \in \mathcal{P}_h^k$ then for $t = n_h^k(B)$ and episodes $k_1 < k_2 < \dots < k_t < k$ where B and its ancestors were encountered at step h before episode k then*

$$\left| \sum_{i=1}^t \alpha_t^i \left(V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] \right) \right| \leq H \sqrt{2 \sum_{i=1}^t (\alpha_t^i)^2 \log(4HK/\delta)}.$$

PROOF. Consider the sequence

$$k_i = \min(k, \min\{\hat{k} : n_h^{\hat{k}}(B^a) = i \text{ and } B^a \text{ is an ancestor of } B \text{ or } B \text{ itself}\}).$$

Clearly k_i is the episode for which B or its ancestors were encountered at step h for the i -th time (as once a ball is split it is never chosen by the algorithm again). Setting

$$Z_i = \mathbb{1}_{[k_i \leq k]} \left(V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] \right)$$

then Z_i is a martingale difference sequence with respect to the filtration \mathcal{F}_i which we denote as the information available to the agent up to an including the step k_i . Moreover, as the sum of a martingale difference sequence is a martingale then for any $\tau \leq K$, $\sum_{i=1}^\tau Z_i$ is a martingale. Noticing that the difference between subsequent terms is bounded by $H\alpha_\tau^i$ and Azuma-Hoeffding's inequality we see that for a fixed $\tau \leq K$

$$\begin{aligned}
\mathbb{P} \left(\left| \sum_{i=1}^\tau \alpha_\tau^i Z_i \right| \leq H \sqrt{2 \sum_{i=1}^\tau (\alpha_\tau^i)^2 \log \left(\frac{4HK}{\delta} \right)} \right) \\
\geq 1 - 2 \exp \left(- \frac{2H^2 \sum_{i=1}^\tau (\alpha_\tau^i)^2 \log \left(\frac{4HK}{\delta} \right)}{2H^2 \sum_{i=1}^\tau (\alpha_\tau^i)^2} \right) \\
= 1 - \frac{\delta}{2HK}.
\end{aligned}$$

Since the inequality holds for fixed $\tau \leq K$ the inequality must also hold for the random stopping time $\tau = t = n_h^k(B) \leq K$. Moreover, for this stopping time each of the indicator functions will be 1.

Taking a union bound over the number of episodes and over all H the result follows. We only need to union bound over the number of episodes instead of the number of balls as the inequality is satisfied for all balls not selected in a given round as it inherits its concentration from its parent ball because the value for t does not change.

We also notice that for $t > 0$ that $\sum_{i=1}^t (\alpha_i^i)^2 \leq \frac{2H}{t}$ by Lemma 3.1 and so

$$H \sqrt{2 \sum_{i=1}^t (\alpha_i^i)^2 \log(4HK/\delta)} \leq H \sqrt{2 \frac{2H}{t} \log(4HK/\delta)} = 2 \sqrt{\frac{H^3 \log(4HK/\delta)}{t}}.$$

□

The next lemma uses the activation rule and Assumption 2 to bound the difference between the optimal Q functions at a point (x, a) in B from the points sampled in ancestors of B by the algorithm. This corresponds to the accumulation of discretization errors in the algorithm by accumulating estimates over a ball.

LEMMA E.6. *For any $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and ball $B \in \mathcal{P}_h^k$ with $(x, a) \in \text{dom}(B)$ if B and its ancestors were encountered at step h in episodes $k_1 < k_2 < \dots < k_t < k$ where $t = n_h^k(B)$ then*

$$\sum_{i=1}^t \alpha_i^i |Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a)| \leq \frac{4Ld_{\max}}{\sqrt{t}}.$$

PROOF. Denote by $B_h^{k_i}$ as the ball selected in step h of episode k_i and $n_h^{k_i}$ as $n_h^{k_i}(B_h^{k_i})$. Then each $B_h^{k_i}$ is an ancestor of B and so both (x, a) and $(x_h^{k_i}, a_h^{k_i})$. Hence by the Lipschitz assumption (Assumption 2) we have that $|Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a)| \leq L \text{diam}(B_h^{k_i}) \leq 2Lr(B_h^{k_i})$.

However, $r(B_h^{k_i}) \leq \frac{d_{\max}}{\sqrt{i}}$. Indeed, by the re-partition rule we split when $n_h^{k_i} = \left(\frac{d_{\max}}{r(B_h^{k_i})}\right)^2$ (and afterwards $B_h^{k_i}$ is not chosen again) and so $n_h^{k_i} \leq \left(\frac{d_{\max}}{r(B_h^{k_i})}\right)^2$. Square rooting this gives that $\sqrt{n_h^{k_i}} \leq \frac{d_{\max}}{r(B_h^{k_i})}$. However, as $n_h^{k_i} = i$ we get that $r(B_h^{k_i}) \leq \frac{d_{\max}}{\sqrt{i}}$. Using this we have that

$$\begin{aligned} \sum_{i=1}^t \alpha_i^i |Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a)| &\leq \sum_{i=1}^t \alpha_i^i 2Lr(B_h^{k_i}) \\ &\leq 2 \sum_{i=1}^t \alpha_i^i Ld_{\max} \frac{1}{\sqrt{i}} \\ &\leq 4Ld_{\max} \frac{1}{\sqrt{t}} \text{ by Lemma 3.1.} \end{aligned}$$

□

The next lemma provides an upper and lower bound on the difference between $Q_h^k(B)$ and $Q_h^*(x, a)$ for any $(x, a) \in \text{dom}(B)$. It also shows that our algorithm is *optimistic*, in that the estimates are upper bounds for the true quality and value function for the optimal policy [22].

LEMMA E.7 (EXPANDED VERSION OF LEMMA 5.1 FROM THE MAIN PAPER). *For any $\delta \in (0, 1)$ if $\beta_t = 2 \sum_{i=1}^t \alpha_i^i b(i)$ then*

$$\beta_t \leq 8 \sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 16 \frac{Ld_{\max}}{\sqrt{t}}$$

and

$$\beta_t \geq 4\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 8\frac{Ld_{\max}}{\sqrt{t}}.$$

Moreover, with probability at least $1 - \delta/2$ the following holds simultaneously for all $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ and ball B such that $(x, a) \in \text{dom}(B)$ where $t = n_h^k(B)$ and $k_1 < \dots < k_t$ are the episodes where B or its ancestors were encountered previously by the algorithm

$$0 \leq Q_h^k(B) - Q_h^\star(x, a) \leq \mathbb{1}_{[t=0]}H + \beta_t + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i})$$

PROOF. First consider $\beta_t = 2 \sum_{i=1}^t \alpha_t^i b(i)$. By definition of the bonus term as

$$b(t) = 2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 2\frac{Ld_{\max}}{\sqrt{t}}$$

we have the following (where we use Lemma 3.1):

$$\begin{aligned} \beta_t &= 2 \sum_{i=1}^t \alpha_t^i b(i) \\ &= 2 \sum_{i=1}^t \alpha_t^i \left(2\sqrt{\frac{H^3 \log(4HK/\delta)}{i}} + 4\frac{Ld_{\max}}{\sqrt{i}} \right) \\ &\leq 4 \left(2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 4\frac{Ld_{\max}}{\sqrt{t}} \right) \\ &= 8\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 16\frac{Ld_{\max}}{\sqrt{t}} \end{aligned}$$

We can similarly lower-bound the expression by $4\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 8\frac{Ld_{\max}}{\sqrt{t}}$ using the lower bound from Lemma 3.1.

We start with the upper bound on $Q_h^k(B) - Q_h^\star(x, a)$. Indeed, by Lemma E.4 we have

$$\begin{aligned} Q_h^k(B) - Q_h^\star(x, a) &= \mathbb{1}_{[t=0]}(H - Q_h^\star(x, a)) + \sum_{i=1}^t \alpha_t^i \left((V_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) \right. \\ &\quad \left. + V_{h+1}^\star(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^\star(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] + b(i) + Q_h^\star(x_h^{k_i}, a_h^{k_i}) - Q_h^\star(x, a) \right). \end{aligned}$$

However, with probability at least $1 - \delta/2$ by Lemma E.5 and Lemma E.6 we get

$$\begin{aligned} &\leq \mathbb{1}_{[t=0]}H + \sum_{i=1}^t \alpha_t^i b(i) + 2\sqrt{\frac{H^3 \log(4HK/\delta)}{t}} + 4\frac{Ld_{\max}}{\sqrt{t}} + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) \\ &\leq \mathbb{1}_{[t=0]}H + \sum_{i=1}^t \alpha_t^i b(i) + \frac{\beta_t}{2} + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^\star)(x_{h+1}^{k_i}) \end{aligned}$$

Using that $\sum_{i=1}^t \alpha_t^i b(i) = \beta_t/2$ establishes the upper bound.

For the lower bound we show the claim by induction on $h = H + 1, H, \dots, 1$.

Indeed, for the base case when $h = H + 1$ then $Q_{H+1}^\star(x, a) = 0 = Q_{H+1}^k(B)$ for every k , ball B , and $(x, a) \in \mathcal{S} \times \mathcal{A}$ trivially as at the end of the episode the expected future reward is always zero.

Now, assuming the claim for $h+1$ we show the claim for h . First consider $\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) - V_{h+1}^*(x_{h+1}^{k_i})$. We show that $\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) - V_{h+1}^*(x_{h+1}^{k_i}) \geq 0$. By the Bellman Equations (1) we know that

$$V_{h+1}^*(x_{h+1}^{k_i}) = \sup_{a \in \mathcal{A}} Q_{h+1}^*(x_{h+1}^{k_i}, a) = Q_{h+1}^*(x_{h+1}^{k_i}, \pi_{h+1}^*(x_{h+1}^{k_i})).$$

If $\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) = H$ then the inequality trivially follows as $V^*(x_{h+1}^{k_i}) \leq H$. Thus by the update in the algorithm we can assume that $\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) = \max_{B \in \text{RELEVANT}_{h+1}^{k_i}(x_{h+1}^{k_i})} Q_h^{k_i}(B)$. Now let B^* be the ball with smallest radius in $\mathcal{P}_{h+1}^{k_i}$ such that $(x_{h+1}^{k_i}, \pi_{h+1}^*(x_{h+1}^{k_i})) \in B^*$. Such a ball exists as $\mathcal{P}_{h+1}^{k_i}$ covers $\mathcal{S} \times \mathcal{A}$ by Lemma E.1. Moreover $(x_{h+1}^{k_i}, \pi_{h+1}^*(x_{h+1}^{k_i})) \in \text{dom}(B^*)$ as B^* was taken to have the smallest radius. Thus, $Q_{h+1}^{k_i}(B^*) \geq Q_{h+1}^*(x_{h+1}^{k_i}, \pi_{h+1}^*(x_{h+1}^{k_i}))$ by the induction hypothesis. Hence we have that

$$\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) \geq Q_{h+1}^{k_i}(B^*) \geq Q_{h+1}^*(x_{h+1}^{k_i}, \pi_{h+1}^*(x_{h+1}^{k_i})) = V_{h+1}^*(x_{h+1}^{k_i}).$$

Putting all of this together with Lemma E.4 then with probability $1 - \delta/2$ we have (using Lemma E.5 and definition of $\beta_t/2$) that

$$\begin{aligned} Q_h^k(B) - Q_h^*(x, a) &= \mathbb{1}_{[t=0]}(H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i (\mathbf{V}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) \\ &\quad + V_{h+1}^*(x_{h+1}^{k_i}) - \mathbb{E} \left[V_{h+1}^*(\hat{x}) \mid x_h^{k_i}, a_h^{k_i} \right] + b(i) + Q_h^*(x_h^{k_i}, a_h^{k_i}) - Q_h^*(x, a) \\ &\geq \sum_{i=1}^t \alpha_t^i (\mathbf{V}_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + \frac{\beta_t}{2} - 2\sqrt{H^3 \log(4HK/\delta)/t} - 4 \frac{Ld_{\max}}{\sqrt{t}} \\ &\geq \frac{\beta_t}{2} - \frac{\beta_t}{2} \geq 0 \end{aligned}$$

where in the last line we used that $\mathbf{V}_{h+1}^{k_i}(x_{h+1}^{k_i}) \geq V_{h+1}^*(x_{h+1}^{k_i})$ from before. \square

The next Lemma extends the results from Lemma E.7 to lower bounds of $\mathbf{V}_h^k - V_h^*$ over the entire state space \mathcal{S} .

COROLLARY E.8. *For any $\delta \in (0, 1)$ with probability at least $1 - \delta/2$ the following holds simultaneously for all $(x, h, k) \in \mathcal{S} \times [H] \times [K]$*

$$\mathbf{V}_h^k(x) - V_h^*(x) \geq 0$$

PROOF. The proof of the lower bound follows from the argument used in the proof of Lemma E.7. \square

The next lemma uses Azuma-Hoeffding's inequality to bound a martingale difference sequence which arises in the proof of the regret bound.

LEMMA E.9. *For any $\delta \in (0, 1)$ with probability at least $1 - \delta/2$ we have that*

$$\sum_{h=1}^H \sum_{k=1}^K \left| \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^*(x_{h+1}^k) - V_{h+1}^{\pi^k}(x_h^k, a_h^k)) \right| \leq 2\sqrt{2H^3 K \log(4HK/\delta)}.$$

PROOF. First consider $Z_h^k = \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^*(x_{h+1}^k) - V_{h+1}^{\pi^k}(x_h^k, a_h^k))$. Similar to the proof of Lemma E.5 we notice that Z_h^k is a martingale difference sequence due to the fact that the next state is drawn from the distribution $\mathbb{P}_h(\cdot \mid x_h^k, a_h^k)$. Using that $|Z_h^k| \leq 2H$ we have that

$$\begin{aligned}
& \mathbb{P} \left(\sum_{h=1}^H \sum_{k=1}^K \left| \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^*(x_{h+1}^k) - V_{h+1}^{\pi^k}(x_h^k, a_h^k)) \right| > \sqrt{8H^3 K \log(4HK/\delta)} \right) \\
& \leq 2 \exp \left(-\frac{8H^3 K \log(4HK/\delta)}{2HK(2H)^2} \right) \\
& = 2 \exp \left(-\frac{8H^3 K \log(4HK/\delta)}{8H^3 K} \right) \\
& = 2 \frac{\delta}{4KH} \leq \delta/2.
\end{aligned}$$

Thus with probability at least $1 - \delta/2$ we have that

$$\sum_{h=1}^H \sum_{k=1}^K \left| \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^*(x_{h+1}^k) - V_{h+1}^{\pi^k}(x_h^k, a_h^k)) \right| \leq \sqrt{8H^3 K \log(4HK/\delta)}$$

as claimed. \square

This next lemma provides a bound on the sum of errors accumulated from the confidence bounds β_t throughout the algorithm. This term gives rise to the N_r covering terms from Theorem 4.1. This Lemma can also show the third condition required for a partitioning algorithm to achieve the regret bound as described in Section 6.

LEMMA E.10. *For every $h \in [H]$, if B_h^k is the ball selected by the algorithm in step h episode k and $n_h^k = n_h^k(B_h^k)$ then*

$$\sum_{k=1}^K \beta_{n_h^k} \leq 32 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} N_r \frac{d_{\max}}{r} + \frac{Kr_0}{d_{\max}} \right).$$

PROOF. Using Lemma E.7,

$$\begin{aligned}
\sum_{k=1}^K \beta_{n_h^k} & \leq \sum_{k=1}^K 8 \sqrt{\frac{H^3 \log(4HK/\delta)}{n_h^k}} + 16 \frac{Ld_{\max}}{\sqrt{n_h^k}} \\
& \leq 16 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \sum_{k=1}^K \frac{1}{\sqrt{n_h^k}}.
\end{aligned}$$

We thus turn our attention to $\sum_{k=1}^K 1/\sqrt{n_h^k}$. Rewriting the sum in terms of the radius of all balls activated by the algorithm we get for an arbitrary $r_0 \in (0, d_{\max}]$,

$$\begin{aligned}
\sum_{k=1}^K \frac{1}{\sqrt{n_h^k}} & = \sum_{r=d_{\max}2^{-i}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{n_h^k(B)}} \\
& = \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{n_h^k(B)}} + \sum_{\substack{r=d_{\max}2^{-i} \\ r < r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{n_h^k(B)}},
\end{aligned}$$

taking the two cases when $r \geq r_0$ and $r < r_0$ separately. We first start with $r < r_0$. Then,

$$\begin{aligned}
& \sum_{\substack{r=d_{\max}2^{-i} \\ r < r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{n_h^k(B)}} \\
& \leq \sum_{\substack{r=d_{\max}2^{-i} \\ r < r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{\frac{1}{4}d_{\max}^2/r^2}} \text{ by Lemma E.2} \\
& \leq \sum_{\substack{r=d_{\max}2^{-i} \\ r < r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{2r_0}{d_{\max}} \leq \frac{2Kr_0}{d_{\max}}
\end{aligned}$$

by bounding the number of terms in the sum by the number of episodes K .

For the case when $r \geq r_0$ we get

$$\begin{aligned}
& \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{k: B_h^k=B} \frac{1}{\sqrt{n_h^k(B)}} \leq \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \sum_{i=1}^{\frac{3}{4}\left(\frac{d_{\max}}{r}\right)^2} \frac{1}{\sqrt{i + \frac{1}{4}\left(\frac{d_{\max}}{r}\right)^2}} \text{ by Lemma E.2} \\
& \leq \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} \int_1^{\frac{3}{4}\left(\frac{d_{\max}}{r}\right)^2} \frac{1}{\sqrt{x + \frac{1}{4}\left(\frac{d_{\max}}{r}\right)^2}} dx \\
& = \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} \sum_{\substack{B \in \mathcal{P}_h^K \\ r(B)=r}} 2 \frac{d_{\max}}{r} \\
& \leq \sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} 2N_r \frac{d_{\max}}{r}
\end{aligned}$$

as by Lemma E.1 the centers of balls of radius r are at a distance at least r from each other and thus form an r -packing of $\mathcal{S} \times \mathcal{A}$. Hence, the total number of balls of radius r is at most N_r . Thus we get that (taking the inf over r_0 arbitrary):

$$\sum_{k=1}^K \frac{1}{\sqrt{n_h^k}} \leq \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} 2N_r \frac{d_{\max}}{r} + \frac{2Kr_0}{d_{\max}} \right).$$

Plugging this back into the first inequality gives that

$$\begin{aligned}
\sum_{k=1}^K \beta_{n_h^k} &\leq 16 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \sum_{k=1}^K \frac{1}{\sqrt{n_h^k}} \\
&\leq 16 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} 2N_r \frac{d_{\max}}{r} + \frac{2Kr_0}{d_{\max}} \right) \\
&= 32 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} N_r \frac{d_{\max}}{r} + \frac{Kr_0}{d_{\max}} \right).
\end{aligned}$$

□

E.2 Regret Analysis

We start with a lemma which was highlighted in the proof sketch from Section 5.

LEMMA E.11 (LEMMA 5.2 FROM THE MAIN PAPER). *For any $\delta \in (0, 1)$ if $\beta_t = 2 \sum_{i=1}^t \alpha_i^i b(i)$ then with probability at least $1 - \delta/2$, for all $h \in [H]$,*

$$\sum_{k=1}^K (\mathbf{V}_h^k - V_h^{\pi^k})(x_h^k) \leq \sum_{k=1}^K (H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k) + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k).$$

PROOF OF LEMMA E.11. By equations 6 and 1, and the definition of the selection rule by the algorithm it follows that for any h and k that

$$\begin{aligned}
\mathbf{V}_h^k(x_h^k) - V_h^{\pi^k}(x_h^k) &\leq \max_{B \in \text{RELEVANT}_h^k(x_h^k)} \mathbf{Q}_h^k(B) - Q_h^{\pi^k}(x_h^k, a_h^k) \\
&= \mathbf{Q}_h^k(B_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k) \\
&= \mathbf{Q}_h^k(B_h^k) - Q_h^*(x_h^k, a_h^k) + Q_h^*(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k).
\end{aligned}$$

First, by Equation 1 we know that $Q_h^*(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k) = \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right]$. Moreover, as $(x_h^k, a_h^k) \in \text{dom}(B_h^k)$ we can use Lemma 5.1 and get for $t = n_h^k(B_h^k)$ and episodes $k_1 < \dots < k_t$ where B_h^k or its ancestors were previously encountered

$$\begin{aligned}
&\mathbf{Q}_h^k(B_h^k) - Q_h^*(x_h^k, a_h^k) + Q_h^*(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k) \\
&\leq \mathbb{1}_{[t=0]}H + \beta_t + \sum_{i=1}^t \alpha_t^i (\mathbf{V}_{h+1}^{k_i} - V_{h+1}^*(x_{h+1}^{k_i})) + \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] \\
&\leq \mathbb{1}_{[t=0]}H + \beta_t + \sum_{i=1}^t \alpha_t^i (\mathbf{V}_{h+1}^{k_i} - V_{h+1}^*(x_{h+1}^{k_i})) + (V_{h+1}^* - V_{h+1}^{\pi^k})(x_{h+1}^k) + \xi_{h+1}^k
\end{aligned}$$

where $\xi_{h+1}^k = \mathbb{E} \left[V_{h+1}^*(\hat{x}) - V_{h+1}^{\pi^k}(\hat{x}) \mid x_h^k, a_h^k \right] - (V_{h+1}^* - V_{h+1}^{\pi^k})(x_{h+1}^k)$. Taking the sum over all episodes k and letting $n_h^k = n_h^k(B_h^k)$ and the respective episodes $k_i(B_h^k)$ as the time B_h^k or its ancestors were

selected for the i 'th time,

$$\begin{aligned} \sum_{k=1}^K \mathbf{V}_h^k(x_h^k) - V_h^{\pi^k}(x_h^k) &\leq \sum_{k=1}^K \left(\mathbb{1}_{[n_h^k=0]} H + \beta_{n_h^k} \right) + \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i (\mathbf{V}_{h+1}^{k_i(B_h^k)} - V_{h+1}^\star)(x_{h+1}^{k_i(B_h^k)}) \\ &\quad + \sum_{k=1}^K \left((V_{h+1}^\star - V_{h+1}^{\pi^k})(x_{h+1}^k) + \xi_{h+1}^k \right). \end{aligned} \quad (8)$$

For the second term we rearrange the summation using the observation used in the proof from [10, 25]. For every $k' \in [K]$ the term $(\mathbf{V}_{h+1}^{k'} - V_{h+1}^\star)(x_{h+1}^{k'})$ appears in the summand when $k = n_h^{k'}$. The next time it appears when $k = n_h^{k'} + 1$ and so on. Hence by rearranging and using Lemma 3.1, it follows that

$$\begin{aligned} \sum_{k=1}^K \sum_{i=1}^{n_h^k} \alpha_{n_h^k}^i (\mathbf{V}_{h+1}^{k_i(B_h^k)} - V_{h+1}^\star)(x_{h+1}^{k_i(B_h^k)}) &\leq \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) \sum_{t=n_h^k}^{\infty} \alpha_t^{n_h^k} \\ &\leq \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k). \end{aligned}$$

Using these inequalities in Equation 8 we have that

$$\begin{aligned} \sum_{k=1}^K \mathbf{V}_h^k(x_h^k) - V_h^{\pi^k}(x_h^k) &\leq \sum_{k=1}^K (H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k) + \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) \\ &\quad + \sum_{k=1}^K (V_{h+1}^\star - V_{h+1}^{\pi^k})(x_{h+1}^k). \end{aligned} \quad (9)$$

However, noticing that $V_{h+1}^{\pi^k}(x_{h+1}^k) \leq V_{h+1}^\star(x_{h+1}^k)$ we have

$$\begin{aligned} &\left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) + \sum_{k=1}^K (V_{h+1}^\star - V_{h+1}^{\pi^k})(x_{h+1}^k) \\ &= \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) + \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k) - (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) \\ &= \frac{1}{H} \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^\star)(x_{h+1}^k) + \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k) \\ &\leq \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k). \end{aligned}$$

Substituting this back into Equation 9 we get that

$$\sum_{k=1}^K (\mathbf{V}_h^k - V_h^{\pi^k})(x_h^k) \leq \sum_{k=1}^K (H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k) + \left(1 + \frac{1}{H} \right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k).$$

□

With the machinery in place we are now ready to show the general version of Theorem 4.1. We restate it here for convenience.

THEOREM E.12. *For any any sequence of initial states $\{x_1^k \mid k \in [K]\}$, and any $\delta \in (0, 1)$ with probability at least $1 - \delta$ Adaptive Q-learning (Alg 1) achieves regret guarantee:*

$$R(K) \leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} \\ + 96H \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} N_r \frac{d_{\max}}{r} + \frac{Kr_0}{d_{\max}} \right)$$

where N_r is the r -covering number of $\mathcal{S} \times \mathcal{A}$ with respect to the metric \mathcal{D} .

PROOF. By definition of the regret we have that $R(K) = \sum_{k=1}^K (V_1^\star(x_1^k) - V_1^{\pi^k}(x_1^k))$. By Lemma E.8 we know with probability at least $1 - \delta/2$ that for any (x, a, h, k) that $\mathbf{V}_h^k(x) - V_h^\star(x) \geq 0$. Hence we have that $R(K) \leq \sum_{k=1}^K (\mathbf{V}_1^k(x_1^k) - V_1^{\pi^k}(x_1^k))$.

The main idea of the rest of the proof is to upper bound $\sum_{k=1}^K (\mathbf{V}_h^k(x_h^k) - V_h^{\pi^k}(x_h^k))$ by the next step $\sum_{k=1}^K (\mathbf{V}_{h+1}^k(x_{h+1}^k) - V_{h+1}^{\pi^k}(x_{h+1}^k))$. For any fixed $(h, k) \in [H] \times [K]$ let B_h^k be the ball selected at episode k step h and $t = n_h^k(B)$. Using Lemma E.11 we know for any $h \in [H]$ that

$$\sum_{k=1}^K (\mathbf{V}_h^k - V_h^{\pi^k})(x_h^k) \leq \sum_{k=1}^K (H \mathbb{1}_{[n_h^k=0]} + \beta_{n_h^k} + \xi_{h+1}^k) + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K (\mathbf{V}_{h+1}^k - V_{h+1}^{\pi^k})(x_{h+1}^k).$$

For the first term we notice that

$$\sum_{k=1}^K \mathbb{1}_{[n_h^k=0]} H \leq H$$

as the indicator is 1 only when $k = 1$ for the first iteration when there is only a single ball covering the entire space.

Substituting this in and expanding this relationship out and using the fact that $\mathbf{V}_{H+1}^k(x_{H+1}^k) - V_{H+1}^{\pi^k}(x_{H+1}^k) = 0$ as $V_{H+1}^k = V_{H+1}^{\pi^k} = 0$ gives

$$R(K) \leq \sum_{k=1}^K (V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k)) \leq H \sum_{h=1}^H \left(1 + \frac{1}{H}\right)^{h-1} + \sum_{h=1}^H \left(1 + \frac{1}{H}\right)^{h-1} \sum_{k=1}^K (\beta_{n_h^k} + \xi_{h+1}^k).$$

Indeed, we can show this by induction on H . For the case when $H = 1$ then we get (using that $\mathbf{V}_2^k(x) = V_2^{\pi^k}(x) = 0$)

$$\begin{aligned} \sum_{k=1}^K (V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k)) &\leq H + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K (\mathbf{V}_2^k(x_2^k) - V_2^{\pi^k}(x_2^k)) + \sum_{k=1}^K (\beta_{n_1^k} + \xi_2^k) \\ &= H \sum_{h=1}^1 \left(1 + \frac{1}{H}\right)^{h-1} + \sum_{h=1}^1 \left(1 + \frac{1}{H}\right)^{h-1} \sum_{k=1}^K (\beta_{n_1^k} + \xi_2^k). \end{aligned}$$

For the step case then

$$\begin{aligned}
& \sum_{k=1}^K V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k) \\
& \leq H + \left(1 + \frac{1}{H}\right) \sum_{k=1}^K \left(V_2^k(x_2^k) - V_2^{\pi^k}(x_2^k)\right) + \sum_{k=1}^K \left(\beta_{n_1^k} + \xi_2^k\right) \\
& \leq H + \left(1 + \frac{1}{H}\right) \left(H \sum_{h=1}^{H-1} \left(1 + \frac{1}{H}\right)^{h-1} + \sum_{h=1}^{H-1} \left(1 + \frac{1}{H}\right)^{h-1} \sum_{k=1}^K \left(\beta_{n_{h+1}^k} + \xi_{h+2}^k\right) \right) + \sum_{k=1}^K \left(\beta_{n_1^k} + \xi_2^k\right) \\
& = H \sum_{h=1}^H \left(1 + \frac{1}{H}\right)^{h-1} + \sum_{h=1}^H \left(1 + \frac{1}{H}\right)^{h-1} \sum_{k=1}^K \left(\beta_{n_h^k} + \xi_{h+1}^k\right).
\end{aligned}$$

Moreover, noticing that $\sum_{h=1}^H \left(1 + \frac{1}{H}\right)^{h-1} \leq 3H$ and that $\left(1 + \frac{1}{H}\right)^H \leq \left(1 + \frac{1}{H}\right)^H \leq 3$ we get

$$\sum_{k=1}^K \left(V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k)\right) \leq 3H^2 + 3 \sum_{h=1}^H \sum_{k=1}^K \left(\beta_{n_h^k} + \xi_{h+1}^k\right).$$

However, $\sum_{k=1}^K \sum_{h=1}^H \xi_{h+1}^k \leq 2\sqrt{2H^3K \log(4HK/\delta)}$ with probability at least $1 - \delta/2$ by Lemma E.9. Hence by with probability $1 - \delta$ by combining the two high probability guarantees we have

$$\sum_{k=1}^K \left(V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k)\right) \leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} + 3 \sum_{h=1}^H \sum_{k=1}^K \beta_{n_h^k}.$$

We use Lemma E.10 to bound $\sum_{k=1}^K \beta_{n_h^k}$.

Combing all of the pieces in the final regret bound we get with probability at least $1 - \delta$ that

$$\begin{aligned}
R(K) & \leq \sum_{k=1}^K (V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k)) \\
& \leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} + 3 \sum_{h=1}^H \sum_{k=1}^K \beta_{n_h^k} \\
& \leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} \\
& \quad + 3 \sum_{h=1}^H 16 \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} 2N_r \frac{d_{\max}}{r} + \frac{2Kr_0}{d_{\max}} \right) \\
& \leq 3H^2 + 6\sqrt{2H^3K \log(4HK/\delta)} \\
& \quad + 96H \left(\sqrt{H^3 \log(4HK/\delta)} + Ld_{\max} \right) \inf_{r_0 \in (0, d_{\max}]} \left(\sum_{\substack{r=d_{\max}2^{-i} \\ r \geq r_0}} N_r \frac{d_{\max}}{r} + \frac{Kr_0}{d_{\max}} \right)
\end{aligned}$$

To recover the term established in Theorem 4.1 we simply take $r_0 = O\left(K^{\frac{-1}{d_c+2}}\right)$. \square

F EXPERIMENTAL RESULTS AND FIGURES

F.1 Oil Problem with Quadratic Survey Function

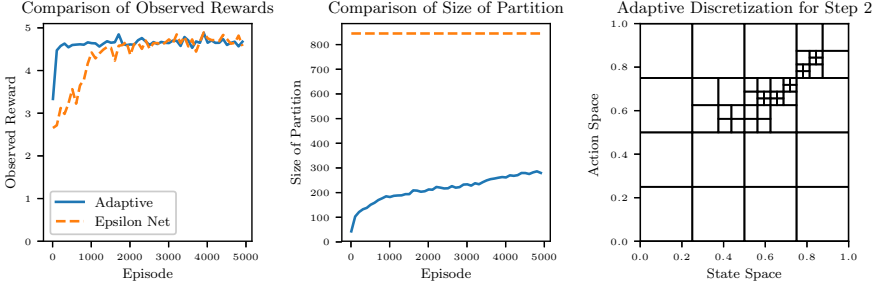


Fig. 4. Comparison of the algorithms on the oil problem with quadratic survey function. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - (a - 0.75)^2 - |x - a|)_+$ (see Section 7.1). The adaptive algorithm creates a fine partition of the space around the optimal point of 0.75.

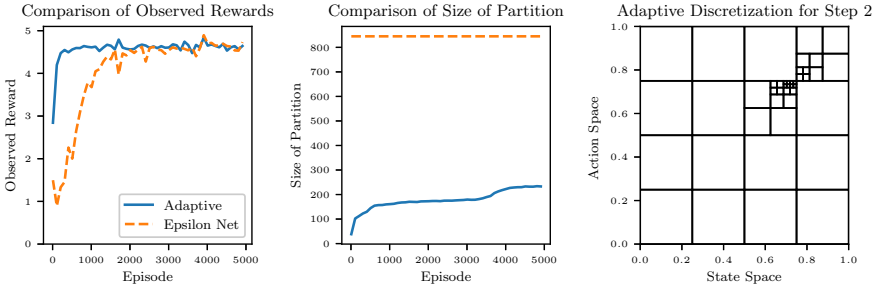


Fig. 5. Comparison of the algorithms on the oil problem with quadratic survey function. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - 10(a - 0.75)^2 - |x - a|)_+$ (see Section 7.1). The epsilon net algorithm suffers by exploring more parts of the space.

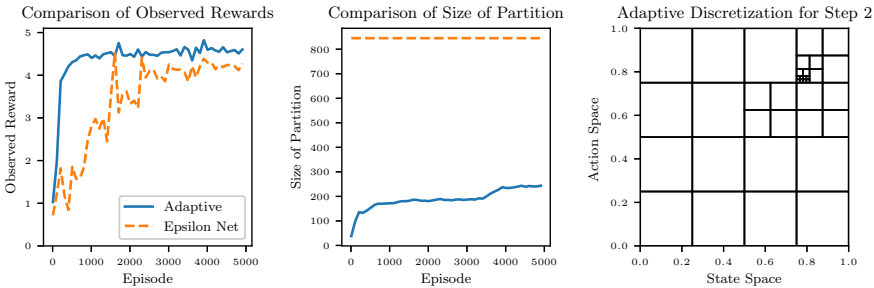


Fig. 6. Comparison of the algorithms on the oil problem with quadratic survey function. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - 50(a - 0.75)^2 - |x - a|)_+$ (see Section 7.1). The epsilon net algorithm suffers a fixed discretization error as the mesh is not fine enough to capture the peak reward.

F.2 Oil Problem with Laplace Survey Function

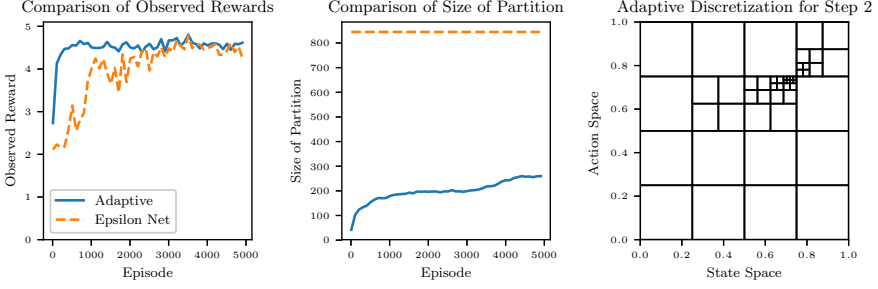


Fig. 7. (Duplicate of Figure 2) Comparison of the algorithms on the oil discovery problem with survey function $f(x) = e^{-|x-0.75|}$. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - e^{-|a-0.75|} - |x - a|)_+$ (see Section 7.1). The adaptive algorithm quickly learns the location of the optimal point 0.75 and creates a fine partition of the space around the optimal.

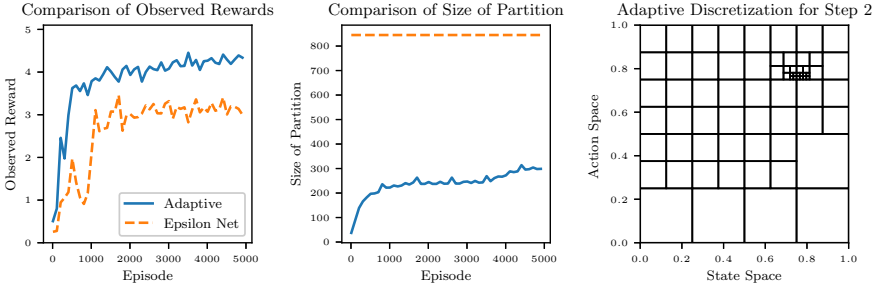


Fig. 8. Comparison of the algorithms on the oil problem with Laplace survey function. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - 10e^{-|a-0.75|} - |x - a|)_+$ (see Section 7.1). The adaptive algorithm learns a fine partition around the optimal point 0.75 while the ϵ -Net algorithm suffers from a large discretization error.

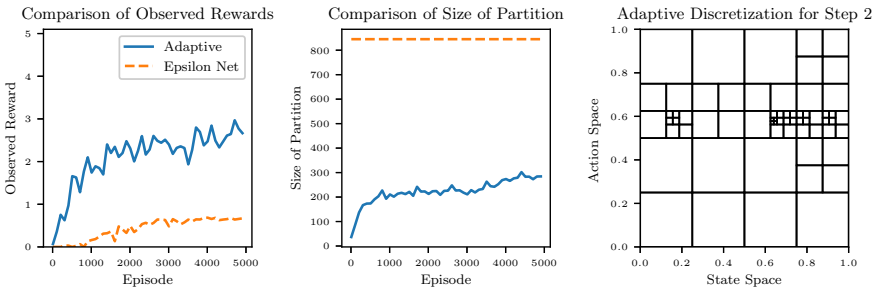


Fig. 9. Comparison of the algorithms on the oil problem with Laplace survey function. The transition kernel is $\mathbb{P}_h(x' | x, a) = \mathbb{1}_{[x'=a]}$ and reward function is $r(x, a) = (1 - 50e^{-|a-0.75|} - |x - a|)_+$ (see Section 7.1). The epsilon net algorithm suffers from a large discretization error as the mesh is not fine enough to capture the high reward region.

F.3 Ambulance Problem with Uniform Arrivals

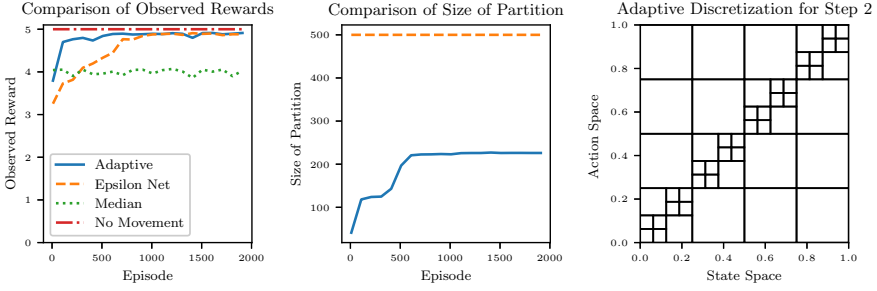


Fig. 10. Comparison of the algorithms on the ambulance problem with Uniform(0, 1) arrivals and reward function $r(x, a) = 1 - |x - a|$ (see Section 7.1). Clearly, the no movement heuristic is the optimal policy but the adaptive algorithm learns a fine partition across the diagonal of the space where the optimal policy lies.

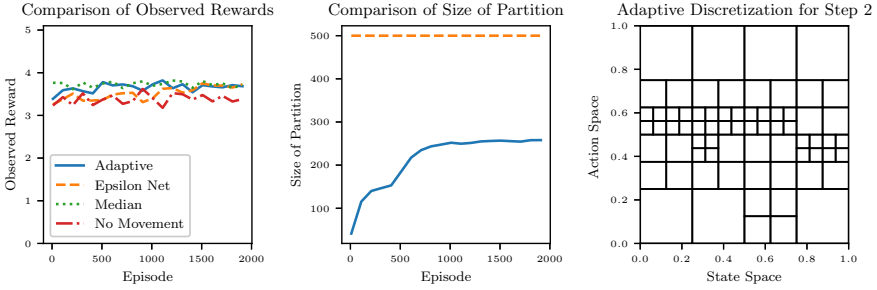


Fig. 11. Comparison of the algorithms on the ambulance problem with Uniform(0, 1) arrival distribution and reward function $r(x', x, a) = 1 - .25|x - a| - .75|x' - a|$ (see Section 7.1). All algorithms perform sub optimally but the adaptive algorithm is able to learn a mixed policy between no movement and traveling to the median.

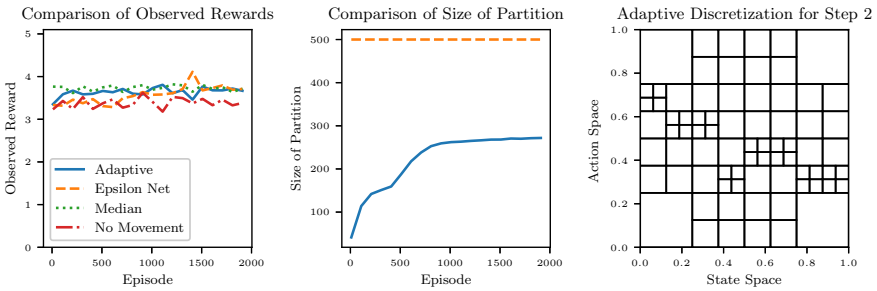


Fig. 12. Comparison of the algorithms on the ambulance problem with Uniform(0, 1) arrival distribution and reward function $r(x', x, a) = 1 - |x' - a|$ (see Section 7.1). The median policy performs best, and the adaptive algorithm is beginning to learn a finer partition around the median of the arrival distribution (0.5).

F.4 Ambulance Problem with Beta Arrivals

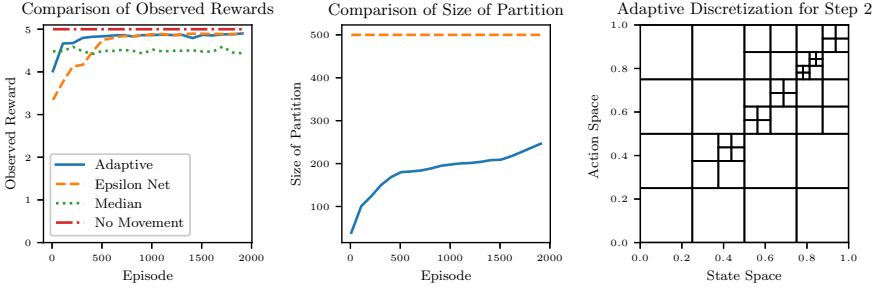


Fig. 13. (Duplicate of Figure 3) Comparison of the algorithms on the ambulance problem with Beta(5, 2) arrival distribution and reward function $r(x, a) = 1 - |x - a|$ (see Section 7.1). Clearly, the no movement heuristic is the optimal policy but the adaptive algorithm learns a fine partition across the diagonal of the space where the optimal policy lies.

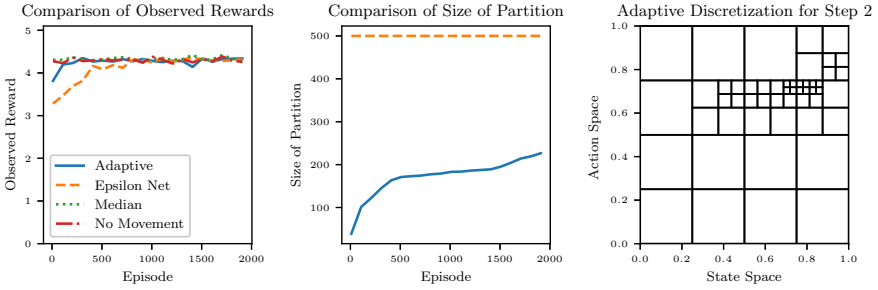


Fig. 14. Comparison of the algorithms on the ambulance problem with Beta(5, 2) arrival distribution and reward function $r(x', x, a) = 1 - .25|x - a| - .75|x' - a|$ (see Section 7.1). All algorithms perform sub optimally but the adaptive algorithm is able to learn a mixed policy of traveling to the median.

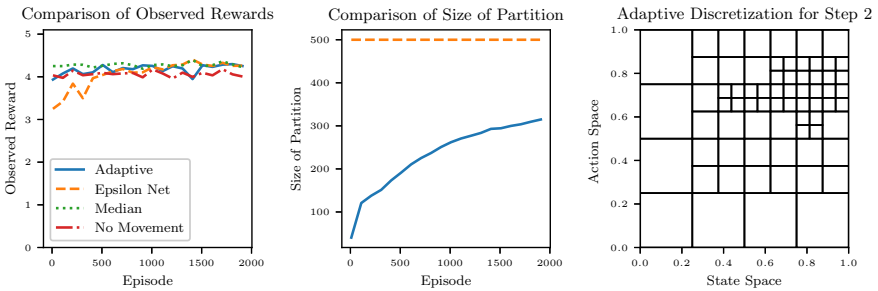


Fig. 15. Comparison of the algorithms on the ambulance problem with Beta(5, 2) arrival distribution and reward function $r(x', x, a) = 1 - |x' - a|$ (see Section 7.1). The median policy performs best, and the adaptive algorithm is beginning to learn a finer partition around the median of the arrival distribution (≈ 0.7).

Received August 2019; revised September 2019; accepted October 2019