Journal of Machine Learning Research 23 (2022) 1-54 Submitted 7/21; Revised 8/22; Published 9/22

Simple Agent, Complex Environment: Efficient Reinforcement Learning with Agent States

Shi Dong SDONG15@STANFORD.EDU
Benjamin Van Roy BVR@STANFORD.EDU
Stanford University

Zhengyuan Zhou ZZHOU@STERN.NYU.EDU
New York University

Editor: Alekh Agarwal

Abstract

We design a simple reinforcement learning (RL) agent that implements an optimistic version of Q-learning and establish through regret analysis that this agent can operate with some level of competence in *any* environment. While we leverage concepts from the literature on provably efficient RL, we consider a *general* agent-environment interface and provide a novel agent design and analysis. This level of generality positions our results to inform the design of future agents for operation in complex real environments. We establish that, as time progresses, our agent performs competitively relative to policies that require longer times to evaluate. The time it takes to approach asymptotic performance is polynomial in the complexity of the agent's state representation and the time required to evaluate the best policy that the agent can represent. Notably, there is no dependence on the complexity of the environment. The ultimate per-period performance loss of the agent is bounded by a constant multiple of a measure of distortion introduced by the agent's state representation. This work is the first to establish that an algorithm approaches this asymptotic condition within a tractable time frame.

Keywords: Reinforcement learning, Q-learning, dynamic programming, regret analysis, agent design.

1. Introduction

Reinforcement learning agents have demonstrated remarkable success in simulated environments. For example, the recently developed MuZero agent (Schrittwieser et al., 2020) learns to interact effectively with any of a broad range of environment simulators and delivers superhuman performance in playing chess, go, shogi, and arcade games. Continuing innovations in this area aim to produce agents that can engage with increasingly complex environments – ultimately, environments like the physical world or the World Wide Web – which pose far greater complexity than the agent can represent.

There is a growing mathematical literature that focuses on establishing efficiency guarantees, typically in terms of sample complexity or regret bounds (Kearns and Singh (2002); Jaksch et al. (2010) represent early instances). Indeed, data efficiency remains an impediment to carrying the success of reinforcement learning from simulated to real environments, in which agents must learn within reasonable time frames. As such, the mathematical literature ought to inform future agent designs. However, work in this area has tended to focus on restrictive classes of environments, and further, to produce bounds that depend on the number of environment states, which is effectively infinite in a complex environment.

In this paper, we aim to bridge the divide. In particular, we extend ideas from the mathematical literature while relaxing common restrictions. In doing so, we establish results that offer insight into how a simple agent can operate effectively in an arbitrarily complex environment. While we consider a problem formulation that has been previously studied, this work contributes to multiple fronts: framing of learning objectives, agent design, and performance analysis.

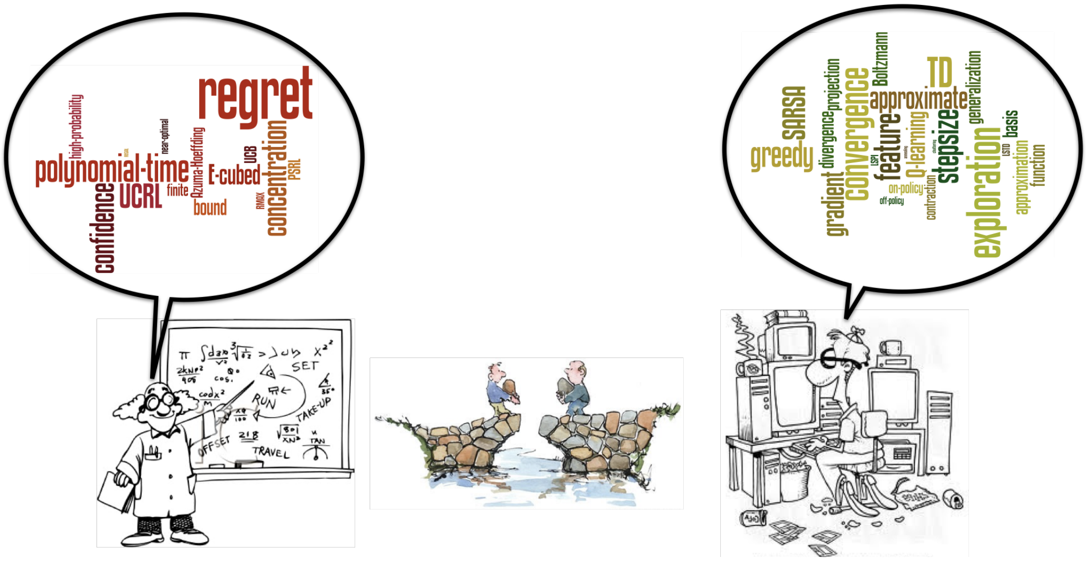
regret high-probability polynomial-time near-optimal Azuma-Hoeffding E-cubed UCB concentration PSRL confidence UCRL finite bound RMAX
greedy SARSA gradient divergence projection convergence Boltzmann approximate TD feature LSPI Q-learning on-policy off-policy contraction stepsize generalization exploration basis LSTD approximation function
SET
RUN
TAKE-UP
OFFSET
TRAVEL
TAN

Figure 1: Bridging the divide: "provably efficient" reinforcement learning versus "practical" agent design.

1.1 Complex Environments

We consider an interface, as illustrated in Figure 2, which is defined by a finite action set $\mathcal{A}$ and a finite observation set $\mathcal{O}$. The agent interacts with the environment by executing at each time t an action A_t and then registering an observation O_{t+1}, generating a single stream of experience $(A_0, O_1, A_1, O_2, \ldots)$. At each time t, the agent selects A_t based on its history $H_t = (A_0, O_1, \ldots, A_{t-1}, O_t)$. The initial history $H_0 = ()$ is empty.

The interface we have described is very general. An agent can engage in this manner with arbitrarily complex environments. As an example, consider an agent that interacts with the World Wide Web via a computer terminal. Each action could encode a keystroke or mouse click or movement, while observations could take the form of pixels rendered by a monitor. In such a context, the environment would likely be far more complex than the agent.

Environment dynamics are characterized by a function ρ, which assigns a probability $\rho(o|H_t, A_t) = \mathbb{P}(O_{t+1} = o|\mathcal{E}, H_t, A_t)$ to each observation $o \in \mathcal{O}$. Hence, an environment is specified by a tuple $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$, with fixed sets $\mathcal{A}$ and $\mathcal{O}$ and an observation probability function ρ. In order to accommodate complex real environments, such formulation enables us to relax several restrictive assumptions commonly made in the literature:

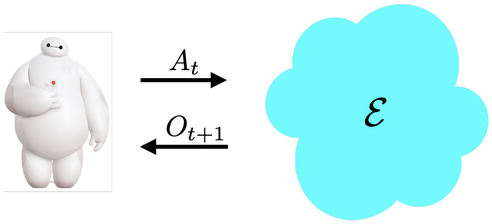

Figure 2: The agent-environment interface.

1. We do not assume the environment is a Markov decision process (MDP), which would require observation probabilities $\rho(\cdot|H_t, A_t)$ to depend on history only through the most recent observation.
2. We do not assume that the environment exhibits episodic behavior, which would require that the environment occasionally "renews."
3. We do not assume that the performance of an optimal policy can be accurately estimated within a manageable time frame. In a complex environment, the required time can be intractably large or even infinite.
4. We do not require that the agent be supplied with the duration T of operation as input. We consider instead a single endless stream of experience, calling for agents to perform well over any long horizon.

1.2 Policies and Performance

A *policy* π is a mapping from histories to action probabilities, with the probability assigned to action a at history h denoted by $\pi(a|h)$. Let $\mathcal{P}$ denote the set of all policies. We denote by $\widehat{\pi} \in \mathcal{P}$ the policy executed by the agent. Agent design amounts to specifying this policy, typically in terms of an algorithm that samples each action A_t according to $\widehat{\pi}(\cdot|H_t)$.

The designer's preferences are expressed in terms of a reward function r. For each history H_t, action A_t, and observation O_{t+1}, this function prescribes a reward $R_{t+1} = r(H_t, A_t, O_{t+1})$. We will characterize the performance of a policy in terms of expected rewards. To formalize this notion, we build on a general probabilistic framework, the details of which are presented in Appendix A. In this framework, actions A_t and observations O_{t+1} are random variables. The observation probability function ρ is also a random variable, as it is unknown to the agent designer, and consequently, the environment $\mathcal{E}$ is a random variable. As formally defined in the appendix, we use a subscript to indicate that a probability or an expectation is evaluated with actions selected by a particular policy. For example, action probabilities satisfy $\mathbb{P}_\pi(A_t = a|H_t) = \pi(A_t = a|H_t)$, and the expected return over T timesteps under policy π is written as $\mathbb{E}_\pi[\sum_{t=0}^{T-1} R_{t+1}]$. When expressing probabilities and expectations under $\widehat{\pi}$, we suppress subscripts. For example, $\mathbb{P}(A_t = a|H_t) = \mathbb{P}_{\widehat{\pi}}(A_t = a|H_t) = \widehat{\pi}(a|H_t)$ and $\mathbb{E}\big[\sum_{t=0}^{T-1} R_{t+1}\big] = \mathbb{E}_{\widehat{\pi}}\big[\sum_{t=0}^{T-1} R_{t+1}\big]$. With this notation, we denote the average reward of a policy $\pi \in \mathcal{P}$ by

$$\lambda_\pi = \liminf_{T\to\infty} \mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=0}^{T-1} R_{t+1}\middle|\mathcal{E}\right], \tag{1}$$

and the optimal average reward by $\lambda_* = \sup_{\pi\in\mathcal{P}} \lambda_\pi$. We quantify the agent's performance relative to a *reference policy* $\pi \in \mathcal{P}$ over T timesteps in an environment $\mathcal{E}$ in terms of regret:

$$\text{Regret}_\pi(T) = \mathbb{E}\left[\sum_{t=0}^{T-1}\big(\lambda_\pi - R_{t+1}\big)\middle|\mathcal{E}\right]. \tag{2}$$

We will also consider a notion of regret relative to a *reference policy class* $\mathcal{P}' \subseteq \mathcal{P}$:

$$\text{Regret}_{\mathcal{P}'}(T) = \sup_{\pi\in\mathcal{P}'} \text{Regret}_\pi(T). \tag{3}$$

Note that $\text{Regret}_{\mathcal{P}}(T)$ is simply the regret relative to the optimal average reward λ_*, and if there exists an optimal policy π_* then $\text{Regret}_{\pi_*}(T) = \text{Regret}_{\mathcal{P}}(T)$. Note that the expressions defining regret are random variables, as they depend on the environment $\mathcal{E}$. From the perspective of an agent designer, a reasonable goal would be to attain low expected regret $\mathbb{E}\big[\text{Regret}_{\mathcal{P}}(T)\big]$ for all long durations T, where the expectation is taken over randomness in $\mathcal{E}$. However, in this paper, we will examine the performance of a fixed agent in the case where the distribution of $\mathcal{E}$ is a point mass, i.e. $\mathbb{P}(\mathcal{E} = e) = 1$, where e can be any environment. As such, we will establish almost sure upper bounds of the random variable $\text{Regret}_{\mathcal{P}}(T)$. Naturally, such bounds would imply upper bounds on $\mathbb{E}\big[\text{Regret}_{\mathcal{P}}(T)\big]$, but we note that in practice, agent designers are not restricted to fixed agents and should be able to leverage prior knowledge on $\mathcal{E}$ to improve agent design. This topic is nevertheless beyond the scope of this work.

Our regret bounds necessarily depend on the time required to assess policies. In a complex environment, the time required to assess optimal or near-optimal policies can be arbitrarily large or even infinite. As such, we develop bounds that depend instead on the time required to assess policies in reference classes. In particular, our bounds indicate that, as time progresses and the agent accumulates experience, it can perform well relative to policies that take longer to assess.

1.3 A Simple Agent

A practical agent must operate with bounded memory and per-timestep computation. With these constraints, the agent cannot retain and repeatedly process an ever-growing history. Rather, the agent maintains only an *agent state* X_t that suffices to produce its actions. Since X_t represents all the agent retains from history, it must be updated incrementally, according to

$$X_{t+1} = \widehat{f}(X_t, A_t, O_{t+1}, U_{t+1}),$$

for some agent state update function $\widehat{f}$, where U_{t+1} represents algorithmic randomness. As discussed in Lu et al. (2021), in popular agent designs (e.g., DQN (Mnih et al., 2015), MuZero (Schrittwieser et al., 2020), MPO (Abdolmaleki et al., 2018; Song et al., 2020)), the agent state can be partitioned into three components:

$$\textbf{agent state } X_t = \Big(\textbf{situational state } S_t, \textbf{epistemic state } P_t, \textbf{algorithmic state } Z_t\Big).$$

The situational state is meant to capture salient information about the agent's current situation in the environment. The epistemic state retains the agent's knowledge about the environment. The algorithmic state can record information unrelated to the environment, such as readings from the agent's internal clock or internally generated random numbers. Rewards computed by such an agent depend on history through the situational state. Letting $\mathcal{S}$ denote the finite set of situational states, the reward function takes the form $r : \mathcal{S}\times\mathcal{A}\times\mathcal{O} \to \mathbb{R}$, generating rewards according to $R_{t+1} = r(S_t, A_t, O_{t+1})$. Note that such reward function is a special case of that introduced in Section 1.2, where $R_{t+1} = r(H_t, A_t, O_{t+1})$. In this work, we focus on the case where rewards are computed from situational states instead of directly from histories. Since the set of situational states is finite, allowing rewards to depend on situational states ensures that the computation of reward function r does not require infinite memory.

In this paper, we design and analyze a simple agent, which can engage with any environment after being instantiated with the following inputs:

1. an initial situational state $S_0 \in \mathcal{S}$ and an update function $f : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \mapsto \mathcal{S}$,
2. a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \mapsto [0, 1]$.

While we will provide a precise specification later in the paper, here we offer a rough description of how the agent operates. Our agent updates its situational state according to $S_{t+1} = f(S_t, A_t, O_{t+1})$ and uses this to compute rewards, as illustrated in Figure 3. Note that function f is assumed to be given to the agent at the beginning and fixed throughout the agent's lifetime.

With the sequence of situational states defined recursively through f, the situational state dynamics need not be Markovian; in particular, we can have $\mathbb{P}(S_{t+1} = s|\mathcal{E}, S_t, A_t) \neq \mathbb{P}(S_{t+1} = s|\mathcal{E}, H_t, A_t)$. Our agent's epistemic state $P_t = (Q_t, N_t)$ is comprised of an action value function $Q_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and a count function $N_t : \mathcal{S} \times \mathcal{A} \to \mathbb{Z}_+$. Our agent's algorithmic state includes only the current time t. Action values Q_t are updated via an optimistic discounted Q-learning algorithm, with the discount factor and degree of optimism increasing over time. The agent updates N_t to track visitation counts, which are used to determine a suitable degree of optimism. Each action A_t is sampled uniformly from the set $\arg\max_{a\in\mathcal{A}} Q_t(S_t, a)$ of greedy actions. Hence, at any time, the agent can be seen as executing a policy $\pi_t(\cdot|H_t)$ for which action probabilities depend on the history H_t only through the situational state S_t.

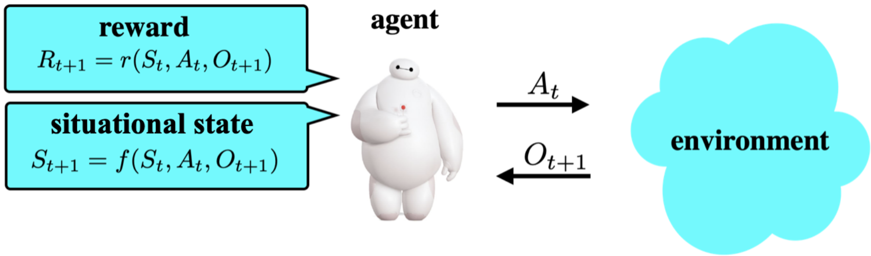
reward
$R_{t+1} = r(S_t, A_t, O_{t+1})$
situational state
$S_{t+1} = f(S_t, A_t, O_{t+1})$
agent
A_t
O_{t+1}
environment

Figure 3: Our agent maintains a situational state and uses that to compute rewards.

It is worth emphasizing that our agent is not designed to offer state-of-the-art performance in simulated or real environments. Rather, our motivation is to design an agent that is amenable to theoretical analysis, with an aim to generate insights that inform the design of future state-of-the-art agents.

1.4 Example: Service Rate Control

Let us consider a didactic example that, while exceedingly simple, *serves* to elucidate our notation and framework. The example involves an agent operating a service station, as illustrated in Figure 4. At each time, there can be at most one customer present, and the agent applies a service mode – *fast* or *slow*. Each customer pays $1 upon arrival. No cost is incurred when the slow mode is applied or when there is no customer being served. The fast mode of service incurs a cost of $0.50 per timestep. To maximize average reward, an agent must make choices that balance revenue against the cost of service.

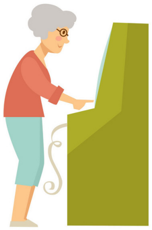

Figure 4: A service station serving a customer.

This problem is one of service rate control, as studied in operations research (see, e.g., (Weber and Stidham Jr, 1987; Stidham Jr and Weber, 1989; Jo, 1989; Sennott, 2009)). However, such work has tended to focus on agents that are effective when applied to particular stylized models that govern arrival and service rates. Our approach instead adapts to *any* statistical structure, and as such, does not suffer from misspecification. Work on reinforcement learning for control of queueing systems (Moallemi et al., 2008; Raeis et al., 2021) shares this spirit. We should note that it is only in order to convey ideas in a simple and transparent manner that we focus on such a simple service system: our agent can be applied to much more complex environments, for example, involving multiple servers and queues.

1.4.1 Agent-Environment Interface

From the agent's perspective, the service station can be viewed as an environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$. Actions $\mathcal{A} = \{\text{fast}, \text{slow}\}$ identify service modes and observations $\mathcal{O} = \{\text{arrival}, \neg\text{arrival}\} \times \{\text{departure}, \neg\text{departure}\}$ indicate arrivals and departures. Hence, A_t is the service mode applied over timestep t and O_{t+1} indicates any arrival or departure occurring by the end of the timestep. The function ρ specifies observation probabilities conditioned on history, which are initially unknown. For example, the designer may be uncertain about customer arrival rates and how they depend on history.

1.4.2 Situational State Dynamics

We consider a situational state $S_t \in \mathcal{S} = \{0, 1\}$ that simply indicates presence of a customer. Since observations record arrivals and departures, there is a function f for which $S_{t+1} = f(S_t, A_t, O_{t+1})$. The service is initially vacant, so $S_0 = 0$. Profit can be written as $R_{t+1} = r(S_t, A_t, O_{t+1})$ for some function r. Of special interest are policies that select actions based only on situational state; that is, the set of policies for which $\mathbb{P}_\pi(A_{t+1}|H_t) = \mathbb{P}_\pi(A_{t+1}|S_t)$. Let us denote this set by $\tilde{\mathcal{P}}$.

1.4.3 Baseline Agents

We consider agents designed to learn policies within $\tilde{\mathcal{P}}$. While this class of policies is simple enough so that an agent could perform a nearly exhaustive search, we will restrict attention to approaches that can scale to settings involving much larger sets of situational states. Two simple agents of this kind will serve as baselines for comparison. Let $\pi_\epsilon \in \tilde{\mathcal{P}}$ be a policy that in the absence of a customer applies the slow mode, and otherwise samples slow or fast with probabilities $1 - \epsilon$ and ϵ. Each of our baseline agents begins by executing π_ϵ, with $\epsilon = 0$ – that is, by applying the slow service mode over every timestep. The first agent increases ϵ after gathering data over a long duration and using that data to estimate the arrival rate, if the estimate warrants increasing the service rate. This agent's analysis is *static*, in the sense that it does not entail any experimentation and instead assumes the arrival rate will remain fixed. The second agent additionally tries a small value of $\epsilon > 0$ for some duration in order to estimate the derivative $\mathrm{d}\lambda_{\pi_\epsilon}/\mathrm{d}\epsilon$ of average reward. If this derivative is positive, it increases ϵ. One significant difference relative to the first agent is that, through its use of the derivative, this second agent anticipates the impact small increases in ϵ bear on the arrival rate. As such, the second agent is representative of approaches used in the policy gradient literature, as discussed in (Sutton and Barto, 2018) and references therein.

1.4.4 Environment Dynamics

We will study the behavior of agents given environment dynamics characterized by a specific observation probability function ρ_*, with the corresponding realized environment denoted by $e_* = (\mathcal{A}, \mathcal{O}, \rho_*)$. We provide a detailed specification in Appendix B and assume for the purposes of this analysis that $\mathbb{P}(\mathcal{E} = e_*) > 0$. In this environment, the customer arrival rate depends on the maximum service time experienced among the most recent dozen customers served. The idea here is that long service times hurt reputation, which in turn reduces the number of customers seeking service. Service times are impacted by the agent's choices: with the fast mode, service is always completed in a single timestep, while with the slow mode, service is completed over the next timestep with probability 1/2. Given our specification of ρ_*, the maximal average reward is \$0.50 per timestep. This is achieved by applying the fast mode of service over every timestep, in which case each customer is served over a single timestep and a new customer arrives as soon as the previous one departs.

1.4.5 Performance

Figure 5 plots cumulative moving average rewards attained by an optimistic Q-learning agent, which we will later present, averaged over two hundred independent simulations. The figure also plots the maximum average reward and the average reward attained by always applying the slow service mode. The baseline agents never choose to deviate from the slow service mode and therefore realize average reward close to the latter.

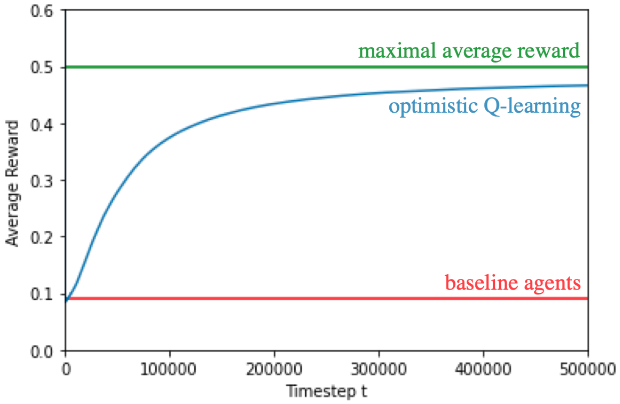

Figure 5: Cumulative moving average rewards attained by an optimistic Q-learning agent, the maximal average reward, and the average reward attained by always applying the slow service mode, which approximates behavior of the baseline agents.

These results convey potential benefits of an agent designed to address general environments. The optimistic Q-learning agent eventually figures out that its choices drive future arrival rates sufficiently to improve performance. The baseline agents do not demonstrate that level of sophistication.

2. Contributions and Related Literature

The formulation of agent-environment interactions studied in this paper is very general, involving a single stream of experience, without restrictive assumptions commonly made in the literature, as discussed in Section 1.1. It is important for theoretical work to relax such assumptions if it is to inform the design of agents that can operate in complex real environments. While this formulation bears close resemblance to those studied by McCallum (1995); Hutter (2004); Daswani et al. (2013, 2014); Lu et al. (2021), such formulations have not been a focus of work on provably efficient reinforcement learning. Our work is the first to extend regret analysis tools to this setting.

Building upon such general modeling of agent-environment interactions, this paper makes a range of contributions, innovating on framing of learning objectives, agent design, and performance analysis, as well as generating qualitative insights that can inform practical agent design. In this section, we summarize these contributions and their relations to prior literature.

2.1 Framing of Learning Objectives

In the literature on provably efficient reinforcement learning, it is common to study agent performance through regret analysis. However, the manner in which regret bounds are typically framed does not suitably accommodate complex environments. We develop concepts that allow us to frame meaningful learning objectives for such contexts.

2.1.1 Averaging Time

To intelligently choose between policies, an agent must assess their relative performance. Regret bounds established in the literature typically reflect this requirement via dependence on statistics that bound the time required to assess an optimal policy. Examples include, episode duration (Osband et al., 2013, 2019; Azar et al., 2017; Jin et al., 2018), diameter (Jaksch et al., 2010), or span (Bartlett and Tewari, 2012; Ouyang et al., 2017; Wei et al., 2020). In a complex environment, the time required to assess an optimal policy can be intractably large or even infinite. As such, we will derive bounds that instead depend on *reward averaging times* of policies in reference classes.

Let $\lambda_\pi(h,T)$ denote the expected average reward over T timesteps starting at history h so that $\lambda_\pi = \liminf_{T\to\infty} \lambda_\pi(H_0,T)$. We define the reward averaging time τ_π of a policy $\pi \in \mathcal{P}$ to be the smallest value $\tau \in [0,\infty]$ such that

$$|\lambda_\pi(h,T) - \lambda_\pi| \leq \frac{\tau}{T}, \tag{4}$$

for all $h \in \mathcal{H}$ and $T \geq 0$. This is closely related to a concept introduced in Kearns and Singh (2002), which defines a notion of averaging time that is a function of a tolerance parameter associated with the error $|\lambda_\pi(h,T) - \lambda_\pi|$. Our definition relies instead on a single scalar statistic τ_π. If we let π_* be an optimal policy, τ_{π_*} is essentially equivalent to the notion of span introduced in (Bartlett and Tewari, 2012).

Note that there might exist a history $h \in \mathcal{H}$ and a policy π such that $\lambda_\pi(h,T)$ does not converge to λ_π as $T \to \infty$, resulting in $\tau_\pi = \infty$. While we do not need to rule out such cases in our analysis, we will see later that for certain policies π, $\tau_\pi = \infty$ leads to a vacuous regret bound $\mathrm{Regret}(T) \leq \infty$. This makes intuitive sense, since an infinite reward averaging time implies that the agent might not be able to reverse its previous mistakes, and consequently could "get stuck" in suboptimality indefinitely.

2.1.2 Distortion

A distinctive element of our formulation is in the agent's instantiation with a situational state update function. This serves to simplify the agent's experience by extracting useful features from history and enables productive behavior in arbitrarily complex environments.

In particular, instead of the number of environment states, our regret bounds will depend on the number of situational states and the distortion incurred by using them to predict optimal discounted value.

In what follows, we will let $\phi(h)$ denote the situational state that would be generated after experiencing a history $h \in \mathcal{H}$. Specifically, $\phi : \mathcal{H} \mapsto \mathcal{S}$ is a mapping defined such that, if

$$h = (a_0, o_1, a_1, o_2, \ldots, a_{t-1}, o_t)$$

is a history sequence containing t timesteps, and $\{s_1, \ldots, s_t\}$ is the sequence of situational states generated by h, i.e.

$$\begin{aligned} s_1 &= f(S_0, a_0, o_1), \\ s_2 &= f(s_1, a_1, o_2), \\ &\cdots \\ s_t &= f(s_{t-1}, a_{t-1}, o_t), \end{aligned}$$

then $\phi(h) = s_t$. It is worth emphasizing that the mapping ϕ is determined by both S_0 and f. To avoid cluttering, we will suppress such dependence in the notations.

For each discount factor $\gamma \in [0, 1)$, history $h \in \mathcal{H}$, and action $a \in \mathcal{A}$, denote the optimal discounted action value by $Q_*^\gamma(h, a)$. The discount factor γ weights the reward realized after k timesteps by γ^k and can be thought of as prescribing an effective planning horizon of $\tau = 1/(1 - \gamma)$. We define the *distortion* for an effective planning horizon τ by

$$\Delta_\tau = \max_{(s,a) \in \mathcal{S} \times \mathcal{A}} \left(\sup_{h \in \mathcal{H}: \phi(h) = s} Q_*^\gamma(h, a) - \inf_{h \in \mathcal{H}: \phi(h) = s} Q_*^\gamma(h, a) \right), \tag{5}$$

where $\gamma = 1 - 1/\tau$. This is the maximum difference between optimal action values across histories that lead to the same situational state and offers a measure of error introduced when predicting optimal action values based on situational state instead of history. This sort of distortion measure has long been used in analysis of approximate dynamic programming algorithms that aggregate environment states (Gordon, 1995; Tsitsiklis and Van Roy, 1996; Van Roy, 2006), though in this case we instead aggregate histories.

Our framing requires a stronger notion of distortion, defined by

$$\overline{\Delta}_\tau = \sup_{\tau' \geq \tau} \Delta_{\tau'}. \tag{6}$$

This quantifies the accuracy with which situational states can predict optimal action values for all planning horizons of duration τ or greater. This distortion measure offers a useful statistic for characterizing performance of agents that are able to plan effectively over increasing horizons as data accumulates.

It is worth mentioning that the aggregation of history and the definition of distortion bear a resemblance to the notion of Q^*-irrelevant abstraction defined in (Li et al., 2006). While Li et al. (2006) presume that the environment is modeled by a *finite-state* MDP, in this work the set of histories is *infinite*. We would like to highlight the conceptual value of such a distinction, since our goal is analysing a concrete agent that operates with finite memory. And that is why in this work, instead of simply assuming that there exists an abstraction function mapping from histories to situational states, we consider a situational state update function f that allows the agent to recursively compute the next situational state based on the previous one. In such a way, we are able to ensure that the agent is able to maintain and update its situational state with finite memory. It is worth mentioning that such a distinction does not affect regret analysis.

Readers familiar with RL theory might also connect this work with (Li, 2009), which comprehensively studies the Q^*-irrelevant abstraction under the PAC-MDP framework. We note that the result in this work applies to more general settings, since it does not impose a fixed discount factor on the sequence of rewards. Moreover, our theory shows that, compared with delayed Q-learning, which is the algorithm studied in (Li, 2009), the ultimate per-period performance loss of our agent enjoys a much more graceful dependence on the abstraction gap (defined in Section 8.2.3 in (Li, 2009)).

Note that the environment that we defined can be viewed as an MDP with infinitely many MDP states, and the aggregation of history that we are considering amounts to state aggregation. In a broader context, state aggregation can be viewed as a form of function approximation, and a large body of theoretical literature has been dedicated to function approximation in MDPs. One of the most relevant works in this category is (Jiang et al., 2017), in which the authors show that if the environment dynamics *approximately* admit a low-rank factorization and the class of functions to approximate state values is in accordance with such a factorization, then the number of samples required to learn an ϵ-optimal policy in the environment is polynomial with respect to the rank. Specifically, this result implies that the sample complexity of learning a near-optimal policy with state aggregation is polynomial in the number of aggregates. Other recent works along this line include (Jin et al., 2020), (Agarwal et al., 2020), (Wang et al., 2020), and (Zanette et al., 2020). However, algorithms in these works with sample complexity guarantees mainly serve theoretical purposes and have little bearing on practical agent design, whereas one of the main targets of our work is to shed light on elements in empirically successful agents. Readers interested in theoretical results on function approximation can refer to the recent paper (Zhou et al., 2021), which provides a comprehensive review.

A limitation of our framing is in its use of a fixed situational state update function f. While this is consistent with the manner in which some practical agents operate – for example, the DQN agent of (Mnih et al., 2015) takes its situational state to be some number of recent video frames – there is likely value to adapting the way in which situational state is updated based on what is learned about the environment, which is encoded in the agent's epistemic state. The MuZero agent (Schrittwieser et al., 2020) does adapt its update function in this way. That agent represents the update function in terms of a recurrent neural network, with weights adapted over time based on interactions with the environment. Despite this limitation, our framing represents a significant step, advancing the mathematical literature in a direction that may inform future agent designs.

2.1.3 Reference Classes

We frame as agent design objectives a notion of competing effectively with policies from particular reference classes, with effectiveness measured through the lens of regret as a function of averaging times, distortions, and $\mathcal{S}$ and $\mathcal{A}$. As opposed to a single scalar objective, the spirit here is to offer a framework for studying trade-offs and to derive interpretable regret bounds that generate insight that can inform agent designers. To understand this spirit, it may be helpful to draw an analogy with the field of optimization. While optimization problems are framed in terms of precise scalar objectives, the design of optimization algorithms tends to be formulated in terms of measures of computational complexity and solution quality as a function of numbers of decision variables and constraints, as well as other salient problem characteristics.

One reference class we introduced earlier, denoted by $\tilde{\mathcal{P}}$, consists of all policies π for which $\mathbb{P}_\pi(A_{t+1}|H_t) = \mathbb{P}_\pi(A_{t+1}|S_t)$. In other words, these are the policies that select actions based on situational state instead of history. Ideally, the situational state should suffice for predicting what the agent requires to make optimal decisions, in which case $\tilde{\mathcal{P}}$ would include an optimal policy. Let $\tilde{\pi} \in \tilde{\mathcal{P}}$ be a policy for which $\lambda_{\tilde{\pi}} = \sup_{\pi \in \tilde{\mathcal{P}}} \lambda_\pi$. We will think of the agent as trying to learn a high-performing policy from within $\tilde{\mathcal{P}}$, and as such, it is natural to expect that $\lambda_{\widehat{\pi}} \leq \lambda_{\tilde{\pi}}$, where $\widehat{\pi}$ is the policy executed by the agent, as is defined in Section 1.2. As we will discuss in Appendix D, there exist environments and situational state dynamics such that $\lambda_* - \lambda_{\tilde{\pi}} \geq \overline{\Delta}_{\tau_{\tilde{\pi}}}$, and consequently, if $\lambda_{\widehat{\pi}} \leq \lambda_{\tilde{\pi}}$, the average regret satisfies

$$\liminf_{T\to\infty} \frac{\text{Regret}_{\mathcal{P}}(T)}{T} = \lambda_* - \lambda_{\widehat{\pi}} \geq \lambda_* - \lambda_{\tilde{\pi}} \geq \overline{\Delta}_{\tau_{\tilde{\pi}}}. \tag{7}$$

In light of this fundamental limitation of the policy class $\tilde{\mathcal{P}}$, we frame as an objective optimizing the dependence of average regret on the distortion $\overline{\Delta}_{\tau_{\tilde{\pi}}}$.

The aforementioned objective calls for the agent to eventually compete effectively with the best policy among those that select actions based on situational state. A second objective we frame calls for the agent to attain that eventual level of performance quickly. The time required depends on the time it takes to compare policies, which can be bounded by averaging times. As discussed earlier, it is important to avoid dependence on the averaging time of an optimal policy as well as the number of environment states, each of which can be intractably large or even infinite in a complex environment. We instead consider regret bounds that depend on the number of situational states and the averaging time of $\tilde{\pi}$. In particular, we consider regret bounds of the form

$$\text{Regret}_{\mathcal{P}}(T) \leq \texttt{foobar}(T, \mathcal{S}, \mathcal{A}, \tau_{\tilde{\pi}}, \overline{\Delta}_{\tau_{\tilde{\pi}}}),$$

where `foobar` is a metasyntactic function and, with some abuse of notation, we use $\mathcal{S}$ and $\mathcal{A}$ to denote set cardinalities. An understanding of how regret depends on the arguments can guide designs that more quickly learn to perform well relative to $\tilde{\pi}$.

We additionally consider, for each $\tau \geq 1$, a reference class $\mathcal{P}_\tau = \{\pi \in \mathcal{P} : \tau_\pi \leq \tau\}$, consisting of policies with averaging times no greater than τ. For these classes, we consider bounds of the form

$$\text{Regret}_{\mathcal{P}_\tau}(T) \leq \texttt{foobar}(T, \mathcal{S}, \mathcal{A}, \tau, \overline{\Delta}_\tau),$$

for a different function `foobar`. Such bounds offer insight into how agents can quickly learn to perform well relative to policies with any particular averaging time. An agent ought to be able to compete against policies in $\mathcal{P}_\tau$ within some time that grows with τ, and such regret bounds reflect that relationship and draw attention to balancing associated trade-offs.

2.2 Agent Design

Our agent implements a variant of Q-learning (Watkins, 1989). Early analyses of Q-learning focused on asymptotic convergence guarantees under the assumption that the agent tries each action at each environment state infinitely often (Watkins, 1989; Watkins and Dayan, 1992; Tsitsiklis, 1994; Jaakkola et al., 1994). More recently, research on Q-learning has merged with concepts from the literature on regret analysis, leading to provably efficient variations (Jin et al., 2018; Wei et al., 2020). These *optimistic* Q-learning agents ensure a level of efficiency by using carefully chosen step sizes and perturbing action value updates to maintain optimistic estimates. This merging presents an opportunity to bridge the efficient reinforcement learning literature with practical agent design, as Q-learning is more aligned with the state-of-the-art than other algorithms that have been studied in the mathematical literature.

While we build on this line of work to design a new optimistic Q-learning agent that is suitable for complex environments, our agent relies on several algorithmic innovations. While the agents of (Jin et al., 2018; Wei et al., 2020) maintain action values at each *environment state*, ours maintains action values at each *situational state*. Further, the algorithm of Jin et al. (2018) is designed for fixed-horizon episodic environments and that of Wei et al. (2020) operates with a fixed discount factor that depends on the horizon T. Our algorithm is designed for general environments, and while it does make use of a discount factor, the discount factor increases over time to generate effective behavior over increasingly long planning horizons. Further, while step sizes used in (Jin et al., 2018; Wei et al., 2020) depend on the horizon T, our agent is designed to operate indefinitely rather than over a predetermined horizon T, and as such, uses step sizes that do not depend on T.

2.3 Performance Analysis

Critical contributions of this paper lie in our performance analysis. While the results will be presented in Section 4, here we discuss a few key implications. Firstly, we establish that, if $\tau_{\tilde{\pi}} < \infty$, our agent attains average regret

$$\limsup_{T\to\infty} \frac{\mathrm{Regret}_{\mathcal{P}}(T)}{T} = \lambda_* - \lambda_{\widehat{\pi}} \leq 4\overline{\Delta}_{\tau_{\tilde{\pi}}}. \tag{8}$$

This is exactly four times the lower bound of (7). It is also interesting to relate this upper bound to Theorem 19 in Section D, which indicates that, for all $\epsilon > 0$, there exists an environment, a set of situational states, a situational state update function and a reward function, such that particular approximate dynamic programming (ADP) methods one might apply (e.g., Whitt (1978); Gordon (1995); Tsitsiklis and Van Roy (1996); Munos and Szepesvári (2008)) yield a policy μ for which $\lambda_* - \lambda_\mu \geq \tau_{\tilde{\pi}}\overline{\Delta}_{\tau_{\tilde{\pi}}} - \epsilon$, which is generally far worse that $4\overline{\Delta}_{\tau_{\tilde{\pi}}}$. Further, Van Roy (2006) suggests that a *temporal-difference fixed point* would yield a policy μ' that satisfies $\lambda_* - \lambda_{\mu'} \leq \overline{\Delta}_{\tau_{\tilde{\pi}}}$, but it is not known whether such a fixed point can be determined by a computationally tractable algorithm. It is intriguing that our agent – which is computationally tractable and itself based on a temporal-difference method – attains average regret within a factor of four of that. We provide a brief discussion on this topic at the end of Appendix D.

Specialized to the case where the distortion $\overline{\Delta}_{\tau_{\tilde{\pi}}} = 0$, our analysis implies the following:

$$\mathrm{Regret}_{\mathcal{P}}(T) \lesssim \left(\sqrt{\mathcal{SA}} + \tau_{\tilde{\pi}}\right) T^{4/5} + \mathcal{SA}T^{1/5} + \tau_{\tilde{\pi}}^5, \tag{9}$$

where $\lesssim$ indicates omission of constant and poly-logarithmic factors. In this case, since situational states enable exact predictions of optimal value, the regret grows sublinearly in T, meaning that the agent eventually learns a globally optimal policy. The dependence on T is worse than the usual $T^{1/2}$ scaling, which appears in results pertaining to episodic environments (Jin et al. (2018); Zhang et al. (2020)). In our formulation, a $T^{1/2}$ scaling is unachievable without additional problem-dependent terms in the regret bound that scale exponentially with $\mathcal{S}$ and $\mathcal{A}$ (Jaksch et al. (2010); Wei et al. (2020)). It is worth noting that, while Wei et al. (2020) considers an average reward objective, though with zero distortion, and provides a regret bound that scales with $T^{2/3}$ rather than $T^{4/5}$, the algorithm crucially relies on knowledge of a fixed duration T. Our agent and analysis can also be modified to attain a $T^{2/3}$ scaling given a fixed duration T.

Combining (8) and (9), we can see that besides $\mathcal{S}, \mathcal{A}$ and T, the bound only depends on $\tau_{\tilde{\pi}}$, the reward averaging time of the best policy in the reference class. Previous regret bounds for tabular reinforcement learning scale with the number of states or the reward averaging time of an optimal policy. In a complex environment, these quantities can be arbitrarily large or infinite. Interestingly, our bound ensures that the agent is able to learn efficiently in spite of that.

We further establish that, for all $\tau \geq 1$,

$$\mathrm{Regret}_{\mathcal{P}_\tau}(T) \lesssim \left(\sqrt{\mathcal{SA}} + \tau\right) T^{4/5} + \mathcal{SA}T^{1/5} + \tau^5 + \overline{\Delta}_\tau T. \tag{10}$$

Recall that $\mathcal{P}_\tau$ is the class of policies with reward averaging times no greater than τ and $\mathrm{Regret}_{\mathcal{P}_\tau}(T)$ quantifies regret relative to that class. This bound offers insight into how, over time, the agent can learn to perform competitively against policies with larger reward averaging times. To understand this, let us focus on a special case where $\Delta_\tau = 0$ for all τ. In this case, the bound implies that, for all $\epsilon \in (0, 1)$, setting $\tau = \epsilon T^{1/5}$,

$$\limsup_{T\to\infty} \frac{\mathrm{Regret}_{\mathcal{P}_\tau}(T)}{T} \lesssim \epsilon. \tag{11}$$

Hence, for sufficiently large T, the agent's average reward approximates that of the best policy with reward averaging time no greater than $\epsilon T^{1/5}$.

2.4 Qualitative Insights

While it shares elements common to state-of-the-art agents, our agent is far simpler. Our motivation was not to produce another state-of-the-art agent, but rather to offer a context amenable to analyses that can inform design of future state-of-the-art agents. We now discuss some key insights supported by our results.

First of all, our results demonstrate that it is possible for an agent to operate effectively within a tractable time frame through a single endless stream of interactions with an arbitrarily complex environment. Previous results either rely on the fact that the environment mixes in a modest amount of time (Jin et al., 2018; Jaksch et al., 2010; Zhang et al., 2020) or that the horizon T of operation is fixed and known to the agent (Wei et al., 2020). Further, previous results focus on MDPs, and while there has also been related work on POMDPs (Jafarnia-Jahromi et al., 2021; Kara and Yuksel, 2020; Subramanian et al., 2022), those results are relevant only when there is a tractable number of environment states. Our bounds do not depend on the environment's mixing time or number of states. Among other things, our results imply that an agent can perform well even in an environment that is so complex that the performance of an optimal policy would take forever to estimate.

Secondly, we are the first to establish that an algorithm with average regret bounded by a constant multiple of distortion approaches such asymptotic performance within a tractable time frame. An example in (Van Roy, 2006) implies that certain common ADP algorithms, which require that environment dynamics be known, do not output a policy $\pi \in \tilde{\mathcal{P}}$ such that $\lambda_* - \lambda_\pi$ is within a constant multiple of $\overline{\Delta}$. Indeed, previous analyses of ADP algorithms instead bound $\lambda_* - \lambda_\pi$ by a multiple of $\tau\overline{\Delta}$, where τ is some notion of averaging time that depends on environment complexity (Whitt, 1978; Gordon, 1995; Tsitsiklis and Van Roy, 1996). This scaling by τ is far worse than a constant, with τ becoming arbitrarily large in complex environments. In real environments, it is impractical to attain zero distortion, and therefore, some degree of impact on performance is inevitable. Our result offers insight into how to avoid scaling by τ.

An intriguing aspect of our agent design is that the effective planning horizon increases with time, allowing the agent to eventually optimize performance over arbitrarily long horizons. Our agent's effective planning horizon scales with $t^{1/5}$, and this rate leads to our regret bound. The notion that planning may benefit from restricting the effective horizon based on the quantity of data gathered has also been observed by Jiang et al. (2015).

Our regret bounds depend on the distortion induced by a fixed situational state update function. However, some state-of-the-art agents leverage the ability of neural networks to adapt this update function (Nachum et al., 2018; Schrittwieser et al., 2020). While our results do not directly address such adaptation, they do offer insight into the way in which that can influence agent performance.

3. Value Functions

Central to the theory of MDPs are value functions. While value functions are typically considered to be functions of environment state, we consider instead functions of history. In this section we define these value functions and characterize them as solutions to Bellman equations.

Throughout this section, we consider a fixed discount factor $\gamma \in [0,1)$ and environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$. To simplify notation, we will use (h,a,o) to denote the history generated by concatenating action a and observation o to history h. For each $a \in \mathcal{A}$, we define an $\mathcal{H} \times \mathcal{H}$ transition matrix P_a, with entries

$$P_{ahh'} = \begin{cases} \rho(o|h,a) & \text{if } h' = (h,a,o) \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

for each $h, h' \in \mathcal{H}$. Similarly, for each policy $\pi \in \mathcal{P}$, we define a transition matrix P_π, with

$$P_{\pi hh'} = \sum_{a \in \mathcal{A}} \Big(\pi(a|h) \cdot P_{ahh'} \Big), \quad \forall h, h' \in \mathcal{H}, \tag{13}$$

for each $h, h' \in \mathcal{H}$. Further, for each action $a \in \mathcal{A}$ and policy $\pi \in \mathcal{P}$, let $\overline{r}_a$ and $\overline{r}_\pi$ be $\mathcal{H}$-dimensional vectors, with components given by

$$\overline{r}_{ah} = \sum_{o \in \mathcal{O}} \Big(\rho(o|h,a) \cdot r\big(\phi(h), a, o\big) \Big) \qquad \text{and} \qquad \overline{r}_{\pi h} = \sum_{a \in \mathcal{A}} \Big(\pi(a|h) \cdot \overline{r}_{ah} \Big). \tag{14}$$

For each policy $\pi \in \mathcal{P}$, let

$$V_\pi^\gamma(h) = \sum_{t=0}^{\infty} \Big(\gamma^t \cdot \big(P_\pi^t \overline{r}_\pi\big)(h) \Big) \qquad \text{and} \qquad Q_\pi^\gamma(h,a) = \overline{r}_{ah} + \sum_{h' \in \mathcal{H}} \Big(P_{ahh'} \cdot V_\pi^\gamma(h') \Big). \tag{15}$$

These functions represent expected discounted rewards starting at history h if either all subsequent actions are selected by π or only after an action a is executed. By taking the supremum over policies, we obtain optimal values:

$$V_*^\gamma(h) = \sup_{\pi \in \mathcal{P}} V_\pi^\gamma(h) \qquad \text{and} \qquad Q_*^\gamma(h,a) = \sup_{\pi \in \mathcal{P}} Q_\pi^\gamma(h,a), \quad \forall h \in \mathcal{H}, a \in \mathcal{A}. \tag{16}$$

The following proposition, which follows from Proposition 2.1.1 in Bertsekas (2018), characterizes V_*^γ and Q_*^γ as unique solutions among the set of bounded functions to the Bellman equations.

Proposition 1 *The pair (V_*^γ, Q_*^γ) uniquely solves the system of equations*

$$\begin{array}{ll} V(h) = \max_{a' \in \mathcal{A}} Q(h,a') & \forall h \in \mathcal{H} \\ Q(h,a) = \overline{r}_{h,a} + \gamma \cdot \sum_{h' \in \mathcal{H}} \Big(P_{ahh'} \cdot V(h') \Big) & \forall h \in \mathcal{H}, a \in \mathcal{A}. \end{array}$$

among all pairs of bounded functions $V : \mathcal{H} \to \mathbb{R}$ and $Q : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$.

We close this section with an important lemma, which ties together three concepts relating to a policy π: the long-term expected average reward λ_π, the reward averaging time τ_π, and the discounted value function V_π^γ. The lemma closely resembles Theorem 4.1 in de Farias and Van Roy (2006), and we omit the proof.

Lemma 2 *For all $\pi \in \mathcal{P}$, $h \in \mathcal{H}$ and $\gamma \in [0,1)$, $\left| V_\pi^\gamma(h) - \frac{\lambda_\pi}{1-\gamma} \right| \leq \tau_\pi$.*

This result establishes that the reward averaging time bounds the difference between discounted value $V_\pi^\gamma(h)$ and average reward λ_π scaled by the effective horizon $1/(1-\gamma)$.

4. Agent Design and Performance Analysis

In this section we present and study our optimistic Q-learning agent. Similarly with agents that have demonstrated success in large-scale simulations, ours learns to predict action values. However, rather than a neural network representation, our agent maintains a lookup table containing one prediction per (situational) state-action pair. Actions are selected greedily with respect to these predictions. Upon each observation, the agent incrementally adjusts the prediction assigned to its previous state-action pair based on a temporal difference.

The agent predicts discounted value. However, in order to eventually maximize average reward, the associated discount factor increases over time and approaches one. The idea is for the agent to plan, at any given time, over a particular effective horizon. This horizon increases as the agent gathers more data, which enables planning over longer horizons with greater confidence.

4.1 Discounted Q-Learning

As a prelude to our primary agent, we introduce a simpler one that serves didactic purposes. This simper agent plans over a fixed effective horizon τ and is designed to operate over a fixed duration $T \gg \tau$, with both these variables required as input when instantiating the agent. In particular, the agent executes Algorithm 1 (`discounted_q_learning`). The effective horizon τ prescribes a discount factor $\gamma = 1 - 1/\tau$. The agent starts with an initial situational state S_0. Over each timestep, the agent increments the visitation count $N(s, a)$, computes the next situational state $s' = f(s, a, o)$, and updates the prediction $Q(s, a)$ via a discounted Q-learning iteration, with discount factor $\gamma = 1 - 1/\tau$.

The Q-learning update of Line 11 adjusts the action value in response to a temporal difference. Two elements of this update warrant further discussion. One is the step size α, which is given by $(1 + 2\tau)/(N(s, a) + 2\tau)$. This step size sequence is adapted from that used in Jin et al. (2018) and has a number of desirable properties, as will be established in Lemma 9. In particular, these properties ensure that estimation errors do not accumulate exponentially as the agent updates action values. A second key element is the optimistic boost added to the temporal difference, which is given by $\beta/\sqrt{N(s, a)}$. This term injects optimism to ensure that predictions are likely to be optimistic, in terms of dominating Q_*^γ. As the number of visits $N(s, a)$ to a state-action pair increases, uncertainty around its prediction decreases, and this is reflected in the denominator $\sqrt{N(s, a)}$.

Let $\widehat{\pi}^{\tau,T} \in \mathcal{P}$ be the policy implemented by an agent that executes Algorithm 1 with effective planning horizon τ and operation duration T. Let the regret relative to a reference policy $\pi \in \mathcal{P}$ experienced by this agent over T timesteps be denoted by

$$\mathrm{Regret}_\pi^\tau(T) = \mathbb{E}_{\widehat{\pi}^{\tau,T}}\left[\sum_{t=0}^{T-1} \left(\lambda_\pi - R_{t+1}\right)\Big|\mathcal{E}\right]. \tag{17}$$

Recall that $\tau_\pi = \inf_{T \geq 0} |\lambda_\pi(h, T) - \lambda_\pi|$ is the reward averaging time of policy π and

$$\Delta_\tau = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left(\sup_{h\in\mathcal{H}:\phi(h)=s} Q_*^\gamma(h, a) - \inf_{h\in\mathcal{H}:\phi(h)=s} Q_*^\gamma(h, a)\right),$$

Algorithm 1 `discounted_q_learning`

```
Input:  $S_0$   initial situational state
        $f$     situational state update function
        $r$     reward function
        $\tau$  effective planning horizon
        $T$     duration of operation
$\gamma \leftarrow 1 - 1/\tau$
$\beta \leftarrow \tau^{3/2} \cdot 4\sqrt{\log(2T^2)}$
$s \leftarrow S_0$
$Q(\cdot, \cdot) \leftarrow \tau$
for $t = 1, 2, \ldots, T$ do
    $a \leftarrow$ sample_unif$\big(\arg\max_{a' \in \mathcal{A}} Q(s, a')\big)$
    execute action $a$ and register observation $o$
    $N(s, a) \leftarrow N(s, a) + 1$
    $\alpha \leftarrow \frac{1+2\tau}{N(s,a)+2\tau}$
    $s' \leftarrow f(s, a, o)$
    $Q(s, a) \leftarrow Q(s, a) + \alpha \cdot \left(r(s, a, o) + \gamma \cdot \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a) + \frac{\beta}{\sqrt{N(s,a)}}\right)$
    $Q(s, a) \leftarrow \min(Q(s, a), \tau)$
    $s \leftarrow s'$
end for
```

where $\gamma = 1 - 1/\tau$, is the distortion introduced in predicting the optimal value over effective horizon τ based on the situational state instead of history. We have the following regret bound.

Theorem 3 *For all $\tau \geq 1, T \geq 1$ and $\pi \in \mathcal{P}$, we have*

$$\mathrm{Regret}_\pi^\tau(T) \leq 24\tau^{3/2} \cdot \sqrt{\mathcal{S}\mathcal{A}T \log(2T^2)} + \Big[3\Delta_\tau + \tau_\pi/\tau\Big] \cdot T + \big[\mathcal{S}\mathcal{A} + 5 + 2\log(T)\big] \cdot \tau.$$

While Algorithm 1 requires the effective planning horizon τ and the duration T of operation as input, we establish in Section 4.2 a regret bound for a more sophisticated agent that does not require τ or T as input. The agent relaxes the need for these parameters by operating with an effective planning horizon that increases over time.

4.2 Growing the Horizon

Rather than targeting fixing the duration of operation and the effective planning horizon, as done by Algorithm 1 (`discounted_q_learning`), we can design an agent that operates effectively over any duration by planning over a growing horizon. We now study our primary agent, which executes Algorithm 2 (`growing_horizon_q_learning`) to accomplish this. The agent is instantiated with only three inputs: an initial situational state, a situational state update function, and a reward function. It is worth noting that the agent interacts with the environment through a single stream of experience, with no resets or reinitialization of the situational state. While the Q-learning update of Line 14 is looks identical to that of Algorithm 1, the effective horizon τ – and thus, the discount factor γ – and optimism coefficient β now change over time.

Algorithm 2 `growing_horizon_q_learning`

```
Input:  $S_0$   initial situational state
        $f$     situational state update function
        $r$     reward function
$s \leftarrow S_0$
$Q(\cdot,\cdot) \leftarrow 1, \quad N(\cdot,\cdot) \leftarrow 0$
for $t = 1, 2, \ldots$ do
    $\tau \leftarrow \texttt{foo}_1(t)$
    $\beta \leftarrow \texttt{foo}_2(t)$
    $Q(\cdot,\cdot) \leftarrow Q(\cdot,\cdot) + \texttt{foo}_3(t)$
    $N(\cdot,\cdot) \leftarrow N(\cdot,\cdot) \cdot \texttt{foo}_4(t)$
    $a \leftarrow \texttt{sample\_unif}\big(\arg\max_{a' \in \mathcal{A}} Q(s, a')\big)$
    execute action $a$ and register observation $o$
    $N(s,a) \leftarrow N(s,a) + 1$
    $\alpha \leftarrow \frac{1+2\tau}{N(s,a)+2\tau}$
    $s' \leftarrow f(s, a, o)$
    $\gamma \leftarrow 1 - 1/\tau$
    $Q(s,a) \leftarrow Q(s,a) + \alpha \cdot \left(r(s,a,o) + \gamma \cdot \max_{a' \in \mathcal{A}} Q(s', a') - Q(s,a) + \frac{\beta}{\sqrt{N(s,a)}}\right)$
    $Q(s,a) \leftarrow \min(Q(s,a), \tau)$
    $s \leftarrow s'$
end for
```

The algorithm calls subroutines $\texttt{foo}_1$ through $\texttt{foo}_4$, which govern evolution of the effective planning horizon τ the optimism coefficient β, and suitably adjust action values Q and visitation counts N in tandem with changes in τ and β. In particular,

- $\texttt{foo}_1(t)$ prescribes the effective planning horizon;
- $\texttt{foo}_2(t)$ prescribes the optimism coefficient;
- $\texttt{foo}_3(t)$ increases all action values so that they remain optimistic as the effective planning horizon increases;
- $\texttt{foo}_4(t)$ deemphasizes less recent temporal differences, which were based on a substantially different discount factor.

A sequence of *change points*, beginning with $T_0 = 1$ and continuing with $T_k = 20 \cdot 2^{k-1}$ for $k = 1, 2, 3, \ldots$, underlie these functions. To specify the functions, it is helpful to define notation for the *most recent change point* at each time t. In particular, with the index of the most recent change point given by $k_t = \max\{k \geq 0 : T_k \leq t\}$ if $t > 0$ and $k_0 = 0$, the most recent change point is T_{k_t}. The first of these functions, which provides the effective planning horizon τ, is

$$\texttt{foo}_1(t) = T_{k_t}^{1/5}. \tag{18}$$

The optimism coefficient β is similarly updated at changed points according to

$$\texttt{foo}_2(t) = 4T_{k_t}^{3/10}\sqrt{\log(2T_{k_t}^2)}. \tag{19}$$

Note that $T_{k_t}^{3/10} = \texttt{foo}_1^{3/2}(t)$, so the optimism coefficient scales with the effective planning horizon raised to a power of 3/2 times a logarithmic term. To ensure that the action values remain optimistic, they are incremented by the same amount as the effective horizon; this is accomplished by

$$\texttt{foo}_3(t) = T_{k_t}^{1/5} - T_{k_{t-1}}^{1/5}. \tag{20}$$

Finally, to simplify analysis, we reset state-action counts at change points by multiplying them by

$$\texttt{foo}_4(t) = \mathbf{1}(T_{k_t} = T_{k_{t-1}}). \tag{21}$$

The count becomes one upon the next visit to any state-action pair, and the resulting step size $\alpha = 1$ replaces the action value with the temporal difference, effectively forcing the agent to forget all experience preceding the change point. It is important to note that these choices of $\texttt{foo}_1$ through $\texttt{foo}_4$ were designed to facilitate analysis rather to produce the most effective agent. We will discuss alternative choices in the next section that may improve performance.

We denote by $\widehat{\pi}$ the policy executed by Algorithm 2 with the subroutines specified above. Recall that $\text{Regret}_\pi(T)$ is the regret experienced by $\widehat{\pi}$ relative to a reference policy $\pi \in \mathcal{P}$ and that $\overline{\Delta}_{\tau_\pi} = \sup_{\tau \geq \tau_\pi} \Delta_\tau$ is the maximum distortion over effective horizons equal to or exceeding the reward averaging time τ_π. The following theorem is the main theoretical result of this paper.

Theorem 4 *For all $\pi \in \mathcal{P}$ and $T \geq 1$,*

$$\text{Regret}_\pi(T) \leq \left(120\sqrt{\mathcal{SA}\log(2T^2)} + 5\tau_\pi\right)T^{4/5} + 3\overline{\Delta}_{\tau_\pi}T + \left(54\mathcal{SA} + 18\log(T)\right)T^{1/5} + 2\tau_\pi^5.$$

Algorithm 2 can be viewed as operating over a sequence of episodes, delineated by change points and with the effective planning horizon and optimism coefficient fixed over each. As such, it can be thought of as instantiating and applying 1 over each episode. Despite that, Theorem 4 is not follow directly from Theorem 3. The reason is that the latter applies to an agent that begins with an empty history, whereas an agent that is instantiated at some change point does not. Theorem 3 in Appendix C.4 bridges this gap, offering a generalization to Theorem 3 that applies to an agent starting with an arbitrary history $h \in \mathcal{H}$ so long as its situational state is initialized to $\phi(h)$.

Two corollaries of Theorem 4 facilitate interpretation of its implications. The first characterizes regret relative to reference classes $\mathcal{P}_\tau$, each of which consists of policies for which reward averaging times do not exceed τ. Since $\text{Regret}_{\mathcal{P}_\tau}(T) = \sup_{\pi \in \mathcal{P}_\tau} \text{Regret}_\pi(T)$, we have the following corollary.

Corollary 5 *For all $\tau \geq 1$ and $T \geq 1$,*

$$\text{Regret}_{\mathcal{P}_\tau}(T) \leq \left(120\sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + 5\tau\right) T^{4/5} + 3\overline{\Delta}_\tau T + \left(54\mathcal{S}\mathcal{A} + 18\log(T)\right) T^{1/5} + 2\tau^5. \tag{22}$$

It follows from this corollary that, for all $\epsilon > 0, \tau \geq 1$ and some polynomial $\texttt{poly}(\cdot,\cdot,\cdot)$, the agent attains average reward within $3\overline{\Delta}_\tau + \epsilon$ of $\sup_{\pi\in\mathcal{P}_\tau} \lambda_\pi$ within $\tau^5 \cdot \texttt{poly}(\mathcal{S}, \mathcal{A}, 1/\epsilon)$ timesteps. Hence, within time that scales with τ^5, the agent attains average reward competitive with any policy with reward averaging time τ. Also implicit in this observation is that, over time, the agent becomes competitive with policies that require longer times to evaluate.

A second corollary bounds regret relative to the optimal average reward λ_*. This follows from Theorem 4 and our next lemma, which is a consequence of Corollary 5.1 of (Van Roy, 2006).

Lemma 6 *For all $\tau \geq 1$, $\lambda_* - \lambda_{\tilde{\pi}} \leq \overline{\Delta}_\tau$.*

Applying Lemma 6 and taking π to be $\tilde{\pi}$, we arrive at the following corollary to Theorem 4.

Corollary 7 *For all $T \geq 1$,*

$$\text{Regret}_{\mathcal{P}}(T) \leq \left(120\sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + 5\tau_{\tilde{\pi}}\right) T^{4/5} + 4\overline{\Delta}_{\tau_{\tilde{\pi}}} T + \left(54\mathcal{S}\mathcal{A} + 18\log(T)\right) T^{1/5} + 2\tau_{\tilde{\pi}}^5. \tag{23}$$

This corollary conveys another intriguing property of our result: the agent approaches its asymptotic performance in time that scales with $\tau_{\tilde{\pi}}^5$. In particular, this time does not depend on the reward averaging time of an optimal policy, which in a complex environment could be intractably large or even infinite. The dependence is instead on the reward averaging time of $\tilde{\pi}$, which is determined by situational, rather than environment, state dynamics.

4.3 Scheduling Schemes

The functions $\texttt{foo}_1$ through $\texttt{foo}_4$ prescribe schedules for adjusting the effective planning horizon, the optimism coefficient, and value and count functions. The particular choices specified in the previous section as part of Algorithm 2 were designed to facilitate regret analysis. Indeed, their irregular structure, with abrupt adjustments occurring at particular change points, was introduced solely to simplify analysis by partitioning the stream into episodes. More natural choices involving "smooth" schedules may substantially improve realized performance while satisfying similar or improved regret bounds. Further, the rate at which the effective horizon grows with time plays an important role, and a rate of $t^{1/5}$ may be onerously slow, requiring a very long time to develop plans that span reasonable horizons.

To illustrate the importance of these schedules, let us revisit the service rate control example of Section 1.4. Simulation results reported in that section, which demonstrated the capability of optimistic Q-learning to improve performance over time, made use of particular smooth schedules:

$$\begin{aligned}\texttt{foo}_1(t) =& 1.5t^{1/5},\\ \texttt{foo}_2(t) =& 0.44t^{3/10}\sqrt{\log(2t^2)},\\ \texttt{foo}_3(t) =& 1.5(t^{1/5} - (t-1)^{1/5}),\\ \texttt{foo}_4(t) =& 1.\end{aligned}$$

Note that, while it may be beneficial to modify the rate at which the planning horizon grows, for the purposes of our current study, we retain the $t^{1/5}$ rate and only tune other aspects of the schedules. Figure 6 compares results reported in Section 1.4.5 against the schedules of Algorithm 2. Each plot represents an average over two hundred simulated trajectories. While the latter agent eventually improves performance, that requires a very long time due to its impractical schedules.

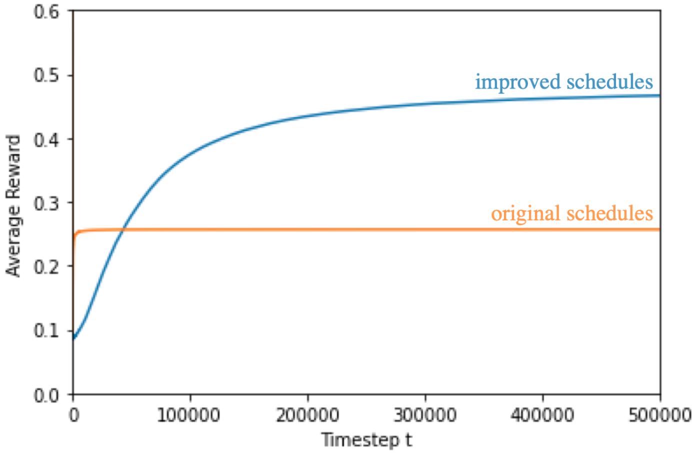

Figure 6: Performance of Algorithm 2 with its original schedules versus improved smooth schedules.

It may be surprising that Algorithm 2 performs so poorly despite satisfying regret bounds of the previous section. Indeed, as is common to mathematical results on efficient reinforcement learning, such regret bounds tend to be very weak. They typically do not offer accurate predictions of realized performance, and given the level of inaccuracy, they do not offer precise guidance on agent design. However, these bounds and the analyses that lead to them, can be useful for developing qualitative understanding and insights, as we have discussed in earlier sections.

5. Closing Remarks

We presented and studied a simple agent that through a general agent-environment interface interacts over a single stream of experience. Our results bound regret realized by the agent. These bounds bear implication on asymptotic performance and the rate at which the agent approaches that level of performance. Importantly, these bounds do not depend on the number of environment states or their mixing time. One interesting insight that emerges involves the relation between the agent's effective planning horizon and its duration T of past experience: the agent plans effectively over a horizon that grows with $T^{1/5}$. There are a number of directions in which the results in this work can be strengthened or extended. We will discuss a few in this section.

Our agent uses a particularly simple representation for the action value function, comprised of a fixed, prespecified situational state update function and a lookup table over situational states. State-of-the-art agents adapt the situational state update function based on the agent's experience and generalize over situational states, typically by using a neural network instead of a lookup table, and these extensions allow for much larger situational state spaces and can greatly improve performance.

The agent that we analyze discards all previous experience whenever the effective planning horizon is increased. This is impractical and done only to facilitate analysis. It ought to be possible to analyze a variation that more gradually phases out the influence of past data, along the lines discussed in Section 4.3.

To maximize long-term average reward, it may be natural for the agent to learn a differential value function directly, as is studied in Wan et al. (2020), rather than discounted value functions, as does our agent. However, an open issue is whether an agent can explore the environment efficiently when doing so. We believe that this is a problem that is worth further investigation.

Finally, we suspect that the $T^{4/5}$ term in our regret bound, which reflects the rate at which the agent approaches its asymptotic performance, is not fundamental and can be improved with a better agent design and a more nuanced analysis. We note that this dependence stems from the subroutines $\texttt{foo}_1$ (18) through $\texttt{foo}_4$ (21) that the agent uses to adjust the planning horizon. As we mentioned in Section 4.2, these settings induce an effective planning horizon that grows with $T^{1/5}$, suggesting that it takes time τ^5 to plan effectively over a horizon of length τ. We conjecture that there exists an environment in which any agent requires time τ^3 to do this, which would translate to a $T^{2/3}$ instead of $T^{4/5}$ term in the regret lower bound. Such a lower bound could shed light on the limits of learning and could offer useful insight to agent designers on how long the agent ought to plan, given the duration of past experience.

Acknowledgements

Financial support from Army Research Office (ARO) grant W911NF2010055, National Science Foundation grant CCF-2106508, and the Digital Twin research grant from Bain & Company are gratefully acknowledged. Shi Dong was partly supported by the Herb and Jane Dwight Stanford Graduate Fellowship. Zhengyuan Zhou acknowledges the generous support from New York University's Spring 2022 Center for Global Economy and Business faculty research grant. We thank J. Michael Harrison, Satinder Singh, John Tsitsiklis, and Xiuyuan Lu for stimulating discussions and helpful feedback. We also thank Alex Cloud for pointing out a mistake in a previous version of this paper.

Appendix A. Probabilistic Framework

In this appendix, we define our probabilistic framework and notation. We will define all random quantities with respect to a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The probability of an event $\mathscr{F} \in \mathcal{F}$ is denoted by $\mathbb{P}(\mathscr{F})$. For all events $\mathscr{F}, \mathscr{G} \in \mathcal{F}$ with $\mathbb{P}(\mathscr{G}) > 0$, the probability of $\mathscr{F}$ conditioned on $\mathscr{G}$ is denoted by $\mathbb{P}(\mathscr{F}|\mathscr{G})$.

A random variable is a function with the set of outcomes Ω as its domain. For all random variable Z, $\mathbb{P}(Z \in \mathcal{Z})$ denotes the probability of the event that Z lies within a set $\mathcal{Z}$. The probability $\mathbb{P}(\mathscr{F}|Z = z)$ is of the event $\mathscr{F}$ conditioned on the event $Z = z$. When Z takes values in $\mathbb{R}^K$ and has a density p_Z, though $\mathbb{P}(Z = z) = 0$ for all z, conditional probabilities $\mathbb{P}(\mathscr{F}|Z = z)$ are well-defined whenever $p_Z(z) > 0$, and will be denoted by $\mathbb{P}(\mathscr{F}|Z = z)$. For fixed $\mathscr{F}$, this is a function of z. We denote the value, evaluated at $z = Z$, by $\mathbb{P}(\mathscr{F}|Z)$, which is itself a random variable. Even when $\mathbb{P}(\mathscr{F}|Z = z)$ is ill-defined for some z, $\mathbb{P}(\mathscr{F}|Z)$ is well-defined because problematic events occur with zero probability.

For each possible realization z, the probability $\mathbb{P}(Z = z)$ that $Z = z$ is a function of z. We denote the value of this function evaluated at Z by $\mathbb{P}(Z)$. Note that $\mathbb{P}(Z)$ is itself a random variable because is it depends on Z. For random variables Y and Z and possible realizations y and z, the probability $\mathbb{P}(Y = y|Z = z)$ that $Y = y$ conditioned on $Z = z$ is a function of (y, z). Evaluating this function at (Y, Z) yields a random variable, which we denote by $\mathbb{P}(Y|Z)$.

Particular random variables appear routinely throughout the paper. One is the environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$. While $\mathcal{A}$ and $\mathcal{O}$ are deterministic sets that define the agent-environment interface, the observation probability function ρ is a random variable. This randomness reflects the agent designer's epistemic uncertainty about the environment. We often consider probabilities $\mathbb{P}(\mathscr{F}|\mathcal{E})$ of events $\mathscr{F}$ conditioned on the environment $\mathcal{E}$.

Our main results involve upper bounding random variables in the form of $\mathbb{E}[\cdot|\mathcal{E}]$, which is a function of $\mathcal{E}$. To establish $\mathbb{E}[\cdot|\mathcal{E}] \leq B$ with probability one, a sufficient argument is that for all environment e, if the distribution of $\mathcal{E}$ has a positive mass on e, then $\mathbb{E}[\cdot|\mathcal{E} = e] \leq B$. In Appendix C, when readers encounter notations like $\mathbb{E}[\cdot|\mathcal{E} = e]$, it may be helpful to think of an accompanying quantifier "for all distributions of $\mathcal{E}$ where e enjoys a positive mass."

A policy π assigns a probability $\pi(a|h)$ to each action a for each history h. For each policy π, random variables $A_0^\pi, O_1^\pi, A_1^\pi, O_2^\pi, \ldots$, represent a sequence of interactions generated by selecting actions according to π. In particular, with $H_t^\pi = (A_0^\pi, O_1^\pi, \ldots, O_t^\pi)$ denoting the history of interactions through time t, we have $\mathbb{P}(A_t^\pi|H_t^\pi) = \pi(A_t^\pi|H_t^\pi)$ and $\mathbb{P}(O_{t+1}^\pi|H_t^\pi, A_t^\pi, \mathcal{E}) = \rho(O_{t+1}^\pi|H_t^\pi, A_t^\pi)$. As shorthand, we generally suppress the superscript π and instead indicate the policy through a subscript of $\mathbb{P}$. For example,

$$\mathbb{P}_\pi(A_t|H_t) = \mathbb{P}(A_t^\pi|H_t^\pi) = \pi(A_t^\pi|H_t^\pi),$$

and

$$\mathbb{P}_\pi(O_{t+1}|H_t, A_t, \mathcal{E}) = \mathbb{P}(O_{t+1}^\pi|H_t^\pi, A_t^\pi, \mathcal{E}) = \rho(O_{t+1}^\pi|H_t^\pi, A_t^\pi).$$

The dependence on π extends to algorithmic state Z_t^π, situational state S_t^π, and epistemic state P_t^π, and we use the same conventions to suppress superscripts when appropriate.

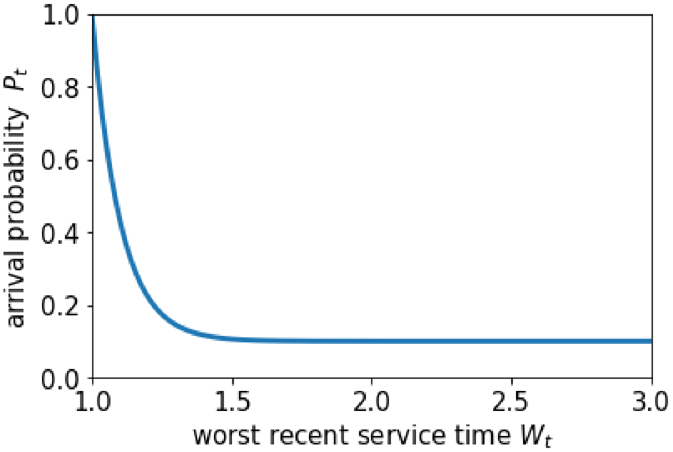

Figure 7: The customer arrival probability is a decreasing function of the maximum service time experienced among the most recent dozen customers served.

When expressing expectations, we use the same subscripting notation as with probabilities. For example, the expectation of a reward $R^{\pi}_{t+1} = r(S^{\pi}_t, A^{\pi}_t, O^{\pi}_{t+1})$ conditioned on the environment $\mathcal{E}$, state S^{π}_t, and action A^{π}_t is written as $\mathbb{E}[R^{\pi}_{t+1}|\mathcal{E}, S^{\pi}_t, A^{\pi}_t] = \mathbb{E}_{\pi}[R_{t+1}|\mathcal{E}, S_t, A_t]$.

Much of the paper studies properties of interactions under a specific policy $\widehat{\pi}$. When it is clear from context, we suppress superscripts and subscripts that indicate this. For example, $H_t = H^{\widehat{\pi}}_t$, $A_t = A^{\widehat{\pi}}_t$, $O_{t+1} = O^{\widehat{\pi}}_{t+1}$. Further,

$$\mathbb{P}(A_t|H_t) = \mathbb{P}_{\widehat{\pi}}(A_t|H_t) = \widehat{\pi}(A_t|H_t).$$

Appendix B. Service Rate Control Example

This appendix supplements the discussion of Section 1.4. In particular, we provide a precise characterization of environment dynamics and establish that the two baseline agents described in Section 1.4 do not deviate from the slow mode of service. We also present a third, more sophisticated, baseline agent and establish that even that does not learn to deviate from the slow mode.

B.1 Environment Dynamics

The service station is initially vacant, and customers may arrive starting at the end of the first timestep. At each time, the arrival probability depends on maximum service time experienced among the most recent 12 customers served. We denote this statistic by W_t and initialize with $W_0 = 1$. The customer arrival probability decreases as W_t increases, as illustrated in Figure 7. In particular, conditioned on W_t, the probability that a customer arrives at time t, if the service station is vacant then, is $P_t = 0.1 + 0.9e^{-10(W_t - 1)}$.

The choice of service mode impacts service times: with the fast mode, service is always completed in a single timestep, while with the slow mode, the service is completed over the next timestep with probability 1/2. As such, the observation probabilities conditioned on $S_t = 0$ are given by

$\rho_*(o \mid H_t, \cdot)$	departure	¬departure
arrival	0	P_t
¬arrival	0	$1 - P_t$

and, conditioned on $S_t = 1$,

$\rho_*(o \mid H_t, \text{fast})$	departure	¬departure
arrival	P_t	0
¬arrival	$1 - P_t$	0

$\rho_*(o \mid H_t, \text{slow})$	departure	¬departure
arrival	$P_t/2$	0
¬arrival	$(1 - P_t)/2$	1/2

It is easy to verify that the long-run average reward is maximized if the agent applies the fast mode of service over every timestep. This policy minimizes service times, with each customer waiting for precisely one timestep. Consequently, under this policy, W_t converges to 1, as does the arrival probability. The long-run average reward is therefore $1 - 0.5 = 0.5$.

B.2 Analysis of Baseline Agents

We now study the performance of the two baseline agents introduced in Section 1.4, as well a more sophisticated variant. Recall that these agents at each time apply a policy π_ϵ, which selects the fast mode with some probability ϵ, which can vary with time. Each of these agents begins with knowledge of service completion probabilities: 1/2 and 1 for the slow and fast modes, respectively. Throughout the discussion, Let W_∞ and P_∞ denote random variables sampled from the steady-state distributions of W_t and P_t, respectively.

Let c_ϵ be the service completion probability over any timestep when a customer is served under policy π_ϵ. In particular, $c_\epsilon = \epsilon + \frac{1}{2}(1-\epsilon) = \frac{1+\epsilon}{2}$. The average reward λ_{π_ϵ} is then given by

$$\lambda_{\pi_\epsilon} = \frac{\text{profit per customer}}{\text{mean interarrival time}} = \frac{1 - 0.5\frac{\epsilon}{c_\epsilon}}{\frac{1}{c_\epsilon} - 1 + \frac{1}{\mathbb{E}_{\pi_\epsilon}[P_\infty \mid \mathcal{E} = e_*]}}. \tag{24}$$

The first agent applies π_0 and only deviates if warranted after observing data over a long duration, assuming that the arrival probability is fixed. Whether it decides to increase ϵ depends on its arrival probability estimate. Under the policy π_0, a customer's service time is 1 with probability 1/2 and, otherwise, at least 2. Since W_t is the largest among 12 service times, $\mathbb{P}_{\pi_0}(W_\infty = 1 \mid \mathcal{E} = e_*) = 1/4096$ and $\mathbb{P}_{\pi_0}(W_\infty \geq 2 \mid \mathcal{E} = e_*) = 4095/4096$. Consequently,

$$\mathbb{E}_{\pi_0}\left[P_\infty \mid \mathcal{E} = e_*\right] = \mathbb{E}_{\pi_0}\left[0.1 + 0.9e^{-10(W_\infty - 1)} \mid \mathcal{E} = e_*\right] < 1.$$

To keep things simple, suppose the agent's estimate of this steady-state arrival probability is exactly $\mathbb{E}_{\pi_0}\left[P_\infty \mid \mathcal{E} = e_*\right]$. As such, the agent's *estimate* of the average reward under π_ϵ is

$$\hat{\lambda}_{\pi_\epsilon} = \frac{\text{profit per customer}}{\text{mean interarrival time}} = \frac{1 - 0.5\epsilon\frac{1}{c_\epsilon}}{\frac{1}{c_\epsilon} - 1 + \frac{1}{\mathbb{E}_{\pi_0}[P_\infty \mid \mathcal{E} = e_*]}} = \frac{1}{1 - \epsilon + \frac{1+\epsilon}{\mathbb{E}_{\pi_0}[P_\infty \mid \mathcal{E} = e_*]}}. \tag{25}$$

Note that the difference between Equation (25) and Equation (24) is due to the first agent's use of an arrival probability that results from π_0 rather than π_ϵ. Since $\mathbb{E}_{\pi_0}\left[P_\infty \mid \mathcal{E} = e_*\right] < 1$, $\hat{\lambda}_{\pi_\epsilon}$ is strictly decreasing in $\epsilon \in [0,1]$. As such, the agent does not deviate from the slow mode.

The second agent additionally tries a small value of $\epsilon > 0$ for some duration in order to estimate the derivative $\mathrm{d}\lambda_{\pi_\epsilon}/\mathrm{d}\epsilon$ of the average reward at $\epsilon = 0$. If this derivative is positive, it increases ϵ. Here, we show that this derivative is negative, and hence the second agent does not deviate from the slow mode. As such, one can easily check that for each positive integer w and for all $\epsilon \in [0,1]$, we have:

$$\mathbb{P}_{\pi_\epsilon}(W_\infty \leq w \mid \mathcal{E} = e_*) = \left(1 - \left(\frac{1-\epsilon}{2}\right)^w\right)^{12}. \tag{26}$$

Recalling that $c_\epsilon = \frac{1+\epsilon}{2}$ and following Equation (24), we have:

$$\lambda_{\pi_\epsilon} = \frac{1 - 0.5\epsilon\frac{1}{c_\epsilon}}{\frac{1}{c_\epsilon} - 1 + \frac{1}{\mathbb{E}_{\pi_\epsilon}[P_\infty | \mathcal{E} = e_*]}} = \frac{G(\epsilon)}{(1-\epsilon)G(\epsilon) + (1+\epsilon)}, \tag{27}$$

where $G(\epsilon) = \mathbb{E}_{\pi_\epsilon}[P_\infty \mid \mathcal{E} = e_*] = \sum_{w=1}^{\infty} \mathbb{P}_{\pi_\epsilon}(W_\infty = w \mid \mathcal{E} = e_*)\left(0.1 + 0.9e^{-10(w-1)}\right)$. Consequently, taking its derivative with respect to ϵ and evaluating it $\epsilon = 0$ yields:

$$\left.\frac{\mathrm{d}\lambda_{\pi_\epsilon}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = \frac{\frac{\mathrm{d}}{\mathrm{d}\epsilon}G(\epsilon)|_{\epsilon=0} - G(0) + G(0)^2}{\big[G(0)+1\big]^2} < -0.072. \tag{28}$$

As such, the second agent will not deviate from π_0.

Finally, we can even consider a third agent, which is similar to the second agent except it additionally estimates the second derivative $\mathrm{d}^2\lambda_{\pi_\epsilon}/\mathrm{d}\epsilon^2$. It then chooses ϵ to maximize a second-order Taylor expansion of λ_{π_ϵ} around $\epsilon = 0$ subject to the constraint $0 \leq \epsilon \leq 1$. This agent again ends up always selecting the slow mode of service, because even exploiting second-order information suggests that staying with π_0 is the best thing to do. To see this, we evaluate the second derivative of λ_{π_ϵ} at $\epsilon = 0$, yielding $\left.\frac{1}{2}\frac{\mathrm{d}^2\lambda_{\pi_\epsilon}}{\mathrm{d}\epsilon^2}\right|_{\epsilon=0} < 0.0716$. As such, when using a second-order polynomial for extrapolation, one would get $\tilde{\lambda}_{\pi_\epsilon} = \lambda_{\pi_0} - \left.\frac{\mathrm{d}\lambda_{\pi_\epsilon}}{\mathrm{d}\epsilon}\right|_{\epsilon=0}\epsilon + \left.\frac{1}{2}\frac{\mathrm{d}^2\lambda_{\pi_\epsilon}}{\mathrm{d}\epsilon^2}\right|_{\epsilon=0}\epsilon^2 < \lambda_{\pi_0} - 0.072\epsilon + 0.0716\epsilon^2 < 0$, thereby yielding a strictly smaller value than λ_{π_0} for all $\epsilon \in (0,1]$.

In summary, the first baseline agent represents what might be produced by a conservative designer, who demands to see empirical evidence justifying fast service before ever trying that. The second agent is representative of approaches used in the policy gradient literature, as discussed in (Sutton and Barto, 2018) and references therein. The third agent pursues a more sophisticated approach entailing estimation and use of the second derivative in addition to the gradient. Per the analysis given above, all three agents end up choosing the slow-only policy, and hence perform poorly relative to our optimistic Q-learning agent, which adapts action values $Q_t(S_t, A_t)$ to predict future return and select actions.

Appendix C. Proofs

C.1 Proof of Lemma 2

The lemma is restated below.

Lemma 8 *For all $\pi \in \mathcal{P}$, $h \in \mathcal{H}$ and $\gamma \in [0,1)$,* $\left|V_\pi^\gamma(h) - \frac{\lambda_\pi}{1-\gamma}\right| \leq \tau_\pi$.

To prove this lemma, recall that, for a fixed policy $\pi \in \mathcal{P}$, discount factor $\gamma \in [0,1)$, and history $h \in \mathcal{H}$,

$$V_\pi^\gamma(h) = \sum_{t=0}^{\infty} \Big(\gamma^t \cdot \big(P_\pi^t \overline{r}_\pi\big)(h)\Big). \tag{29}$$

For simplicity, let $r_\ell = \gamma^\ell \cdot \big(P_\pi^\ell \overline{r}_\pi\big)(h)$. We have that

$$V_\pi^\gamma(h) = \sum_{\ell=0}^{\infty} \gamma^\ell r_\ell. \tag{30}$$

By the definition of τ_π, for all $\ell \geq 0$,

$$\tau_\pi \geq \left|\sum_{k=0}^{\ell} \big(r_k - \lambda_\pi\big)\right|. \tag{31}$$

Hence,

$$\begin{aligned}
\left|V_\pi^\gamma(h) - \frac{\lambda_\pi}{1-\gamma}\right| &= \left|\sum_{\ell=0}^{\infty} \gamma^\ell \cdot \big(r_\ell - \lambda_\pi\big)\right| \\
&= \left|\sum_{\ell=0}^{\infty} (1-\gamma)\gamma^\ell \cdot \sum_{k=0}^{\ell} \big(r_k - \lambda_\pi\big)\right| \\
&\leq \sum_{\ell=0}^{\infty} (1-\gamma)\gamma^\ell \cdot \left|\sum_{k=0}^{\ell} \big(r_k - \lambda_\pi\big)\right| \\
&\leq \sum_{\ell=0}^{\infty} (1-\gamma)\gamma^\ell \cdot \tau_\pi \\
&= \tau_\pi,
\end{aligned}$$

which is our desired result.

C.2 Properties of the Learning Rates

In this subsection, we generalize a useful lemma from (Jin et al., 2018) on properties of the learning rates. Let

$$\alpha_k^i = \alpha_i \cdot \prod_{\ell=i+1}^{k} (1-\alpha_\ell), \quad i = 1, \ldots, k, \tag{32}$$

where $(\alpha_\ell : \ell = 1, 2, \dots)$ is the learning rate sequence $\alpha_\ell = (1 + 2\tau)/(\ell + 2\tau)$, and

$$\alpha_k^0 = \mathbf{1}\{k = 0\}. \tag{33}$$

Naturally, $\sum_{i=0}^k \alpha_k^i = 1$. We also have the following:

Lemma 9

(a) For all $k \geq 1$, $\frac{1}{\sqrt{k}} \leq \sum_{i=1}^k \frac{\alpha_k^i}{\sqrt{i}} \leq \frac{2}{\sqrt{k}}$;

(b) For all $k \geq 1$, $\max_{i=1,\dots,k} \alpha_k^i \leq \frac{4\tau}{k}$ and $\sum_{i=1}^k (\alpha_k^i)^2 \leq \frac{4\tau}{k}$;

(c) For all $i \geq 1$, $\sum_{k=i}^\infty \alpha_k^i = 1 + \frac{1}{2\tau}$.

Proof. The proof of Lemma 4.1 in (Jin et al., 2018) covers the case where $H = 2\tau$ is a positive integer. We note that their proof of parts (a) and (b) also applies for all $H > 1$. Thus, what is left for us here is showing that part (c) holds for all real numbers $H > 1$. To this end, we first establish that, for all positive real numbers $b > a$,

$$\frac{a}{b-a} = \sum_{i=1}^{\infty} \prod_{j=1}^{i} \frac{a+j-1}{b+j}. \tag{34}$$

In fact, we can show by induction on positive integer ℓ that

$$\frac{a}{b-a} = \sum_{i=1}^{\ell} \prod_{j=1}^{i} \frac{a+j-1}{b+j} + \frac{a}{b-a} \prod_{j=1}^{\ell} \frac{a+j}{b+j}. \tag{35}$$

When $\ell = 1$, we have that

$$\frac{a}{b-a} - \frac{a}{b+1} = \frac{a}{b-a} \cdot \left[1 - \frac{b-a}{b+1}\right] = \frac{a}{b-a} \cdot \frac{a+1}{b+1}. \tag{36}$$

Hence, (35) holds for $\ell = 1$. Now suppose that (35) holds for ℓ. There is

$$\begin{aligned}
\frac{a}{b-a} - \sum_{i=1}^{\ell+1} \prod_{j=1}^{i} \frac{a+j-1}{b+j} &= \left\{ \frac{a}{b-a} - \sum_{i=1}^{\ell} \prod_{j=1}^{i} \frac{a+j-1}{b+j} \right\} - \prod_{j=1}^{\ell+1} \frac{a+j-1}{b+j} \\
&= \frac{a}{b-a} \prod_{j=1}^{\ell} \frac{a+j}{b+j} - \prod_{j=1}^{\ell+1} \frac{a+j-1}{b+j} \qquad (37) \\
&= \left\{ \frac{a}{b-a} \prod_{j=1}^{\ell} \frac{a+j}{b+j} \right\} \cdot \left(1 - \frac{b-a}{b+\ell+1}\right) \\
&= \frac{a}{b-a} \prod_{j=1}^{\ell+1} \frac{a+j}{b+j}, \qquad (38)
\end{aligned}$$

where (37) follows from our induction hypothesis. Thus, (35) also holds for $\ell+1$, concluding our induction. Following (35), we have

$$\frac{a}{b-a}-\sum_{i=1}^{\infty}\prod_{j=1}^{i}\frac{a+j-1}{b+j}=\lim_{\ell\to\infty}\left\{\frac{a}{b-a}-\sum_{i=1}^{\infty}\prod_{j=1}^{i}\frac{a+j-1}{b+j}\right\}=\lim_{\ell\to\infty}\frac{a}{b-a}\prod_{j=1}^{\ell}\frac{a+j}{b+j}. \tag{39}$$

However,

$$\log\prod_{j=1}^{\ell}\frac{a+j}{b+j}=\sum_{j=1}^{\ell}\log\left(1-\frac{b-a}{b+j}\right)\le-\sum_{j=1}^{\ell}\frac{b-a}{b+j}. \tag{40}$$

The right-hand side goes to $-\infty$ as $\ell\to\infty$, implying that

$$\lim_{\ell\to\infty}\prod_{j=1}^{\ell}\frac{a+j}{b+j}=0. \tag{41}$$

Thus, (34) follows from (39). Now we have

$$\begin{aligned}\sum_{k=i}^{\infty}\alpha_k^i &= \frac{H+1}{H+i}\cdot\left\{1+\sum_{k=i}^{\infty}\prod_{j=0}^{k-i}\frac{i+j}{H+i+j+1}\right\}\\ &= \frac{H+1}{H+i}\cdot\left\{1+\sum_{k=1}^{\infty}\prod_{j=1}^{k}\frac{i+j-1}{H+i+j}\right\}\\ &= \frac{H+1}{H+i}\cdot\left\{1+\frac{i}{H}\right\}\\ &= \frac{H+1}{H}\\ &= 1+\frac{1}{2\tau},\end{aligned} \tag{42}$$

as we have claimed in part (c). ■

C.3 Regret Analysis of the Discounted Q-Learning Agent

In this section, we focus ourselves on the discounted variant of the agent in Section 4.1. Throughout this section we assume that the discount factor $\gamma=1-1/\tau\in[0,1)$ is fixed. We also consider a hypothetical setting, in which the history starts from an arbitrary $h\in\mathcal{H}$ and the agent starts from situational state $\phi(h)$. Since every time the agent changes the discount factor, the environment history is not reset to H_0, our main goal here is to demonstrate that our result holds regardless of the initial history, as long as the agent starts from the corresponding agent state. In order to simplify notations, in this section we always condition on the environment being a generic fixed environment $e=(\mathcal{A},\mathcal{O},\rho)$, and $\mathbb{P}(\cdot|\mathcal{E}=e)$, $\mathbb{E}[\cdot|\mathcal{E}=e]$ are shortened as $\mathbb{P}(\cdot)$, $\mathbb{E}[\cdot]$, respectively. We let $H_0=h$ be a fixed, possibly non-empty history, and $S_0=\phi(h)$. We will also omit the superscript γ on value

Algorithm 3 Discounted Q-learning subroutine

```
Input: $f, r, \gamma, \beta, Q'$
initialize history to $h$
$t = 0, \quad s \leftarrow \phi(h)$
$Q \leftarrow Q', \quad N(\cdot,\cdot) \leftarrow 0$
$V(s) \leftarrow \max_{a' \in \mathcal{A}} Q(s, a'), \quad \forall s \in \mathcal{S}$
while True do
    $a \leftarrow \texttt{sample\_unif}(\arg\max_{a' \in \mathcal{A}} Q(s, a'))$
    $N(s, a) \leftarrow N(s, a) + 1$
    $\alpha \leftarrow \frac{2+(1-\gamma)}{2+N(s,a)\cdot(1-\gamma)}$
    execute action $a$ and register observation $o$
    $s' \leftarrow f(s, a, o)$
    $Q(s, a) \leftarrow (1-\alpha) \cdot Q(s, a) + \alpha \cdot \left[r(s, a, o) + \gamma \cdot V(s') + \frac{\beta}{\sqrt{N(s,a)}}\right]$
    $Q(s, a) \leftarrow \min\left\{Q(s, a),\ 1/(1-\gamma)\right\}$
    $V(s) \leftarrow \max_{a' \in \mathcal{A}} Q(s, a')$
    $s \leftarrow s', \quad t \leftarrow t + 1$
end while
```

functions V_*^γ, Q_*^γ, V_π^γ and Q_π^γ. Readers should keep in mind that all value functions in this section are with respect to discount factor γ.

Specifically, we will consider Algorithm 3, which is identical to Algorithm 1 except that the initial history can be arbitrary, and that we use $Q'(s, a)$ to initialize $Q(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Let $H_1, H_2, \ldots$ be the history trajectory of Algorithm 3, i.e.

$$H_t = \big(h, A_0, O_1, \ldots, A_{t-1}, O_t\big), \quad t = 1, 2, \ldots, \tag{43}$$

and let

$$S_t = \phi(H_t), \quad R_t = r(S_{t-1}, A_{t-1}, O_t), \quad t = 1, 2, \ldots. \tag{44}$$

Also let $V_t(s)$ be the value of situational state s at timestep t immediately *after* the update

$$Q(S_{t-1}, A_{t-1}) \leftarrow (1-\alpha) \cdot Q(S_{t-1}, A_{t-1}) + \alpha \cdot \left(R_t + \gamma \cdot V(S_t) + \frac{\beta}{\sqrt{N(S_{t-1}, A_{t-1})}}\right), \tag{45}$$

$$Q(S_{t-1}, A_{t-1}) \leftarrow \min\left\{Q(S_{t-1}, A_{t-1}),\ \frac{1}{1-\gamma}\right\}, \tag{46}$$

and

$$V(S_{t-1}) \leftarrow \max_{a' \in \mathcal{A}} Q(S_{t-1}, a'), \tag{47}$$

and let $V_t(h)$ be a shorthand for $V_t(\phi(h))$. Similarly are $Q_t(s,a)$ and $Q_t(h,a)$ defined. Note that since the actions are selected greedily, we have

$$Q_t(H_t, A_t) = \max_{a\in\mathcal{A}} Q_t(H_t, a) = V_t(H_t). \tag{48}$$

Finally, we let $\mathbf{P}$ be the *transition operator*, such that for all functions $g : \mathcal{H} \mapsto \mathbb{R}$ and history-action pairs $(h,a) \in \mathcal{H}\times\mathcal{A}$,

$$\mathbf{P}g(h,a) = \sum_{h'\in\mathcal{H}} \Big(P_{ahh'} \cdot g(h')\Big). \tag{49}$$

Let $\mathring{\pi}$ be the policy corresponding to Algorithm 3. Recall that the distortion with respect to effective planning horizon $\tau \geq 1$ is defined as

$$\Delta_\tau = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left(\sup_{h\in\mathcal{H}:\phi(h)=s} Q_*^{1-1/\tau}(h,a) - \inf_{h\in\mathcal{H}:\phi(h)=s} Q_*^{1-1/\tau}(h,a) \right). \tag{50}$$

To avoid cluttering, for $\gamma \in [0,1)$ we will simply use Δ_γ to represent Δ_τ with $\tau = 1/(1-\gamma)$. Our aim is to show the following result:

Theorem 10 *For all $T \geq 1$, if Algorithm 3 is executed with $\gamma \in [0,1)$,*

$$\beta = \frac{4}{(1-\gamma)^{3/2}}\sqrt{\log(2T^2)}, \tag{51}$$

and Q' such that for some $\iota \geq 0$,

$$Q_*(h,a) - \frac{\iota}{1-\gamma} \leq Q'\big(\phi(h),a\big) \leq \frac{1}{1-\gamma}, \quad \forall h\in\mathcal{H}, a\in\mathcal{A}, \tag{52}$$

then for all initial history $h\in\mathcal{H}$,

$$\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \leq \frac{24}{(1-\gamma)^{\frac{5}{2}}}\cdot\sqrt{\mathcal{SA}T\cdot\log(2T^2)} + \frac{3\tilde{\Delta}_\gamma T}{1-\gamma} + \frac{\mathcal{SA}+3}{(1-\gamma)^2}, \tag{53}$$

where $\tilde{\Delta}_\gamma = \max\{\Delta_\gamma, \iota\}$.

C.3.1 Regret Decomposition

In this subsection we will prove the following lemma, which decomposes the left-hand side of (53) and paves the way for further analysis.

Lemma 11 *For all $T \geq 1$,*

$$\begin{aligned}\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \;\leq\; & \frac{1}{1-\gamma}\cdot\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - Q_*(H_t,A_t)\Big)\right] \\ & + \frac{1}{(1-\gamma)^2}.\end{aligned} \tag{54}$$

Proof. We have

$$\begin{aligned}\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \;&=\; \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \\ &\quad - \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - V_*(H_t)\Big)\right]. \end{aligned} \tag{55}$$

Taking a closer look at the first term on the right-hand side, since $\mathring{\pi}$ is greedy with respect to V_t for each t,

$$\begin{aligned}\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \;&=\; \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - Q_{\mathring{\pi}}(H_t, A_t)\Big)\right] \\ &=\; \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - Q_*(H_t, A_t)\Big)\right] \\ &\quad + \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(Q_*(H_t, A_t) - Q_{\mathring{\pi}}(H_t, A_t)\Big)\right] \\ &=\; \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_t(H_t) - Q_*(H_t, A_t)\Big)\right] \\ &\quad + \gamma \cdot \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_{t+1}) - V_{\mathring{\pi}}(H_{t+1})\Big)\right]. \end{aligned} \tag{56}$$

Combining (55) and (56), we have

$$\begin{aligned}(1-\gamma)\cdot\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \;&\leq\; \mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - Q_*(H_t, A_t)\Big)\right] \\ &\quad + \Big(V_*(H_T) - V_{\mathring{\pi}}(H_T)\Big). \end{aligned} \tag{57}$$

Dividing both sides by $1-\gamma$ and considering that $V_*(H_T) - V_{\mathring{\pi}}(H_T) \leq 1/(1-\gamma)$, we arrive at (54). ■

For simplicity, let

$$\chi_k = V_k(H_k) - V_*(H_k) + \frac{\tilde{\Delta}_\gamma}{1-\gamma} \tag{58}$$

and

$$\xi_k = Q_k(H_k, A_k) - Q_*(H_k, A_k) \tag{59}$$

for each $k \geq 0$. Using these notations, (54) can be written equivalently as

$$\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t) - V_{\mathring{\pi}}(H_t)\Big)\right] \leq \frac{1}{1-\gamma}\cdot\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\big(\xi_k - \chi_k\big)\right] + \frac{\tilde{\Delta}_\gamma T}{(1-\gamma)^2} + \frac{1}{(1-\gamma)^2}. \tag{60}$$

C.3.2 Establishing Near-Optimism

In this subsection we show that at each timestep t, the value function V_t is almost optimistic uniformly across all histories. We have the following result.

Lemma 12 *For all $T \geq 1$, if Algorithm 3 is executed with $\gamma \in [0, 1)$,*

$$\beta_\delta = \frac{4}{(1-\gamma)^{3/2}} \sqrt{\log \frac{2T}{\delta}}, \tag{61}$$

and Q' such that for some $\iota \geq 0$,

$$Q_*^\gamma(h, a) - \frac{\iota}{1-\gamma} \leq Q'(h, a) \leq \frac{1}{1-\gamma}, \quad \forall h \in \mathcal{H}, a \in \mathcal{A}, \tag{62}$$

then with probability at least $1 - \delta$, for all $h \in \mathcal{H}, a \in \mathcal{A}$ and $0 \leq t \leq T$,

$$V_t(h) \geq V_*(h) - \frac{\tilde{\Delta}_\gamma}{1-\gamma} \quad \textit{and} \quad Q_t(h, a) \geq Q_*(h, a) - \frac{\tilde{\Delta}_\gamma}{1-\gamma},$$

where $\tilde{\Delta}_\gamma = \max\{\Delta_\gamma, \iota\}$.

Proof. For the moment let us fix $h \in \mathcal{H}$ and $a \in \mathcal{A}$. Let $\hat{Q}_k$ be the Q-value of $\big(\phi(h), a\big)$ after it has been updated k times, with $\hat{Q}_0 = 1/(1-\gamma)$ being the initial value. Further, for each $k = 1, 2, \ldots$, let t_k be the timestep at which $\big(\phi(h), a\big)$ is updated. Note that we have

$$\phi(H_{t_k}) = \phi(h), \quad A_{t_k} = a. \tag{63}$$

From the update rule in Algorithm 3, when $n \geq 1$,

$$\hat{Q}_n = \sum_{i=1}^{n} \alpha_n^i \cdot \left(R_{t_i+1} + \gamma \cdot V_{t_i}\big(H_{t_i+1}\big) + \frac{\beta_\delta}{\sqrt{i}} \right). \tag{64}$$

Thus, when $n \geq 1$,

$$\begin{aligned}
\hat{Q}_n - Q_*(h,a) &= \sum_{i=1}^{n} \alpha_n^i \cdot \Big(R_{t_i+1} + \gamma \cdot V_{t_i}\big(H_{t_i+1}\big) + \frac{\beta_\delta}{\sqrt{i}} - Q_*(h,a) \Big) \\
&\geq \sum_{i=1}^{n} \alpha_n^i \cdot \Big(R_{t_i+1} + \gamma \cdot V_{t_i}\big(H_{t_i+1}\big) + \frac{\beta_\delta}{\sqrt{i}} \Big) \\
&\quad - \sum_{i=1}^{n} \alpha_n^i \cdot \Big(Q_*(H_{t_i}, A_{t_i}) + \tilde{\Delta}_\gamma \Big) \qquad (65) \\
&= \sum_{i=1}^{n} \alpha_n^i \cdot \Big(\gamma \cdot V_{t_i}\big(H_{t_i+1}\big) - \gamma \cdot \mathbf{P}V_*(H_{t_i}, A_{t_i}) \Big) \\
&\quad + \Big\{ \sum_{i=1}^{n} \alpha_n^i \cdot \frac{\beta_\delta}{\sqrt{i}} \Big\} - \tilde{\Delta}_\gamma \qquad (66) \\
&\geq \sum_{i=1}^{n} \alpha_n^i \cdot \Big(V_{t_i}(H_{t_i+1}) - V_*(H_{t_i+1}) \Big) \\
&\quad + \sum_{i=1}^{n} \alpha_n^i \cdot \Big(V_*(H_{t_i+1}) - \mathbf{P}V_*(H_{t_i}, A_{t_i}) \Big) + \frac{\beta_\delta}{\sqrt{n}} - \tilde{\Delta}_\gamma, \qquad (67)
\end{aligned}$$

where (65) follows from (63) and that

$$\phi(H_{t_i}) = \phi(h) \quad \Rightarrow \quad |Q_*(h,a) - Q_*(H_{t_i}, a)| \leq \Delta_\gamma \leq \tilde{\Delta}_\gamma, \ \forall a \in \mathcal{A}; \qquad (68)$$

and (66) follows from the fact that

$$Q_*(H_{t_i}, A_{t_i}) = R_{t_i+1} + \gamma \cdot \mathbf{P}V_*(H_{t_i}, A_{t_i}). \qquad (69)$$

Consider the following sequence:

$$G_k = \sum_{i=1}^{k} \alpha_n^i \cdot \Big(V_*(H_{t_i+1}) - \mathbf{P}V_*(H_{t_i}, A_{t_i}) \Big), \quad k = 1, \ldots, n, \qquad (70)$$

with $G_0 = 0$. We have that, for $k \geq 1$,

$$\begin{aligned}
\mathbb{E}[G_k | G_{k-1}] &= \mathbb{E}\left[\alpha_n^k \cdot \Big(V_*(H_{t_k+1}) - \mathbf{P}V_*(H_{t_k}, A_{t_k}) \Big) \right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\alpha_n^k \cdot \Big(V_*(H_{t_k+1}) - \mathbf{P}V_*(H_{t_k}, A_{t_k}) \Big) \Big| H_{t_k}, A_{t_k} \right] \right] \\
&= 0, \qquad (71)
\end{aligned}$$

implying that $\{G_k : k = 0, \ldots, n\}$ is a martingale. As a result, it follows from Azuma-Hoeffding inequality that, with probability at least $1 - \delta$,

$$\begin{aligned}
\big|G_n - G_0\big| &= \left| \sum_{i=1}^{n} \alpha_n^i \Big(V_*(H_{t_i+1}) - \mathbf{P}V_*(H_{t_i}, A_{t_i}) \Big) \right| \\
&\leq \frac{4}{(1-\gamma)^{3/2}} \cdot \frac{1}{\sqrt{n}} \cdot \sqrt{\log \frac{2}{\delta}}, \qquad (72)
\end{aligned}$$

where we used assertion (b) of Lemma 9 and the fact that

$$\Big|V_*(H_{t_i+1}) - \mathbf{P}V_*(H_{t_i}, A_{t_i})\Big| \leq \frac{1}{1-\gamma}. \tag{73}$$

Scaling δ to δ/T and applying union bounds, we have that, with probability at least $1-\delta$, simultaneously for all $h \in \mathcal{H}, a \in \mathcal{A}$ and $n \geq 1$, as long as $\big(s(h), a\big)$ is updated not more than n times in timesteps $1, 2 \ldots, T$,

$$\left|\sum_{i=1}^{n} \alpha_n^i \bigg(V_*(H_{t_i+1}) - \mathbf{P}V_*(H_{t_i}, A_{t_i})\bigg)\right| \leq \frac{4}{(1-\gamma)^{3/2}} \cdot \frac{1}{\sqrt{n}} \cdot \sqrt{\log \frac{2T}{\delta}}. \tag{74}$$

We denote the above event by $\mathfrak{E}$. Recall that we choose

$$\beta_\delta = \frac{4}{(1-\gamma)^{3/2}} \cdot \sqrt{\log \frac{2T}{\delta}}. \tag{75}$$

As a result, following (67), conditioned on event $\mathfrak{E}$,

$$\hat{Q}_n - Q_* \geq \gamma \cdot \sum_{i=1}^{n} \alpha_n^i \cdot \big(V_{t_i}\big(H_{t_i+1}\big) - V_*(H_{t_i+1})\big) - \tilde{\Delta}_\gamma. \tag{76}$$

We will now show our desired result by induction. Assume that event $\mathfrak{E}$ occurs. At $t = 0$, from our requirements on Q', obviously there is $V_0(h) \geq V_*(h) - \tilde{\Delta}_\gamma/(1-\gamma)$ for all $h \in \mathcal{H}$. Suppose that the result holds for all $t < t'$. At $t = t'$, for all $(h, a) \in \mathcal{S} \times \mathcal{A}$, as long as $\big(\phi(h), a\big)$ is updated $n \geq 1$ times in timesteps $1, 2 \ldots, t'$, from (76) we have

$$\begin{aligned} Q_{t'}(h,a) - Q_*(h,a) \quad &\geq \quad \gamma \cdot \sum_{i=1}^{n} \alpha_n^i \cdot \big(V_{t_i}\big(H_{t_i+1}\big) - V_*(H_{t_i+1})\big) - \tilde{\Delta}_\gamma \\ &\geq \quad \gamma \cdot \left(-\frac{\tilde{\Delta}_\gamma}{1-\gamma}\right) - \tilde{\Delta}_\gamma \\ &= \quad -\frac{\tilde{\Delta}_\gamma}{1-\gamma}. \end{aligned} \tag{77}$$

Otherwise, if $\big(\phi(h), a\big)$ is not updated in timesteps $1, 2, \ldots, t'$, then $V_{t'}(h) = V_0(h) \geq V_*(h) - \tilde{\Delta}_\gamma/(1-\gamma)$. This leads to

$$V_{t'}(h) - V_*(h) = \min\left\{\max_{a' \in \mathcal{A}} Q_{t'}(h, a'),\ \frac{1}{1-\gamma}\right\} - V_*(h) \geq -\frac{\tilde{\Delta}_\gamma}{1-\gamma}$$

for all $h \in \mathcal{H}$. Therefore, the result holds for all $0 \leq t \leq T$. ■

Note that one direct implication of Lemma 12 is that $\chi_t \geq 0$ for all $t = 0, 1, \ldots, T$.

C.3.3 A High-Probability Bound

In this subsection we will prove the following lemma.

Lemma 13 *If Algorithm 3 is executed with γ, β_δ and Q' specified in Lemma 12, then with probability at least $1-\delta$,*

$$\sum_{t=0}^{T-1}\big(\xi_t-\chi_t\big)\le\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1}{1-\gamma}+\frac{24}{(1-\gamma)^{3/2}}\cdot\sqrt{\mathcal{SA}T\cdot\log\frac{2T}{\delta}},$$

where ξ_t and χ_t are defined in (59) *and* (58)*, respectively.*

Proof. For the moment let us fix $t\in\{0,1,\ldots,T\}$, and consider the situational state-action pair $\big(\phi(H_t),A_t\big)$, which has been updated $n\ge 1$ times before (and including) timestep t. Let $1\le t_1<\cdots<t_n\le t$ be the timestep in which $\big(\phi(H_t),A_t\big)$ is updated. Recall that

$$\phi(H_{t_i})=\phi(H_t),\quad A_{t_i}=A_t,\quad\forall i=1,\ldots,n. \tag{78}$$

We have that, conditioned on event $\mathfrak{E}$,

$$\begin{aligned}
Q_t(H_t,A_t)-Q_*(H_t,A_t) &= \sum_{i=1}^{n}\alpha_n^i\cdot\left(R_{t_i+1}+\gamma\cdot V_{t_i}\big(H_{t_i+1}\big)+\frac{\beta_\delta}{\sqrt{i}}-Q_*(H_t,A_t)\right)\\
&\le \left\{\sum_{i=1}^{n}\alpha_n^i\cdot\Big[\gamma\cdot V_{t_i}\big(h_{t_i+1}\big)-\gamma\cdot\mathbf{P}V_*\big(H_{t_i},A_{t_i}\big)\Big]\right\}\\
&\quad+\tilde{\Delta}_\gamma+\sum_{i=1}^{n}\alpha_n^i\cdot\frac{\beta_\delta}{\sqrt{i}} \qquad\qquad (79)\\
&\le \left\{\gamma\cdot\sum_{i=1}^{n}\alpha_n^i\cdot\Big[V_{t_i}\big(H_{t_i+1}\big)-\mathbf{P}V_*\big(H_{t_i},A_{t_i}\big)\Big]\right\}\\
&\quad+\tilde{\Delta}_\gamma+\frac{2\beta_\delta}{\sqrt{n}} \qquad\qquad (80)\\
&= \left\{\gamma\cdot\sum_{i=1}^{n}\alpha_n^i\cdot\Big[V_{t_i}\big(H_{t_i+1}\big)-V_*\big(H_{t_i+1}\big)\Big]\right\}+\tilde{\Delta}_\gamma+\frac{2\beta_\delta}{\sqrt{n}}\\
&\quad+\left\{\gamma\cdot\sum_{i=1}^{n}\alpha_n^i\cdot\Big[V_*\big(H_{t_i+1}\big)-\mathbf{P}V_*\big(H_{t_i},A_{t_i}\big)\Big]\right\}\\
&\le \left\{\gamma\cdot\sum_{i=1}^{n}\alpha_n^i\cdot\Big[V_{t_i}\big(H_{t_i+1}\big)-V_*\big(H_{t_i+1}\big)\Big]\right\}+\tilde{\Delta}_\gamma+\frac{3\beta_\delta}{\sqrt{n}}, \qquad (81)
\end{aligned}$$

where (79) follows from the fact that

$$\phi(H_{t_i})=\phi(H_t)\quad\Rightarrow\quad|Q_*(H_t,a)-Q_*(H_{t_i},a)|\le\Delta_\gamma\le\tilde{\Delta}_\gamma,\ \forall a\in\mathcal{A}; \tag{82}$$

inequality (80) follows from assertion (a) of Lemma 9; and (81) follows from that, conditioned on the event $\mathfrak{E}$,

$$\left|\sum_{i=1}^{n} \alpha_n^i \cdot \Big[V_*\big(H_{t_i+1}\big) - \mathbf{P}V_*\big(H_{t_i}, A_{t_i}\big)\Big]\right| \leq \frac{\beta_\delta}{\sqrt{n}}. \tag{83}$$

Combining (48) and (81), we have

$$\begin{aligned} V_t(H_t) - V_*(H_t) &\leq V_t(H_t) - Q_*(H_t, A_t) \\ &\leq Q_t(H_t, A_t) - Q_*(H_t, A_t) \\ &\leq \left\{\gamma \cdot \sum_{i=1}^{n} \alpha_n^i \cdot \Big[V_{t_i}\big(H_{t_i+1}\big) - V_*\big(H_{t_i+1}\big)\Big]\right\} + \tilde{\Delta}_\gamma + \frac{3\beta_\delta}{\sqrt{n}}. \end{aligned} \tag{84}$$

For each $i = 1, \ldots, n$, if $\phi(H_{t_i+1}) = \phi(H_{t_i})$, then

$$\begin{aligned} V_{t_i}\big(H_{t_i+1}\big) - V_*\big(H_{t_i+1}\big) &= V_{t_i}\big(H_{t_i}\big) - V_*\big(H_{t_i+1}\big) \\ &\leq V_{t_i}\big(H_{t_i}\big) - V_*\big(H_{t_i}\big) + \tilde{\Delta}_\gamma. \end{aligned} \tag{85}$$

Otherwise, if $\phi(H_{t_i+1}) \neq \phi(H_{t_i})$, then $\phi(H_{t_i+1})$ is not updated at timestep $t_i + 1$, leading to

$$V_{t_i}\big(H_{t_i+1}\big) = V_{t_i+1}\big(H_{t_i+1}\big). \tag{86}$$

Combining the two cases, we can claim that there exists $1 \leq w_1 < \cdots < w_n \leq t + 1$, such that

$$\begin{aligned} V_t(H_t) - V_*(H_t) &\leq Q_t(H_t, A_t) - Q_*(H_t, A_t) \\ &\leq \left\{\gamma \cdot \sum_{i=1}^{n} \alpha_n^i \cdot \Big[V_{w_i}\big(H_{w_i}\big) - V_*\big(H_{w_i}\big)\Big]\right\} + 2\tilde{\Delta}_\gamma + \frac{3\beta_\delta}{\sqrt{n}}. \end{aligned} \tag{87}$$

Note that (87) only applies to times t after which has been updated at least once. Suppose that $\big(\phi(H_t), A_t\big)$ is first updated at time $t + 1$ (meaning that it has not been updated even once prior to timestep t); then, naturally there should be

$$V_t(H_t) - V_*(H_t) \leq \frac{1}{1-\gamma}. \tag{88}$$

Recall that n is the number of times at which the value of $\big(\phi(H_t), A_t\big)$ is updated before (and including) timestep t. We now let t to take values in $0, 1, \ldots, T$, and replace n and w_i by n_t and $w_{t,i}$ respectively to reflect their dependence on t. Summing all sides of (87) from $t = 0$ to $t = T - 1$, and considering (88), we have

$$\begin{aligned} \sum_{t=0}^{T-1}\Big(V_t(H_t) - V_*(H_t)\Big) &\leq \sum_{t=0}^{T-1}\Big(Q_t(H_t, A_t) - Q_*(H_t, A_t)\Big) \\ &\leq \left\{\gamma \cdot \sum_{t=0}^{T-1}\sum_{i=1}^{n_t} \alpha_{n_t}^i \cdot \Big[V_{w_{t,i}}\big(H_{w_{t,i}}\big) - V_*\big(H_{w_{t,i}}\big)\Big]\right\} \\ &\quad + 2\tilde{\Delta}_\gamma T + \frac{\mathcal{S}\mathcal{A}}{1-\gamma} + \sum_{t=0}^{T-1}\frac{3\beta_\delta}{\sqrt{n_t}}, \end{aligned} \tag{89}$$

where we note that there can be at most $\mathcal{SA}$ many "first visits." We now want to determine whether $V_T(H_T) - V_*(H_T)$ appears in the summation on the right-hand side of (89). Based on (85) and (86), if it appears, there must be

$$w_{T-1,n_{T-1}} = T, \tag{90}$$

meaning that the situational state-action pair $\big(\phi(H_T), A_T\big)$ has never been visited in $t = 0, 1, \ldots, T-1$, which leads to $V_T(H_T) - V_*(H_T) \leq 1/(1-\gamma)$. Therefore, we can claim that

$$\begin{aligned}\sum_{t=0}^{T-1}\Big(V_t(H_t) - V_*(H_t)\Big) \quad &\leq \quad \left\{\gamma \cdot \sum_{t=0}^{T-1}\sum_{i=1}^{n_t}\alpha_{n_t}^i \cdot \Big[V_{w_{t,i}}\big(H_{w_{t,i}}\big) - V_*\big(H_{w_{t,i}}\big)\Big]\right\} \\ &\qquad + 2\tilde{\Delta}_\gamma T + \frac{\mathcal{SA}+1}{1-\gamma} + \sum_{t=0}^{T-1}\frac{3\beta_\delta}{\sqrt{n_t}},\end{aligned} \tag{91}$$

where for all t and i, $w_{t,i} \leq t-1$.

We have shown that conditioned on event $\mathfrak{E}$, $\chi_k \geq 0$ for all k. Inequality (89) implies that

$$\begin{aligned}\sum_{t=0}^{T-1}\xi_t \quad &\leq \quad \left\{\gamma \cdot \sum_{t=0}^{T-1}\sum_{i=1}^{n_t}\alpha_{n_t}^i \cdot \left(\chi_{w_{t,i}} - \frac{\tilde{\Delta}_\gamma}{1-\gamma}\right)\right\} + 2\tilde{\Delta}_\gamma T + \frac{\mathcal{SA}+1}{1-\gamma} + \sum_{t=0}^{T-1}\frac{3\beta_\delta}{\sqrt{n_t}}, \\ &= \quad \left\{\gamma \cdot \sum_{t=0}^{T-1}\sum_{i=1}^{n_t}\alpha_{n_t}^i \cdot \chi_{w_{t,i}}\right\} + \frac{2-3\gamma}{1-\gamma} \cdot \tilde{\Delta}_\gamma T + \frac{\mathcal{SA}+1}{1-\gamma} + \sum_{t=0}^{T-1}\frac{3\beta_\delta}{\sqrt{n_t}},\end{aligned} \tag{92}$$

Examining the first term on the right-hand side of (92), from Lemma 9 (c), there should be

$$\sum_{t=0}^{T-1}\sum_{i=1}^{n_t}\alpha_{n_t}^i \cdot \chi_{w_{t,i}} \leq \frac{3-\gamma}{2}\sum_{t=0}^{T-1}\chi_t. \tag{93}$$

In terms of the last term in (92), letting $n_T(s,a)$ be the number of times situational state-action pair (s,a) is updated at times $t = 1, \ldots, T$, we have

$$\begin{aligned}\sum_{t=0}^{T-1}\frac{1}{\sqrt{n_t}} \quad &= \quad \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{m=1}^{n_T(s,a)}\frac{1}{\sqrt{m}} \\ &\leq \quad \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} 2\sqrt{n_T(s,a)} \\ &\leq \quad 2\sqrt{\mathcal{SA}\cdot\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} n_T(s,a)} \\ &= \quad 2\sqrt{\mathcal{SA}T},\end{aligned} \tag{94}$$

where in the final step we used the fact that

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} n_T(s,a) = T.$$

Now we can revisit (92), starting from which there is

$$\begin{aligned}\sum_{t=0}^{T-1}\xi_t &\leq \left\{\gamma\cdot\sum_{t=0}^{T-1}\sum_{i=1}^{n_t}\alpha_{n_t}^i\cdot\chi_{w_{t,i}}\right\}+\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1}{1-\gamma}+\sum_{t=0}^{T-1}\frac{3\beta_\delta}{\sqrt{n_t}}\\ &\leq \frac{\gamma(3-\gamma)}{2}\cdot\left(\sum_{t=0}^{T-1}\chi_t\right)+\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1}{1-\gamma}+6\beta_\delta\sqrt{\mathcal{SA}T}. \qquad (95)\end{aligned}$$

Equivalently,

$$\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\leq\left(\frac{\gamma(3-\gamma)}{2}-1\right)\cdot\left(\sum_{t=0}^{T-1}\chi_t\right)+\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1}{1-\gamma}+6\beta_\delta\sqrt{\mathcal{SA}T}. \quad (96)$$

Recall that conditioned on event $\mathfrak{E}$, $\chi_t\geq 0$ for all t, and further we have

$$\frac{\gamma(3-\gamma)}{2}-1<0,\quad \forall\gamma\in(0,1).$$

Thus, we can drop the first term in (96) and deduce that, with probability $1-\delta$,

$$\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\leq\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1}{1-\gamma}+\frac{24}{(1-\gamma)^{3/2}}\cdot\sqrt{\mathcal{SA}T\cdot\log\frac{2T}{\delta}}. \qquad (97)$$

C.3.4 Finishing the Proof of Theorem 10

It remains to show that the high-probability bound in Lemma 13 also implies a bound on the expected sum of $\xi_t-\chi_t$. For all $\delta>0$, let $\mathfrak{E}_\delta$ denote the event in (74). Because

$$\xi_t-\chi_t=V_*(H_t)-Q_*(H_t,A_t)-\frac{\tilde{\Delta}_\gamma}{1-\gamma}\leq\frac{1}{1-\gamma}, \qquad (98)$$

we have that

$$\mathbb{E}\left[\left\{\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\right\}\mathbf{1}(\mathfrak{E}^c)\right]\leq\frac{T}{1-\gamma}\cdot\left(1-\mathbb{P}(\mathfrak{E}_\delta)\right)=\frac{\delta T}{1-\gamma}. \qquad (99)$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\right] &= \mathbb{E}\left[\left\{\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\right\}\mathbf{1}(\mathfrak{E})\right]+\mathbb{E}\left[\left\{\sum_{t=0}^{T-1}\left(\xi_t-\chi_t\right)\right\}\mathbf{1}(\mathfrak{E}^c)\right]\\ &\leq \frac{24}{(1-\gamma)^{3/2}}\cdot\sqrt{\mathcal{SA}T\cdot\log\frac{2T}{\delta}}+\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{SA}+1+\delta T}{1-\gamma}.\end{aligned} \qquad (100)$$

Letting $\delta=1/T$, we have that

$$\beta=\frac{4}{(1-\gamma)^{3/2}}\sqrt{\log(2T^2)}, \qquad (101)$$

and

$$\mathbb{E}\left[\sum_{t=0}^{T-1}\big(\xi_t-\chi_t\big)\right]\le\frac{24}{(1-\gamma)^{3/2}}\cdot\sqrt{\mathcal{S}\mathcal{A}T\cdot\log(2T^2)}+\frac{2-3\gamma}{1-\gamma}\cdot\tilde{\Delta}_\gamma T+\frac{\mathcal{S}\mathcal{A}+2}{1-\gamma}. \tag{102}$$

Plugging the above inequality into (60), we arrive at

$$\begin{aligned}\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(V_*(H_t)-V_{\mathring{\pi}}(H_t)\Big)\right] &\le \frac{24}{(1-\gamma)^{\frac{5}{2}}}\cdot\sqrt{\mathcal{S}\mathcal{A}T\cdot\log(2T^2)}\\ &\quad+\left(\frac{2-3\gamma}{(1-\gamma)^2}+\frac{1}{(1-\gamma)^2}\right)\cdot\tilde{\Delta}_\gamma T\\ &\quad+\frac{\mathcal{S}\mathcal{A}+2}{(1-\gamma)^2}+\frac{1}{(1-\gamma)^2}\\ &= \frac{24}{(1-\gamma)^{\frac{5}{2}}}\cdot\sqrt{\mathcal{S}\mathcal{A}T\cdot\log(2T^2)}\\ &\quad+\frac{3\tilde{\Delta}_\gamma T}{1-\gamma}+\frac{\mathcal{S}\mathcal{A}+3}{(1-\gamma)^2},\end{aligned}$$

which concludes the proof of Theorem 10. ■

C.4 From Discounted Return to Average Reward

In this section we still continue to study the discounted Q-learning subroutine Algorithm 3, but we shift our focus to the learning performance with respect to the average reward. Our goal is to show the following stronger version of Theorem 3.

Theorem 14 *For all $\tau\ge1$ and $T>\tau\cdot\log(T)$, if Algorithm 3 is executed with $\gamma=1-1/\tau$,*

$$\beta=4\tau^{3/2}\sqrt{\log(2T^2)}, \tag{103}$$

and Q' such that for some $\iota\ge0$,

$$Q_*^\gamma(h,a)-\iota\tau\le Q'\big(\phi(h),a\big)\le\tau,\quad\forall h\in\mathcal{H},a\in\mathcal{A}, \tag{104}$$

then for all $\pi'\in\mathcal{P}$ and initial history $h\in\mathcal{H}$, we have that

$$\begin{aligned}\mathbb{E}_{\mathring{\pi}}\left[\sum_{t=0}^{T-1}\Big(\lambda_{\pi'}-R_{t+1}\Big)\right] &\le 24\tau^{3/2}\cdot\sqrt{\mathcal{S}\mathcal{A}T\cdot\log(2T^2)}+3\Big[\tilde{\Delta}_\tau+\tau_{\pi'}/\tau\Big]\cdot T\\ &\quad+\Big[\mathcal{S}\mathcal{A}+5+2\log(T)\Big]\cdot\tau,\end{aligned} \tag{105}$$

where $\tilde{\Delta}_\tau=\max\{\Delta_\tau,\iota\}$.

Proof. First notice that for all $t\ge0$,

$$\mathbb{E}_{\mathring{\pi}}\Big[V_{\mathring{\pi}}^\gamma(H_t)\Big]=\mathbb{E}\left[\sum_{\ell=0}^{\infty}\gamma^\ell R_{t+\ell+1}\right]. \tag{106}$$

Thus, we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=0}^{T-1} V^{\gamma}_{\pi'}(H_t) - V^{\gamma}_{\mathring{\pi}}(H_t)\right] &\geq \mathbb{E}\left[\sum_{t=0}^{T-1}\left(\frac{\lambda_{\pi'}}{1-\gamma} - \tau_{\pi'}\right)\right] - \mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{\ell=0}^{\infty}\gamma^{\ell} R_{t+\ell+1}\right] \quad (107)\\
&= \mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{\ell=0}^{\infty}\gamma^{\ell}\cdot\big(\lambda_{\pi'} - R_{t+\ell+1}\big)\right] - \tau_{\pi'}\cdot T\\
&= \mathbb{E}\left[\sum_{t=0}^{T-1}\frac{1-\gamma^{t+1}}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right]\\
&\quad + \mathbb{E}\left[\sum_{t=T}^{\infty}\gamma^{t+1-T}\frac{1-\gamma^{T}}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right]\\
&\quad - \tau_{\pi'}\cdot T, \qquad (108)
\end{aligned}$$

where (107) is the result of Lemma 2. Notice that, since $|\lambda_{\pi'} - R_t| \leq 1$,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=T}^{\infty}\gamma^{t+1-T}\frac{1-\gamma^{T}}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right] &\geq -\sum_{t=T}^{\infty}\gamma^{t+1-T}\frac{1-\gamma^{T}}{1-\gamma}\\
&= -\frac{1-\gamma^{T}}{1-\gamma}\cdot\frac{\gamma}{1-\gamma}\\
&\geq -\frac{1}{(1-\gamma)^2}. \qquad (109)
\end{aligned}$$

Let $T_0^{\gamma} = \lfloor \log(T)/(1-\gamma)\rfloor$, then for all $t > T_0^{\gamma}$,

$$\gamma^{t} \leq \gamma^{\frac{\log T}{1-\gamma}} \leq \left(\frac{1}{e}\right)^{\log T} = \frac{1}{T}. \qquad (110)$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\frac{1-\gamma^{t+1}}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right] &= \mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\frac{1}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right]\\
&\quad - \mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\frac{\gamma^{t+1}}{1-\gamma}\cdot\big(\lambda_{\pi'} - R_{t+1}\big)\right]\\
&\geq \frac{1}{1-\gamma}\cdot\mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\big(\lambda_{\pi'} - R_{t+1}\big)\right] - \sum_{t=T_0^{\gamma}}^{T-1}\frac{\gamma^{t+1}}{1-\gamma}\\
&\geq \frac{1}{1-\gamma}\cdot\mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\big(\lambda_{\pi'} - R_{t+1}\big)\right] - \frac{1}{(1-\gamma)T}\cdot T\\
&\geq \frac{1}{1-\gamma}\cdot\mathbb{E}\left[\sum_{t=T_0^{\gamma}}^{T-1}\big(\lambda_{\pi'} - R_{t+1}\big)\right] - \frac{1}{(1-\gamma)}. \qquad (111)
\end{aligned}$$

On the other hand,

$$\mathbb{E}\left[\sum_{t=0}^{T_0^\gamma-1}\frac{1-\gamma^{t+1}}{1-\gamma}\cdot\big(\lambda_{\pi'}-R_{t+1}\big)\right]\geq-\frac{1}{1-\gamma}\cdot T_0^\gamma\geq-\frac{\log(T)}{(1-\gamma)^2}. \tag{112}$$

From Theorem 10, there is also

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{T-1}V_{\pi'}^{\gamma}(H_t)-V_{\mathring{\pi}}^{\gamma}(H_t)\right] &\leq \mathbb{E}\left[\sum_{t=0}^{T-1}V_{*}^{\gamma}(H_t)-V_{\mathring{\pi}}^{\gamma}(H_t)\right]\\ &\leq 24\tau^{5/2}\cdot\sqrt{\mathcal{SA}T\log(2T^2)}+3\tilde{\Delta}_\tau\cdot\tau\cdot T+\Big[\mathcal{SA}+3\Big]\cdot\tau^2,\end{aligned} \tag{113}$$

where we note that $\tilde{\Delta}_\gamma=\tilde{\Delta}_\tau$ since $\gamma=1-1/\tau$. Combining (108)-(113), we have that

$$\begin{aligned}\tau\cdot\mathbb{E}\left[\sum_{t=T_0^\gamma}^{T-1}\big(\lambda_{\pi'}-R_{t+1}\big)\right] &\leq 24\tau^{5/2}\cdot\sqrt{\mathcal{SA}T\log(2T^2)}+3\tilde{\Delta}_\tau\cdot\tau\cdot T+\Big[\mathcal{SA}+3\Big]\cdot\tau^2\\ &\quad+\Big[\log(T)+1\Big]\cdot\tau^2+\tau+\tau_{\pi'}\cdot T.\end{aligned} \tag{114}$$

For $T>\tau\cdot\log(T)\geq T_0^\gamma$,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{T-1}\lambda_{\pi'}-R_{t+1}\right] &\leq 24\tau^{3/2}\cdot\sqrt{\mathcal{SA}T\log(2T^2)}+3\tilde{\Delta}_\tau\cdot T+\Big[\mathcal{SA}+3\Big]\cdot\tau\\ &\quad+\Big[\log(T)+1\Big]\cdot\tau+1+\tau_{\pi'}/\tau\cdot T+T_0^\gamma\\ &\leq 24\tau^{3/2}\cdot\sqrt{\mathcal{SA}T\log(2T^2)}+3\tilde{\Delta}_\tau\cdot T+\Big[\mathcal{SA}+5\Big]\cdot\tau\\ &\quad+2\log(T)\cdot\tau+\tau_{\pi'}/\tau\cdot T,\end{aligned} \tag{115}$$

which is what we claim in Theorem 14. ■

C.5 Proof of Lemma 6

The lemma is restated below.

Lemma 15 *For all $\tau\geq1$, $\lambda_*-\lambda_{\tilde{\pi}}\leq\overline{\Delta}_\tau$.*

In fact, for a fixed $\tau\geq1$, let $\gamma=1-1/\tau<1$. Then, for all $\gamma'\in(\gamma,1)$ and $h_1,h_2\in\mathcal{H}$, as long as $\phi(h_1)=\phi(h_2)$, there should be

$$\left|V_*^{\gamma'}(h_1)-V_*^{\gamma'}(h_1)\right|\leq\overline{\Delta}_\tau. \tag{116}$$

From Theorem 7.1 in Van Roy (2006), for all $\epsilon>0$ and $\gamma'\in(\gamma,1)$, there exists a policy $\tilde{\pi}_\epsilon^{\gamma'}\in\tilde{\mathcal{P}}$ and a distribution $\mu_{\epsilon,\gamma'}$ on $\mathcal{H}$, such that, for all $h\in\mathcal{H}$,

$$\mu_{\epsilon,\gamma'}(h)>0, \tag{117}$$

and

$$(1-\gamma')\sum_{h\in\mathcal{H}} \mu_{\epsilon,\gamma'}(h)\left[V_*^{\gamma'}(h) - V_{\tilde{\pi}_\epsilon^{\gamma'}}^{\gamma'}(h)\right] \le \gamma'\overline{\Delta}_\tau + \epsilon. \tag{118}$$

Taking the limit $\gamma' \uparrow 1$ and noticing that, for all $\pi \in \mathcal{P}$ and $h \in \mathcal{H}$,

$$\lim_{\gamma'\uparrow 1}(1-\gamma')V_\pi^{\gamma'}(h) = \lambda_\pi, \tag{119}$$

we arrive at

$$\limsup_{\gamma'\uparrow 1} \lambda_{\pi_*^{\gamma'}}(h) - \lambda_{\tilde{\pi}_\epsilon^{\gamma'}}(h) \le \overline{\Delta}_\tau + \epsilon. \tag{120}$$

This means that, for all $\iota > 0$, there exists $\gamma_\iota \in (\gamma, 1)$, such that

$$\lambda_{\pi_*^{\gamma_\iota}}(h) - \lambda_{\tilde{\pi}_\epsilon^{\gamma_\iota}}(h) \le \overline{\Delta}_\tau + \epsilon + \iota. \tag{121}$$

Since $\lambda_{\tilde{\pi}_\epsilon^{\gamma_\iota}} \le \lambda_{\tilde{\pi}}$, the above inequality implies that, for all $\epsilon, \iota > 0$,

$$\lambda_{\pi_*^{\gamma_\iota}}(h) - \lambda_{\tilde{\pi}}(h) \le \overline{\Delta}_\tau + \epsilon + \iota. \tag{122}$$

Now notice that, following the result in Blackwell (1962), when $\gamma_\iota \uparrow 1$, there should be $\lambda_{\pi_*^{\gamma_\iota}}(h) \to \lambda_*(h)$. Hence, we can take $\epsilon \downarrow 0$ and $\iota \downarrow 0$, which gives us

$$\lambda_*(h) - \lambda_{\tilde{\pi}}(h) \le \overline{\Delta}_\tau, \tag{123}$$

as we desire. ■

C.6 Concluding the Proof of Theorem 4

In this section we complete the final steps towards Theorem 4, which we restate below.

Theorem 4 *For all $\pi \in \mathcal{P}$ and $T \ge 1$,*

$$\mathrm{Regret}_\pi(T) \le \left(120\sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + 5\tau_\pi\right)T^{4/5} + 3\overline{\Delta}_{\tau_\pi}T + (54\mathcal{S}\mathcal{A} + 18\log(T))\,T^{1/5} + 2\tau_\pi^5.$$

With the set of subroutines $\texttt{foo}_1$ to $\texttt{foo}_4$ specified in (18)–(21), the interactions between the agent and the environment can be viewed through epochs $k = 1, 2\ldots$, with each epoch corresponding to executing Algorithm 3 with a fixed a fixed discount factor $\gamma_k = 1 - 1/T_k^{1/5}$, which is equivalent to a fixed effective planning horizon $\tau_k = T_k^{1/5}$, where T_k are the change points, with $T_0 = 1$ and $T_k = 20 \times 2^{k-1}$, $k \ge 1$. In the previous subsections we have analysed the average-reward regret of Algorithm 3 when the discount factor is fixed. In this subsection we will allow the discount factor to change and use a "doubling trick" to bound the total regret of $\widehat{\pi}$. Throughout this section, we assume that the total number of timesteps T is fixed. We start with the following useful lemma.

Lemma 16 *Let $a, S > 0, b \geq 0$ be integers, $\zeta \in (0,1)$ and $f : \mathbb{Z}_+ \mapsto \mathbb{R}$ is such that $f(x) \leq x^\zeta$ for all x. Consider the sequence $t_j = a \cdot 2^{b+j}$, $j = 0, 1, \ldots$ and let k be the minimum index such that $t_0 + t_1 + \cdots + t_k \geq S$. If $k \geq 1$, then*

$$f(t_0) + \cdots + f(t_k) \leq \frac{2^{2\zeta}}{2^\zeta - 1} \cdot S^\zeta. \tag{124}$$

Proof. First notice that

$$t_0 + t_1 + \cdots + t_k \leq 3S. \tag{125}$$

This is because based on the definition,

$$t_0 + t_1 + \cdots + t_{k-1} = a \cdot 2^b \cdot (2^k - 1) \geq a \cdot 2^{b+k-1} = \frac{1}{2} t_k. \tag{126}$$

Hence, if $\sum_{t=0}^k t_k > 3S$, then $\sum_{t=0}^{k-1} t_k > S$, contradicting the minimality of k. Since $\sum_{t=0}^k t_k = a \cdot 2^b \cdot (2^{k+1} - 1)$, we have

$$k \leq 1 + \log_2 \frac{S}{a \cdot 2^b}. \tag{127}$$

Therefore,

$$f(t_0) + \cdots + f(t_k) \leq \sum_{t=0}^k t_k^\zeta \leq a^\zeta \cdot \frac{2^{(k+1)\zeta}}{2^\zeta - 1} \leq \frac{2^{2\zeta}}{2^\zeta - 1} \cdot S^\zeta, \tag{128}$$

as desired. ■

In light of our requirements on Q' in Algorithm 3, we also need the following result.

Lemma 17 *For all γ_1, γ_2 such that $0 \leq \gamma_1 < \gamma_2 < 1$,*

$$Q_*^{\gamma_1}(h,a) - \frac{1}{1-\gamma_1} \geq Q_*^{\gamma_2}(h,a) - \frac{1}{1-\gamma_2}. \tag{129}$$

Proof. For $i = 1, 2$, let $\pi_i \in \mathcal{P}$ be such that, for all $h \in \mathcal{H}$, $\pi_i(\cdot|h)$ is the uniform distribution over $\arg\max_{a' \in \mathcal{A}} Q_*^{\gamma_i}(h, a')$. From Proposition 1, $(\pi_i : i = 1, 2)$ exists, and

$$Q_*^{\gamma_i}(h,a) = \overline{r}_{h,a} + \sum_{h' \in \mathcal{H}} P_{ahh'} \cdot \sum_{t=0}^{\infty} \gamma_i^{t+1} \big(P_{\pi_i}^t \overline{r}_{\pi_i}\big)(h'), \quad \forall i \in \{1,2\}, h \in \mathcal{H}, a \in \mathcal{A}. \tag{130}$$

As a result, we have

$$\begin{aligned}
Q_*^{\gamma_2}(h,a) - Q_*^{\gamma_1}(h,a) &\leq Q_{\pi_2}^{\gamma_2}(h,a) - Q_{\pi_2}^{\gamma_1}(h,a) \\
&= \sum_{h' \in \mathcal{H}} P_{ahh'} \cdot \sum_{t=0}^{\infty} \Big\{ \big(\gamma_2^{t+1} - \gamma_1^{t+1}\big) \cdot \big(P_{\pi_2}^t \overline{r}_{\pi_2}\big)(h') \Big\} \\
&\leq \sum_{h' \in \mathcal{H}} P_{ahh'} \cdot \sum_{t=0}^{\infty} \Big\{ \big(\gamma_2^{t+1} - \gamma_1^{t+1}\big) \cdot 1 \Big\} \\
&= \frac{\gamma_2}{1-\gamma_2} - \frac{\gamma_1}{1-\gamma_1} \\
&= \frac{1}{1-\gamma_2} - \frac{1}{1-\gamma_1},
\end{aligned} \tag{131}$$

which is what we desire. ■

The above lemma, together with Lemma 12 and the subroutine $\texttt{foo}_3$ defined in (20), ensures that at the beginning of epoch k,

$$Q\big(\phi(h), a\big) \geq Q_*^{\gamma_k}(h, a) - \overline{\Delta}_{\tau_k} \cdot \tau_k, \quad \forall k = 0, 1, \ldots, h \in \mathcal{H}, a \in \mathcal{A}. \tag{132}$$

Let $\mathcal{T}_k$ denote the timesteps in the k-th epoch, $k = 0, 1, \ldots, L$, where L is the index of the epoch that contains timestep $T - 1$ (which is the last epoch that we are concerned about), i.e.

$$\begin{aligned}
\mathcal{T}_0 &= \{0, 1, \ldots, 19\}, \\
\mathcal{T}_1 &= \{20, 21, \ldots, 39\}, \\
\mathcal{T}_2 &= \{40, 41, \ldots, 79\}, \\
&\cdots \\
\mathcal{T}_L &= \{20 \cdot 2^{L-1}, 20 \cdot 2^{L-1} + 1, \ldots, T - 1\}.
\end{aligned}$$

Letting $|\mathcal{T}|$ be the length of epoch $\mathcal{T}$, we have that $|\mathcal{T}_k| = T_k$ for $1 \leq k \leq L-1$ and $|\mathcal{T}_0| = 20$, $|\mathcal{T}_L| = T - 20 \cdot 2^{L-1}$. For a fixed reference policy $\pi \in \mathcal{P}$, let

$$\mathcal{R}_\pi(\mathcal{T}_k) = \mathbb{E}\left[\sum_{t \in \mathcal{T}_k} \big(\lambda_\pi - R_{t+1}\big)\Big|\mathcal{E}\right]. \tag{133}$$

For $k \geq 1$, let

$$\gamma_k = 1 - \frac{1}{T_{k-1}^{1/5}} = 1 - \frac{1}{|\mathcal{T}_k|^{1/5}} \tag{134}$$

be the discount factor used in the k-th epoch (where $\gamma_0 = 0$ is the discount factor in the 0-th epoch). By Theorem 14, either

$$\begin{aligned}
\mathcal{R}_\pi(\mathcal{T}_k) &\leq 24\tau_k^{3/2} \cdot \sqrt{\mathcal{S}\mathcal{A}|\mathcal{T}_k| \log(2|\mathcal{T}_k|^2)} + \Big[3\overline{\Delta}_{\tau_\pi} + \tau_\pi/\tau_k\Big] \cdot |\mathcal{T}_k| \\
&\quad + \Big[\mathcal{S}\mathcal{A} + 2\log(|\mathcal{T}_k|) + 5\Big] \cdot \tau_k \\
&\leq \Big(24 \cdot \sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + \tau_\pi\Big) \cdot |\mathcal{T}_k|^{4/5} + 3\overline{\Delta}_{\tau_\pi} \cdot |\mathcal{T}_k| \\
&\quad + \Big[6\mathcal{S}\mathcal{A} + 2\log(T)\Big] \cdot |\mathcal{T}_k|^{\frac{1}{5}},
\end{aligned} \tag{135}$$

or, if $\tau_k \leq \tau_\pi$,

$$\mathcal{R}_\pi(\mathcal{T}_k) \leq |\mathcal{T}_k|. \tag{136}$$

Let

$$\mathbb{I}_k = \begin{cases} 1 & \text{if } \tau_k > \tau_\pi \\ 0 & \text{otherwise} \end{cases}, \tag{137}$$

and let

$$g(t) = \Big(24 \cdot \sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + \tau_{\tilde{\pi}_*}\Big) \cdot t^{\frac{4}{5}} + 3\Delta \cdot t + \Big(6\mathcal{S}\mathcal{A} + 2\log(T)\Big) \cdot t^{\frac{1}{5}}. \tag{138}$$

We arrive at

$$\begin{aligned}
\mathrm{Regret}_\pi(T) &\leq \sum_{k=0}^{L} \mathcal{R}_\pi(\mathcal{T}_k) \\
&\leq \sum_{k=0}^{L} |\mathcal{T}_k| \cdot \mathbf{1}(\mathbb{I}_k = 0) + g(|\mathcal{T}_k|) \cdot \mathbf{1}(\mathbb{I}_k = 1) \\
&= \sum_{k=0}^{L} |\mathcal{T}_k| \cdot \mathbf{1}\left\{\tau_k \leq \tau_\pi\right\} + g(|\mathcal{T}_k|) \cdot \mathbf{1}\left\{\tau_k > \tau_\pi\right\} \\
&= \sum_{k=0}^{L} |\mathcal{T}_k| \cdot \mathbf{1}\left\{|\mathcal{T}_k| \leq \tau_\pi^5\right\} + g(|\mathcal{T}_k|) \cdot \mathbf{1}\left\{|\mathcal{T}_k| > \tau_\pi^5\right\} \\
&\leq 2\tau_\pi^5 + \sum_{k=0}^{L} g(|\mathcal{T}_k|) \cdot \mathbf{1}\left\{|\mathcal{T}_k| > \tau_\pi^5\right\}. \qquad (139)
\end{aligned}$$

By Lemma 16,

$$\sum_{k=0}^{L} g(|\mathcal{T}_k|) \cdot \mathbf{1}\left\{|\mathcal{T}_k| > \tau_\pi^5\right\} \leq \Big(120\sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + 5\tau_\pi\Big) \cdot T^{\frac{4}{5}} + 3\Delta T + \Big(54\mathcal{S}\mathcal{A} + 18\log(T)\Big) \cdot T^{\frac{1}{5}}.$$

Therefore, we have

$$\begin{aligned}
\mathrm{Regret}_\pi(T) \leq\ & \Big(120\sqrt{\mathcal{S}\mathcal{A}\log(2T^2)} + 5\tau_\pi\Big) \cdot T^{\frac{4}{5}} + \Big(54\mathcal{S}\mathcal{A} + 18\log(T)\Big) \cdot T^{\frac{1}{5}} \\
& + 3\Delta \cdot T + 2\tau_\pi^5, \qquad (140)
\end{aligned}$$

which justifies our claim in Theorem 4. ■

Appendix D. Results on Approximate Dynamic Programming

Theorem 18 *There exists an environment $e_0 = (\mathcal{A}, \mathcal{O}, \rho)$, a set of situational states $\mathcal{S}$, a situational state update function f and a reward function r, such that, conditioned on $\mathcal{E} = e_0$,*

$$\lambda_* - \lambda_{\tilde{\pi}} = \overline{\Delta}_\tau, \quad \forall \tau \geq 1. \tag{141}$$

Proof. Consider an environment with two actions $\mathcal{A} = \{1, 2\}$ and two observations $\mathcal{O} = \{0, 1\}$. The observation probabilities are given by

$$\rho(1|h, a) = \begin{cases} 1 & \text{if } a_{|h|-1}(h) \neq a \\ 0 & \text{otherwise} \end{cases}, \quad \forall h \neq H_0, \tag{142}$$

and

$$\rho(0|H_0, a) = 1. \tag{143}$$

In other words, the observation is deterministically 1 if and only if the agent takes a different action from the one that it took in the previous timestep, and the observation is always 0 in the first timestep. There is only one situational state $\mathcal{S} = \{1\}$, and the situational state update function is

$$f(s, a, o) = 1, \quad \forall s, a, o. \tag{144}$$

The reward is equal to the observation, i.e.

$$r(s, a, o) = o. \tag{145}$$

Notice that $\lambda_* = 1$ in this environment, which can be attained by alternating between the two actions in each timestep. We let this environment be e_0.

We claim that, conditioned on $\mathcal{E} = e_0$, $\Delta_\gamma = 1$ for all $\gamma \in (0, 1)$. Since there is only one situational state, we only have to verify that, for all history pairs (h_1, h_2) and action $a \in \{1, 2\}$,

$$\left|Q_*^\gamma(h_1, a) - Q_*^\gamma(h_2, a)\right| \leq 1. \tag{146}$$

Indeed, for all history h and action a, we have

$$Q_*^\gamma(h, a) = \begin{cases} \frac{1}{1-\gamma} & \text{if } h \neq H_0 \text{ and } a \neq a_{|h|-1}(a) \\ \frac{\gamma}{1-\gamma} & \text{otherwise} \end{cases}. \tag{147}$$

Since $Q_*^\gamma(h, a)$ can only take the above two values, (146) obviously holds. Thus,

$$\overline{\Delta}_\tau = 1, \quad \forall \tau \geq 1. \tag{148}$$

However, there are only two policies in $\tilde{\mathcal{P}}$, one always taking action 1, and the other always taking action 2. Either policy results in an all-zero reward sequence, implying that $\lambda_{\tilde{\pi}} = 0$. Thus,

$$\lambda_* - \lambda_{\tilde{\pi}} = 1, \tag{149}$$

as we desire. ■

Next we show that an intuitive learning algorithm may generate a policy whose performance is much worse than $\tilde{\pi}$, and this happens even when the environment dynamics are known.

The learning algorithm proceeds as follows. In iteration k, the algorithm independently samples $o_1^{(k)}, \ldots, o_M^{(k)} \in \mathcal{O}$ according to the uniform distribution. The value function is then updated via

$$V^{(k)} \leftarrow \underset{V:\mathcal{S}\mapsto\mathbb{R}}{\arg\min} \sum_{m=1}^{M} \left[V\big(\phi(o_m^{(k)})\big) - \max_{a\in\mathcal{A}} \left\{\overline{r}_{ao_m^{(k)}} + \gamma \cdot \mathbf{P}\big(V^{(k-1)} \circ \phi\big)(o_m^{(k)}, a)\right\}\right]^2, \tag{150}$$

where $\gamma \in (0,1)$ is a fixed discount factor. Note that if the environment dynamics function ρ is known, then the algorithm is able to compute $\overline{r}$ and $\mathbf{P}$, otherwise it has to use fitted values of $\overline{r}$ and $\mathbf{P}$ in (150). This algorithm is a version of the fitted value iteration algorithm with ℓ_2-norm loss function, as is studied in (Munos and Szepesvári, 2008). It is worth mentioning that, in general, fitted value iteration need not converge. However, since the update rule (150) amonts to fitted value iteration with aggregated states, the sequence $\big(V^{(k)} : k = 0, 1, 2, \dots\big)$ always converges in probability to $V^{(\infty)} : \mathcal{S} \to \mathbb{R}$ as per Munos and Szepesvári (2008). Using an example adapted from Van Roy (2006), we have the following result.

Theorem 19 *If the value function is updated via* (150)*, then for all* $\gamma \in (0,1)$ *and* $\epsilon > 0$*, there exists integer* M *and an environment* e *such that, conditioned on* $\mathcal{E} = e$*, for all* $\tau \geq 1$ *and* $\pi^{(\infty)}$ *greedy with respect to* $V^{(\infty)}$*,*

$$\lambda_* - \lambda_{\pi^{(\infty)}} \geq \tau_{\pi^{(\infty)}} \cdot \gamma \cdot \overline{\Delta}_\tau - \epsilon. \tag{151}$$

Proof. For integer $N \geq 2$, we can consider an MDP with $2N$ states $\{1, 2, \dots, 2N\}$ and two actions $\{1, 2\}$. Note that this MDP can be viewed as an environment $e_N = (\mathcal{A}, \mathcal{O}, \rho)$, where $\mathcal{A} = \{1, 2\}$, $\mathcal{O} = \{1, 2, \dots, 2N\}$ with observations corresponding to the MDP states (which we will simply call "states" hereafter), and ρ depends on the history only through the most recent observation. The $2N$ observations (or states) induce $2N$ equivalence classes in the set of histories $\mathcal{H}$, where $h \sim h'$ if and only if their last observations are the same. Henceforth we will use observation o to denote the equivalence class in $\mathcal{H}$ induced by o. We will also not distinguish between "observations" and "states."

In every timestep, with probability ϵ_1 the system "resets" itself, and the next state is drawn uniformly from $\{1, 2, \dots, 2N\}$. Conditioned on that the system does not reset, from states $3, 5, \dots, 2N-1$ the system transitions deterministically to state 1, and from states $4, 6, \dots, 2N$ the system transitions deterministically to state 2, regardless of the action; from state 1 the system transitions to state 2 with probability ϵ_2 under both actions; and from state 2, the system transitions to state 1 deterministically under action 1, and stays in state 2 deterministically under action 2. The environment dynamics ρ can thus be written as

$$\rho(1|o,a) = \begin{cases} (1-\epsilon_1)(1-\epsilon_2) + \frac{\epsilon_1}{2N} & \text{if } o = 1 \\ 1 - \epsilon_1 + \frac{\epsilon_1}{2N} & \text{if } \big(o \in \{3, 5, \dots, 2N-1\}\big) \text{ or } \big(o = 2, a = 1\big) \\ 0 & \text{otherwise} \end{cases},$$

$$\rho(2|o,a) = \begin{cases} (1-\epsilon_1)\epsilon_2 + \frac{\epsilon_1}{2N} & \text{if } o = 1 \\ 1 - \epsilon_1 + \frac{\epsilon_1}{2N} & \text{if } \big(o \in \{4, 6, \dots, 2N\}\big) \text{ or } \big(o = 2, a = 2\big) \\ 0 & \text{otherwise} \end{cases},$$

and

$$\rho(x|o,a) = \frac{\epsilon_1}{2N}, \quad \forall x \in \{3, 4, 5, \dots, 2N\}, o \in \mathcal{O}, a \in \mathcal{A}.$$

Let $\delta, \kappa > 0$. From every state in $\{4, 6, \ldots, 2N\}$, the agent receives reward δ if it takes action 1, and reward $-\kappa$ if it takes action 2. From every state in $\{3, 5, \ldots, 2N-1\}$, the agent receives reward $-\delta$ regardless of the action. Additionally, the agent also receives reward $-\kappa$ if it takes action 2 in state 2. In all other scenarios the agent receives zero reward. The situational state space is $\mathcal{S} = \{1, 2\}$, with $\phi(o) = 1$ whenever $o \in \{1, 3, \ldots, 2N-1\}$ and $\phi(o) = 2$ whenever $o \in \{2, 4, \ldots, 2N\}$.

Conditioned on $\mathcal{E} = e_N$, for a fixed $\gamma \in [0, 1)$, we can verify that

$$Q_*^\gamma(1, a) = 0, \quad a \in \{1, 2\},$$

and

$$Q_*^\gamma(k, a) = -\delta, \quad \forall k \in \{3, 5, \ldots, 2N-1\}, a \in \{1, 2\}.$$

There is also

$$Q_*^\gamma(2, 1) = 0, \quad Q_*^\gamma(2, 2) = -\kappa,$$

and

$$Q_*^\gamma(k, 1) = \delta, \quad Q_*^\gamma(k, 2) = -\kappa, \quad \forall k \in \{4, 6, \ldots, 2N\}.$$

Since all states in $\{1, 3, \ldots, 2N-1\}$ are mapped to situational state 1 and all states in $\{2, 4, \ldots, 2N\}$ are mapped to situational state 2, we have that

$$\overline{\Delta}_\tau = \delta,$$

for all $\tau \geq 1$.

The optimal average reward in environment e_N is $\lambda_* = 0$, which is attained by applying action 1 in every state. We can consider $\pi' \in \widetilde{\mathcal{P}}$, which chooses action 1 in situational state 1 and action 2 in situational state 2. Apparently $\lambda_{\pi'} < -\kappa/2$. In addition, Van Roy (2006) established that whenever $\kappa < 2\gamma\delta/(1-\gamma)$ and $\epsilon_1 = 1 - \gamma$, there exists N and M such that $\pi^{(\infty)} = \pi'$. Thus,

$$\lambda_* - \lambda_{\pi^{(\infty)}} = \lambda_* - \lambda_{\pi'} > \kappa/2. \tag{152}$$

Since in this environment $\tau_{\pi'} = 1/\epsilon_1 = 1/(1-\gamma)$, it follows that

$$\tau_{\pi^{(\infty)}} \cdot \gamma \cdot \overline{\Delta}_\tau = \tau_{\pi'} \cdot \gamma \cdot \overline{\Delta}_\tau = \frac{\gamma\delta}{1-\gamma}. \tag{153}$$

Therefore, for all $\epsilon > 0$, we can choose

$$\kappa = \frac{2\gamma\delta}{1-\gamma} - 2\epsilon, \tag{154}$$

which allows (151). ■

Finally, we would like to provide a brief discussion over the dependence on the distortion factor Δ_τ. In (Van Roy, 2006), it is shown that if then environment is an MDP, and we consider the fixed-point solution to a version of approximate value iteration that takes into account the invariant distribution of appropriate policies, then the greedy policy with respect to such fixed-point solution always attains the optimal dependence on distortion. More specifically, the projection in approximate value iteration has to incorporate the invariant distribution of the greedy policy with respect to the current values. However, approximate value iteration is a planning algorithm in which the values of all state-action pairs in the MDP are updated simultaneously. When it comes to learning algorithms, where value updates depend on the state-action pair visited in each timestep, temporal-difference learning, which constitutes the core of our agent, implicitly carries out such projection. Readers interested in this connection can refer to Section 8 in (Van Roy, 2006), and the set of lecture slides on real-time value iteration with aggregated states (Russo, 2020).

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. FLAMBE: Structural complexity and representation learning of low rank MDPs. *arXiv preprint arXiv:2006.10814*, 2020.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

Peter L Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. *arXiv preprint arXiv:1205.2661*, 2012.

Dimitri P Bertsekas. *Abstract dynamic programming.* Athena Scientific, 2018.

David Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, pages 719–726, 1962.

Mayank Daswani, Peter Sunehag, and Marcus Hutter. Q-learning for history-based reinforcement learning. In *Asian Conference on Machine Learning*, pages 213–228. PMLR, 2013.

Mayank Daswani, Peter Sunehag, Marcus Hutter, et al. Feature reinforcement learning: state of the art. In *Sequential decision-making with big data: papers from the AAAI-14 workshop.* Association for the Advancement of Artificial Intelligence, 2014.

Daniela Pucci de Farias and Benjamin Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620, 2006.

Geoffrey J Gordon. Stable function approximation in dynamic programming. In *Machine Learning Proceedings 1995*, pages 261–268. Elsevier, 1995.

Marcus Hutter. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Springer Science & Business Media, 2004.

Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

Mehdi Jafarnia-Jahromi, Rahul Jain, and Ashutosh Nayyar. Online learning for unknown partially observable MDPs. *arXiv preprint arXiv:2102.12661*, 2021.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1181–1189. Citeseer, 2015.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? *arXiv preprint arXiv:1807.03765*, 2018.

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

Kyung Y Jo. A Lagrangian algorithm for computing the optimal service rates in Jackson queuing networks. *Computers & operations research*, 16(5):431–440, 1989.

Ali Devran Kara and Serdar Yuksel. Near optimality of finite memory feedback policies in partially observed Markov decision processes. *arXiv preprint arXiv:2010.07452*, 2020.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

Lihong Li. *A unifying framework for computational reinforcement learning theory*. Rutgers The State University of New Jersey-New Brunswick, 2009.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4(5):9, 2006.

Xiuyuan Lu, Benjamin Van Roy, Vikranth Dwaracherla, Morteza Ibrahimi, Ian Osband, and Zheng Wen. Reinforcement learning, bit by bit. *arXiv preprint arXiv:2103.04047*, 2021.

R Andrew McCallum. Instance-based utile distinctions for reinforcement learning with hidden state. In *Machine Learning Proceedings 1995*, pages 387–395. Elsevier, 1995.

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

Ciamac C Moallemi, Sunil Kumar, and Benjamin Van Roy. Approximate and data-driven dynamic programming for queueing networks. unpublished manuscript, 2008.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-optimal representation learning for hierarchical reinforcement learning. *arXiv preprint arXiv:1810.01257*, 2018.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *arXiv preprint arXiv:1306.0940*, 2013.

Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown Markov decision processes: A Thompson sampling approach. *arXiv preprint arXiv:1709.04570*, 2017.

Majid Raeis, Ali Tizghadam, and Alberto Leon-Garcia. Queue-learning: A reinforcement learning approach for providing quality of service. *arXiv preprint arXiv:2101.04627*, 2021.

Daniel Russo. Online variants of value iteration and approximation via state aggregation. `https://djrusso.github.io/Dynamic-Optimization-Course/DP_slides_7.pdf`, 2020. Lecture Slides.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020.

Linn I Sennott. *Stochastic dynamic programming and the control of queueing systems*, volume 504. John Wiley & Sons, 2009.

H. Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin Riedmiller, and Matthew M. Botvinick. V-MPO: On-policy maximum a posteriori policy optimization for discrete and continuous control. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SylOlp4FvH`.

Shaler Stidham Jr and Richard R Weber. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations research*, 37(4):611–625, 1989.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23:12–1, 2022.

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

John N Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine learning*, 16(3):185–202, 1994.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, 1996.

Benjamin Van Roy. Performance loss bounds for approximate value iteration with state aggregation. *Mathematics of Operations Research*, 31(2):234–244, 2006.

Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward Markov decision processes. *arXiv preprint arXiv:2006.16318*, 2020.

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

Christopher J.C.H. Watkins. *Learning from delayed rewards.* PhD thesis, 1989.

Christopher J.C.H. Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Richard R Weber and Shaler Stidham Jr. Optimal control of service rates in networks of queues. *Advances in applied probability*, pages 202–218, 1987.

Chen-Yu Wei, Mehdi Jafarnia Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward Markov decision processes. In *International Conference on Machine Learning*, pages 10170–10180. PMLR, 2020.

Ward Whitt. Approximations of dynamic programs, I. *Mathematics of Operations Research*, 3(3):231–243, 1978.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Zihan Zhang, Xiangyang Ji, and Simon S Du. Is reinforcement learning more difficult than bandits? A near-optimal algorithm escaping the curse of horizon. *arXiv preprint arXiv:2009.13503*, 2020.

Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021.