


Scaling Up Q-Learning via Exploiting State–Action Equivalence

Yunlian Lyu ^{1,2,*} , Aymeric Côme ³, Yijie Zhang ¹ and Mohammad Sadegh Talebi ¹

¹ Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark; yizh@di.ku.dk (Y.Z.); m.shahi@di.ku.dk (M.S.T.)

² School of Computer Science and Engineering, University of Electronic Science and Technology of China, Xiyuan Ave., Chengdu 611731, China

³ Inria Rennes, Bretagne Atlantique Campus Universitaire de Beaulieu, Avenue du Général Leclerc, 35042 Rennes, France; aymeric.come@inria.fr

* Correspondence: yunlianmoon@gmail.com

Abstract: Recent success stories in reinforcement learning have demonstrated that leveraging structural properties of the underlying environment is key in devising viable methods capable of solving complex tasks. We study off-policy learning in discounted reinforcement learning, where some equivalence relation in the environment exists. We introduce a new model-free algorithm, called QL-ES (Q-learning with equivalence structure), which is a variant of (asynchronous) Q-learning tailored to exploit the equivalence structure in the MDP. We report a non-asymptotic PAC-type sample complexity bound for QL-ES, thereby establishing its sample efficiency. This bound also allows us to quantify the superiority of QL-ES over Q-learning analytically, which shows that the theoretical gain in some domains can be massive. We report extensive numerical experiments demonstrating that QL-ES converges significantly faster than (structure-oblivious) Q-learning empirically. They imply that the empirical performance gain obtained by exploiting the equivalence structure could be massive, even in simple domains. To the best of our knowledge, QL-ES is the first provably efficient model-free algorithm to exploit the equivalence structure in finite MDPs.

Keywords: reinforcement learning; Markov decision process; Q-learning; equivalence structure



Citation: Lyu, Y.; Côme, A.; Zhang, Y.; Talebi, M.S. Scaling Up Q-Learning via Exploiting State–Action Equivalence. *Entropy* **2023**, *25*, 584. <https://doi.org/10.3390/e25040584>

Academic Editor: Adam Lipowski

Received: 31 January 2023

Revised: 1 March 2023

Accepted: 1 March 2023

Published: 29 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reinforcement learning (RL) aims to develop computer systems with the ability to learn how to behave optimally, or nearly so, in an unknown dynamic environment. An RL task typically involves an agent interacting with the environment, which is often modeled as a Markov decision process (MDP), and the agent’s goal is to find a policy maximizing some notion of reward. In most settings in RL, the MDP is initially unknown beyond its state and action spaces. Hence, the agent aims to learn a near-optimal policy using the experiences collected from the environment.

A classical setting in RL is off-policy learning [1], where one tries to learn the optimal action–value function (i.e., Q-function) through the data collected under some *behavior* or *logging* policy. Perhaps the most famous off-policy learning algorithm is the celebrated Q-learning algorithm [2], whose improved variants, combined with deep neural networks as function approximators, played key roles in many recent breakthroughs in RL [3,4]. More precisely, Q-learning and its variants fall under the category of *model-free* methods, in which one tries to directly estimate the optimal value function—without estimating the true model (MDP)—from the collected experience, from which a near-optimal policy could be straightforwardly derived. This approach is in stark contrast to the *model-based* counterpart, where one first attempts to estimate the unknown model parameters (i.e., MDP parameters that include transition probabilities and rewards) from the collected experience and then finds an optimal policy in the estimated model.

Off-policy learning in finite MDPs is by now well understood, and the existing literature (e.g., [5–9]) exhibit algorithms that admit PAC-type sample complexity guarantees.

Precisely speaking, these algorithms are guaranteed to return a near-optimal policy, with respect to a prescribed accuracy, with high probability if the amount of collected experience exceeds a certain (algorithm-dependent) function of the MDP parameters (and relevant input parameters). Most of these works study unstructured tabular MDPs, where the advertised sample complexity bounds scale, among other things, with the size of the state–action space. Thus, despite their appealing performance guarantees, most of these algorithms only work reasonably well when the size of the underlying MDP is small. On the other hand, many practical tasks can be modeled by MDPs with huge state spaces (or even infinite), but they often exhibit some structural properties. Ignoring such structural properties and directly applying the above algorithms would lead to a prohibitively large sample complexity bound, which may imply a huge learning phase in the worst case. Alternatively, one could leverage the structure in the MDP to speed up the exploration. In fact, exploiting the structure allows the agent to use the collected observations from the environment to reduce the uncertainty in the model parameter for many *similar* state–action pairs at each time slot. As a result, the learning performance would depend on the effective size of the state space (or the effective number of unknown parameters). Various notions of structures have been studied in MDPs, which include the Lipschitz continuity of MDP parameters (e.g., rewards and transition functions) [10–13], factorization structure [14–16], and equivalence relations [17–22]. These works reveal that exploiting the underlying structure in the environment in various RL tasks leads to massive empirical performance gain (over structure-oblivious algorithms) and to significantly improved performance bounds. However, exploiting structure often poses additional challenges.

This work is motivated by tabular RL problems, where the (potentially large) state–action space admits a natural partitioning such that within each element of the partition (or class), the state–action pairs have *similar* transition probabilities. There exist several ways to characterize the similarity between the transition distribution of two state–action pairs. Here, we consider a notion implying that they are (almost) identical up to some permutation. As we shall see in later sections, this notion of structure induces an equivalence relation in the state–action space. This model has been considered in prior work [18,23,24], where model-based algorithms were presented to exploit such a structure in the context of regret minimization in episodic or average-reward MDPs. However, their proposed ideas and techniques are specific to a model-based approach, where the model parameters are directly estimated, and cannot be used to incorporate the knowledge of the equivalence structure into a model-free algorithm. Model-free algorithms have played a pivotal role in the recent success of RL to solving complex tasks arising in real-world applications (e.g., autonomous driving and continuous control [25]). Hence, it sounds promising to study the gain one could obtain using model-free methods when leveraging the equivalence structure, thereby extending the theoretical analysis in [18,24] beyond model-based methods.

Contributions. We make the following contributions. We study off-policy learning in discounted finite MDPs, admitting some equivalence structure in their state–action space. We introduce a new model-free algorithm, called QL-ES (Q-learning with equivalence structure), which is a variant of (asynchronous) Q-learning tailored to exploit the equivalence structure in the MDP, when a prior knowledge on the structure is provided to the agent. We report a non-asymptotic PAC-type sample complexity bound for QL-ES, thereby establishing its sample efficiency. This bound also allows us to quantify the superiority of QL-ES over Q-learning analytically. As it turns out, the sample efficiency gain of QL-ES over Q-learning is captured by an MDP-dependent quantity ζ that is defined in terms of the associated covering times in the MDP; see Section 5 for details. Analytically establishing the dependence of the gain ratio ζ on the number S of states in a given MDP seems difficult, although it is possible to numerically compute it. Nonetheless, we present a simple example where $\zeta = O(S)$, thus showcasing that QL-ES in some domains may require *much* fewer (by a factor of S) samples than Q-learning. Furthermore, we numerically compute ζ for a few families of MDPs built using standard environments (with increasing S), thereby showcasing the theoretical superiority of QL-ES over Q-learning. Through extensive nu-

merical experiments on standard domains, we show that Q-function estimates under QL-ES converge much faster than those obtained from (structure-oblivious) Q-learning. These results demonstrate that the empirical performance gain from exploiting the equivalence structure could be massive, even in simple domains. To our best knowledge, QL-ES is the first provably efficient model-free algorithm to exploit the equivalence structure in MDPs.

2. Related Work

Similarity and equivalence in MDPs. There is a rich literature on learning and exploiting various notions of structure in MDPs, where the aim is to leverage structure to alleviate the computational cost of finding an optimal policy (in the known MDP setting) or to speed up exploration (in the RL setting). Many such algorithms fall into the category of state abstraction (or aggregation) [26,27]. Approximate homomorphism is proposed to construct beneficial abstract models in MDPs [28]. In the known MDP setting, Refs. [29,30] appear to be the first presenting the notion of equivalence between states based on *stochastic bi-simulation*. The authors of [31,32] use *bi-simulation metrics* as quantitative analogues of the equivalence relations to partition the state space by capturing similarities. In the RL setting, Refs. [18–20,33,34] investigate *model-based* algorithms that rely on the grouping of similar states (or state–action pairs) to speed up exploration. Ref. [20] is the first to present an average-reward RL algorithm (in the regret setting), where the confidence intervals of similar states are aggregated. Ref. [18] studies regret minimization in average-reward MDPs with equivalence structure and presents the C-UCRL algorithm, which is capable of exploiting the structure. The regret bound for C-UCRL depends on the number of classes in the MDP rather than the size of the state–action space. A similar equivalence structure was studied in [17] in the context of multi-task RL, where similarities of the transition dynamics across tasks were extracted and exploited to speed up learning. Ref. [24] studies the efficiency of hierarchical RL in the regret setting in scenarios where the hierarchical structure is defined with respect to the notion of equivalence; more precisely, it assumes that the underlying MDP can be decomposed into *equivalent* sub-MDPs—i.e., smaller MDPs with identical reward and transition functions up to some known bijection mappings. Closest to our work, in terms of the structure definition, is [18]. However, we restrict ourselves to a model-free approach where the model-based machinery presented in [18] does not apply. Finally, we mention that there is some literature on exploiting equivalence in deep RL (e.g., [21,22]). However, none of these works study provably efficient learning methods to our best knowledge.

Q-learning and its variants. We provide a very brief overview of the works studying theoretical analysis of Q-learning and its variants. Q-learning [2] has been around for more than three decades as a cheap and popular model-free method to solve finite, unknown discounted MDPs without estimating the model. Its convergence was investigated in an asymptotic flavor [35,36], and more recently in the non-asymptotic (finite-sample) regime in a series of work, including [9,37–40]. To the best of our knowledge, Ref. [9] reports the sharpest PAC-type sample complexity bound for the classical Q-learning. Some of these works present variants of Q-learning with improved sample complexity bounds using a variety of techniques, such as acceleration and variance reduction [9,40,41]. Although the concept of equivalence in MDPs is not new, there is no work reporting PAC-type sample complexity bounds for model-free algorithms combined with equivalence relations, to our knowledge.

3. Problem Formulation

In this section, we present some necessary background and formulate the reinforcement learning problem considered in this paper. We use the following notations throughout. For a set B , $\Delta(B)$ denotes the set of all probability distributions over B . For an event E , $\mathbb{I}E$ denotes the indicator function of E : namely, it equals 1 if E holds, and 0 otherwise.

3.1. Discounted Markov Decision Processes

Let $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ be an infinite-horizon discounted MDP [42], where \mathcal{S} denotes a discrete state space with cardinality S , \mathcal{A} denotes a discrete action space with cardinality A , and $\gamma \in (0, 1)$ is a discount factor. $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ represents the transition function such that $P(s'|s, a)$ denotes the probability of transiting to state s' when action $a \in \mathcal{A}$ is chosen in state $s \in \mathcal{S}$. Further, $R : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ denotes the reward function supported on $[0, 1]$ such that $R(s, a)$ denotes the reward distribution when choosing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. We denote by $\bar{R}(s, a)$ the mean of $R(s, a)$. A stochastic (or randomized) policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ is a mapping that maps a state to a probability distribution over \mathcal{A} . For a policy π , the value function of π is a mapping $V^\pi : \mathcal{S} \rightarrow \mathbb{R}$ defined as

$$V^\pi(s) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right], \quad s \in \mathcal{S},$$

where for all $t \geq 0$, $a_t \sim \pi(s_t)$, $s_{t+1} \sim P(\cdot | s_t, a_t)$, and $r_t \sim R(s_t, a_t)$, and where the expectation is taken with respect to the randomness in rewards, next states, and actions sampled from π . The action-value function of a policy π , denoted by $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, is defined as

$$Q^\pi(s, a) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a \right], \quad s \in \mathcal{S}, a \in \mathcal{A}.$$

The optimal value function is denoted by V^* and satisfies $V^*(s) = \max_{\pi} V^\pi(s)$ for all $s \in \mathcal{S}$. It is well known that in any finite MDP, there exists a stationary deterministic policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ such that $V^{\pi^*} = V^*$, which is called an optimal policy [42]. Similarly, the optimal state-action value function is defined as $Q^*(s, a) = \max_{\pi} Q^\pi(s, a)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. An optimal policy π^* satisfies $\pi^*(s) \in \arg \max_a Q^*(s, a)$ for all $s \in \mathcal{S}$. Furthermore, Q^* is the unique solution to the optimal Bellman equation [42]:

$$Q^*(s, a) = \bar{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s' | s, a) \max_{a' \in \mathcal{A}} Q^*(s', a'), \quad s \in \mathcal{S}, a \in \mathcal{A}.$$

3.2. The Off-Policy Learning Problem and Q-Learning

We consider the off-policy learning problem as follows. The agent is provided with some dataset \mathcal{D} collected according to some *behavior or logging* policy π_b . Precisely speaking, \mathcal{D} takes the form of trajectory $\{(s_t, a_t, r_t)\}_{t \geq 0}$, where s_0 is some initial state, and where for each $t \geq 0$, $a_t \sim \pi_b(s_t)$, $s_{t+1} \sim P(\cdot | s_t, a_t)$, and $r_t \sim R(s_t, a_t)$. The agent is given an accuracy parameter $\varepsilon > 0$ and a failure probability parameter $\delta \in (0, 1)$, and its goal is to find an ε -optimal policy using as few samples as possible from \mathcal{D} .

We need to impose some assumptions on the behavior policy π_b to ensure that it is possible to learn a near-optimal policy only using \mathcal{D} efficiently with PAC-type guarantees. To state the assumptions, we introduce some necessary definitions, which are borrowed from standard textbooks on Markov chains (e.g., [43]) but are also standard in the theoretical analysis of Q-learning (e.g., [9,39]). Let \mathcal{X} be a finite set. The *total variation distance* between two distributions μ and ν defined over \mathcal{X} is $d_{TV}(\mu, \nu) := \frac{1}{2} \sum_{x \in \mathcal{X}} |\mu(x) - \nu(x)|$. Now, consider an ergodic Markov chain $(X_t)_{t \geq 1}$ with state space \mathcal{X} and transition function $p \in \Delta(\mathcal{X})$, and let μ be the unique stationary distribution of the chain. The Markov chain is said to be *uniformly ergodic* if there exist some $\rho < 1$ and $M < \infty$ such that for all $t > 0$,

$$\sup_{x \in \mathcal{X}} d_{TV}(\mu, p^t(\cdot | x)) \leq M\rho^t,$$

where $p^t(\cdot | x)$ is the distribution of X_t given $X_0 = x$.

Similar to [9], we assume that the Markov chain induced by π_b is uniformly ergodic. This property ensures that all the states are visited infinitely often, and that convergence to the stationary distribution is performed at a geometric pace. This property is needed for the result presented in Section 5.

The Q-learning algorithm. The Q-learning algorithm [35] is perhaps the most famous model-free algorithm for learning an optimal policy in unknown tabular MDPs. As a model-free method, it directly learns the optimal Q-function Q^* of the MDP (without estimating P and R), which can be used to derive a policy. The algorithm maintains an estimate Q_t of the optimal Q^* at each time step t . Specifically, it starts from an arbitrary choice for $Q_0 \in \mathbb{R}^{S \times A}$ and updates Q_t , at each $t \geq 0$, as

$$Q_{t+1}(s_t, a_t) = \begin{cases} (1 - \alpha_t)Q_t(s, a) + \alpha_t \left(r_t + \gamma \max_{a' \in \mathcal{A}} Q_t(s_{t+1}, a') \right), & (s, a) = (s_t, a_t), \\ Q_{t+1}(s, a) = Q_t(s, a), & (s, a) \neq (s_t, a_t), \end{cases} \quad (1)$$

where α_t is a suitably chosen learning rate. Precisely speaking, the update Equation (1) corresponds to the asynchronous variant of Q-learning. The classical asymptotic performance analysis of Q-learning (in, for example, [36]) indicates that if (i) π_b is exploratory enough such that all state–action pairs are visited infinitely often and (ii) $(\alpha_t)_{t \geq 0}$ satisfies the following conditions, known as the Robbins–Monro conditions [36,44]:

$$\alpha_t \geq 0, \quad \sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty,$$

then $Q_t \xrightarrow{t \rightarrow \infty} Q^*$ almost surely, for any choice of $Q_0 \in \mathbb{R}^{S \times A}$. For example, one such choice of the learning rate is as follows: for all $t \geq 0$, $\alpha_t = \frac{1}{N_t(s_t, a_t) + 1}$, where for any (s, a) , $N_t(s, a)$ denotes the number of visits to (s, a) up to time t : $N_t(s, a) = \sum_{\tau=0}^{t-1} \mathbb{I}\{(s_\tau, a_\tau) = (s, a)\}$. The pseudo-code of Q-learning is described in Algorithm 1, where the learning rate sequence $(\alpha_t)_{t \geq 0}$ is considered as input.

Algorithm 1 Q-learning [2].

Input: dataset \mathcal{D} , maximum iterations T , learning rates $(\alpha_t)_{t \geq 0}$
Initialization: $Q_0 = 0 \in \mathbb{R}^{S \times A}$
for $t = 0, 1, \dots, T$ **do**
 Sample action $a_t \sim \pi_b(s_t)$ and observe $r_t \sim R(s_t, a_t)$ and $s_{t+1} \sim P(\cdot | s_t, a_t)$.
 Compute Q_{t+1} using (1).
end for

It is worth remarking that some studies consider off-policy learning in the online setting, where data are collected from the environment while executing the algorithm. In such online settings, it is possible to choose a_t according to an adaptive (randomized) policy π_t (usually defined as a function of the current estimate Q_t), instead of sampling it from a fixed behavior policy. In doing so, the aim is to balance exploration and exploitation so as to collect higher rewards while learning the Q-function. A notable example is the ϵ -greedy policy, where at time t , a_t is chosen greedily with respect to $Q_t(s_t, \cdot)$ with probability $1 - \epsilon$, and chosen uniformly at random (from \mathcal{A}) with probability ϵ . In the theoretical part of this paper, we consider a fixed behavior policy.

3.3. Similarity and Equivalence Classes

We now present a definition of the equivalence structure considered in this paper. We start by stating the following definition of similarity in finite MDPs as introduced in [18]. A similar definition is provided in [23].

Definition 1 (Similar state–action pairs [18]). *Two state–action pairs (s, a) and (s', a') are called θ -similar if there exist mappings $\sigma_{s,a} : \{1, \dots, S\} \rightarrow \mathcal{S}$ and $\sigma_{s',a'} : \{1, \dots, S\} \rightarrow \mathcal{S}$ such that*

$$\|P(\sigma_{s,a}(\cdot) | s, a) - P(\sigma_{s',a'}(\cdot) | s', a')\|_1 \leq \theta.$$

We refer to $\sigma_{s,a}$ as the **profile mapping** (or for short, **profile**) for (s, a) , and denote by $\sigma = (\sigma_{s,a})_{s,a}$ the set of profile mappings across $\mathcal{S} \times \mathcal{A}$.

We stress that $\sigma_{s,a}$ in Definition 1 may not be unique in general. The case of 0-similarity is of particular interest: it is evident from Definition 1 that if (s, a) and (s', a') are 0-similar, then $P(\cdot|s, a)$ and $P(\cdot|s', a')$ are identical up to some permutation from $\mathcal{S} \times \mathcal{A}$ to $\mathcal{S} \times \mathcal{A}$. Furthermore, 0-similarity induces a partition of the state–action space $\mathcal{S} \times \mathcal{A}$ as formalized below.

Definition 2 (Equivalence structure [18]). 0-similarity is an equivalence relation and induces a canonical partition of $\mathcal{S} \times \mathcal{A}$. We refer to such a canonical partition as **equivalence structure** and denote it by \mathcal{C} . We further define $C := |\mathcal{C}|$.

We provide an example to help understand Definition 2. Consider the RiverSwim environment [45] with 6 states and $\mathcal{A} = \{L, R\}$ (see Figure 1). The two pairs (s_1, R) and (s_6, R) are 0-similar since $P(\cdot|s_1, R) = [0.6, 0.4, 0, 0, 0, 0]$ and $P(\cdot|s_6, R) = [0, 0, 0, 0, 0.6, 0.4]$, so there exist permutations $\sigma_{s_1, R}$ and $\sigma_{s_6, R}$ such that $P(\sigma_{s_1, R}(\cdot)|s_1, R) = P(\sigma_{s_6, R}(\cdot)|s_6, R)$. Additionally, all pairs $(s_i, L), i = 1, \dots, 6$ are 0-similar, and so are $(s_i, R), i = 2, \dots, 5$. We thus identify an equivalence structure \mathcal{C} of $\mathcal{S} \times \mathcal{A}$ as follows: $\mathcal{C} = \{c_1, c_2, c_3\}$ with $c_1 = \{(s_1, R), (s_6, R)\}$, $c_2 = \{(s_i, R), i = 2, \dots, 5\}$, and $c_3 = \{(s_i, L), i = 1, \dots, 6\}$.

Note that for any finite MDP, Definition 2 trivially holds with $\mathcal{C} = \mathcal{S} \times \mathcal{A}$. There are many interesting environments that non-trivially admit the notion of equivalence structure in Definition 2. In such MDPs, it is often the case that the size C of the structure is much smaller than the size of the state–action space, i.e., $C \ll SA$. For example, in a generic RiverSwim with S states, one has $C = 3$. Another example admitting an equivalence structure is the classical grid-world MDP, which is detailed in Section 6.2.

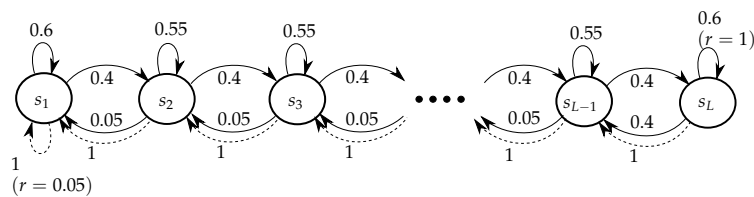


Figure 1. The L -state RiverSwim environment [45].

Off-policy learning in MDPs with equivalence structures. In this work, we assume that the underlying MDP M admits an equivalence structure \mathcal{C} as introduced above. In other words, the transition function P is such that $\mathcal{S} \times \mathcal{A}$ can be partitioned into $C := |\mathcal{C}|$ classes, where the pairs in each class $c \in \mathcal{C}$ are 0-similar. We make the following assumption regarding the agent’s prior knowledge about \mathcal{C} .

Assumption 1. The agent has prior knowledge on \mathcal{C} .

Let $c(s, a)$ denote the class that the pair (s, a) belongs to. Assumption 1 implies that the agent knows $c(s, a)$ for any pair (s, a) and the associated profile mapping $\sigma_{s,a}$. Note, however, that the agent does not know the actual transition probabilities. Armed with such prior knowledge, we are interested in devising a model-free algorithm that is capable of leveraging the structure in M to improve the learning performance. We expect that the corresponding speed-up in learning the optimal Q-function could be significant in MDPs with $C \ll SA$.

We also make the following assumption regarding the reward function to ease the presentation. (This assumption has often been made in the literature on theoretical RL (e.g., [6,46]) since the main challenge in RL arises from unknown transition probabilities.)

Assumption 2. The agent knows the reward function R .

4. The QL-ES Algorithm

In this section, we present a variant of Q-learning that exploits the equivalence structure in the environment to speed up the learning of the optimal Q-function. We call this algorithm QL-ES, which is short for ‘Q-learning with equivalence structure’.

QL-ES follows the same machinery of QL but is also built on the idea that the knowledge on \mathcal{C} and the corresponding profile mappings allows for using the triplet (s_t, a_t, s_{t+1}) collected at each time t to update potentially multiple entries of Q_t . Precisely speaking, the Q-function update for a given pair (s, a) requires a sample from $R(s, a)$ and a sample from $P(\cdot|s, a)$. Since the agent perfectly knows \mathcal{C} , it can determine $c(s_t, a_t)$, i.e., the class (s_t, a_t) belongs to. Hence, it knows all other pairs belonging to the same class as (s_t, a_t) . Then using s_{t+1} (sampled from $P(\cdot|s_t, a_t)$) at time t , the agent can construct samples for all other pairs in $c(s_t, a_t)$ as follows. If $(s, a) \in c(s_t, a_t)$, then there is a mapping $\sigma_{s,a}$ and a state $s_{t+1}^{(sa)}$ such that $P(\sigma_{s_t, a_t}(s_{t+1})|s_t, a_t) = P(\sigma_{s,a}(s_{t+1}^{(sa)})|s, a)$. In other words, the sample s_{t+1} obtained from $P(\cdot|s_t, a_t)$ is equivalent to obtaining a fresh sample $s_{t+1}^{(sa)}$ from $P(\cdot|s, a)$. As $\sigma_{s,a}$ and σ_{s_t, a_t} are known, the agent finds $s_{t+1}^{(sa)} := \sigma_{s,a}^{-1}(\sigma_{s_t, a_t}(s_{t+1}))$, where σ^{-1} denotes the inverse mapping of σ . In other words, $s_{t+1}^{(sa)}$ acts as a *counterfactual next-state* for $(s, a) \in c(s_t, a_t)$ thanks to the knowledge on \mathcal{C} .

The agent thus uses (s_t, a_t, s_{t+1}) to update Q_t for all $(s, a) \in c(s_t, a_t)$. In summary, we update Q_t as follows: For all $t \geq 0$,

$$Q_{t+1}(s_t, a_t) = \begin{cases} (1 - \alpha_t)Q_t(s, a) + \alpha_t \left(\bar{R}(s, a) + \gamma \max_{a' \in \mathcal{A}} Q_t(s_{t+1}^{(sa)}, a') \right), & (s, a) \in c(s_t, a_t), \\ Q_{t+1}(s, a) = Q_t(s, a), & (s, a) \notin c(s_t, a_t), \end{cases} \quad (2)$$

where $(\alpha_t)_{t \geq 0}$ is a sequence of suitably chosen learning rates, as in (1). The pseudo-code of QL-ES is provided in Algorithm 2.

Algorithm 2 QL-ES

Input: dataset \mathcal{D} , maximum iterations T , learning rates $(\alpha_t)_{t \geq 0}$, equivalence structure \mathcal{C}
Initialization: $Q_0 = 0 \in \mathbb{R}^{S \times A}$.
for $t = 0, 1, 2, \dots, T$ **do**
 Sample action $a_t \sim \pi_b(s_t)$ and observe $s_{t+1} \sim P(\cdot|s_t, a_t)$.
 Find $c(s_t, a_t)$.
 for $(s, a) \in c(s_t, a_t)$ **do**
 $s_{t+1}^{(sa)} = \sigma_{s,a}^{-1}(\sigma_{s_t, a_t}(s_{t+1}))$
 Compute Q_{t+1} using (2)
 end for
end for

When the underlying MDP admits some equivalence structure, QL-ES performs multiple updates of Q-function at any slot, in contrast to structure-oblivious Q-learning that updates only the Q-function of the current state–action pair. Thus, we expect learning the optimal Q-function under QL-ES to be faster than Q-learning; this will be corroborated by the numerical experiments in Section 6. It is also worth mentioning that QL-ES is never worse than Q-learning, as for the trivial partition $\mathcal{C} = \mathcal{S} \times \mathcal{A}$, which holds for any finite MDP, QL-ES reduces to Q-learning.

Remark 1. The multiple updates used in QL-ES can be straightforwardly combined with many other variants of Q-learning, such as Speedy Q-learning [46] and UCB-QL [41].

We finally remark that some works in the literature on Q-learning use learning rates of the form $\alpha_t = f(N_t(s_t, a_t))$, where $N_t(s, a) = \sum_{\tau=0}^{t-1} \mathbb{I}\{(s_\tau, a_\tau) = (s, a)\}$ and where f is some suitable function f satisfying the Robbins–Monro conditions, e.g., $\alpha_t = \frac{1}{N_t(s_t, a_t) + 1}$. Such

learning rates in the case of QL-ES can be modified to $\alpha_t = f(N_t(c(s_t, a_t)))$, where for any $c \in \mathcal{C}$, $N_t(c) := \sum_{\tau=0}^{t-1} \mathbb{I}\{(s_\tau, a_\tau) \in c\}$.

5. Theoretical Guarantee for QL-ES

In this section, we investigate the theoretical guarantee of QL-ES in terms of sample complexity in the PAC setting. Specifically, we are interested in characterizing the deviation between the optimal Q-function Q^* and its estimate Q_T computed by QL-ES after T time steps. A relevant notion of deviation often studied in the literature (see, e.g., [37,38]) is the ℓ_∞ -distance between Q_T and Q^* :

$$\|Q^* - Q_T\|_\infty = \max_{s,a} |Q^*(s, a) - Q_T(s, a)| \tag{3}$$

which captures the worst error (with respect to Q^*) among various pairs. One may study the rate at which the error function $\|Q^* - Q_T\|_\infty$ decays as a function of T . Alternatively, one may characterize the PAC sample complexity defined as the number T of steps needed until Q_T satisfies $\|Q^* - Q_T\|_\infty \leq \epsilon$ with probability at least $1 - \delta$, for pre-specified ϵ and δ . We consider the latter case.

Let us first recall the classic definition of cover time t_{cover} , which is a standard notion in the literature on Markov chains as well as those studying theoretical guarantees of Q-learning (and its variants) [9,38,39]. Let $t_1 \geq 0$ and let $t_2 > t_1$ denote the first time step such that all state–action pairs are visited at least once with probability at least $\frac{1}{2}$. Then, the cover time t_{cover} is defined as the maximum value of $t_2 - t_1$ over all initial pairs (s_{t_1}, a_{t_1}) . Note that t_{cover} depends on both the MDP M and the behavior policy π_b . More precisely, it depends on the mixing properties of the Markov chain induced by π_b on M . Further, we have $t_{\text{cover}} \geq SA$.

Next, we introduce a notion of cover time for equivalence classes, which is relevant to the performance analysis of QL-ES. We believe it can be of independent interest.

Definition 3. Let M be a finite MDP and \mathcal{C} be an equivalence structure in M . Given $t_1 \geq 0$, let $t_2 > t_1$ denote the first time step such that for each $c \in \mathcal{C}$, some state–action pair in c is visited at least once with probability at least $\frac{1}{2}$. Then, the cover time with respect to the equivalence structure \mathcal{C} in M , denoted by $t_{\text{cover},\mathcal{C}}$, is defined as the maximum value of $t_2 - t_1$ over all initial choices of $c(s_{t_1}, a_{t_1})$ (i.e., the class the initial pair (s_{t_1}, a_{t_1}) belongs to).

The following theorem provides a non-asymptotic sample complexity for QL-ES. It concerns constant learning rates, i.e., $\alpha_t = \alpha$ for all $t \geq 0$, where α may depend on ϵ and δ , among other things.

Theorem 1. There exist some universal constants κ_0, κ_1 such that for any $\delta \in (0, 1)$ and $\epsilon \in (0, \frac{1}{1-\gamma}]$, we have $\|Q^* - Q_T\|_\infty \leq \epsilon$ with probability greater than $1 - \delta$, provided that the number T of steps and learning rate α jointly satisfy

$$T \geq \frac{\kappa_0 t_{\text{cover},\mathcal{C}}}{(1-\gamma)^5 \epsilon^2} \log^2\left(\frac{CT}{\delta}\right) \log\left(\frac{1}{(1-\gamma)^2 \epsilon}\right), \quad \alpha = \min\left\{\frac{\kappa_1 (1-\gamma)^4 \epsilon^2}{\gamma^2 \log(CT/\delta)}, \frac{1}{2}\right\}.$$

A proof of this theorem is provided in Appendix A. Our proof is an adaptation of the of the proof of Theorem 2 in [9], which concerns the sample complexity of Q-learning.

Comparison with sample complexity of Q-learning. Theorem 1 tells us that the number of steps to have $\|Q^* - Q_T\|_\infty \leq \epsilon$ with high probability depends on $t_{\text{cover},\mathcal{C}} \epsilon^{-2} (1-\gamma)^{-5}$ (up to some logarithmic factors), where $t_{\text{cover},\mathcal{C}}$, defined in Definition 3, is the cover time with respect to \mathcal{C} . Comparing this result against the sample of complexity of Q-

learning (e.g., Theorem 2 in [9]) reveals that using QL-ES yields an improvement over Q-learning by a factor of $\text{const.} \times \zeta$, where

$$\zeta := \zeta(M, \mathcal{C}, \pi_b) := \frac{t_{\text{cover}}}{t_{\text{cover}, \mathcal{C}}}.$$

This ratio ζ is a problem-dependent constant (depending on both M and (\mathcal{C}, σ)). It also depends on the behavior policy π_b in view of the definitions of the cover times. It is evident that $\zeta \geq 1$ for any choice of M and \mathcal{C} . For a given MDP M , the ratio $\zeta(M, \mathcal{C}, \pi_b)$ can be numerically computed; we report numerical values of ζ for several domains in Section 6.3. On the other hand, deriving the analytical bounds on the ratio $\zeta(M, \mathcal{C}, \pi_b)$ for any M appears to be complicated and tedious, if possible at all. Nonetheless, it is possible to construct simple problem instances, where one can derive analytical bounds on ζ .

Figure 2 portrays one such example; this example is a simple Markov chain but can be easily extended to become an MDP. Easy calculations show that $t_{\text{cover}} = (S - 1)/\delta$ and $t_{\text{cover}, \mathcal{C}} = 1$ so that $\zeta = (S - 1)/\delta$. Hence, one here has $\zeta = O(S)$. This simple example demonstrates that the gain of QL-ES over Q-learning in some domains could be as large as $O(S)$, the size of the state space. Additionally, Theorem 1 reveals that in such domains, the theoretical sample complexity bound of QL-ES does not depend on S but on \mathcal{C} , the number of classes in \mathcal{C} .

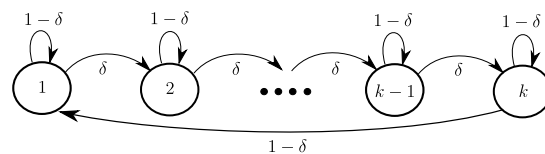


Figure 2. An illustrative example where $\zeta = O(S)$.

We refer the reader to the results in Section 6.3, where we present numerical bounds on ζ in some MDPs, which serve as the standard domain in the RL literature.

6. Simulation Results

This section is devoted to reporting numerical experiments conducted to examine the performance of QL-ES against the (structure-oblivious) Q-learning algorithm. First, we present the considered evaluation metrics and environments. Then we present numerical assessment of ζ for these environments. Finally, we report empirical sample complexities of QL-ES and Q-learning in the environments.

6.1. Evaluation Metrics

We consider two evaluation metrics in the experiments:

- (i) *Max-norm Q-value Error* defined as $\|Q^* - Q_t\|_\infty$;
- (ii) *Total Policy Error* defined as $\|\pi^* - \pi_t^{\text{greedy}}\|_1$, where π_t^{greedy} denotes the greedy policy w.r.t. Q_t , i.e., $\pi_t^{\text{greedy}}(s) := \arg \max_a Q_t(s, a)$ for all s .

The metric (i), which is in line with the definition of sample complexity studied in Section 5, captures the maximum difference between Q_t and Q^* over all state–action pairs and allows us to empirically study the convergence speed of Q_t toward Q^* . The second metric captures the quality of the estimate Q_t in terms of inferred policies. Evidently, the quantity $\|\pi^* - \pi_t^{\text{greedy}}\|_1$ returns the number of states at which π_t^{greedy} prescribes a sub-optimal action. Hence, the metric (ii) may capture how bad the policy derived from Q_t (i.e., π_t^{greedy}) would be, compared to π^* , had we stopped at time step t . Equivalently, we may compute the metric (ii) via

$$\sum_{s \in \mathcal{S}} \mathbb{I} \left\{ Q_t(s, \pi^*(s)) - \max_{a \neq \pi^*(s)} Q_t(s, a) < 0 \right\}. \tag{4}$$

Working with (4) is preferred, as then, one may not worry about how ties (in $\arg \max$) are broken when either π_i or π^* is not unique.

6.2. Environments

We consider two environments: *RiverSwim* and *GridWorld*. These are classical MDPs widely used in the RL literature. Both render suitability to demonstrate the numerical performance of QL-ES since each allows us to define a family of MDPs with progressive difficulty levels.

RiverSwim and variants. A generic RiverSwim MDP with L states is shown in Figure 1, which extends the classical 6-state RiverSwim presented in [45]. This MDP is constructed so that efficient exploration is required to obtain the optimal policy. The larger the number L of states, the more exploration is required. The L -state RiverSwim (with $L \geq 3$) admits an equivalence structure with $C = 3$ regardless of L . We consider RiverSwim instances with various L so as to have MDPs with progressive difficulty levels while having a fixed number of classes. In some experiments, we consider a slightly modified version of RiverSwim, which we shall call *Perturbed RiverSwim*. It is identical to RiverSwim (Figure 1) except that in any state s_i , where $i < L$ is even, $p(s_i|s_i, R) = 0.65$ and $p(s_{i+1}|s_i, R) = 0.3$. It is clear that there are $C = 4$ classes in a L -state Perturbed RiverSwim.

GridWorld. We also consider 2-room and 4-room grid-world MDPs with different grid sizes. Figure 3 shows a 7×7 2-room and a 9×9 4-room grid-worlds, respectively. In both environments, the agent starts at the upper-left corner (in red) and is supposed to reach the lower-right corner (in yellow), where it is given a reward of 1 and then sent back to the initial red state. At each step, the agent has four possible actions (hence, $A = 4$): Going up, left, down, or right. Black squares indicate the wall where the agent is not able to penetrate through. After executing a given action, the agent has a probability of 0.1 to stay in the same state, has a probability of 0.7 to move to the desired direction, and has a probability of 0.06 and 0.14 to move to the other two possible directions. If the wall blocks the agent, it stays where it is, and the transition probability of the next state is added to that of the current state.

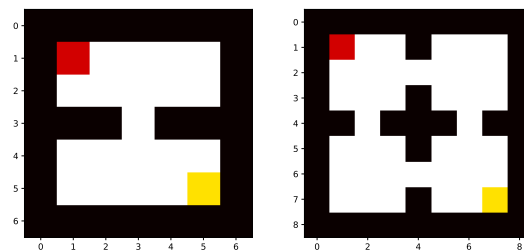


Figure 3. The 2-room grid world (left) and 4-room grid world (right) with walls in black, initial state in red, and goal state in yellow.

It is clear that the grid-world MDPs above admit some equivalence structure. In the case of 2-room (respectively, 4-room), the state–action space is of size 84 (respectively, 160), while the number of classes remains 8 in both. In Table 1, we also present six examples of grid-world environments with walls defined according to the way mentioned above. In the introduced 2-room and 4-room MDPs, the number of state–action pairs changes with the increase in the grid size, while the number of classes remains fixed.

Table 1. State–action space and equivalence classes comparison in grid-world MDPs.

Environment	States	7×7	9×9	11×11	20×20	50×50	100×100
2-room	SA	84	172	292	1228	9028	3.8×10^4
2-room	C	8	8	8	8	8	8
4-room	SA	80	160	272	1172	8852	3.7×10^4
4-room	C	8	8	8	8	8	8

6.3. Bounds on the Ratio ζ

We recall from Section 5 that the theoretical gain of QL-ES over Q-learning in terms of sample efficiency is captured by the problem-dependent quantity $\zeta = \frac{t_{\text{cover}}}{t_{\text{cover},\mathcal{C}}}$. In this subsection, we compute ζ for the introduced environments with the aim of providing insights into the growth of ζ as the number S of states grows. Specifically, we consider RiverSwim, Perturbed RiverSwim (introduced in Section 6.2), and GridWorld MDPs, each with growing number of states. In each case, we report empirical values for t_{cover} and $t_{\text{cover},\mathcal{C}}$ together with the corresponding 95% confidence intervals. The empirical t_{cover} is computed as the median value (across 100 independent runs for every possible initial state-action pair) of the number of steps it takes to discover all state-action pairs starting from a given initial state-action pair. A similar procedure is used for $t_{\text{cover},\mathcal{C}}$.

Tables 2–4 summarize empirical values of t_{cover} and $t_{\text{cover},\mathcal{C}}$ (together with the associated 95% confidence intervals denoted by CI) for RiverSwim, Perturbed RiverSwim, and 2-room GridWorld, respectively, with varying number of states in each case. In the case of GridWorld, we ran a uniform agent (sampling each action uniformly), whereas in RiverSwim MDPs, the agent samples R (resp. L) with probability 0.8 (resp. 0.2).

These results reveal that $t_{\text{cover},\mathcal{C}}$ is much smaller than t_{cover} in all cases. Furthermore, they indicate that while t_{cover} grows rapidly as S increases (in any family of the MDPs considered), $t_{\text{cover},\mathcal{C}}$ experiences a much smaller growth. As for the ratio ζ , we report ζ_{LCB} as the lower confidence bound obtained by dividing the lower value in the CI for t_{cover} by the upper value in the CI for $t_{\text{cover},\mathcal{C}}$. This is a rather conservative estimate of the true ζ but ensures that $\zeta \leq \zeta_{\text{LCB}}$ with probability at least 0.9. The reported values demonstrate that in these environments, ζ grows rapidly as the size of state space grows. This observation verifies that the theoretical gain of QL-ES over Q-learning can be significant.

Table 2. Empirical values of t_{cover} , $t_{\text{cover},\mathcal{C}}$, and ζ_{LCB} for RiverSwim with S states.

S	6	10	14	20
t_{cover}	131, CI = [97, 158]	513, CI = [340, 658]	2529, CI = [1867, 3196]	12792, CI = [7577, 15688]
$t_{\text{cover},\mathcal{C}}$	12, CI = [9, 16]	33, CI = [26, 38]	56, CI = [46, 66]	113, CI = [94, 133]
ζ_{LCB}	$\frac{97}{16} \approx 6.1$	$\frac{340}{38} \approx 8.9$	$\frac{1867}{66} \approx 28.3$	$\frac{7577}{133} \approx 57.0$

Table 3. Empirical values of t_{cover} , $t_{\text{cover},\mathcal{C}}$, and ζ_{LCB} for Perturbed RiverSwim with S states.

S	6	10	14	20
t_{cover}	116, CI = [97, 136]	557, CI = [351, 655]	2011, CI = [1616, 2451]	11856, CI = [7282, 17103]
$t_{\text{cover},\mathcal{C}}$	14, CI = [12, 17]	32, CI = [23, 37]	58, CI = [46, 67]	114, CI = [99, 130]
ζ_{LCB}	$\frac{97}{17} \approx 5.7$	$\frac{351}{37} \approx 9.5$	$\frac{1616}{67} \approx 24.1$	$\frac{7282}{130} \approx 56.0$

Table 4. Empirical values of t_{cover} , $t_{\text{cover},\mathcal{C}}$, and ζ_{LCB} for 2-room GridWorld with S states.

S	21	43	71	111
t_{cover}	2164, CI = [1939, 2355]	5480, CI = [4890, 5889]	11310, CI = [9817, 12735]	22793, CI = [20571, 24882]
$t_{\text{cover},\mathcal{C}}$	205, CI = [125, 251]	418, CI = [329, 481]	877, CI = [613, 1011]	1338, CI = [1101, 1551]
ζ_{LCB}	$\frac{1939}{251} \approx 7.7$	$\frac{4890}{481} \approx 10.2$	$\frac{9817}{1011} \approx 9.7$	$\frac{20571}{1551} \approx 13.3$

6.4. Experimental Results with Exact Equivalence Structure

We now turn to reporting experimental results for QL-ES and Q-learning in RiverSwim and GridWorld. In the following figures, QL indicates the standard Q-learning algorithm (Algorithm 1). We used a constant learning rate $\alpha = 0.05$ and ϵ -greedy policy (with values of ϵ to be specified later). Furthermore, all the results are averaged over 100 independent runs, and the corresponding 95% confidence intervals are shown.

Figure 4 presents the max-norm Q-value error and total policy error under both QL-ES and Q-learning in a 6-state RiverSwim, where we set $\gamma = 0.9$ and $\epsilon = 0.5$. It is evident that QL-ES significantly outperforms Q-learning. The Q-value error under Q-learning decays at

a very slow rate until about 4×10^5 steps. After this step, the decay rate increases tangibly. In contrast, the Q-value error under QL-ES decays at a much faster speed. Under Q-learning, the total policy error remains above 5 until time step 4×10^5 , which implies that only one state has learned its optimal policy. On the contrary, the total policy error under QL-ES drops to the vicinity of 0 very quickly. These results verify that the empirical gain of leveraging the equivalence structures in MDPs, in terms of the number of samples, can be significant.

To demonstrate the scalability of QL-ES, in Figure 5, we present the Q-value error under QL-ES in RiverSwim instances with 6, 20, and 40 states. As the figure shows, although the error in the 6-state RiverSwim starts decaying much earlier than the others, all of them exhibit a similar rate of decay. Moreover, the curves corresponding to 20-state and 40-state instances are almost indistinguishable. This result showcases MDPs where the sample complexity of QL-ES does not scale with the size of the state–action space and is mostly determined by the number of classes.

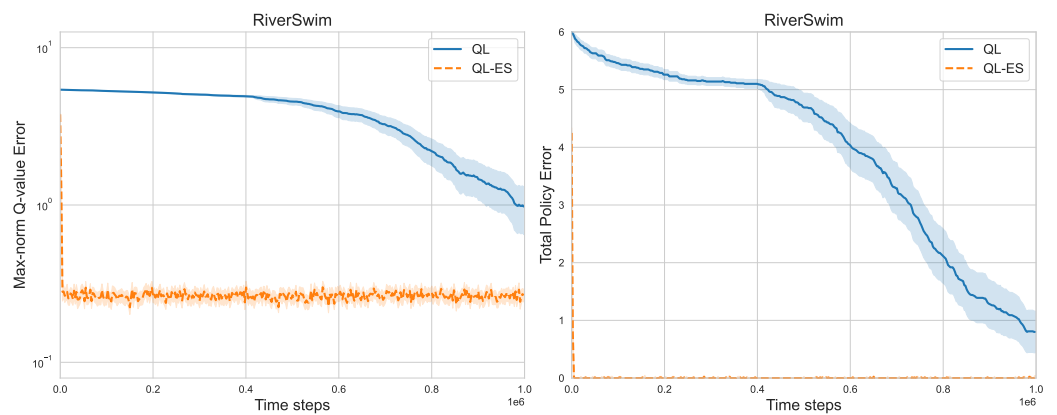


Figure 4. Results in 6-state RiverSwim.

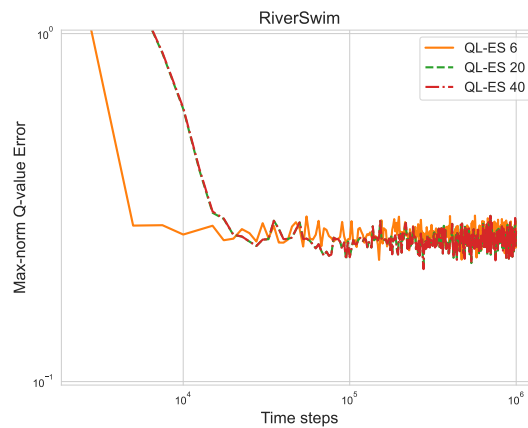


Figure 5. Comparison between RiverSwim domains.

We now turn to the results for GridWorld MDPs. Figures 6 and 7 show the results for QL-ES and Q-learning in 2-room and 4-room GridWorld MDPs, where we used $\gamma = 0.85$ and $\epsilon = 0.2$ in the 2-room and $\gamma = 0.85$ and $\epsilon = 0.3$ in the 4-room.

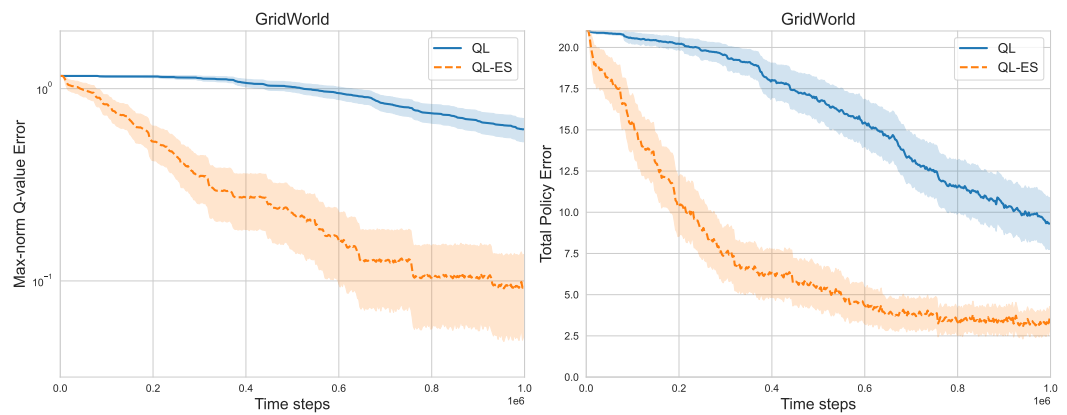


Figure 6. Results in 2-room grid world.

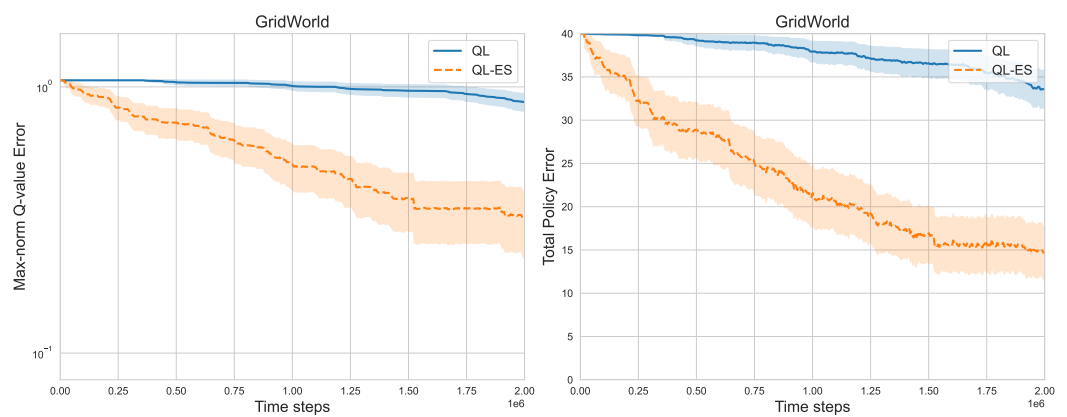


Figure 7. Results in 4-room grid world.

As in RiverSwim MDPs, QL-ES significantly outperforms Q-learning in the grid-world environments. For Q-learning, the Q-value error remains considerable, even for 10^6 samples. Although both QL-ES and Q-learning do not fully learn an optimal policy by the end of the run, the total policy error decays much faster under QL-ES. Overall, the results demonstrate that exploiting equivalence structure is beneficial in grid-world MDPs. Comparing Figures 6 and 7, it is evident that QL-ES still obtains relatively better performance than Q-learning with the increase in state space. Moreover, similar trends are expected when conducting this experiment in larger grid-world MDPs.

6.5. The Gain in the Case of θ -Similar Pairs

We now investigate the case where the MDP may not admit any equivalence structure but admits θ -similarity across its state–action space; see Definition 1. To this effect, we introduce *Modified RiverSwim* and *Modified GridWorld* defined as follows. The *Modified RiverSwim* is identical to 6-state RiverSwim (Figure 1), except that its non-zero transition probabilities under (s_2, R) and (s_4, R) are changed from $[0.05, 0.55, 0.4]$ to $[0.15, 0.3, 0.55]$. The *Modified GridWorld* is identical to 7×7 2-room grid-world, except that the non-zero transition probabilities under (s_1, down) , (s_5, down) , and (s_7, down) are set to $[0.6, 0.05, 0.2, 0.15]$ instead of $[0.7, 0.06, 0.14, 0.1]$. Modified RiverSwim admits an equivalence structure, but we remark that it satisfies θ -similarity with $\theta = 0.1$. Similarly, Modified GridWorld satisfies θ -similarity with $\theta = 0.1$. Figure 8 shows the results in Modified RiverSwim (with $\gamma = 0.95$) and Modified GridWorld ($\gamma = 0.85$). It is evident that in both cases, QL-ES still achieves smaller Q-value error than Q-learning.

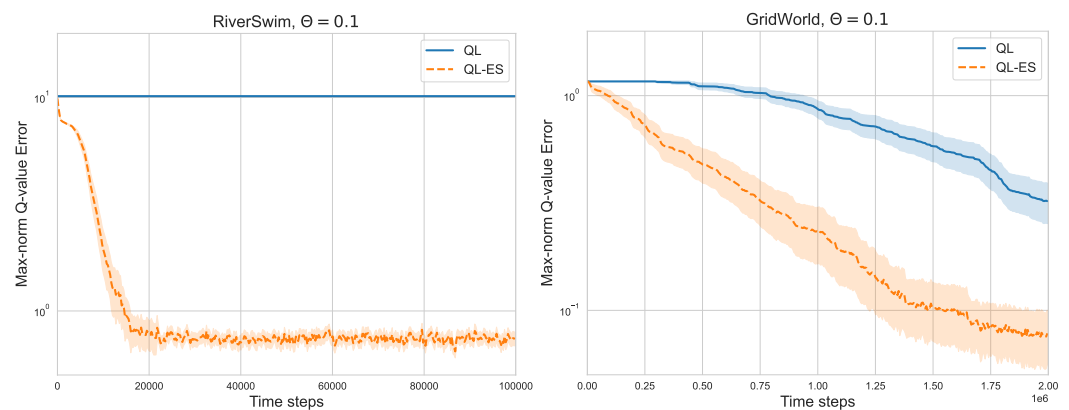


Figure 8. Results with θ -similar pairs: Modified RiverSwim (left) and Modified GridWorld (right).

6.6. The Impact of Partially Using the Structure

Considering MDPs with huge state–action spaces, it is necessary to take into account the feasibility of using only a few equivalent pairs. This thus naturally leads to the question as to whether only using a few equivalent pairs would lead to a reasonable performance gain. Therefore, here we investigate the convergence speed when choosing a subset of equivalent state–action pairs at each time step, rather than considering all the state–action pairs in the same class.

As shown in Figure 9, we can still obtain reasonable performance only considering a few equivalent pairs. The numbers in brackets represent how many equivalent pairs are used. In RiverSwim with 6 states ($SA = 12$), the performance of QLES(3) and QLES(4) is comparable to QL-ES. Interestingly, we already observe a significant improvement over Q-learning using QLES(1), i.e., when using only one additional observation in the Q-learning update.

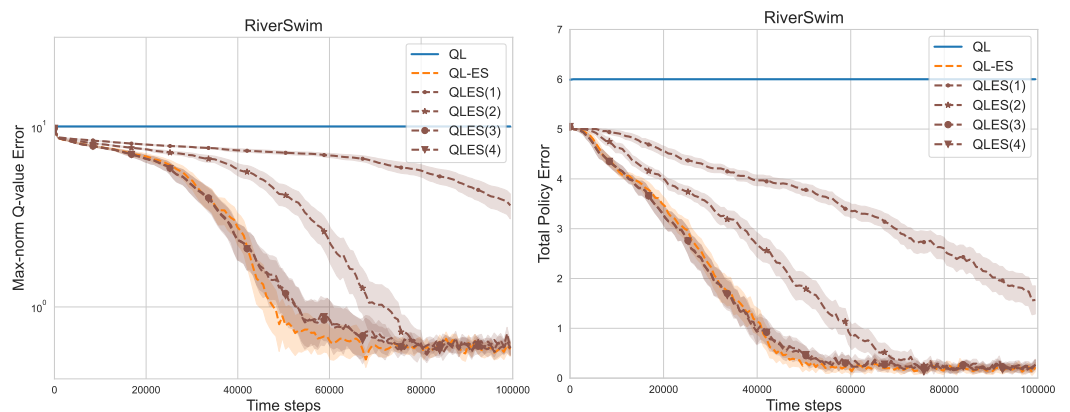


Figure 9. Results in RiverSwim: $S = 6$.

Meanwhile, the total policy error is less than or close to one. This shows that the optimal policy is correctly learned in almost all the states.

In 20-state RiverSwim ($SA = 40$), algorithms with few equivalent pairs can still achieve excellent performance, albeit not as good as QL-ES. Additionally, as Figure 10 shows, using more equivalent pairs leads to better sample efficiencies in MDPs with large state space. Concerning QLES(3)-QLES(12), the Q-value error gets smaller when more equivalent pairs are used.

From the perspective of policy, even QLES(3) and QLES(6) manage to find optimal actions significantly faster than Q-learning. In addition, QLES(10) is far superior to QLES(6) because of the quite smaller Q-value error and total policy error. The results show that algorithms with a suitable number of equivalent pairs are sufficient to learn an optimal policy in most states reasonably fast.

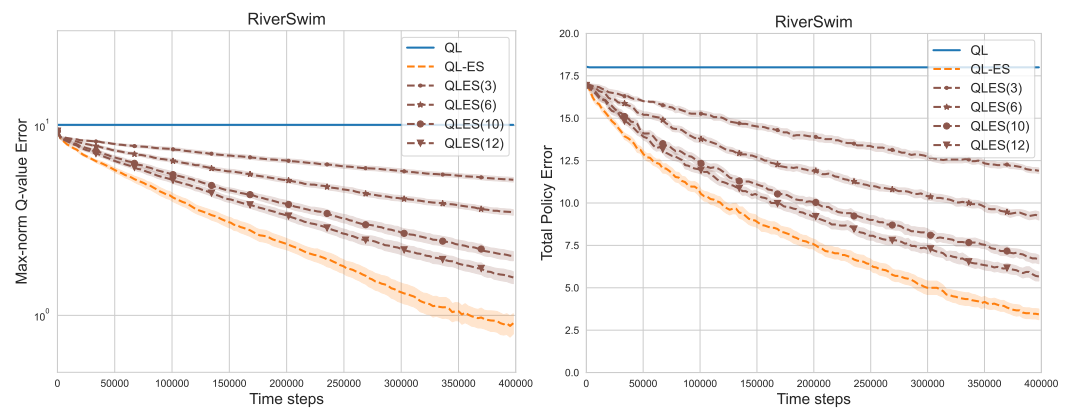


Figure 10. Results in RiverSwim: $S = 20$.

7. Conclusions

We studied off-policy learning in discounted Markov decision processes, where some equivalence structure exists in the state–action space. We presented a model-free algorithm called QL-ES, which is a natural extension of the classical asynchronous Q-learning but capable of exploiting the equivalence structure. We presented a high-probability sample complexity bound for QL-ES, and discussed how it improves that of Q-learning. As demonstrated, there exist problem instances on which the improvement over Q-learning could be a multiplicative factor of S , the size of the state space. Through extensive numerical experiments in standard domains, we demonstrated that QL-ES significantly improves over (structure-oblivious) Q-learning. These results revealed that exploiting state–action equivalence favors faster convergence of the Q-function and policy learning in large MDPs. A limitation of our approach is the need for the prior knowledge on the structure. To the best of our knowledge, existing methods for learning the equivalence structure are all model-based. Hence, an interesting question is whether it is possible to exploit the equivalence structure *without prior knowledge on the structure using only model-free algorithms*. Devising such model-free algorithms (or otherwise establishing an impossibility result) is an interesting yet challenging topic for future work. Another interesting direction for future work is to investigate ways to combine the knowledge of the equivalence structure with function approximation methods. Finally, another avenue for future work is to study model-free algorithms for the regret minimization setting in average-reward MDPs (e.g., [47,48]).

Author Contributions: Conceiving and designing the experiments, Y.L., Y.Z. and A.C.; performing the experiments, Y.L. and Y.Z.; writing original draft preparation, Y.L., A.C. and M.S.T.; providing proofs, A.C. and M.S.T.; reviewing the paper and provided suggestions, M.S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers for their constructive comments. Yunlian Lyu was supported by the China Scholarship Council (CSC) and the Department of Computer Science at the University of Copenhagen. Aymeric Côme was supported by the French government through the Program “Investissement d’avenir” (I-SITE ULNE/ANR-16-IDEX-0004 ULNE) managed by the National Research Agency. This work was partially done while Aymeric Côme was completing an internship at the Department of Computer Science at the University of Copenhagen.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Sample Complexity of QL-ES: Proof of Theorem 1

The following proof for Theorem 1 is a direct adaptation from Theorem 2 of [9]. We closely follow the notations and definitions used in [9].

Appendix A.1. Notations

Before presenting the result and going into the details of the proof, some notations must be introduced. In the following, letters in bold, such as \mathbf{Q} , shall stand for matrices and vectors, and for a matrix $\mathbf{M} \in \mathbb{R}^{N \times L}$, $\mathbf{M}(i)$ is the i -th row of \mathbf{M} . Additionally, as long as there is no confusion, applying operations such as $\sqrt{\cdot}$ or $|\cdot|$ to a matrix is the entry-wise operation: for example, $|\mathbf{M}| = (|m_{ij}|)_{i,j}$. Finally, $\mathbf{1}$ is the vector full of ones, and \mathbf{I} is the identity matrix.

Moreover, the following quantities appear in the proof:

$$t_{\text{cover,all}} = t_{\text{cover,C}} \log\left(\frac{T}{\delta}\right) \tag{A1}$$

$$t_{\text{th}} = \max\left\{\frac{2t_{\text{cover,all}}}{\alpha} \log\left(\frac{1}{(1-\gamma)^2 \varepsilon}\right), t_{\text{cover,all}}\right\} \tag{A2}$$

$$\rho = (1-\gamma)(1-(1-\alpha)^{\frac{1}{2}}) \tag{A3}$$

Let us introduce the short-hand $c_t := c(s_t, a_t)$ for any $t \geq 0$. In order to analyze the update in QL-ES in a matrix-form way, we introduce

$$\Lambda_t((s, a), (s, a)) = \begin{cases} \alpha & \text{if } (s, a) \in c_{t-1} \\ 0 & \text{otherwise} \end{cases} \tag{A4}$$

$$\mathbf{P}_t((s, a), s') = \begin{cases} 1 & \text{if } (s, a) \in c_{t-1} \text{ and } s' = \sigma_{s,a}^{-1}(\sigma_{s_{t-1}, a_{t-1}}(s_t)) \\ 0 & \text{otherwise} \end{cases} \tag{A5}$$

and we will use \mathbf{Q}_t for the Q-value matrix (of size $S \times A$) and \mathbf{P} for the transition matrix (of size $SA \times S$). Notice that $\Lambda_t \in [0, 1]^{SA \times SA}$ is diagonal, and $\mathbf{P}_t \in [0, 1]^{SA \times S}$. Then we have:

$$\mathbf{Q}_t = (\mathbf{I} - \Lambda_t)\mathbf{Q}_{t-1} + \Lambda_t(\mathbf{r} + \gamma\mathbf{P}_t\mathbf{V}_{t-1}),$$

where $\mathbf{V}_t \in \mathbb{R}^S$ is the vector of value estimates at time t , whose s -th element is given by $\max_a \mathbf{Q}_t(s, a)$. Finally, we are interested in bounding $\Delta_t = \mathbf{Q}_t - \mathbf{Q}^*$.

The following lemma showcases the role of the cover time, and the proof is given in Appendix B.1:

Lemma A1. Define the event

$$\mathcal{K}_l = \{\exists c \in \mathcal{C} \text{ s.t. } c \text{ is not visited within iterations } (lt_{\text{cover,all}}, (l+1)t_{\text{cover,all}})\}$$

and $L = \lfloor \frac{T}{t_{\text{cover,all}}} \rfloor$. Then: $\mathbb{P}(\cup_{l=0}^L \mathcal{K}_l) \leq \delta$.

Appendix A.2. Proof of Theorem 1

As established in the proof of [9] (Theorem 2) (see Equation (39) there), we have

$$\Delta_t = (\mathbf{I} - \Lambda_t)\Delta_{t-1} + \gamma\Lambda_t(\mathbf{P}_t - \mathbf{P})\mathbf{V}^* + \gamma\Lambda_t\mathbf{P}_t(\mathbf{V}_{t-1} - \mathbf{V}^*),$$

which, using an inductive argument, yields

$$\Delta_t = \underbrace{\gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i (\mathbf{P}_i - \mathbf{P}) \mathbf{V}^*}_{\beta_{1,t}} + \underbrace{\gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{P}_i (\mathbf{V}_{i-1} - \mathbf{V}^*)}_{\beta_{2,t}} + \underbrace{\prod_{j=1}^t (\mathbf{I} - \Lambda_j) \Delta_0}_{\beta_{3,t}}.$$

Hence, $|\Delta_t| \leq |\beta_{1,t}| + |\beta_{2,t}| + |\beta_{3,t}|$, where \leq holds element-wise. Furthermore, as in the proof of [9] (Theorem 2),

$$\|\mathbf{P}_i(\mathbf{V}_{i-1} - \mathbf{V}^*)\|_\infty \leq \|\mathbf{P}_i\|_1 \|\mathbf{V}_{i-1} - \mathbf{V}^*\|_\infty = \|\mathbf{Q}_{i-1} - \mathbf{Q}^*\|_\infty = \|\Delta_{i-1}\|_\infty,$$

since $\|\mathbf{P}_i\|_1 = 1$ in view of the definition of \mathbf{P}_i . Hence,

$$|\beta_{2,t}| \leq \gamma \sum_{i=1}^t \|\Delta_{i-1}\|_\infty \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1}. \tag{A6}$$

The following two lemmas provide upper bounds on $|\beta_{1,t}|$ and $|\beta_{3,t}|$.

Lemma A2. *There exists a universal constant $\kappa > 0$ such that, for any $0 < \delta < 1$,*

$$\forall 1 \leq t \leq T, \quad |\beta_{1,t}| \leq \tau_1 \|\mathbf{V}^*\|_\infty \mathbf{1} \tag{A7}$$

with probability at least $1 - \delta$, where $\tau_1 = \kappa \gamma \sqrt{\alpha \log \left(\frac{CT}{\delta}\right)}$.

Lemma A3. *For all $t > 0$, we have $|\beta_{3,t}| \leq \|\Delta_0\|_\infty \mathbf{1}$. Furthermore, with probability at least $1 - \delta$,*

$$\forall t \in [t_{\text{cover,all}}, T], \quad |\beta_{3,t}| \leq (1 - \alpha)^{\frac{t}{2t_{\text{cover,all}}}} \|\Delta_0\|_\infty \mathbf{1}.$$

Lemma A2 is proven in Appendix B.2. The proof of Lemma A3 is the same as the proof of [9] (Lemma 6) but using appropriate definitions of $t_{\text{cover,all}}$ and $K_t(s, a)$ for the case of QL-ES; it is thus omitted.

The two lemmas above together with (A6) imply that with probability greater than $1 - 2\delta$,

$$|\Delta_t| \leq \begin{cases} \gamma \sum_{i=1}^t \|\Delta_{i-1}\|_\infty \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} + \tau_1 \|\mathbf{V}^*\|_\infty \mathbf{1} + \|\Delta_0\|_\infty \mathbf{1} & t < t_{\text{cover,all}} \\ \gamma \sum_{i=1}^t \|\Delta_{i-1}\|_\infty \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} + \tau_1 \|\mathbf{V}^*\|_\infty \mathbf{1} + (1 - \alpha)^{t/(2t_{\text{cover,all}})} \|\Delta_0\|_\infty \mathbf{1} & t_{\text{cover,all}} \leq t \leq T \end{cases} \tag{A8}$$

Now, a refined recursive analysis is conducted (in Appendix B.3):

Lemma A4. *Let*

$$u_0 = \frac{\|\Delta_0\|_\infty}{1 - \gamma}, \quad u_t = \|\mathbf{v}_t\|_\infty \tag{A9}$$

$$\mathbf{v}_t = \begin{cases} \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} u_{i-1} + \|\Delta_0\|_\infty \mathbf{1} & 1 \leq t \leq t_{\text{th}} \\ \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} u_{i-1} & t > t_{\text{th}}. \end{cases} \tag{A10}$$

Then with probability greater than $1 - 2\delta$,

$$\|\Delta_t\|_\infty \leq \frac{\tau_1 \|\mathbf{V}^*\|_\infty}{1 - \gamma} + u_t + \varepsilon. \tag{A11}$$

This lemma is the consequence of a direct computational induction from (A8), using the fact that $(1 - \alpha)^{\frac{t}{2t_{\text{cover,all}}}} \leq (1 - \gamma)\varepsilon$ when $t \geq t_{\text{th}}$, and $\|\Delta_0\|_\infty = \|\mathbf{Q}^*\|_\infty \leq \frac{1}{1 - \gamma}$.

Lemma A5. *Let $w_k = (1 - \rho)^k \frac{\|\Delta_0\|_\infty}{1 - \gamma}$ for all $k \in \mathbb{N}$. Then, with probability $1 - 2\delta$:*

$$u_t \leq w_{k_t} \quad \text{with } k_t = \max \left\{ 0, \left\lfloor \frac{t - t_{\text{th}}}{t_{\text{cover,all}}} \right\rfloor \right\} \tag{A12}$$

The proof for Lemma A5 relies once again on computational arguments and an induction, as showcased in Appendix B.4.

Finally, the following theorem is the last milestone of the proof, obtained from the two previous lemmas.

Theorem A1. For any $\varepsilon \in (0, \frac{1}{1-\gamma}]$, $\delta \in (0, 1)$, there exists a universal constant $\kappa > 0$ such that with probability at least $1 - 6\delta$, for all $t \leq T$:

$$\|\mathbf{Q}_t - \mathbf{Q}^*\|_\infty \leq (1 - \rho)^k \frac{\|\mathbf{Q}_0 - \mathbf{Q}^*\|_\infty}{1 - \gamma} + \frac{\kappa\gamma}{1 - \gamma} \|\mathbf{V}^*\|_\infty \sqrt{\alpha \log\left(\frac{CT}{\delta}\right)} + \varepsilon \tag{A13}$$

with $k = \max\left\{0, \lfloor \frac{t - t_{th}}{t_{cover,all}} \rfloor\right\}$.

It is easy to upper bound the second term in (A13) with ε by setting $\alpha = \frac{(1-\gamma)^4 \varepsilon^2}{\kappa^2 \gamma^2 \log(CT/\delta)}$. Then, using $(1 - \rho)^k \leq e^{-\rho k}$ and noting that $\rho \geq \frac{1}{4}(1 - \gamma)\alpha$ for $\alpha < \frac{1}{2}$ allow us to upper bound the first term by ε , thanks to the definition of k , as long as

$$t \geq t_{th} + t_{cover,all} + \frac{4t_{cover,all}}{(1 - \gamma)\alpha} \log\left(\frac{\|\Delta_0\|_\infty}{\varepsilon(1 - \gamma)}\right) = T, \tag{A14}$$

which concludes this proof of Theorem 1.

Appendix B. Proofs of Technical Lemmas

Appendix B.1. Proof of Lemma A1

This proof reproduces the one of Lemma 5 in [9] but tailored to the case of QL-ES. We set $t_l = t_{cover,C^l}$, and define

$$\mathcal{H}_l = \{\exists c \in \mathcal{C} \text{ that is not visited within } (t_l, t_{l+1}]\}$$

for any integer $l > 0$. From the definition of $t_{cover,C}$, for any $c' \in \mathcal{C}$,

$$\mathbb{P}(\mathcal{H}_l | c_{t_l} = c') \leq \frac{1}{2}.$$

Hence, using the Markov property, for any integer $L > 0$,

$$\begin{aligned} \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_L) &= \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \mathbb{P}(\mathcal{H}_L | \mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \\ &= \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \sum_{c' \in \mathcal{C}} \mathbb{P}(\mathcal{H}_L | c_{t_L} = c') \mathbb{P}(c_{t_L} = c' | \mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \\ &\leq \frac{1}{2} \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \sum_{c' \in \mathcal{C}} \mathbb{P}(c_{t_L} = c' | \mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}) \\ &= \frac{1}{2} \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_{L-1}). \end{aligned}$$

An easy recursive derivation gives $\mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_L) \leq 2^{-L}$. Finally:

$$\mathbb{P}(\exists c \in \mathcal{C} \text{ not visited between } (0, t_{cover,all}]) \leq \mathbb{P}(\mathcal{H}_1 \cap \dots \cap \mathcal{H}_{\log_2 \frac{T}{\delta}}) \leq \frac{1}{2^{\log_2 \frac{T}{\delta}}} = \frac{\delta}{T}$$

from which we easily deduce the result by using a straightforward union bound.

Appendix B.2. Proof of Lemma A2

The proof is an adaptation of the proof of Lemma 1 in [9] to the case of Q-value updates in QL-ES. Similar to the proof of Lemma 1 in [9], we begin with looking at the (s, a) -th element of $\beta_{1,t}$:

$$\beta_{1,t}(s, a) = \gamma \sum_{k=1}^{K_t(s,a)} (1 - \alpha)^{K_t(s,a)-k} \alpha (\mathbf{P}_{t_k(s,a)+1}(s, a) - \mathbf{P}(s, a)) \mathbf{V}^*,$$

where $t_k(s, a)$ denotes the time step when $c(s, a)$ is visited for the k -th time, and where $K_t(s, a)$ denotes the number of times (s, a) is updated during the t first time steps: $K_t(s, a) = \max\{k \mid t_k(s, a) \leq t\}$. We simplify the notation below with $t_k(s, a) = t_k$. Now, we claim that

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \quad |\beta_{1,t}| \leq \gamma \sqrt{\alpha \log\left(\frac{CT}{\delta}\right)} \|\mathbf{V}^*\|_\infty.$$

Firstly, the vectors $\mathbf{P}_{t_k+1}(s, a), k = 1, \dots, K$ are independent and identically distributed for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ and any $K \in \mathbb{N}^*$. Indeed, for any $i_1, \dots, i_K \in \mathcal{S}$,

$$\begin{aligned} & \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K\right) \\ &= \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K-1 \text{ and } \sigma_{s,a}^{-1}(\sigma_{s_{t_K}, a_{t_K}}(s_{t_K+1})) = i_K\right) \\ &= \sum_{m \in \mathbb{N}^*} \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K-1 \text{ and } t_K = m \text{ and } \sigma_{s,a}^{-1}(\sigma_{s_m, a_m}(s_{m+1})) = i_K\right) \\ &\stackrel{(i)}{=} \sum_{m \in \mathbb{N}^*} \left[\mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K-1 \text{ and } t_K = m\right) \right. \\ &\quad \left. \times \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_m, a_m}(s_{m+1})) = i_K, \mid (s, a) \in c_m\right) \right] \\ &\stackrel{(ii)}{=} \mathbb{P}_{s,a}(i_K) \sum_{m \in \mathbb{N}^*} \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K-1 \text{ and } t_K = m\right) \\ &= \mathbb{P}_{s,a}(i_K) \mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K-1\right) \end{aligned}$$

where (i) uses the Markov property, $(s_{t_k}, a_{t_k}) \in c(s, a)$ from the definition of $t_K = t_K(s, a)$, (ii) uses the equivalence property between (s, a) and (s_m, a_m) , and $\mathbb{P}_{s,a}$ is the transition probability from (s, a) . By induction, one obtains

$$\mathbb{P}\left(\sigma_{s,a}^{-1}(\sigma_{s_{t_k}, a_{t_k}}(s_{t_k+1})) = i_k, \forall 1 \leq k \leq K\right) = \prod_{j=1}^K \mathbb{P}_{s,a}(i_j)$$

which proves the sought independence. Then, following identical lines as in the proof of Lemma 1 in [9] (Lemma 1), we can use Hoeffding’s inequality to bound $\beta_{1,t}$, which yields

$$\left| \sum_{k=1}^K (1 - \alpha)^{K-k} \alpha (\mathbf{P}_{t_k+1}(s, a) - \mathbf{P}(s, a)) \mathbf{V}^* \right| \leq \sqrt{\alpha \log\left(\frac{CT}{\delta}\right)} \|\mathbf{V}^*\|_\infty.$$

The proof is concluded by taking the union bound over all classes $c \in \mathcal{C}$ and all $1 \leq K \leq T$.

Appendix B.3. Proof for Lemma A4

The proof is a straightforward adaptation of the proof of [9] (Lemma 3), where we reproduce most of the steps from [9] for completeness. We prove the lemma by induction on t . The base case $t = 0$ holds trivially:

$$\|\Delta_0\|_\infty \leq \frac{\|\Delta_0\|_\infty}{1-\gamma} \leq \frac{\tau_1\|\mathbf{V}^*\|_\infty}{1-\gamma} + u_0 + \varepsilon.$$

Now let us assume that the recursive hypothesis holds for all $i < t$. Let us define

$$h(t) = \begin{cases} \|\Delta_0\|_\infty & \text{if } t \leq t_{\text{th}} \\ (1-\gamma)\varepsilon & \text{if } t > t_{\text{th}} \end{cases}$$

and we remind that $(1-\alpha)^{\frac{t}{2^{\text{cover,all}}}} \leq (1-\gamma)\varepsilon$ whenever $t \geq t_{\text{th}}$. Therefore, we have

$$\begin{aligned} |\Delta_t| &\leq \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} \left(\frac{\tau_1\|\mathbf{V}^*\|_\infty}{1-\gamma} + u_{i-1} + \varepsilon \right) + \tau_1\|\mathbf{V}^*\|_\infty \mathbf{1} + h(t)\mathbf{1} \\ &= \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} u_{i-1} + \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} \left(\frac{\tau_1\|\mathbf{V}^*\|_\infty}{1-\gamma} + \varepsilon \right) + \tau_1\|\mathbf{V}^*\|_\infty \mathbf{1} + h(t)\mathbf{1}. \end{aligned}$$

Furthermore, define the diagonal matrix $\mathbf{M}_i = \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i$, and denote by $N_i^j(s, a)$ the number of visits to the equivalence class of the state–action pair (s, a) between the i -th and the j -th iterations (including i and j). Then the diagonal entries of \mathbf{M}_i satisfy

$$\mathbf{M}_i((s, a), (s, a)) = \begin{cases} \alpha(1-\alpha)^{N_{i+1}^t(s, a)}, & \text{if } (s, a) \in c_{i-1}, \\ 0, & \text{if } (s, a) \notin c_{i-1}. \end{cases}$$

Introducing $\mathbf{e}_{(s, a)} \in \mathbb{R}^{\mathcal{S}\mathcal{A}}$ as the standard basis vector whose only non-zero entry is the (s, a) -th entry, one obtains the following:

$$\prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} = \mathbf{M}_i \mathbf{1} = \mathbf{M}_i \mathbf{e}_{(s_{i-1}, a_{i-1})} = \alpha(1-\alpha)^{N_{i+1}^t(s_{i-1}, a_{i-1})} \mathbf{e}_{(s_{i-1}, a_{i-1})}$$

and

$$\begin{aligned} \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} &= \sum_{i=1}^t \alpha(1-\alpha)^{N_{i+1}^t(s_{i-1}, a_{i-1})} \mathbf{e}_{(s_{i-1}, a_{i-1})} \\ &= \sum_{(s, a) \in \mathcal{S} \times \mathcal{A}} \left\{ \sum_{i=1}^t \alpha(1-\alpha)^{N_{i+1}^t(s, a)} \mathbb{I}\{(s, a) \in c_{i-1}\} \right\} \mathbf{e}_{(s, a)} \\ &\leq \sum_{(s, a) \in \mathcal{S} \times \mathcal{A}} \sum_{j=0}^\infty \alpha(1-\alpha)^j \mathbf{e}_{(s, a)} = \sum_{j=0}^\infty \alpha(1-\alpha)^j \mathbf{1} = \mathbf{1}. \end{aligned} \tag{A15}$$

Using (A15), we obtain

$$\begin{aligned}
 |\Delta_t| &\leq \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} u_{i-1} + \frac{\gamma \tau_1 \|\mathbf{V}^*\|_\infty}{1-\gamma} \mathbf{1} + \gamma \varepsilon \mathbf{1} + \tau_1 \|\mathbf{V}^*\|_\infty \mathbf{1} + h(t) \mathbf{1} \\
 &= \frac{\tau_1 \|\mathbf{V}^*\|_\infty}{1-\gamma} \mathbf{1} + \gamma \varepsilon \mathbf{1} + \gamma \sum_{i=1}^t \prod_{j=i+1}^t (\mathbf{I} - \Lambda_j) \Lambda_i \mathbf{1} u_{i-1} + h(t) \mathbf{1} \\
 &= \frac{\tau_1 \|\mathbf{V}^*\|_\infty}{1-\gamma} \mathbf{1} + \gamma \varepsilon \mathbf{1} + \mathbf{v}_t + (1-\gamma) \varepsilon \mathbb{I}\{t > t_{\text{th}}\} \mathbf{1} \\
 &\leq \frac{\tau_1 \|\mathbf{V}^*\|_\infty}{1-\gamma} \mathbf{1} + \varepsilon \mathbf{1} + \mathbf{v}_t.
 \end{aligned}$$

With the definition of $u_t = \|\mathbf{v}_t\|_\infty$, we arrive at the desired result.

Appendix B.4. Proof for Lemma A5

This proof follows identical steps as in the proof of Lemma 4 in [9] except that the quantities $N_i^n(s, a)$ used there should be defined as the number of visits to the equivalence class of the state–action pair (s, a) between iteration i and iteration n (including i and n).

References

1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
2. Watkins, C.J.C.H. *Learning from Delayed Rewards*. Ph.D. Thesis, University of Cambridge England, Cambridge, UK, 1989.
3. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
4. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
5. Gheshlaghi Azar, M.; Munos, R.; Kappen, H.J. Minimax PAC Bounds on the Sample Complexity of Reinforcement Learning with a Generative Model. *Mach. Learn.* **2013**, *91*, 325–349. [[CrossRef](#)]
6. Azar, M.G.; Osband, I.; Munos, R. Minimax Regret Bounds for Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, 2017, Sydney, Australia, 6–11 August 2017; pp. 263–272.
7. Zhou, D.; Gu, Q.; Szepesvari, C. Nearly Minimax Optimal Reinforcement Learning for Linear Mixture Markov Decision Processes. In Proceedings of the Conference on Learning Theory, Boulder, CO, USA, 15–19 August 2021; pp. 4532–4576.
8. Agarwal, A.; Kakade, S.; Yang, L.F. Model-Based Reinforcement Learning with a Generative Model is Minimax Optimal. In Proceedings of the Conference on Learning Theory, Virtual, 9–12 July 2020; pp. 67–83.
9. Li, G.; Wei, Y.; Chi, Y.; Gu, Y.; Chen, Y. Sample Complexity of Asynchronous Q-Learning: Sharper Analysis and Variance Reduction. *arXiv* **2021**, arXiv:2006.03041.
10. Ortner, R.; Ryabko, D. Online Regret Bounds for Undiscounted Continuous Reinforcement Learning. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1763–1771.
11. QIAN, J.; Fruit, R.; Pirotta, M.; Lazaric, A. Exploration Bonus for Regret Minimization in Discrete and Continuous Average Reward MDPs. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 4891–4900.
12. Asadi, K.; Misra, D.; Littman, M. Lipschitz Continuity in Model-Based Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 264–273.
13. Ok, J.; Proutiere, A.; Tranos, D. Exploration in Structured Reinforcement Learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 8888–8896.
14. Osband, I.; Van Roy, B. Near-Optimal Reinforcement Learning in Factored MDPs. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 604–612.
15. Talebi, M.S.; Jonsson, A.; Maillard, O. Improved Exploration in Factored Average-Reward MDPs. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 13–15 April 2021; pp. 3988–3996.
16. Rosenberg, A.; Mansour, Y. Oracle-Efficient Regret Minimization in Factored MDPs with Unknown Structure. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11148–11159.
17. Sun, Y.; Yin, X.; Huang, F. Temple: Learning Template of Transitions for Sample Efficient Multi-Task RL. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 9765–9773.
18. Asadi, M.; Talebi, M.S.; Bourel, H.; Maillard, O.A. Model-Based Reinforcement Learning Exploiting State-Action Equivalence. In Proceedings of the Asian Conference on Machine Learning, Nagoya, Japan, 17–19 November 2019; pp. 204–219.
19. Leffler, B.R.; Littman, M.L.; Edmunds, T. Efficient Reinforcement Learning with Relocatable Action Models. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22–26 July 2007; Volume 7, pp. 572–577.

20. Ortner, R. Adaptive Aggregation for Reinforcement Learning in Average Reward Markov Decision Processes. *Ann. Oper. Res.* **2013**, *208*, 321–336. [[CrossRef](#)]
21. Van der Pol, E.; Worrall, D.; van Hoof, H.; Oliehoek, F.; Welling, M. MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 4199–4210.
22. Mondal, A.K.; Nair, P.; Siddiqi, K. Group Equivariant Deep Reinforcement Learning. *arXiv* **2020**, arXiv:2007.03437.
23. Ortner, R.; Ryabko, D.; Auer, P.; Munos, R. Regret Bounds for Restless Markov Bandits. *Theor. Comput. Sci.* **2014**, *558*, 62–76. [[CrossRef](#)]
24. Wen, Z.; Precup, D.; Ibrahimi, M.; Barreto, A.; Van Roy, B.; Singh, S. On Efficiency in Hierarchical Reinforcement Learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6708–6718.
25. Dulac-Arnold, G.; Levine, N.; Mankowitz, D.J.; Li, J.; Paduraru, C.; Gowal, S.; Hester, T. Challenges of Real-World Reinforcement Learning: Definitions, Benchmarks and Analysis. *Mach. Learn.* **2021**, *110*, 2419–2468. [[CrossRef](#)]
26. Li, L.; Walsh, T.J.; Littman, M.L. Towards a Unified Theory of State Abstraction for MDPs. In Proceedings of the International Symposium on Artificial Intelligence and Mathematics, Lauderdale, FL, USA, 4–6 January 2006.
27. Abel, D.; Hershkowitz, D.; Littman, M. Near Optimal Behavior via Approximate State Abstraction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2915–2923.
28. Ravindran, B.; Barto, A.G. Approximate Homomorphisms: A Framework for Non-Exact Minimization in Markov Decision Processes. In Proceedings of the KBCS, New York, NY, USA, 17–18 May 2004 .
29. Dean, T.; Givan, R.; Leach, S. Model Reduction Techniques for Computing Approximately Optimal Solutions for Markov Decision Processes. In Proceedings of the Uncertainty in Artificial Intelligence, Providence, RI, USA, 1–3 August 1997; pp. 124–131.
30. Givan, R.; Dean, T.; Greig, M. Equivalence Notions and Model Minimization in Markov Decision Processes. *Artif. Intell.* **2003**, *147*, 163–223. [[CrossRef](#)]
31. Ferns, N.; Panangaden, P.; Precup, D. Metrics for Finite Markov Decision Processes. In Proceedings of the Uncertainty in Artificial Intelligence, Banff, AB, Canada, 7–11 July 2004; pp. 162–169.
32. Ferns, N.; Panangaden, P.; Precup, D. Bisimulation Metrics for Continuous Markov Decision Processes. *SIAM J. Comput.* **2011**, *40*, 1662–1714. [[CrossRef](#)]
33. Brunskill, E.; Li, L. Sample Complexity of Multi-task Reinforcement Learning. In Proceedings of the Uncertainty in Artificial Intelligence, Bellevue, WA, USA, 11–15 August 2013; p. 122.
34. Mandel, T.; Liu, Y.E.; Brunskill, E.; Popovic, Z. Efficient Bayesian Clustering for Reinforcement Learning. In Proceedings of the International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–16 July 2016; pp. 1830–1838.
35. Watkins, C.J.; Dayan, P. Q-Learning. *Mach. Learn.* **1992**, *8*, 279–292. [[CrossRef](#)]
36. Tsitsiklis, J.N. Asynchronous Stochastic Approximation and Q-Learning. *Mach. Learn.* **1994**, *16*, 185–202. [[CrossRef](#)]
37. Even-Dar, E.; Mansour, Y. Learning Rates for Q-Learning. *J. Mach. Learn. Res.* **2003**, *5*, 1–25.
38. Azar, M.G.; Munos, R.; Ghavamzadeh, M.; Kappen, H. Reinforcement Learning with a Near Optimal Rate of Convergence Technical Report <inria-00636615v2>. 2011. Available online: https://www.researchgate.net/publication/265653590_Reinforcement_Learning_with_a_Near_Optimal_Rate_of_Convergence (accessed on 30 January 2023).
39. Qu, G.; Wierman, A. Finite-Time Analysis of Asynchronous Stochastic Approximation and Q-Learning. In Proceedings of the Conference on Learning Theory, Virtual, 9–12 July 2020; pp. 3185–3205.
40. Devraj, A.M.; Meyn, S.P. Q-Learning with Uniformly Bounded Variance: Large Discounting is Not a Barrier to Fast Learning. *arXiv* **2020**, arXiv:2002.10301.
41. Wang, Y.; Dong, K.; Chen, X.; Wang, L. Q-Learning with UCB Exploration is Sample Efficient for Infinite-Horizon MDP. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
42. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
43. Levin, D.A.; Peres, Y. *Markov Chains and Mixing Times*; American Mathematical Society: Providence, RI, USA, 2017; Volume 107.
44. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [[CrossRef](#)]
45. Strehl, A.L.; Littman, M.L. An Analysis of Model-Based Interval Estimation for Markov Decision Processes. *J. Comput. Syst. Sci.* **2008**, *74*, 1309–1331. [[CrossRef](#)]
46. Azar, M.G.; Munos, R.; Ghavamzadeh, M.; Kappen, H.J. Speedy Q-Learning. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 2411–2419.
47. Jaksch, T.; Ortner, R.; Auer, P. Near-Optimal Regret Bounds for Reinforcement Learning. *J. Mach. Learn. Res.* **2010**, *11*, 1563–1600.
48. Bourel, H.; Maillard, O.; Talebi, M.S. Tightening Exploration in Upper Confidence Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1056–1066.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.