Convergence of regularized agent-state based Q-learning in POMDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we present a framework to understand the convergence of commonly used Q-learning reinforcement learning algorithms in practice. Two salient features of such algorithms are: (i) the Q-table is recursively updated using an agent state (such as the state of a recurrent neural network) which is not a belief state or an information state and (ii) policy regularization is often used to encourage exploration and stabilize the learning algorithm. We investigate the simplest form of such Q-learning algorithms which we call regularized agent-state based Q-learning (RASQL) and show that it converges under mild technical conditions to the fixed point of an appropriately defined regularized MDP, which depends on the stationary distribution induced by the behavioral policy. We also show that a similar analysis continues to work for a variant of RASQL that learns periodic policies. We present numerical examples to illustrate that the empirical convergence behavior matches with the proposed theoretical limit.

1 Introduction

Reinforcement learning (RL) is a useful paradigm in learn optimal control policies via simulation when the system model is not available or when the system is too large for explicitly solving the dynamic programming equations. The simplest setting is the fully-observed setting of Markov decision processes (MDP), where the controller has access to the environment state. Most existing theoretical RL results on convergence of learning algorithms and their rates of convergence and regret bounds, etc. are established for the MDP setting.

However, in many real-world applications, the controller does not have access to the environment state. Examples include autonomous driving, robotics, healthcare, finance, and others. In such settings, the controller has a partial observation of the environment state, so they need to be modeled as a partially observable Markov decision process (POMDP) rather than a MDP.

When the system model is known, the POMDP model can be converted into an MDP by considering the controller's belief on the state of the environment (also called the belief state) as an information state [2,34] However, such a reduction does not work in the RL setting because the belief state depends on the system model, which is unknown. Nonetheless, there have been several empirical works which show that standard RL algorithms for MDPs continue to work for POMDPs if one uses "frame stacking" (i.e., use the last few observations as a state) or recurrent neural networks [12,15,20,41]. In recent years, considerable progress has been made in understanding the properties of such algorithms but a complete theoretical understanding is still lacking.

A common way to model such RL algorithms for POMDPs is to consider the state of the controller as an agent state [6]. Such agent-state based-controllers have also been considered in the planning

Submitted to the 18th European Workshop on Reinforcement Learning (EWRL 2025). Do not distribute.

setting as they can be simpler to implement than belief-state based controllers. We refer the reader to [33] for a review.

A challenge in understanding the convergence of agent-state based RL algorithms for POMDPs is that an agent state is not an information state. So, it is not possible to write a dynamic programming decomposition based on the agent state. So, one cannot follow the typical proof techniques used to evaluate the convergence of RL algorithms for MDPs (where RL algorithms can be viewed as stochastic approximation variant of MDP algorithms such as value iteration and policy iteration to compute the optimal policy).

There is a good understanding of the convergence of agent-state based Q-learning (ASQL) for POMDPs [16, 17, 32] (which is related to Q-learning for non-Markovian environments [3, 5]). There is also some work on understanding the convergence of actor-critic algorithms for POMDPs [18, 36]. However, most practical RL algorithms for POMDPs use some form of policy regularization, while most theoretical analysis is restricted to the unregularized setting.

Regularization adds an auxiliary loss to the per-step rewards. Typically the auxiliary loss depends on the policy but it may also depend on the value function. Regularization is commonly used in RL algorithms for various reasons, including the use of entropy regularization to encourage exploration [1, 9, 30] and improve generalization [13], using KL-regularization to constrain the policy update to be similar to a prior policy [25, 29], and many others. Unified theory for different facets of regularization in MDPs is provided in [7, 23]

Based on the various benefits of regularization in RL for MDPs, it is also commonly used in RL for POMDPs [10, 11, 15, 24, 29, 30, 36, 40]. However, the recent theoretical analysis of RL for POMDPs discussed above do not consider regularization. The objective of this paper to to present initial results on understanding regularization in RL for POMDPs.

There is some recent work on understanding regularization in POMDPs but they either consider the role of entropy regularization in POMDP solvers (when the model information is known) [4, 35], or consider regularization of the belief distribution [22] or observation distribution [39]. These results do not directly provide an understanding of the role of regularization in RL for POMDPs.

In this paper, we revisit Q-learning for POMDPs when the learning agent is using an agent state and using policy regularization. Our main contribution is to show that in this setting, Q-learning converges under mild technical conditions. We characterize the converged limit in terms of the model parameters and choice of behavioral policy used in Q-learning. Recently, it has been argued that periodic policies may perform better when considering agent-state based POMDPs [32]. We show that our analysis extends to a periodic version of regularized Q-learning as well.

Notations. We use uppercase letters to denote random variables (e.g. S, A, etc.), lowercase letters to denote their realizations (e.g. s, a, etc.) and calligraphic letters to denote sets (e.g. S, A; etc.). Subscripts (e.g. S_t, A_t, etc.) denote variables at time t. Similarly, $S_{1:t}$ denotes the collection of random variables from time 1 to t. $\Delta(\mathsf{S})$ denotes the space of probability measures on a set S; $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ denote the probability of an event and the expectation of a random variable, respectively; and $\mathbb{1}(\cdot)$ denotes the indicator function. $|\mathsf{S}|$ denotes the number of elements in S (when it is a finite set). $\mathbb{R}$ denotes real numbers. $[L]$ denotes the set of integers from 0 to $L-1$, where $L \in \mathbb{Z}^+$. $[\![\ell]\!]$ denotes $(\ell \bmod L)$.

2 Background

2.1 Legendre-Fenchel transform (convex conjugate)

We start with a short review of convex conjugates and Legendre-Fenchel transforms [28], which are an important tool to understand regularization in MDPs [7].

Definition 1 For a strongly convex function $\Omega\colon \mathbb{R}^n \to \mathbb{R}$, its convex conjugate $\Omega^\star\colon \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\Omega^\star(q) = \max_{p \in \mathbb{R}^n} \big\{ \langle p, q \rangle - \Omega(p) \big\}.$$

The mapping $\Omega \mapsto \Omega^*$ is called the Legendre-Fenchel transform.

The following is a useful property of the Legendre-Fenchel transform for regularized MDPs:

Lemma 1 (Based on [14, 21]) *Let Δ be a simplex in $\mathbb{R}^n$ and $\Omega\colon \Delta \to \mathbb{R}$ be twice differentiable and a strongly convex function. Let $\Omega^\star\colon \mathbb{R}^n \to \mathbb{R}$ be the Legendre-Fenchel transform of Ω. Then, $\nabla\Omega^\star$ is Lipschitz and satisfies*

$$\nabla\Omega^\star(q) = \arg\max_{p\in\Delta}\big\{\langle p, q\rangle - \Omega(p)\big\}.$$

In Markov decision processes, one often regularizes the policy. Below we describe some of the commonly used policy regularizers. For the purpose of the discussion below, let A be a finite set (later we will take A to be the set of actions of an MDP, but for now we can consider it as a generic set).

Entropy regularization uses the regularizer $\Omega\colon \Delta(\mathsf{A}) \to \mathbb{R}$ given by

$$\Omega(p) = \frac{1}{\beta}\sum_{a\in\mathsf{A}} p(a)\ln p(a)$$

where $\beta \in \mathbb{R}_{>0}$ is a parameter. It's convex conjugate $\Omega^\star\colon \mathbb{R}^{|\mathsf{A}|} \to \mathbb{R}$ is given by

$$\Omega^\star(q) = \frac{1}{\beta}\ln\Big(\sum_{a\in\mathsf{A}} \exp(\beta q(a))\Big).$$

Moreover, from Lemma 1, we get that the argmax in the definition of convex conjugate is achieved by

$$p^\star(a) = \frac{\exp(\beta q(a))}{\sum_{a'\in\mathsf{A}}\exp(\beta q(a'))}.$$

KL regularization uses the regularizer $\Omega\colon \Delta(\mathsf{A}) \to \mathbb{R}$ given by

$$\Omega(p) = \frac{1}{\beta}\sum_{a\in\mathsf{A}} p(a)\ln\frac{p(a)}{p_{\text{REF}}(a)},$$

where $\beta \in \mathbb{R}_{>0}$ is a parameter and $p_{\text{REF}} \in \Delta(\mathsf{A})$ is a reference distribution. It's convex conjugate $\Omega^\star\colon \mathbb{R}^{|\mathsf{A}|} \to \mathbb{R}$ is given by

$$\Omega^\star(q) = \frac{1}{\beta}\ln\Big(\sum_{a\in\mathsf{A}} p_{\text{REF}}(a)\exp(\beta q(a))\Big).$$

Moreover, from Lemma 1, we get that the argmax in the definition of convex conjugate is achieved by

$$p^\star(a) = \frac{p_{\text{REF}}(a)\exp(\beta q(a))}{\sum_{a'\in\mathsf{A}} p_{\text{REF}}(a')\exp(\beta q(a'))}.$$

2.2 Regularized MDPs

In this section, we provide a brief review of regularized Markov decision processes (MDPs), which are a generalization of standard MDPs with an additional "regularization cost" at each stage.

Consider a Markov decision process (MDP) with state $s_t \in \mathsf{S}$, control action $a_t \in \mathsf{A}$, where all sets are finite. The system operates in discrete time. The initial state $s_1 \sim \rho$ and for any time $t \in \mathbb{N}$, we have $\mathbb{P}(s_{t+1} \mid s_{1:t}, a_{1:t}) = \mathbb{P}(s_{t+1} \mid s_t, a_t) =: P(s_{t+1} \mid s_t, a_t)$, where P is a probability transition matrix. The system yields a reward $R_t = r(s_t, a_t) \in [0, R_{\max}]$. The rewards are discounted by a factor $\gamma \in [0, 1)$.

Consider a policy $\pi \in \mathsf{S} \to \Delta(\mathsf{A})$. Let $\Omega\colon \Delta(\mathsf{A}) \to \mathbb{R}$ be a strongly convex function that is used as a policy regularizer. Then, the *regularized performance* of policy π is given by

$$J_\pi^\Omega := \mathbb{E}^\pi\left[\sum_{t=1}^\infty \gamma^{t-1}\big[r(s_t, a_t) - \Omega(\pi(\cdot \mid s_t))\big] \;\middle|\; s_1 \sim \rho\right],$$

where the notation $\mathbb{E}^\pi$ means that the expectation is taken with the joint measure on the system variables induced by the policy π.

The objective in a regularized MDP is to find a policy π that maximizes the regularized performance J_π^Ω defined above. A key step in understanding the optimal solution of the regularized MDP is to

define the regularized Bellman operator $\mathcal{B}^\Omega$ on the space of real-valued functions on $\mathsf{S} \times \mathsf{A}$ as follows. For any $Q : \mathsf{S} \times \mathsf{A} \to \mathbb{R}$,

$$\mathcal{B}^\Omega Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathsf{S}} P(s' \mid s,a) \Omega^\star(Q(s',\cdot)),$$

where $\Omega^\star$ is the Legendre-Fenchel transform of Ω.

Proposition 1 (Based on [7]) *The following results hold:*

(1) *The operator $\mathcal{B}^\Omega$ is a contraction and therefore has a unique fixed point, which we denote by Q^Ω. By definition,*

$$Q^\Omega(s,a) = r(s,a) + \gamma \sum_{s' \in \mathsf{S}} P(s' \mid s,a) \Omega^\star(Q^\Omega(s',\cdot)).$$

(2) *Define the policy $\pi^{\Omega,*} : \mathsf{S} \to \Delta(\mathsf{A})$ as follows: for any $s \in \mathsf{S}$,*

$$\pi^{\Omega,\star}(\,\cdot \mid s) = \nabla \Omega^\star(Q^\Omega(s,\cdot)), = \operatorname*{arg\,max}_{\xi \in \Delta(\mathsf{A})} \left\{ \sum_{a \in \mathsf{A}} \xi(a) Q^\Omega(s,a) - \Omega(\xi) \right\}$$

where the last equality follows from Lemma 1. Then, the policy $\pi^{\Omega,\star}$ is optimal for maximizing the regularized performance J_π^Ω over the set of all policies.

3 System model and regularized Q-learning for POMDPs

3.1 Model for POMDPs

Consider a partially observable Markov decision process (POMDP) with state $s_t \in \mathsf{S}$, control action $a_t \in \mathsf{A}$, and output $y_t \in \mathsf{Y}$, where all sets are finite. The system operates in discrete time with the dynamics given as follows. The initial state $s_1 \sim \rho$ and for any time $t \in \mathbb{N}$, we have

$$\begin{aligned} \mathbb{P}(s_{t+1}, y_{t+1} \mid s_{1:t}, y_{1:t}, a_{1:t}) &= \mathbb{P}(s_{t+1}, y_{t+1} \mid s_t, a_t) \\ &=: P(s_{t+1}, y_{t+1} \mid s_t, a_t), \end{aligned}$$

where P is a probability transition matrix. In addition, at each time the system yields a reward $R_t = r(s_t, a_t) \in [0, R_{\max}]$. The rewards are discounted by a factor $\gamma \in [0,1)$.

Let $\vec{\boldsymbol{\pi}} = (\vec{\pi}_1, \vec{\pi}_2, \dots)$ denote any (history dependent and possibly randomized) policy, i.e., under policy $\vec{\boldsymbol{\pi}}$ the action at time t is chosen as $a_t \sim \vec{\pi}_t(y_{1:t}, a_{1:t-1})$. The performance of policy $\vec{\boldsymbol{\pi}}$ is given by

$$J_{\vec{\boldsymbol{\pi}}} := \mathbb{E}^{\vec{\boldsymbol{\pi}}} \left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \;\middle|\; s_1 \sim \rho \right],$$

where the notation $\mathbb{E}^{\vec{\boldsymbol{\pi}}}$ means that the expectation is taken with the joint measure on the system variables induced by the policy $\vec{\boldsymbol{\pi}}$.

The objective is to find a (history dependent and possibly randomized) policy $\vec{\boldsymbol{\pi}}$ to maximize $J_{\vec{\boldsymbol{\pi}}}$. When the system model is known, the above POMDP model can be converted to a fully observed Markov decision process (MDP) by considering the controller's posterior belief on the system state as an information state [2, 34]. However, when the system model is not known, it is not possible to run reinforcement learning (RL) algorithms on the belief-state MDP because the belief depends on the system model. For that reason, in RL for POMDPs it is often assumed that the controller is an agent-state based controller.

Definition 2 (Agent state) An agent state is a model-free recursively updateable function of the history of observations and actions. In particular, let Z denote the agent state space. Then, the agent state is a process $\{z_t\}_{t \geq 0}$, $z_t \in \mathsf{Z}$, which starts with some initial value z_0, and is then recursively computed as

$$z_{t+1} = \phi(z_t, y_{t+1}, a_t), \quad t \geq 0 \tag{1}$$

where ϕ is a pre-specified agent-state update function.

Some examples of agent-state based controllers are: (i) a finite memory controller, which chooses the actions based on the previous k observations; (ii) a finite state controller, which effectively filters the possible histories to values from a finite set Z. We refer the reader to [33] for a detailed review of agent-state based policies in POMDPs.

We use $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots)$ to denote an agent-state based policy,[1] i.e., a policy where the action at time t is given by $a_t \sim \pi_t(z_t)$. An agent-state based policy is said to be **stationary** if for all t and t', we have $\pi_t(a \mid z) = \pi_{t'}(a \mid z)$ for all $(z, a) \in \mathsf{Z} \times \mathsf{A}$.

If the agent state is an information state, then MDP-based RL algorithms can directly be applied to find optimal stationary solutions [36]. However, in general, an agent state is not an information state, as is the case in frame-stacking or when using recurrent neural networks. In such settings, the dynamics of the agent state process is non Markovian and the standard dynamic programming based argument does not work. It is possible to find the optimal policy by viewing the POMDP with an agent-state based controller as a decentralized control problem and using the designer's approach [19] to compute an optimal agent-state based policy, as is done in [33], but such an approach is intractable for all but small toy problems.

The Q-learning algorithms for POMDPs maintain a Q-table based on the agent states and actions and update the Q-values based on the samples generated by the environment. Since the agent state is non Markovian, it is not clear if such an iterative scheme converges, and if so, to what value. In the next section, we present a formal model for agent state based Q-learning when the agent also uses policy regularization.

3.2 Regularized agent-state based Q-learning for POMDPs

In this section we describe regularized agent-state based Q-learning (RASQL), which is an online off-policy learning approach in which the agent acts according to a fixed behavioral policy to generate a sample path $(z_1, a_1, r_1, z_2, \dots)$ of agent states, actions, and rewards observed by a learning agent.

The learning agent uses a policy regularizer $\Omega \colon \Delta(\mathsf{A}) \to \mathbb{R}$ and maintains a regularized Q-table, which is arbitrarily initialized and then recursively updated as follows:

$$Q_{t+1}(z, a) = Q_t(z, a) + \alpha(z, a) \left[r_t + \gamma \Omega^\star(Q_t(z_{t+1}, \cdot)) - Q_t(z, a) \right], \tag{2}$$

where the learning rate sequence $\{\alpha_t(z, a)\}_{t \geq 1}$ is chosen such that $\alpha_t(z, a) = 0$ whenever $(z, a) \neq (z_t, a_t)$. For instance, if the policy regularizer is the entropy regularizer, then the above iteration corresponds to an agent-state based version of soft-Q-learning [8]. The "greedy" policy at each time is given by $\pi_t(\cdot \mid z) = \nabla \Omega^\star(Q_t(z, \cdot))$. Thus, for entropy regularization, it would correspond to soft-max based on Q_t.

If the $\Omega^\star(Q_t(z_{t+1}, \cdot))$ term in (2) is replaced by $\max_{a' \in \mathsf{A}} Q_t(z_{t+1}, a')$, the iteration in RASQL corresponds to agent-state based Q-learning (ASQL):

$$Q_{t+1}(z_t, a_t) = Q_t(z_t, a_t) + \alpha_t(z_t, a_t) \left[r_t + \gamma \max_{a' \in \mathsf{A}} Q_t(z_{t+1}, a') - Q_t(z_t, a_t) \right].$$

The convergence of ASQL and its variations have been recently studied in [3, 17, 32]. However, the analysis of ASQL does not include regularization. The main result of this paper is to characterize the convergence of RASQL.

4 Main result

We impose the following standard assumptions on the model.

Assumption 1 For all (z, a), the learning rates $\{\alpha_t(z, a)\}_{t \geq 1}$ are measurable with respect to the sigma-algebra generated by $(z_{1:t}, a_{1:t})$ and satisfy $\alpha_t(z, a) = 0$ if $(z, a) \neq (z_t, a_t)$. Moreover, $\sum_{t \geq 1} \alpha_t(z, a) = \infty$ and $\sum_{t \geq 1} (\alpha_t(z, a))^2 < \infty$, almost surely.

Assumption 2 The behavior policy μ is such that the Markov chain $\{(S_t, Y_t, Z_t, A_t)\}_{t \geq 1}$ converges to a limiting distribution ζ_μ, where $\sum_{(s,y)} \zeta_\mu(s, y, z, a) > 0$ for all (z, a) (i.e., all (z, a) are visited infinitely often).

[1] We use $\vec{\boldsymbol{\pi}}$ to denote history dependent policies and $\boldsymbol{\pi}$ to denote agent-state based policies.

Assumption 1 is the standard assumption for convergence of stochastic approximation algorithms [27]. Assumption 2 ensures persistence of excitation and is a standard assumption in convergence analysis of Q-learning [16, 17, 32, 37, 38].

For ease of notation, we will continue to use ζ_μ to denote the marginal and conditional distributions w.r.t. ζ_μ. In particular, for marginals we use $\zeta_\mu(y, z, a)$ to denote $\sum_{s\in\mathsf{S}} \zeta_\mu(s, y, z, a)$ and so on; for conditionals, we use $\zeta_\mu(s|z, a)$ to denote $\zeta_\mu(s, z, a)/\zeta_\mu(z, a)$ and so on. Note that $\zeta_\mu(s, z, y, a) = \zeta_\mu(s, z)\mu(a|z)P(y|s, a)$. Thus, we have that $\zeta_\mu(s|z, a) = \zeta_\mu(s|z)$.

The key idea to characterize the convergence behavior is the following. Given the limiting distribution ζ_μ, we can define an MDP with state space Z, action space A, and per-step reward $r_\mu \colon \mathsf{Z} \times \mathsf{A} \to \mathbb{R}$ and dynamics $P_\mu \colon \mathsf{Z} \times \mathsf{A} \to \Delta(\mathsf{Z})$ given as follows:

$$\begin{aligned} r_\mu(z, a) &\coloneqq \sum_{s\in\mathsf{S}} r(s, a)\zeta_\mu(s \mid z), \\ P_\mu(z'|z, a) &\coloneqq \sum_{(s,y')\in\mathsf{S}\times\mathsf{Y}} \mathbb{1}_{\{z'=\phi(z,y',a)\}} P(y'|s, a)\zeta_\mu(s|z). \end{aligned} \tag{3}$$

Now consider a regularized version of this MDP, where we regularize the policy using Ω. Let Q_μ denote the fixed point of the regularized Bellman operator corresponding to this reguralized MDP, i.e., Q_μ is the unique fixed point of the following (see the discussion in Sec. 2.2):

$$Q_\mu(z, a) = r_\mu(z, a) + \gamma \sum_{z'\in\mathsf{Z}} P_\mu(z' \mid z, a)\Omega^\star(Q_\mu(z', \cdot)). \tag{4}$$

Then, our main result is the following:

Theorem 1 *Under Assumptions 1 and 2, the RASQL iteration (2) converges to Q_μ almost surely.*

PROOF The proof is provided in Appx. A.

Remark 1 Note that Proposition 1 implies that the "greedy" regularized policy with respect to the limit point of $\{Q_t\}_{t\geq 1}$ is given by $\pi^*(\cdot \mid z) = \nabla\Omega^\star(Q_\mu(z, \cdot))$, which typically lies in the interior of $\Delta(\mathsf{A})$ for each z. Thus, the greedy policy is stochastic. This is a big advantage of RASQL compared to ASQL because in ASQL, the greedy policy corresponding to the limit point of the Q-learning iteration is deterministic. As shown in [31] (also see [32,33]), in general for POMDPs with agent-state based controllers, stochastic stationary policies can outperform deterministic stationary policies.

5 Regularized periodic Q-learning

The idea of periodic Q-learning has been explored in [32]. They show that periodic policies can perform better than stationary policies when the agent state is not an information state. Regularized Q-learning can be generalized by regularized periodic Q-learning, since taking the period $L = 1$ reproduces the stationary setting.

Consider the convergence properties when we consider the following regularized periodic agent-state based Q-learning (RePASQL) update for $\ell \in [L]$.

$$Q^\ell_{t+1}(z, a) = Q^\ell_t(z, a) + \alpha^\ell_t(z, a)\left[r_t + \gamma\Omega^\star(Q_t^{[\![\ell+1]\!]}(z', \cdot)) - Q^\ell_t(z, a)\right]. \tag{5}$$

Assumption 3 For all (ℓ, z, a), the learning rates $\{\alpha^\ell_t(z, a)\}_{t\geq 1}$ are measurable with respect to the sigma-algebra generated by $(z_{1:t}, a_{1:t})$ and satisfy $\alpha^\ell_t(z, a) = 0$ if $(\ell, z, a) \neq ([\![t]\!], z_t, a_t)$. Moreover, $\sum_{t\geq 1} \alpha^\ell_t(z, a) = \infty$ and $\sum_{t\geq 1}(\alpha^\ell_t(z, a))^2 < \infty$, almost surely.

Assumption 4 The behavior/exploration policy $\mu = \{\mu^\ell\}_{\ell\in[L]}$ is such that the Markov chain $\{(S_t, Y_t, Z_t, A_t)\}_{t\geq 1}$ converges to a limiting periodic distribution ζ^ℓ_μ, where $\sum_{(s,y)} \zeta^\ell_\mu(s, y, z, a) > 0$ for all (ℓ, z, a) (i.e., all (ℓ, z, a) are visited infinitely often).

By considering this limiting distribution w.r.t. the original model's rewards and dynamics, we can construct an artificial MDP on the agent state for each $\ell \in [L]$, which has the following rewards and

dynamics:

$$r^\ell_\mu(z,a) \coloneqq \sum_{s\in\mathsf{S}} r(s,a)\zeta^\ell_\mu(s \mid z), \tag{6}$$
$$P^\ell_\mu(z'|z,a) \coloneqq \sum_{(s,y')\in\mathsf{S}\times\mathsf{Y}} \mathbb{1}_{\{z'=\phi(z,y',a)\}} P(y'|s,a)\zeta^\ell_\mu(s|z).$$

Now we can extend the same techniques used in regularized MDPs 2.2 to this by defining a regularized Bellman operator $\mathcal{B}^\ell_\mu$ on an arbitrary Q-function $Q \in \mathbb{R}^{|\mathsf{Z}|\times|\mathsf{A}|}$ as follows:

$$\mathcal{B}^\ell_\mu Q(z,a) = r^\ell_\mu(z,a) + \gamma \sum_{z'\in\mathsf{Z}} P^\ell_\mu(z' \mid z,a)\Omega^\star(Q(z',\cdot)).$$

Next define the composition of the sequence of L Bellman operators corresponding to cycle ℓ as is done in [32].

$$\mathbb{B}^\ell_\mu = \mathcal{B}^\ell_\mu \mathcal{B}^{[\![\ell+1]\!]}_\mu \cdots \mathcal{B}^{[\![\ell+L-1]\!]}_\mu.$$

Then we can apply Prop. 1 to $\mathbb{B}^\ell_\mu$. In addition, considering the periodicity of the operators, the same approach followed in [32] can be used to show that $\mathbb{B}^\ell_\mu$ is a contraction and therefore has a unique fixed point denoted by Q^ℓ_μ which is given by

$$Q^\ell_\mu(z,a) = r^\ell_\mu(z,a) + \gamma \sum_{z'\in\mathsf{Z}} P^\ell_\mu(z' \mid z,a) V^{[\![\ell+1]\!]}_\mu(z').$$

Theorem 2 *Under Assumptions 3 and 4, the RePASQL iteration (5) converges to $\{Q^\ell_\mu\}_{\ell\in[L]}$ almost surely.*

PROOF We omit the proof due to space limitations. The proof follows a similar style used in [32].

6 Numerical example

In this section, we present an example to highlight the salient features of our results. First, we describe the POMDP model.

6.1 POMDP model

Consider a POMDP with $\mathsf{S} = \{0,1,2,3\}, \mathsf{A} = \{0,1\}, \mathsf{Y} = \{0,1\}$ and $\gamma = 0.9$. The start state distribution is given by

$$\rho(s) = [0.3, 0.0, 0.2, 0.5]$$

Now consider the reward and transitions when $a = 0$:

$$r(s,0) = (1-\gamma) \times [0.6, 0.0, 0.5, -0.3]$$
$$P(s' \mid s, 0) = \begin{bmatrix} 0.0 & 0.6 & 0.4 & 0.0 \\ 0.8 & 0.0 & 0.2 & 0.0 \\ 0.7 & 0.3 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.0 & 0.8 \end{bmatrix}.$$

Note that s, s' (state, next state) corresponds to the rows, columns of P, respectively. Next, when $a = 1$

$$r(s,1) = (1-\gamma) \times [0.1, -0.3, -0.2, 0.5]$$
$$P(s' \mid s, 1) = \begin{bmatrix} 0.8 & 0.2 & 0.0 & 0.0 \\ 0.4 & 0.0 & 0.6 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.1 & 0.7 & 0.2 & 0.0 \end{bmatrix}.$$

Finally, we have the observations function which maps $s = \{0,3\}$ to $y = 0$ and $s = \{1,2\}$ to $y = 1$.

6.2 Regularized agent-state based Q-learning (RASQL) experiment

For the purpose of providing a simple illustration in this example, we fix the agent state to be the observation of the agent, i.e., $z_t = y_t$. However, in general the theoretical results hold for the general agent-state update rule given in (1). Consider the following fixed exploration policy:

$$\mu(a \mid z) = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}.$$

Note that z, a (observation, action) corresponds to the rows, columns of μ, respectively.

Using μ, we run 25 random seeds on the given POMDP and we perform the RASQL update (2) with a regularization coefficient $(\beta) = 1.0$ for 10^5 timesteps/iterations. We plot the median and quartiles from 25 seeds of the iterates $\{Q_t(z,a)\}_{t\geq 1}$ for each (z,a) pair as well as their corresponding theoretical limits $Q_\mu(z,a)$ (computed using Theorem 1) are shown in Fig. 1. The salient features of these results are as follows:

- RASQL converges to the theoretical limit predicted by Theorem 1.
- The limit Q_μ depends on the exploration policy μ.

Thus, it can be seen from this example that we can precisely characterize the limits of convergence when using regularized Q-learning with an agent-state based representation.

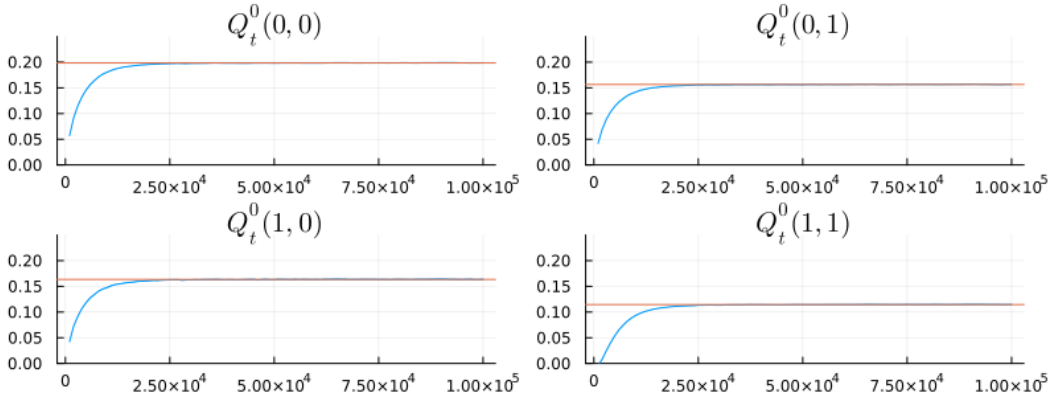

Figure 1: RASQL convergence: Q-values vs. number of iterations. Blue: RASQL iterates, Red: Theoretical limit from Theorem 1.

6.3 Regularized periodic agent-state based Q-learning (RePASQL) experiment

Similar to the RASQL experiment, we fix the agent state to be the observation of the agent, i.e., $z_t = y_t$. Consider the following fixed periodic exploration policy for period $L = 2$:

$$\mu^0(a \mid z) = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}, \quad \mu^1(a \mid z) = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}.$$

Using μ^ℓ, we run 25 random seeds on the given POMDP and we perform the RePASQL update (5) with a regularization coefficient $(\beta) = 1.0$ for 10^5 timesteps/iterations. We plot the median and quartiles from 25 seeds of the iterates $\{Q_t^\ell(z,a)\}_{t\geq 1}$ for each (ℓ, z, a) pair as well as their corresponding theoretical limits $Q_\mu^\ell(z,a)$ (computed using Theorem 2) are shown in Fig. 2. The salient features of these results are as follows:

- RePASQL converges to the theoretical limit predicted by Theorem 2.
- The limits $\{Q_\mu^\ell\}_{\ell\in[L]}$ depend on the periodic exploration policy $\{\mu^\ell\}_{\ell\in[L]}$.

Thus, it can be seen from this example that we can precisely characterize the limits of convergence.

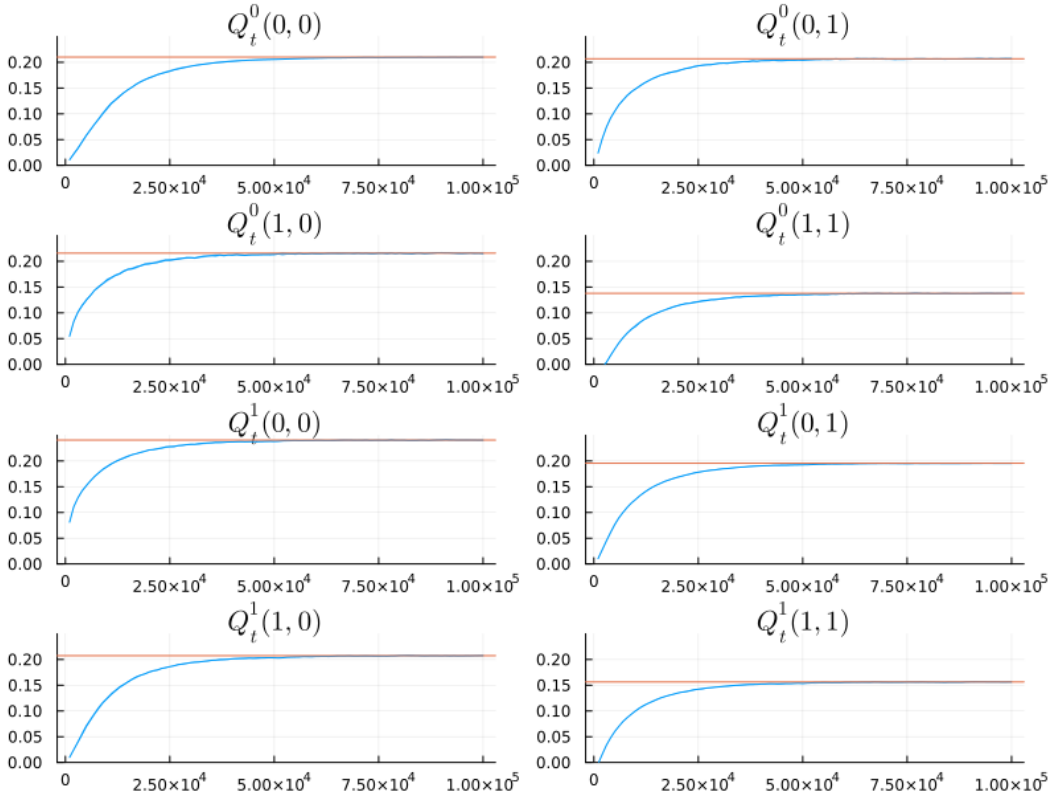

Figure 2: RePASQL convergence: Q-values vs. number of iterations. Blue: RePASQL iterates, Red: Theoretical limit from Theorem 2.

7 Conclusions

In this work, we present theoretical results on the convergence of regularized agent-state based Q-learning (RASQL) under some standard assumptions from the literature. In particular, we show that: 1) RASQL converges and 2) we characterize the solution that RASQL converges to as a function of the model parameters and the choice of exploration policy. We illustrate these ideas on a small POMDP example and show that the Q-learning iterates of RASQL matches with the calculated theoretical limit. We also generalize these ideas to the periodic setting and demonstrate the theoretical and empirical convergence of RePASQL. Thus, in doing so we are able to understand how regularization works when combined with Q-learning for POMDPs that have an agent state that is not an information state.

A noteworthy issue with RASQL/RePASQL is that it inherits the limitations of its predecessor approaches of ASQL and PASQL. In particular, while we are able to prove convergence and characterize the converged solution in RASQL/RePASQL, we cannot guarantee the convergence to the optimal agent-state based solution and this largely depends on the choice of exploration policy and the POMDP dynamics. Even so, seeing how regularization is an important component in several empirical works concerning POMDPs with agent states that are not an information state, we find it useful to establish some useful theoretical properties on the convergence of such algorithms.

References

[1] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[2] Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

[3] Siddharth Chandak, Pratik Shah, Vivek S Borkar, and Parth Dodhia. Reinforcement learning in non-markovian environments. *Systems & Control Letters*, 185:105751, 2024.

[4] Harrison Delecki, Marcell Vazquez-Chanlatte, Esen Yel, Kyle Wray, Tomer Arnon, Stefan Witwicki, and Mykel J Kochenderfer. Entropy-regularized point-based value iteration. *arXiv preprint arXiv:2402.09388*, 2024.

[5] Ali Devran Kera and Serdar Yüksel. Q-learning for stochastic control under general information structures and non-markovian environments. *Transactions on Machine Learning Research*, 2024.

[6] Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Simple agent, complex environment: Efficient reinforcement learning with agent states. *Journal of Machine Learning Research*, 23(255):1–54, 2022.

[7] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pages 2160–2169. PMLR, 2019.

[8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870, 2018.

[10] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations*, 2020.

[11] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[12] Matthew J Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, page 141, 2015.

[13] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

[14] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

[15] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International conference on machine learning*, pages 2117–2126. PMLR, 2018.

[16] Tommi Jaakkola, Satinder Singh, and Michael Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems*, volume 7, pages 345–352. MIT Press, 1994.

[17] Ali Devran Kara and Serdar Yüksel. Convergence of finite memory Q learning for POMDPs and near optimality of learned policies under filter stability. *Mathematics of Operations Research*, November 2022.

[18] Vijay R Konda and John N Tsitsiklis. Onactor-critic algorithms. *SIAM journal on Control and Optimization*, 42(4):1143–1166, 2003.

[19] Aditya Mahajan. *Sequential decomposition of sequential dynamic teams: applications to real-time communication and networked control systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 2008.

[20] Lingheng Meng, Rob Gorbet, and Dana Kulić. Memory-based deep reinforcement learning for pomdps. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5619–5626. IEEE, 2021.

[21] Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pages 3462–3471. PMLR, 2018.

[22] Timothy L Molloy and Girish N Nair. Smoother entropy for active state trajectory estimation and obfuscation in pomdps. *IEEE Transactions on Automatic Control*, 68(6):3557–3572, 2023.

[23] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *Neural Information Processing Systems*, 2017.

[24] Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive RL. *International Conference on Learning Representations*, 2024.

[25] Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, pages 1607–1612, 2010.

[26] LA Prashanth and Shalabh Bhatnagar. Gradient-based algorithms for zeroth-order optimization, 2024.

[27] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[28] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[29] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[31] Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Learning without state-estimation in partially observable markovian decision processes. In *Machine Learning Proceedings 1994*, pages 284–292. Elsevier, 1994.

[32] Amit Sinha, Matthieu Geist, and Aditya Mahajan. Periodic agent-state based Q-learning for pomdps. *Neural Information Processing Systems (Neurips)*, 2024.

[33] Amit Sinha and Aditya Mahajan. Agent-state based policies in pomdps: Beyond belief-state mdps. *Conference on Decision and Control*, 2024.

[34] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[35] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *Advances in neural information processing systems*, 26, 2013.

[36] 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):1–83, 2022.

[37] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16:185–202, 1994.

[38] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[39] Riccardo Zamboni, Duilio Cirino, Marcello Restelli, and Mirco Mutti. The limits of pure exploration in pomdps: When the observation entropy is enough. *Reinforcement Learning Journal*, 2024.

[40] Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, pages 7444–7453. PMLR, 2019.

[41] Pengfei Zhu, Xin Li, Pascal Poupart, and Guanghui Miao. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*, 2017.

A Proof of Theorem 1

The proof argument for Theorem 1 is similar to the proof argument given in [5, 16, 17, 32].

Define an error function between the converged value and the Q-learning iteration $\Delta_{t+1} \coloneqq Q_{t+1} - Q_\mu$. Then, combine (2), (4) and (6) as follows for all (z, a).

$$\begin{aligned}\Delta_{t+1}(z,a) &= Q_{t+1}(z,a) - Q_\mu(z,a) \\ &= (1-\alpha_t(z,a))\Delta_t(z,a) + \alpha_t(z,a)\left[U_t^0(z,a) + U_t^1(z,a) + U_t^2(z,a)\right], \end{aligned} \tag{7}$$

where

$$\begin{aligned} U_t^0(z,a) &\coloneqq [r(S_t,A_t) - r_\mu(z,a)]\,\mathbb{1}_{\{Z_t=z,A_t=a\}}, \\ U_t^1(z,a) &\coloneqq \left[\gamma\Omega^\star(Q_\mu(Z_{t+1},\cdot)) - \gamma\sum_{z'\in\mathsf{Z}} P_\mu(z' \mid z,a)\Omega^\star(Q_\mu(z',\cdot))\right]\mathbb{1}_{\{Z_t=z,A_t=a\}}, \\ U_t^2(z,a) &\coloneqq [\gamma\Omega^\star(Q_t(Z_{t+1},\cdot)) - \gamma\Omega^\star(Q_\mu(Z_{t+1},\cdot))]\,\mathbb{1}_{\{Z_t=z,A_t=a\}}. \end{aligned}$$

Note that we are adding the term $\gamma\Omega^\star(Q_\mu(Z_{t+1},\cdot))\mathbb{1}_{\{Z_t=z,A_t=a\}}$ in $U_t^1(z,a)$ and subtracting it in $U_t^2(z,a)$. We can now view (7) as a linear system with state Δ_t and three inputs $U_t^0(z,a), U_t^1(z,a)$ and $U_t^2(z,a)$. Using the linearity, we can now split the state into three components $\Delta_{t+1} = X_{t+1}^0 + X_{t+1}^1 + X_{t+1}^2$, where the components evolve as follows for $i \in \{0,1,2\}$:

$$X_{t+1}^i(z,a) = (1-\alpha_t(z,a))X_t^i(z,a) + \alpha_t(z,a)U_t^i(z,a).$$

We will now separately show each $\|X_t^i\| \to 0$.

A.1 Convergence of component X_t^0

The proof for the convergence of component X_t^0 is similar to that given in [32]. Details are omitted due to space limitations.

A.2 Convergence of component X_t^1

The proof for the convergence of component X_t^1 is based on the argument given in [32]. Let W_t denote the tuple $(S_t, Z_t, A_t, S_{t+1}, Z_{t+1}, A_{t+1})$. Note that $\{W_t\}_{t\geq 1}$ is a Markov chain and converges to a limiting distribution $\bar{\zeta}_\mu$, where

$$\bar{\zeta}_\mu(s,z,a,s',z',a') = \zeta_\mu(s,z,a)\sum_{y'\in\mathsf{Y}} P(s',y'|s,a)\mathbb{1}_{\{z'=\phi(z,y',a)\}}\mu(a'|z').$$

We use $\bar{\zeta}_\mu(s,z,a,\mathcal{S},\mathcal{Z},\mathcal{A})$ to denote the marginalization over the "future states" and a similar notation for other marginalizations. Note that $\bar{\zeta}_\mu(s,z,a,\mathcal{S},\mathcal{Z},\mathcal{A}) = \zeta_\mu(s,z,a)$.

Define V_t as the value function associated with Q_t, i.e., $V_t(z) \coloneqq \Omega^\star(Q_t(z,\cdot))$. Fix $(z_\circ, a_\circ) \in \times\mathsf{Z}\times\mathsf{A}$ and define

$$h_P(W_t; z_\circ, a_\circ) = \left[\gamma V_\mu(Z_{t+1}) - \gamma\sum_{\bar{z}\in\mathsf{Z}} P_\mu(\bar{z}|z_\circ,a_\circ)V_\mu(\bar{z})\right]\mathbb{1}_{\{Z_t=z_\circ,A_t=a_\circ\}}.$$

Then the process $\{X_t^1(z,a)\}_{t\geq 1}$ is given by the stochastic iteration

$$X_{t+1}^1(z_\circ,a_\circ) = (1-\alpha_t(z_\circ,a_\circ))X_t^1(z_\circ,a_\circ) + \alpha_t(z_\circ,a_\circ)h_P(W_t;z_\circ,a_\circ).$$

As argued earlier, the process $\{W_t\}_{t\geq 1}$ is a Markov chain. Due to Assm. 1, the learning rate $\alpha_t(z_\circ,a_\circ)$ is measurable with respect to the sigma-algebra generated by $(Z_{1:t}, A_{1:t})$ and is therefore also measurable with respect to the sigma-algebra generated by $W_{1:t}$. Thus, the learning rates $\{\alpha_t(z_\circ,a_\circ)\}_{t\geq 1}$ satisfy the conditions of Theorem 2.7 from [26]. Therefore, the theorem implies

that $\{X_t^1(z_\circ, a_\circ)\}_{t\geq 1}$ converges a.s. to the following limit

$$\begin{aligned}
&\lim_{t\to\infty} X_t^1(z_\circ, a_\circ) \\
&= \sum_{\substack{s,z,a\in \mathsf{S}\times\mathsf{Z}\times\mathsf{A} \\ s',z',a'\in \mathsf{S}\times\mathsf{Z}\times\mathsf{A}}} \bar{\zeta}_\mu(s,z,a,s',z',a') h_P(s,z,a,s',z',a';z_\circ,a_\circ) \\
&= \gamma\left[\sum_{z'\in\mathsf{Z}} \bar{\zeta}_\mu(\mathcal{S},z_\circ,a_\circ,\mathcal{S},z',\mathcal{A}) V_\mu(z')\right] - \left[\gamma\bar{\zeta}_\mu(\mathcal{S},z_\circ,a_\circ,\mathcal{S},\mathcal{Z},\mathcal{A}) \sum_{\bar{z}\in\mathsf{Z}} P_\mu(\bar{z}|z_\circ,a_\circ) V_\mu(\bar{z})\right] \\
&= 0
\end{aligned}$$

where the last step follows from the fact that $\bar{\zeta}_\mu(\mathsf{S},z_\circ,a_\circ,\mathsf{S},\mathcal{Z},\mathcal{A}) = \zeta_\mu(z_\circ,a_\circ)$ and $\bar{\zeta}_\mu(\mathsf{S},z_\circ,a_\circ,\mathsf{S},z',\mathsf{A}) = \zeta_\mu(z_\circ,a_\circ)P_\mu(z'|z_\circ,a_\circ)$.

A.3 Convergence of component X_t^2

The convergence of the X_t^2 component is based on [17,32] but requires some additional considerations due to the regularization term. We start by defining:

$$\begin{aligned}
\pi_t(\cdot \mid z) &= \arg\max_{\xi\in\Delta(\mathsf{A})} \sum_{a\in\mathsf{A}} \xi(a) Q_t(z,a) - \Omega(\xi) \\
\pi^\star(\cdot \mid z) &= \arg\max_{\xi\in\Delta(\mathsf{A})} \sum_{a\in\mathsf{A}} \xi(a) Q_\mu(z,a) - \Omega(\xi).
\end{aligned}$$

In the previous steps, we have shown that $\|X_t^i\| \to 0$ a.s., for $i \in \{0,1\}$. Thus, we have that $\|X_t^0 + X_t^1\| \to 0$ a.s. Arbitrarily fix an $\epsilon > 0$. Therefore, there exists a set Ω^1 of measure one and a constant $T(\omega,\epsilon)$ such that for $\omega \in \Omega^1$, all $t > T(\omega,\epsilon)$, and $(z,a) \in \times\mathsf{Z}\times\mathsf{A}$, we have

$$X_t^0(z,a) + X_t^1(z,a) < \epsilon. \tag{8}$$

Now pick a constant C such that

$$\kappa \coloneqq \gamma\left(1 + \frac{1}{C}\right) < 1 \tag{9}$$

Suppose for some $t > T(\omega,\epsilon)$, $\|X_t^2\| > C\epsilon$. Then, for $(z,a) \in \mathsf{Z}\times\mathsf{A}$,

$$\begin{aligned}
&U_t^2(z,a) \\
&= \gamma V_t(Z_{t+1}) - \gamma V_\mu(Z_{t+1}) \\
&= \gamma\Omega^\star(Q_t(Z_{t+1},\cdot)) - \gamma\Omega^\star(Q_\mu(Z_{t+1},\cdot)) \\
&= \gamma\left[\sum_{a\in\mathsf{A}} \pi_t(a \mid Z_{t+1}) Q_t(Z_{t+1},a) - \Omega(\pi_t(\cdot \mid Z_{t+1})) - \sum_{a\in\mathsf{A}} \pi^\star(a \mid Z_{t+1}) Q_\mu(Z_{t+1},a) + \Omega(\pi^\star(\cdot \mid Z_{t+1}))\right] \\
&\overset{(a)}{\leq} \gamma\left[\sum_{a\in\mathsf{A}} \pi_t(a \mid Z_{t+1}) Q_t(Z_{t+1},a) - \Omega(\pi_t(\cdot \mid Z_{t+1})) - \sum_{a\in\mathsf{A}} \pi_t(a \mid Z_{t+1}) Q_\mu(Z_{t+1},a) + \Omega(\pi_t(\cdot \mid Z_{t+1}))\right] \\
&\leq \gamma \sum_{a\in\mathsf{A}} \pi_t(a \mid Z_{t+1}) \left|Q_t(Z_{t+1},a) - Q_\mu(Z_{t+1},a)\right| \\
&\overset{(b)}{\leq} \gamma\|Q_t - Q_\mu\| = \gamma\|\Delta_t\| \\
&\leq \gamma\|X_t^0 + X_t^1\| + \gamma\|X_t^2\| && \text{(10a)} \\
&\overset{(c)}{\leq} \gamma\epsilon + \gamma\|X_t^2\| && \text{(10b)} \\
&\overset{(d)}{\leq} \gamma\left(1 + \frac{1}{C}\right)\|X_t^2\| \overset{(e)}{=} \kappa\|X_t^2\| \overset{(e)}{<} \|X_t^2\|, && \text{(10c)}
\end{aligned}$$

where (a) follows from the fact that we replace the argmax $\pi^\star$ with a different argument π_t in the second term, (b) follows from maximizing over all realizations of Z_{t+1} and $a \in \mathsf{A}$, (c) follows from (8), (d) follows from $\|X_t^2\| > C\epsilon$, (e) follows from (9). Thus, for any $t > T(\omega, \epsilon)$ and $\|X_t^2\| > C\epsilon$:

$$\begin{aligned} & X_{t+1}^2(z,a) = (1-\alpha_t(z,a))X_t^2(z,a) + \alpha_t(z,a)U_t^2(z,a) < \|X_t^2\| \\ \implies & \|X_{t+1}^2\| < \|X_t^2\|. \end{aligned}$$

Hence, when $\|X_t^2\| > C\epsilon$, it decreases monotonically with time. Hence, there are two possibilities: either

1. $\|X_t^2\|$ always remains above $C\epsilon$; or
2. it goes below $C\epsilon$ at some stage.

We consider these two possibilities separately.

A.3.1 Possibility (i): $\|X_t^2\|$ always remains above $C\epsilon$

We will show that $\|X_t^2\|$ cannot remain above $C\epsilon$ forever.

We will now prove that $\|X_t^2\|$ cannot remain above $C\epsilon$ forever. The proof is by contradiction. Suppose $\|X_t^2\|$ remains above $C\epsilon$ forever. As argued earlier, this implies that $\|X_t^2\|$, $t \geq T(\omega, \epsilon)$, is a strictly decreasing sequence, so it must be bounded from above. Let $B^{(0)}$ be such that $\|X_t^2\| \leq B^{(0)}$ for all $t \geq T(\omega, \epsilon)$. Eq. (10c) implies that $\|U_t^2\| < \kappa B^{(0)}$. Then, we have for all $(z, a) \in \mathsf{Z} \times \mathsf{A}$ that

$$\begin{aligned} X_{t+1}^2(z,a) &\leq (1-\alpha_t(z,a))\|X_t^2\| + \alpha_t(z,a)\|U_t^2\| \\ &< (1-\alpha_t(z,a))\|X_t^2\| + \alpha_t(z,a)\kappa\|X_t^2\| \end{aligned}$$

which implies that $\|X_t^2\| \leq \|M_t^{(0)}\|$, where $\{M_t^{(0)}\}_{t \geq T(\omega,\epsilon)}$ is a sequence given by

$$M_{t+1}^{(0)}(z,a) \leq (1-\alpha_t(z,a))M_t^{(0)}(z,a) + \alpha_t(z,a)\kappa B^{(0)}.$$

Theorem 2.4 from [26] implies that $M_t^{(0)}(z,a) \to \kappa B^{(0)}$ and hence $\|M_t^{(0)}\| \to \kappa B^{(0)}$. Now pick an arbitrary $\bar{\epsilon} \in (0, (1-\kappa)C\epsilon)$. Thus, there exists a time $T^{(1)} = T^{(1)}(\omega, \epsilon, \bar{\epsilon})$ such that for all $t > T^{(1)}$, $\|M_t^{(0)}\| \leq B^{(1)} := \kappa B^{(0)} + \bar{\epsilon}$. Since $\|X_t^2\|$ is bounded by $\|M_t^{(0)}\|$, this implies that for all $t > T^{(1)}$, $\|X_t^2\| \leq B^{(1)}$ and, by (10c), $\|U_t^2\| \leq \kappa B^{(1)}$. By repeating the above argument, there exists a time $T^{(2)}$ such that for all $t \geq T^{(2)}$,

$$\|X_t^2\| \leq B^{(2)} := \kappa B^{(1)} + \bar{\epsilon} = \kappa^2 B^{(0)} + \kappa\bar{\epsilon} + \bar{\epsilon},$$

and so on. By (9), $\kappa < 1$ and $\bar{\epsilon}$ is chosen to be less than $C\epsilon$. So eventually, $B^{(m)} := \kappa^m B^{(0)} + \kappa^{m-1}\bar{\epsilon} + \cdots + \bar{\epsilon}$ must get below $C\epsilon$ for some m, contradicting the assumption that $\|X_t^2\|$ remains above $C\epsilon$ forever.

A.3.2 Possibility (ii): $\|X_t^2\|$ goes below $C\epsilon$ at some stage

Suppose that there is some $t > T(\omega, \epsilon)$ such that $\|X_t^2\| < C\epsilon$. Then (10a), (10b) and (9) imply that

$$\|U_t^2\| \leq \gamma\|X_t^0 + X_t^1\| + \gamma\|X_t^2\| \leq \gamma\epsilon + \gamma C\epsilon < C\epsilon.$$

Therefore,

$$X_{t+1}^2(z,a) \leq (1-\alpha_t(z,a))\|X_t^2\| + \alpha_t(z,a)\|U_t^2\| < C\epsilon$$

where the last inequality uses the fact that both $\|U_t^2\|$ and $\|X_{t+1}^2\|$ are both below $C\epsilon$. Thus, we have that

$$X_{t+1}^2(z,a) < C\epsilon.$$

Hence, once $\|X_{t+1}^2\|$ goes below $C\epsilon$, it stays there.

A.3.3 Implication

We have show that for sufficiently large $t > T(\omega, \epsilon)$, $X_t^2(z,a) < C\epsilon$. Since ϵ is arbitrary, this means that for all realizations $\omega \in \Omega^1$, $\|X_t^2\| \to 0$. Thus,

$$\lim_{t\to\infty} \|X_t^2\| = 0, \quad a.s. \tag{11}$$

A.4 Putting everything together

Recall that we defined $\Delta_t = Q_t - Q_\mu$ and in Step 1, we split $\Delta_t = X_t^0 + X_t^1 + X_t^2$. Steps 2 and 3 together show that $\|X_t^0 + X_t^1\| \to 0$, a.s. and Step 3 (11) shows us that $\|X_t^2\| \to 0$, a.s. Thus, by the triangle inequality,

$$\lim_{t\to\infty} \|\Delta_t\| \leq \lim_{t\to\infty} \|X_t^0 + X_t^1\| + \lim_{t\to\infty} \|X_t^2\| = 0,$$

which establishes that $Q_t \to Q_\mu$, a.s.