Multi-Bellman operator for convergence of Q-learning with linear function approximation

Anonymous authors
Paper under double-blind review

Abstract

We investigate the convergence of Q-learning with linear function approximation and introduce the multi-Bellman operator, an extension of the traditional Bellman operator. By analyzing the properties of this operator, we identify conditions under which the projected multi-Bellman operator becomes a contraction, yielding stronger fixed-point guarantees compared to the original Bellman operator. Building on these insights, we propose the multi-Q-learning algorithm, which achieves convergence and approximates the optimal solution with arbitrary precision. This contrasts with traditional Q-learning, which lacks such convergence guarantees. Finally, we empirically validate our theoretical results.

1 Introduction

Reinforcement learning aims to approximate the value of performing different actions in different states, considering the expected sum of time-discounted rewards in a Markovian environment. The importance of this task cannot be overstated, as an accurate value function allows an agent to make optimal decisions by selecting the action with the highest value in a given state (Puterman, 2005). The value function also facilitates environment evaluation and enables comparisons between different environments. If we can store the value of performing each action on each state individually, the Q-learning algorithm converges to the correct value function (Watkins & Dayan, 1992). When it is not possible to store values, for example when there are too many states, Q-learning can be combined with function approximation (Melo & Ribeiro, 2007).

Unfortunately, the combination of Q-learning and function approximation is troublesome, and even when the function approximation space is linear, the algorithm can diverge (Sutton & Barto, 2018). In fact, the approximation problem that Q-learning addresses does not have, in general, a solution (Melo et al., 2008). To address this limitation, we propose an alternative problem and algorithm that effectively solve linear function approximation in reinforcement learning, unlike Q-learning. The problem and algorithm proposed are extensions of the original. To our ends, we start by introducing the multi-Bellman operator, and identify the conditions under which its projection into the function approximation space is contractive. Leveraging the contractivity property of the Multi-Bellman operator, we propose the multi Q-learning algorithm, and demonstrate its convergence to arbitrarily good solutions, both theoretically and empirically.

2 Background

A Markov decision problem (MDP) is a tuple $(\mathcal{X}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{X}$ is a discrete set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}$ is a set of distributions over $\mathcal{X}$ for each state and action, $r : \mathcal{X} \times \mathcal{A} \to [0, 1]$ is a reward function, and γ is a real in $[0, 1)$ called the discount factor (Puterman, 2005). A policy is a function $\pi : \mathcal{X} \to \Delta(A)$, mapping states to distributions over $\mathcal{A}$, and its value is the function $q_\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ such that

$$q_\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid x_0 = x, a_0 = a\right], \tag{1}$$

where the expectation is with respect to trajectories of the MDP, namely rewards r_t and states x_{t+1} that are obtained by performing action a_t on state x_t, with actions a_t that are selected according to π at every t. There is at least one policy π^* that maximizes the value q on every state and action (Puterman, 2005) and we refer to its value as q^*. The value q^* satisfies the Bellman fixed-point equation given by

$$q^* = \mathrm{H}q^*, \tag{2}$$

where H is the Bellman operator, such that

$$(\mathrm{H}q)(x,a) = \mathbb{E}\left[r(x,a) + \gamma \max_{a' \in \mathcal{A}} q\left(x', a'\right)\right], \tag{3}$$

where the expectation is with respect to the next state x' obtained by performing action a on state x.

2.1 Function Approximation

Our goal is to approximate q^*. In other words, we want to find a good parameterized representation of the value of an optimal policy for a given MDP, given a function approximation space $\mathcal{H} = \{h_\omega : \mathcal{Z} \to \mathbb{R}, \omega \in \mathbb{R}^k\}$. Let us consider that every $h_\omega \in \mathcal{H}$ is differentiable in ω, a function $h : \mathcal{Z} \to \mathbb{R}$, a distribution μ over a random z taking values in $\mathcal{Z}$, and the loss $l(\omega) = \frac{1}{2}\left\|h - h_\omega\right\|^2$, where we consider the μ-norm such that $\left\|h\right\|^2 = \mathbb{E}_\mu\left[h^2(z)\right]$. To approximate h is to find an element in the set $\operatorname{Proj} h \subset \mathcal{H}$ such that

$$\operatorname{Proj} h = \arg\min_{h_\omega \in \mathcal{H}} l(\omega). \tag{4}$$

Any ω parameterizing a h_ω in $\operatorname{Proj} h$ must be a critical point and thus verify

$$\nabla_\omega l(\omega) = 0. \tag{5}$$

Linear function approximation Given features $\phi : \mathcal{Z} \to \mathbb{R}^k$, a linear function approximation space is given by the functions such that $h_\omega(z) = \phi^T(z)\omega$. In this case, the gradient of the loss function is given by

$$\nabla_\omega l(\omega) = -\mathbb{E}_\mu\left[\phi(z)\left(h(z) - h_\omega(z)\right)\right]. \tag{6}$$

Solving for $\nabla_\omega l(\omega) = 0$, we obtain that

$$\omega = \mathbb{E}_\mu\left[\phi(z)\phi^T(z)\right]^{-1}\mathbb{E}_\mu\left[\phi(z)h(z)\right]. \tag{7}$$

Thus, if the inverted matrix above exists, the set $\operatorname{Proj} h$ has a single element and we can refer to both as h_{ω^*}. The solution ω^* is also the equilibrium of the dynamical system

$$\dot{\omega} = \nabla_\omega l(\omega), \tag{8}$$

and the limit of the sequence of ω_t obtained by performing a discretized update with z_t i.i.d. from μ

$$\omega_{t+1} = \omega_t + \alpha_t\left[\phi\left(z_t\right)\left(h(z_t) - h_{\omega_t}(z_t)\right)\right], \tag{9}$$

with α_t a small positive real called the learning rate.

Stochastic approximation Without access to h, stochastic approximation performs the update

$$\omega_{t+1} = \omega_t + \alpha_t\left[\phi\left(z_t\right)\left(\tau_t - h_{\omega_t}(z_t)\right)\right], \tag{10}$$

with τ_t a possibly ω_t-dependent estimate of $h\left(z_t\right)$ called the target. If there exists an equilibrium ω^* for the dynamical system

$$\dot{\omega} = \mathbb{E}_\mu\left[\phi\left(z_t\right)\left(\tau_t - h_\omega(z_t)\right)\right], \tag{11}$$

and it is globally asymptotically stable, well-established conditions guarantee the stochastic approximation update converges to ω^* (Borkar, 2008).

2.2 Q-learning with Linear Function Approximation

We focus now on the reinforcement learning setting and consider $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$, and an approximation space of linearly parameterized functions $q_\omega : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ such that

$$q_\omega(x, a) = \phi^T(x, a)\omega. \tag{12}$$

In reinforcement learning, we want to approximate the value q^*, by computing ω^* that satisfies

$$q_{\omega^*} = \operatorname{Proj} q^*. \tag{13}$$

Without knowledge of q^*, we are unable to perform the exact stochastic approximation update,

$$\omega_{t+1} = \omega_t + \alpha_t \left[\phi\left(x_t, a_t\right)\left(q^*(x_t, a_t) - q_{\omega_t}(x_t, a_t)\right)\right]. \tag{14}$$

Leveraging the fact that $q^* = \mathrm{H}q^*$, Q-learning performs instead the stochastic approximation update

$$\omega_{t+1} = \omega_t + \alpha_t \left[\phi\left(x_t, a_t\right)\left(\tau_t - q_{\omega_t}(x_t, a_t)\right)\right], \tag{15}$$

where the target τ_t is a sample for $\left(\mathrm{H}q_{\omega_t}\right)(x_t, a_t)$, i.e.

$$\tau_t = r_t + \gamma \max_{a' \in \mathcal{A}} q_{\omega_t}\left(x_t', a'\right). \tag{16}$$

Since the target is not differentiated, despite possibly depending on ω_t, Sutton & Barto (2018) calls these semi-gradient methods. An equilibrium for Q-learning with linear function approximation must be the fixed-point for the projected Bellman operator, thus verifying the fixed-point equation

$$q_\omega = \operatorname{Proj}\left(\mathrm{H}q_\omega\right). \tag{17}$$

Unfortunately, since the Projected Bellman operator is not generally a contraction, the fixed-point may not exist and, even when it does, the solution may not be an equilibrium of the associated dynamical system

$$\dot{\omega} = \mathbb{E}_\mu\left[\phi\left(x_t, a_t\right)\left(\tau_t - q_\omega(x_t, a_t)\right)\right]. \tag{18}$$

Consequently, Q-learning with linear function approximation can diverge, failing to provide any stable solution. The divergence of Q-learning is evidenced in classic counter-examples where the parameters of the approximator do not approach any solution, either oscillating within a window (Boyan & Moore, 1995; Gordon, 2001) or growing without bound (Tsitsiklis & Van Roy, 1996; Baird, 1995). Currently, the theoretical results that establish convergence of Q-learning restrict the data or the features too much (Szepesvári & Smart, 2004a; Melo et al., 2008), and the proposed variants of Q-learning that are guaranteed to converge under more general conditions (Carvalho et al., 2020; Zhang et al., 2021; Lim et al., 2022) converge to fundamentally biased limit solutions that do not hold good performance guarantees (Chen et al., 2022a).

3 Multi-Bellman Operator

Let us define the multi-Bellman operator H^n, such that

$$\mathrm{H}^1 q = \mathrm{H}q \tag{19}$$

and

$$\mathrm{H}^{n+1} q = \mathrm{H}\left(\mathrm{H}^n q\right). \tag{20}$$

Since $q^* = \mathrm{H}q^*$, it also holds that, for every n in $\mathbb{N}$,

$$q^* = \mathrm{H}^n q^*. \tag{21}$$

Proposition 3.1. *The operator H^n is a contraction in the ∞-norm with contraction factor γ^n.*

Proof. See the proof of B.1 of the appendix. □

While unsurprising, the result has profound implications on the use of linear function approximation.

Assumption 3.2. The features and data are such that the matrix $\mathbb{E}_\mu\left[\phi(x,a)\phi^T(x,a)\right]$ is invertible.

Theorem 3.3. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$, the operator $\operatorname{Proj}\mathrm{H}^n$ is a contraction in the ∞-norm.*

Proof. See the proof of B.2 of the appendix. □

Corollary 3.4. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$, there exists a unique solution $\tilde{\omega}^n$ to the fixed-point equation*

$$q_\omega = \operatorname{Proj}(\mathrm{H}^n q_\omega). \tag{22}$$

The result states that, while the projected Bellman operator may not be contractive, the projected multi-Bellman operator is guaranteed to be contractive for sufficiently large n. Consequently, whereas the fixed-point equation of the projected Bellman operator may fail to have a solution, the fixed-point equation of the projected multi-Bellman operator is guaranteed to have a unique solution.

While the previous result establishes existence and uniqueness of solution, it says nothing about the quality of such solution in light of the reinforcement learning objective. Now, we compare $q_{\tilde{\omega}^n}$ with the optimal solution.

Assumption 3.5. For all $t \in \mathbb{N}$, (x_t, a_t) is i.i.d. from μ.

Theorem 3.6. *For all $n \geq N$, the solution $\tilde{\omega}^n$ is such that*

$$\left\|q^* - q_{\tilde{\omega}^n}\right\|_\infty \leq \frac{1}{1-\lambda\gamma^n}\left\|q^* - q_{\omega^*}\right\|_\infty. \tag{23}$$

where $\lambda = \sigma_{\max}\phi_{\max}^2$, with $\phi_{\max} = \max_{x,a}\|\phi(x,a)\|_2$ and $\sigma_{\max} = \left\|\mathbb{E}_\mu\left[\phi(x,a)\phi^T(x,a)\right]^{-1}\right\|_2$.

Proof. See the proof of B.3 of the appendix. □

Corollary 3.7. *The sequence of $\{\tilde{\omega}\}_{n\geq N}$ is such that*

$$\lim_{n\to\infty}\|\omega^* - \tilde{\omega}^n\| = 0. \tag{24}$$

The results state that the solution $q_{\tilde{\omega}}$ can be made arbitrarily close to the optimal solution q_{ω^*}, and can therefore provide an optimal approximation of q^* in a given function approximation space. In the following, we design a stochastic approximation algorithm that aims to approximate $\tilde{\omega}^n$.

4 Multi Q-learning

For given $n \in \mathbb{N}$, to approximate the solution of

$$q_\omega = \operatorname{Proj}(\mathrm{H}^n q_\omega), \tag{25}$$

we propose to perform the multi Q-learning update

$$\omega_{t+1} = \omega_t + \alpha_t\left[\phi(x_t,a_t)\left(\tau_t^{n,m} - q_{\omega_t}(x_t,a_t)\right)\right], \tag{26}$$

where the target $\tau_t^{n,m} := \tau^{n,m}(x_t,a_t)$ is a sample for $\left(\mathrm{H}^n q_{\omega_t}\right)(x_t,a_t)$ defined recursively as:

$$\tau^{1,m}(x_t,a_t) = r_t + \gamma\max_{a'} q_{\omega_t}(x_t',a') \tag{27}$$

and

$$\tau^{n+1,m}(x_t, a_t) = r_t + \gamma \max_{a'} \left(\frac{1}{m} \sum_{i=1}^{m} \tau_i^{n,m}(x'_t, a') \right), \tag{28}$$

where $\tau_i^{n,m}(x'_t, a')$ are i.i.d. $\tau^{n,m}(x'_t, a')$ and are used to build an average with m samples.

At time step t, instead of building a 1-step greedy target, multi Q-learning builds a target that is obtained by trying, on every state encountered along n-step trajectory starting at x'_t, every action m times, and maximizing the average of the sum of the discounted rewards at each level. We assume access to a fixed replay buffer $\mathcal{D}$, from which state-action pairs are sampled, and a simulator to sample transitions. Our algorithm is most similar to the one of Kearns et al. (2002). In our case, in addition to off-policy learning, we have function approximation and bootstrapping. Our algorithm is also similar to the one of Rosenberg et al. (2023), but with off-policy learning and function approximation. More generally, our n-step updates differ from the usual in that they are policy-independent, which as we will see allows our updates to approximate the multi-Bellman operator accurately. Moreover, our setting introduces additional technical challenges due to maximization at each step and at each level, which does not usually appear in the related works and can lead to biased updates and solution if not properly handled. We provide a pseudo-code of Multi Q-learning in Algorithm 1, which calls a function that builds the target for the update defined in Algorithm 2.

Algorithm 1 Multi Q-learning

Initialize weight vector ω_0
Given a fixed replay buffer $\mathcal{D}$ containing state-action pairs (s_t, a_t)
Access to a simulator that returns (r_t, s'_t) given (s_t, a_t)
For each training iteration:

- **Sample** a random state-action pair (s_t, a_t) from $\mathcal{D}$
- **Simulate** reward r_t and next state s'_t by calling the simulator with (s_t, a_t)
- **Call** build_target to compute the target:

$$\tau^{n,m}(s_t, a_t) = \text{build_target}(n, m, s_t, a_t, \omega_t)$$

- Update weights:

$$\omega_{t+1} \leftarrow \omega_t + \alpha \left[\phi(s_t, a_t) \left(\tau^{n,m}(s_t, a_t) - q_{\omega_t}(s_t, a_t) \right) \right]$$

4.1 Convergence

Contrary to the case of Q-learning, there is a solution to the dynamical system of the multi Q-learning update.

Proposition 4.1. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$, there exists an equilibrium $\hat{\omega}^{n,m}$ for the dynamical system*

$$\dot{w} = \mathbb{E}_\mu \left[\phi(x, a) \left(\tau^{n,m}(x, a) - q_\omega(x, a) \right) \right] \tag{29}$$

Proof. See the proof of C.1 of the appendix. □

We show that multi Q-learning converges to the solution.

Assumption 4.2. The sequence of learning rates $\{\alpha_t\}_{t \in \mathbb{N}}$ satisfies $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$.

Theorem 4.3. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$ and $m \in \mathbb{N}$, the sequence of $\{\omega_t\}_{t \in \mathbb{N}}$ is such that*

$$\lim_{t \to \infty} \|\hat{\omega}^{n,m} - \omega_t\| = 0. \tag{30}$$

Algorithm 2 build_target function

Input: parameters n, m, state s, action a, weight vector ω
Output: Target τ
Simulate reward r and next state s' by calling the simulator with (s, a)
If $n = 1$:

- **Compute** the target:

$$\tau = r + \gamma \max_{a' \in \mathcal{A}} q_\omega(s', a')$$

Else:

- **For each** $i = 1 \dots m$:
  - **Recursively obtain sample:** $\tau_i^{n-1,m}(s', a') = \text{build_target}(n-1, m, s', a', \omega)$
- **Compute** the target:

$$\tau = r + \gamma \max_{a' \in \mathcal{A}} \sum_{i=1}^{m} \tau_i^{n-1,m}(s', a')$$

Return τ

Proof. See the proof of C.2 of the appendix. □

The result establishes conditions under which multi Q-learning converges to $q_{\hat{\omega}^{n,m}}$. The assumptions are commonplace (Tsitsiklis & Van Roy, 1996; Carvalho et al., 2020; Melo et al., 2008). Assumptions 3.2 and 4.2 are mild. Assumption 3.5 is the most restrictive of the three, as it requires the data distribution to have no shift during training. Same as the other authors, we use the assumption to facilitate our technical analysis of the convergence of Q-learning, which still fails in this benign setting. Nevertheless, the replay buffer can slow down the data distribution shift arbitrarily much. In the limit, the case of offline reinforcement learning, the distribution is indeed fixed and samples are i.i.d.. We do believe our result can be extended to the case where the data distribution converges.

4.2 Performance

Unfortunately, in general, $q_{\hat{\omega}^{n,m}}$ does not equal $q_{\tilde{\omega}^n}$. Specifically, when the environment is not deterministic, the target of multi Q-learning is biased, due to the difficulty in estimating the maximum of an expectation. In fact, it is impossible to estimate the expectation of the maximum without bias (Van Hasselt, 2013). We show, however, that as the number m of samples per action grows, regardless of the randomness of the environment, the solution of multi Q-learning converges to the unbiased solution $q_{\tilde{\omega}^n}$.

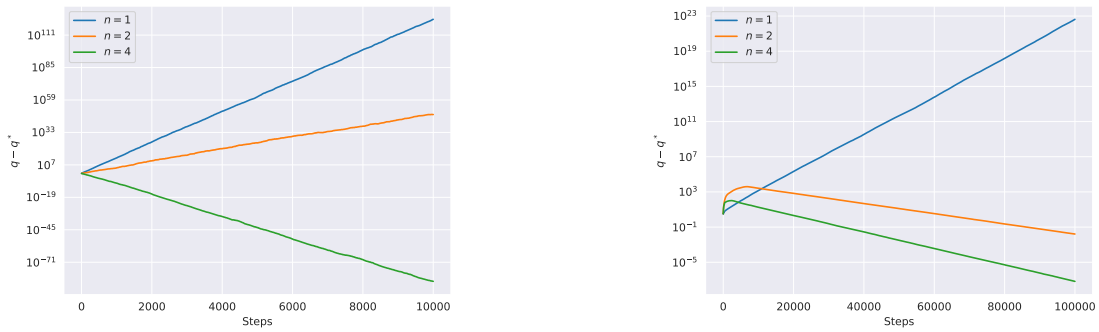
Theorem 4.4. *For all $n \geq N$ and $m \in \mathbb{N}$, we have that*

$$\|q_{\tilde{\omega}^n} - q_{\hat{\omega}^{n,m}}\| \leq \frac{\xi}{1 - \lambda\gamma^n} \frac{\max \sqrt{\text{var}\left(\tau^{n-1,m}\right)}}{\sqrt{m}}, \tag{31}$$

with $\xi = \lambda \frac{\sqrt{|\mathcal{A}|-1}}{1-\gamma}$.

Proof. See the proof of C.8 of the appendix. □

The results states that, for each n, when m increases, the solution of multi Q-learning $q_{\hat{\omega}^{n,m}}$ approaches the fixed-point of the projected Multi-Bellman operator $q_{\tilde{\omega}^n}$. On the other hand, it states that, for each m, when n increases, the distance between the two can grow, but is bounded, since we also have that $\sqrt{\text{var}\left(\tau^{n-1,m}\right)} \leq \frac{1}{1-\gamma}$ for all n and m.



(a) $\omega \to 2\omega$.

(b) Star.

Figure 1: Classic counter-examples. The y-axis shows the difference between the estimated q and q^*. In both examples, Q-learning, that is multi Q-learning with $n = 1$, diverges but, for sufficiently large n, multi Q-learning converges.

Corollary 4.5. *The sequence of* $\{\hat{\omega}^{n,m}\}_{m\in\mathbb{N}}$ *is such that*

$$\lim_{m\to\infty} \|\tilde{\omega}^n - \hat{\omega}^{n,m}\| = 0. \tag{32}$$

From Corollary 3.7, we also knew that, as n grows, the fixed-point of the projected Multi-Bellman operator $q_{\tilde{\omega}^n}$ approaches the optimal solution q_{ω^*}. Putting the two results together, we conclude that, increasing n and m, the solution of multi Q-learning becomes arbitrarily close to q_{ω^*}.

5 Experiments

In this section, we validate our theoretical findings. We start by focusing on the convergence property of multi Q-learning. All results are averages of five runs with standard deviations. Given the linear function approximation setting and relatively small scale of the environments, all experiments can be performed on standard commercial CPUs, with small memory costs, and lasting less than eight hours.

5.1 Convergence

We show that, on well-known problems where Q-learning with linear function approximation fails to converge, multi Q-learning converges if n is sufficiently large, as predicted in Theorem 4.3. We start by using two classical problems that are typically used in the literature to exhibit the divergence of Q-learning with linear function approximation. The first problem was introduced by Tsitsiklis & Van Roy (1996) and the second problem by Baird (1995). Even though both problems are relatively simple and even though the correct solution q^* is within the function approximation space, the parameters of Q-learning diverge to infinity in some conditions. Additionally, we introduce another simple example, which we call the Bias example. This example serves as a complement to the first examples by showcasing the bias that can appear in multi Q-learning, as predicted in 4.4, and how we can minimize it.

$\omega \to 2\omega$ The $\omega \to 2\omega$ is a problem due to Tsitsiklis & Van Roy (1996). The MDP has two states y_1 and y_2 and only one action b_1. Performing the action on any of the two states always takes the agent to the second state. The reward received is always zero. Therefore, q^* is zero. The features $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ such that $\phi(y_1, b_1) = 1$ and $\phi(y_2, b_1) = 2$. We consider a discount factor of 0.9, which is within the interval where Q-learning is originally reported to diverge, and a learning rate of 10^{-1}, and uniform data distribution. Figure 2 depicts the problem.

Star The Star problem is due to Baird (1995). The MDP has seven states y_1 to y_7 and two actions b_1 and b_2. The first action always takes the agent to the last state, the second action takes the agent to any

Figure 2: Transition diagram of the process in the $\omega \to 2\omega$ problem.

of the first five states uniformly. The reward received is always zero. Therefore, q^* is zero. The features are $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{16}$ such that, for j between 1 and 7, for all i between 1 and 16 $\phi_i(y_j, b_2) = \mathbf{1}(i = j + 1)$, for i between 2 and 7 $\phi_i(y_j, b_1) = 2 \cdot \mathbf{1}(i = j + 1)$, for j between 1 and 6 $\phi_1(y_j, b_1) = \mathbf{1}(j \leq 6)$ and $\phi_1(y_7, b_1) = 2$, $\phi_i(y_j, b_1) = 0$ otherwise. We consider a discount factor of 0.98, within the interval at which Q-learning diverges, a learning rate of 10^{-2}, and a data distribution that samples the first action one seventh of the times and the second action six sevenths of the times. Figure 3 depicts the problem.

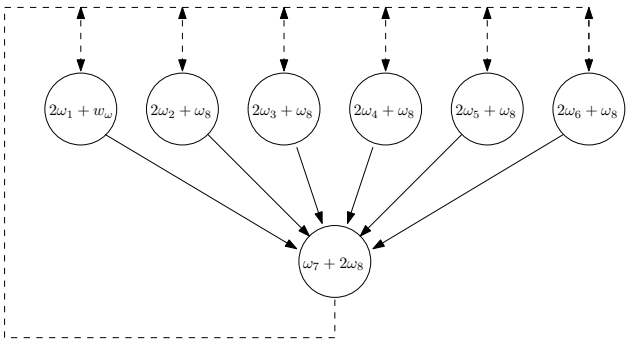


Figure 3: Transition diagram of the process in the Star problem.

We test multi Q-learning for $n \in \{1, 2, 4\}$. We note that $n = 1$ recovers the original Q-learning. We use $m = 1$. The experimental results in Figure 1 validate our theoretical finding that, whereas the projected Bellman operator may fail to have a fixed-point and Q-learning may diverge, multi Q-learning converges for large enough n, as predicted by Theorem 4.3, to the correct solution $q^* = 0$.

Bias In some non-deterministic environments that are not reward-free, the solution of multi-Q-learning may not equal the projected multi-Bellman operator, resulting in a bias that increases with n. To highlight this behavior, we introduce the Bias example. The Bias example is intentionally simple, consisting of only two states and two actions with uniformly random transitions. One state provides a reward of 1, while the other provides 0. This example serves two key purposes. First, it serves to illustrate the potential bias in multi Q-learning when $n > 1$. As shown in prior work (Van Hasselt, 2013), it is generally impossible to estimate the maximum of a set of expected values of random variables without introducing bias. In our case, for fixed n and m, multi Q-learning can converge to a biased solution of the equation $q = \text{Proj}(\text{H}^n q)$, and this bias increases with n. This phenomenon aligns with the predictions of our theoretical results (Theorem 4.4). Second, the Bias problem serves to demonstrate how increasing the parameter m effectively reduces the bias. Our theoretical results (Corollary 4.5) suggest that increasing m—the number of samples used to construct the update target—reduces this bias. The example provides a controlled setting to validate this claim.

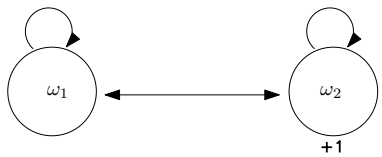
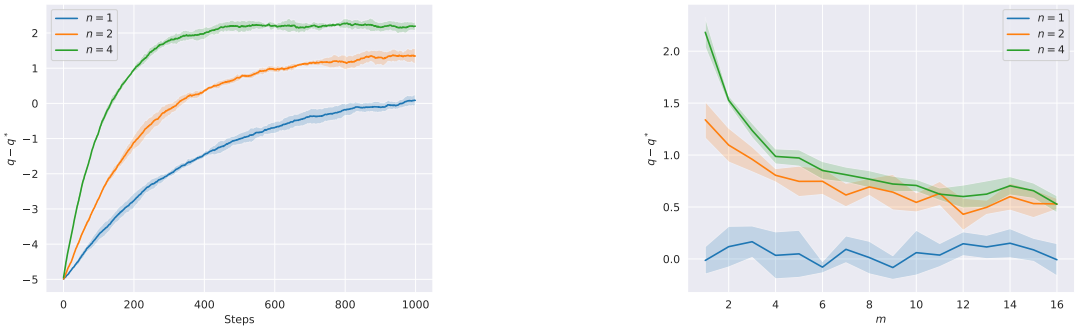


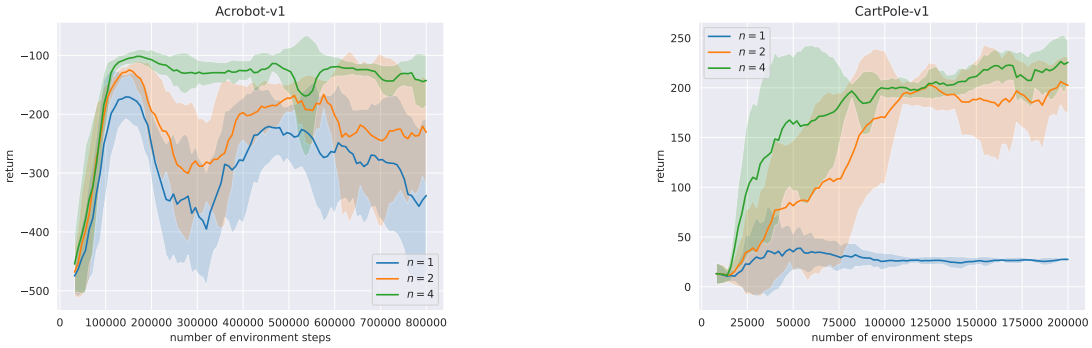
Figure 4: Transition diagram of the process in the Bias problem.



(a) The bias increases with n.

(b) The bias decreases with m.

Figure 5: Bias problem.



(a) Acrobot.

(b) Cartpole.

Figure 6: Classic control. The y-axis shows the average return. The performance increases with n.

We use tabular features, with a discount factor of 0.9 and a learning rate of 0.1. Figure 4 shows the transition diagram for the Bias problem. The results in Figure 5a confirm that, for $m = 1$, the bias increases as n increases. In contrast, Figure 5b shows that, as m increases, the bias decreases, aligning with the theoretical predictions in Corollary 4.5.

5.2 Performance

We also validate that, as n increases, the distance between the fixed-point of the projected Bellman operator and q_{ω^*} decreases, as predicted by Corollary 3.7. We use an ϵ-greedy policy where ϵ decays linearly from 100% to 5% during the first half of interactions and remains constant afterwards. We use a replay buffer with 20% of the total number of timesteps used for the environment. We plot a moving average of the last 5% of the total number of time steps. We use classic deterministic environments, which we describe.

Acrobot Acrobot is a classic control problem proposed by Sutton (1995) where a joint actuates two links such that one end is fixed and the other is free. The actions of the agent are to apply a negative torque to the joint, apply a positive torque to the joint or do nothing. The state space is the tuple composed of sine, cosine and angular velocity of each link. The reward is always minus one unless the free end of the links reaches a target height, after which the agent reaches a terminal state. We discretize each dimension of the state space in four and use gaussian features. We use a discount factor of 0.99 and a learning rate of $3 \cdot 10^{-3}$.

Cartpole Cartpole is a classic control problem proposed by Barto et al. (1983), where a cart balances a pole. The actions of the agent are to push the cart left or right. The state space is a tuple with the position and velocity of the cart and the angle and angular velocity of the pole. The reward is always one unless the pole falls and the agent reaches a terminal state. We discretize each dimension of the state space in two and use gaussian features. We use a discount factor of 0.99 and a learning rate of $3 \cdot 10^{-2}$.

The results in Figure 7 validate the finding that as n increases, the solution of multi Q-learning improves, as predicted by Corollary 3.7.

Altogether, our experimental results confirm that, contrary to the original Q-learning, multi Q-learning converges for sufficiently large n, that the distance between the projected multi-Bellman operator and the optimal $\operatorname{Proj} q^*$ decreases with n, and that the distance between the solution of multi Q-learning and the fixed-point of the projected multi-Bellman operator decreases with m.

6 Related Work

We analyze work related to our own, both with respect to the problem of Q-learning with linear function approximation and our proposed multi Q-learning solution.

Convergence conditions Around the time the divergence of Q-learning was established, some works aimed at reaching conditions under which Q-learning would converge. The first works proposed restrictions of the function approximation spaces themselves. They establish that with some specific choices of features, Q-learning is guaranteed to converge (Singh et al., 1994; Ormoneit & Sen, 2002; Szepesvári & Smart, 2004b). The linear architectures considered across the three works are extensions of what would be the one-hot representation in the tabular case. Afterwards, another work considered again general linear function approximation architectures. Melo et al. (2008) prove that Q-learning with linear function approximation converges if the distribution of state-action pairs the agent uses to learn is sufficiently close to the distribution that the optimal policy induces.

Gradient-TD methods Instead of finding conditions under which Q-learning converges, a different line of works proposes to take a step back and modify the objective that Q-learning is trying to solve. Maei et al. (2010) propose to perform full-gradient descent on a different objective, called the projected Bellman error. The resulting algorithm, called Greedy-GQ, is a gradient-TD methods. Being a full gradient method, Greedy-GQ is provably convergent to a minimum in the approximation space. However, the method can converge to local minima and there is no guarantee that the resulting greedy policy is a good control policy (Scherrer, 2010). Gradient-TD is also less efficient than semi-gradient methods (Mahadevan et al., 2014; Du et al., 2017).

Regularized methods The problem of divergence of Q-learning with function approximation was significantly revived after an empirical success story of Q-learning with deep neural networks (Mnih et al., 2015). One of the components of the renowned deep Q-network (DQN) therein is a target network that aims at compensating the instability generated by the Q-learning updates. While the DQN is not provably convergent, its empirical success inspired theoretical results. The works of Carvalho et al. (2020), Zhang et al. (2021) and Chen et al. (2022a) provided convergence guarantees for variants with target networks. Additionally, another work points out that the target network can be seen as a regularizer (Piché et al., 2021). As hinted by Farahmand (2011), several works prove that various forms of regularization of the Q-values or the parameters themselves can stabilize Q-learning, resulting in a convergent algorithms (Zhang et al., 2021; Lim et al., 2022; Agarwal et al., 2021). However, the introduction of the regularizers biases the solution obtained, even in deterministic environments. Contrarily, the bias of our method only appears in non-deterministic environments and can be minimized arbitrarily much.

Non-linear function approximation The behaviour of Q-learning with linear function approximation has been the focus of several theoretical works. In practice, however, Q-learning is mostly used with non-linear function approximation, especially through neural networks (Mnih et al., 2015). Still, there are works that address this more general setting. A recent work suggests a loss function that is decreasing over time,

assuming the neural network converges to a target network at each step (Wang & Ueda, 2021). However, having a loss function that is monotonically decreasing does not imply that neither the parameters of the approximator are converging nor that the Q-values are converging. Cai et al. (2019) and Xu & Gu (2020) provide finite-time analysis of Q-learning with over-parameterized neural networks that convergence happens only as the size of the network grows to infinity, which is an impractical scenario.

Lookahead In the context of policy iteration, several works hinted at the theoretical benefits of lookahead in the tabular case (De Asis et al., 2018; Efroni et al., 2018a). Efroni et al. (2018b) then identified a problem with soft policy improvement for lookahead policies, which happens if function approximation is used. Specifically, contrarily to what happens with a single-step policy improvement step, a multistep policy improvement step is not necessarily monotonically increasing. However, Efroni et al. (2020) and Winnicki & Srikant (2022) respectively provide finite-time and asymptotic results for lookahead in approximate policy iteration. Rosenberg et al. (2023) proposes an interesting policy-based algorithm with adaptive depth that results in convergence in the tabular case. We focus on the problem of divergence of Q-learning, a value-based algorithm, when used with linear function approximation architectures; the projected multi-Bellman operator that we introduced differs in that it is designed for evaluating the optimal policy. Chen et al. (2022b) also analyze an operator similar to our proposed multi-Bellman operator in the context of off-policy prediction in an actor-critic setting with linear function approximation, leveraging the improved fixed-point guarantees. Due to the maximization that appears on the Bellman operator for control, in our case we lose the property of linearity and, since a maximization and an expectation do not, in general, commute, the Q-learning setting is more challenging and the algorithm we propose is significantly different.

Planning and model-based reinforcement learning Multi-step real time dynamic programming algorithms, as defined by Efroni et al. (2018b) and discussed by Moerland et al. (2020), integrate planning and learning and have several successful practical applications (Silver et al., 2017; 2018). We refer to the survey of Moerland et al. (2023) for varied algorithms that include $Q(\sigma)$ (De Asis et al., 2018), tree-backup (Precup, 2000) and multi-step expected SARSA (Sutton & Barto, 2018). While most such algorithms are policy-based and on-policy, with separate value and policy and behavior-dependent solutions, we have an off-policy, value-based algorithm, which plans exclusively at training time multi Q-learning. Our fixed depth and full breadth setting are under-explored (Moerland et al., 2023).

7 Conclusion

Our work has made significant contributions to addressing the convergence challenge in Q-learning with linear function approximation. By introducing the multi-Bellman operator and demonstrating its contractive nature, we have paved the way for improved convergence properties. The proposed multi Q-learning algorithm effectively approximates the fixed-point solution of the projected multi-Bellman operator. Importantly, we have shown that the algorithm converges under relatively mild conditions.

7.1 Limitations and Future Work

Even though our algorithm can be combined with models that are learned concurrently to the reinforcement learning update to simulate the environment, in our experiments, we used a given simulator of the environment. It is not the case that such simulator is always available in applications. There are several techniques for learning a model of the environment. For example, we can learn the model previously or concurrently to reinforcement learning; the model can be exact or inexact; the model can be parametric or non-parametric. Multi Q-learning can be combined with any of the mentioned model-based approaches to remove the need for a simulator when it is not available.

Multi Q-learning also performs every action on every reached state while building the target. Thus, the computational costs will grow exponentially with the depth n. In case only some actions are performed on each state, we cannot show convergence to an optimal solution. Nevertheless, in some cases, the computational benefits of not performing every action on every state may outweigh the theoretical comfort. An analysis

of this trade-off would be valuable. Future work should also analyze techniques to prune the set of actions sampled at each level of multi Q-learning and reduce the complexity of the algorithm.

It would also be interesting to analyze multi Q-learning with non-linear function approximation, both theoretically and empirically. Specifically, under which conditions and function approximation architectures would our convergence result still hold? How would multi Q-learning with non-linear function approximation compare, in practice, with relevant policy-free reinforcement learning algorithms such as the DQN (Mnih et al., 2015)? And will the target of multi-Q-learning, if combined with the target network, increase the performance of the DQN?

References

Naman Agarwal, Syomantak Chaudhuri, Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Online target q-learning with reverse experience replay: Efficiently finding the optimal policy for linear mdps. *arXiv preprint arXiv:2110.08440*, 2021.

Terje Aven. Upper (lower) bounds on the mean of the maximum (minimum) of a number of random variables. *Journal of applied probability*, 22(3):723–728, 1985.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*, pp. 30–37, 1995.

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13 (5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077.

Vivek Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

Justin A. Boyan and Andrew W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems 7*, pp. 369–376, 1995.

Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/98baeb82b676b662e12a7af8ad9212f6-Paper.pdf`.

Diogo S. Carvalho, Francisco S. Melo, and Pedro A. Santos. A new convergent variant of q-learning with linear function approximation. *Advances in Neural Information Processing Systems*, 33, 2020.

Zaiwei Chen, John Paul Clarke, and Siva Theja Maguluri. Target network and truncation overcome the deadly triad in q-learning. *arXiv preprint arXiv:2203.02628*, 2022a.

Zaiwei Chen, Sajad Khodadadian, and Siva Theja Maguluri. Finite-sample analysis of off-policy natural actor–critic with linear function approximation. *IEEE Control Systems Letters*, 6:2611–2616, 2022b.

Kristopher De Asis, J Hernandez-Garcia, G Holland, and Richard Sutton. Multi-step reinforcement learning: A unifying algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pp. 1049–1058. PMLR, 2017.

Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Beyond the one-step greedy approach in reinforcement learning. In *International Conference on Machine Learning*, pp. 1387–1396. PMLR, 2018a.

Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Multiple-step greedy policies in approximate and online reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018b.

Yonathan Efroni, Mohammad Ghavamzadeh, and Shie Mannor. Online planning with lookahead policies. *Advances in Neural Information Processing Systems*, 33:14024–14033, 2020.

Amir-massoud Farahmand. Regularization in reinforcement learning. *PhD Thesis*, 2011.

Geoffrey J. Gordon. Reinforcement learning with function approximation converges to a region. In *Advances in neural information processing systems*, pp. 1040–1046, 2001.

Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49:193–208, 2002.

Han-Dong Lim, Do Wan Kim, and Donghwan Lee. Regularized q-learning. *arXiv preprint arXiv:2202.05404*, 2022.

Hamid R. Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S. Sutton. Toward off-policy learning control with function approximation. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 719–726, 2010.

Sridhar Mahadevan, Bo Liu, Philip Thomas, Will Dabney, Steve Giguere, Nicholas Jacek, Ian Gemp, and Ji Liu. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv preprint arXiv:1405.6757*, 2014.

Francisco S Melo and M Isabel Ribeiro. Q-learning with linear function approximation. In *Learning Theory: 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings 20*, pp. 308–322. Springer, 2007.

Francisco S. Melo, Sean Meyn, and M. Isabel Ribeiro. An analysis of reinforcement learning with function approximation. In *Proceedings of the 25th International Conference on Machine learning*, pp. 664–671, 2008.

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.

Thomas M Moerland, Anna Deichler, Simone Baldi, Joost Broekens, and Catholijn M Jonker. Think too fast nor too slow: the computational trade-off between planning and reinforcement learning. *arXiv preprint arXiv:2005.07404*, 2020.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2):161–178, 2002.

Alexandre Piché, Joseph Marino, Gian Maria Marconi, Christopher Pal, and Mohammad Emtiyaz Khan. Beyond target networks: Improving deep q-learning with functional regularization. *arXiv preprint arXiv:2106.02613*, 2021.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Martil L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005.

Aviv Rosenberg, Assaf Hallak, Shie Mannor, Gal Chechik, and Gal Dalal. Planning and learning with adaptive lookahead. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9606–9613, 2023.

Bruno Scherrer. Should one compute the temporal difference fix point or minimize the bellman residual? the unified oblique projection view. *arXiv preprint arXiv:1011.4362*, 2010.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Satinder Singh, Tommi Jaakkola, and Michael Jordan. Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, 7, 1994.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, 8, 1995.

C. Szepesvári and W. Smart. Interpolation-based Q-learning. In *Proceedings of the 21st International Conference on Machine learning*, pp. 100–107, 2004a.

Csaba Szepesvári and William D Smart. Interpolation-based q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 100, 2004b.

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

Hado Van Hasselt. Estimating the maximum expected value: an analysis of (nested) cross validation and the maximum sample average. *arXiv preprint arXiv:1302.7175*, 2013.

Zhikang T. Wang and Masahito Ueda. A convergent and efficient deep q network algorithm. *arXiv preprint arXiv:2106.15419*, 2021.

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

Anna Winnicki and R Srikant. Reinforcement learning with unbiased policy evaluation and linear function approximation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 801–806. IEEE, 2022.

Pan Xu and Quanquan Gu. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pp. 10555–10565. PMLR, 2020.

Shangtong Zhang, Hengshuai Yao, and Shimon Whiteson. Breaking the deadly triad with a target network. *arXiv preprint arXiv:2101.08862*, 2021.

A Stochastic approximation

Theorem A.1. *Let us suppose that the following hold for the stochastic approximation setting of Sec. 2.*

1. *The map* $g : \mathbb{R}^k \to \mathbb{R}^k$ *such that*

$$g(\omega) = \mathbb{E}\left[\phi\left(z\right)\left(\tau(\omega) - g_\omega(z)\right)\right]$$

   is Lipschitz;

2. *The sequence of* m_t *such that*

$$m_{t+1} = \phi(z_t)\left(\tau_{t+1} - g_{\omega_t}(z_t)\right) - g(\omega_t);$$

   is a martingale difference sequence with respect to $\{(\tau_s, \omega_s) : s \leq t\}$ *and is such that*

$$\mathbb{E}\left[\|m_{t+1}\|^2 \mid \{(\tau_s, \omega_s) : s \leq t\}\right] \leq c_m\left(1 + \|\omega_t\|^2\right).$$

3. *The o.d.e*

$$\dot{\omega} = g(\omega)$$

has a unique and globally asymptotically stable equilibrium ω^* *such that*

$$\omega^* = \mathbb{E}\left[\phi\left(z\right)\phi^T\left(z\right)\right]^{-1}\mathbb{E}\left[\phi\left(z\right)\tau(\omega)\right];$$

4. *The map* $g_c : \mathbb{R}^k \to \mathbb{R}^k$ *such that*

$$g_c(\omega) = \frac{g(c\omega)}{c}$$

is such that $\lim_{c\to\infty} \|g_c(\omega) - g_\infty(\omega)\|$ *uniformly on compacts for a* $g_\infty : \mathbb{R}^k \to \mathbb{R}^k$ *and that the origin is the unique and globally asymptotically stable equilibrium of*

$$\dot{\omega} = g_\infty(\omega).$$

Then, the sequence of stochastic approximation updates ω_t *converges to* ω^*.

B Multi-Bellman operator

Proposition B.1. *The operator* H^n *is a contraction in the* ∞*-norm with contraction factor* γ^n.

Proof. We establish the result by induction. First, we show that H contracts in the ∞-norm. We start by writing the expression $\|\mathrm{H}q - \mathrm{H}p\|_\infty$ in simpler terms, where q and p are value functions.

$$\begin{aligned}\|\mathrm{H}q - \mathrm{H}p\|_\infty &= \max_{x,a}\left|\mathbb{E}\left[r(x,a) + \gamma\max_{a'\in\mathcal{A}} q\left(x',a'\right)\right] - \mathbb{E}\left[r(x,a) + \gamma\max_{a'\in\mathcal{A}} p\left(x',a'\right)\right]\right| \\ &= \gamma\max_{x,a}\left|\mathbb{E}\left[\max_{a'\in\mathcal{A}} q\left(x',a'\right) - \max_{a'\in\mathcal{A}} p\left(x',a'\right)\right]\right|.\end{aligned}$$

Next, we make use of Jensen's inequality to establish that

$$\|\mathrm{H}q - \mathrm{H}p\|_\infty \leq \gamma\max_{x,a}\mathbb{E}\left[\left|\max_{a'\in\mathcal{A}} q\left(x',a'\right) - \max_{a'\in\mathcal{A}} p\left(x',a'\right)\right|\right].$$

We use that the absolute difference of maxima is less than or equal to the maxima of absolute difference.

$$\|\mathrm{H}q - \mathrm{H}p\|_\infty \leq \gamma\max_{x,a}\mathbb{E}\left[\max_{a'\in\mathcal{A}}\left|q\left(x',a'\right) - p\left(x',a'\right)\right|\right].$$

Then, we use that the expectation of a random function is less than or equal to its maximum.

$$\begin{aligned}\|\mathrm{H}q - \mathrm{H}p\|_\infty &\leq \gamma\max_{x'\in\mathcal{X}}\max_{a'\in\mathcal{A}}\left|q\left(x',a'\right) - p\left(x',a'\right)\right| \\ &= \gamma\max_{x',a'}\left|q\left(x',a'\right) - p\left(x',a'\right)\right| \\ &= \gamma\left\|q - p\right\|_\infty\end{aligned}$$

Now, we make the induction step for $\mathrm{H}^{n+1} = \mathrm{H}\mathrm{H}^n$, observing that

$$\begin{aligned}\left\|\mathrm{H}\left(\mathrm{H}^n q\right) - \mathrm{H}\left(\mathrm{H}^n p\right)\right\|_\infty &\leq \gamma\left\|\mathrm{H}^n q - \mathrm{H}^n p\right\|_\infty \\ &= \gamma\gamma^n\left\|q - p\right\|_\infty \\ &= \gamma^{n+1}\left\|q - p\right\|_\infty.\end{aligned}$$

We conclude the proof. □

Theorem B.2. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$, $\operatorname{Proj} \mathrm{H}^n$ is a contraction in the ∞-norm.*

Proof. We want to show that there exists a function $\lambda : \mathbb{N} \to \mathbb{R}$ such that

$$\|\operatorname{Proj}(\mathrm{H}^n q) - \operatorname{Proj}(\mathrm{H}^n p)\|_\infty \leq \lambda(n) \|q - p\|_\infty$$

and such that, for all n greater than an existing N, $\lambda(n)$ is strictly smaller than 1.

We start by showing that Proj satisfies

$$\|\operatorname{Proj} q - \operatorname{Proj} p\|_\infty \leq \phi_{\max}^2 \sigma_{\max} \|q - p\|_\infty .$$

To see that, we start by recalling that

$$(\operatorname{Proj} q)(x, a) = \phi^T(x, a) \mathbb{E}\left[\phi(x, a)\phi^T(x, a)\right]^{-1} \mathbb{E}\left[\phi(x, a) q(x, a)\right].$$

Then, we have that

$$\|\operatorname{Proj} q - \operatorname{Proj} p\|_\infty = \max_{x,a} |\phi^T(x, a) \mathbb{E}\left[\phi(x, a)\phi^T(x, a)\right]^{-1} \mathbb{E}\left[\phi(x, a)\left(q(x, a) - p(x, a)\right)\right]|.$$

Now, we make use of a Cauchy-Schwarz inequality to say that

$$\|\operatorname{Proj} q - \operatorname{Proj} p\|_\infty \leq \max_{x,a} \left\|\phi^T(x, a) \mathbb{E}\left[\phi(x, a)\phi^T(x, a)\right]^{-1}\right\|_2 \|\mathbb{E}\left[\phi(x, a)\left(q(x, a) - p(x, a)\right)\right]\|_2$$

and we can write, using the definition of the matrix norm induced by the Euclidean norm in $\mathbb{R}^k$

$$\begin{aligned}\|\operatorname{Proj} q - \operatorname{Proj} p\|_\infty \leq \max_{x,a} &\left\|\phi^T(x, a)\right\| \left\|\mathbb{E}\left[\phi(x, a)\phi^T(x, a)\right]^{-1}\right\|_2 \cdot \\ \cdot \quad &\|\mathbb{E}\left[\phi(x, a)\left(q(x, a) - p(x, a)\right)\right]\|_2\end{aligned}$$

We obtain that

$$\begin{aligned}\|\operatorname{Proj} q - \operatorname{Proj} p\|_\infty &\leq \phi_{\max}\sigma_{\max} \cdot \mathbb{E}\left[\|\phi(x, a)\|_2 |q(x, a) - p(x, a)|\right] \\ &\leq \phi_{\max}^2 \sigma_{\max} \|q - p\|_\infty .\end{aligned}$$

To conclude the proof, we start by combining the inequality we just established and Proposition B.1.

$$\begin{aligned}\|\operatorname{Proj}(\mathrm{H}^n q) - \operatorname{Proj}(\mathrm{H}^n p)\| &\leq \phi_{\max}^2 \sigma_{\max} \|\mathrm{H}^n q - \mathrm{H}^n p\|_\infty \\ &\leq \phi_{\max}^2 \sigma_{\max} \gamma^n \|q - p\|_\infty\end{aligned}$$

We finish the proof by using $\lambda(n) = \phi_{\max}^2 \sigma_{\max} \gamma^n$. It is true that $\lambda(n) < 1$ for all $n \geq N$ when we consider $N = \lceil -\log_\gamma \left(\phi_{\max}^2 \sigma_{\max}\right) \rceil$.[1] □

Theorem B.3. *For all $n \geq N$, where N is identified in Theorem 3.3, $\tilde{\omega}^n$ is such that*

$$\|q^* - q_{\tilde{\omega}^n}\|_\infty \leq \frac{1}{1 - \lambda\gamma^n} \|q^* - q_{\omega^*}\|_\infty , \tag{33}$$

Proof. Let us use the triangle inequality with vertex q_{ω^*}.

$$\|q^* - q_{\tilde{\omega}^n}\|_\infty \leq \|q^* - q_{\omega^*}\|_\infty + \|q_{\omega^*} - q_{\tilde{\omega}^n}\|_\infty .$$

Now we focus on the second term of the right-hand side above. We have that

$$\|q_{\omega^*} - q_{\tilde{\omega}^n}\|_\infty = \|\operatorname{Proj}(\mathrm{H}^n q^*) - \operatorname{Proj}(\mathrm{H}^n q_{\tilde{\omega}^n})\|_\infty .$$

From Proposition 3.3, we have that

$$\|q_{\omega^*} - q_{\tilde{\omega}^n}\|_\infty \leq \sigma_{\max}\phi_{\max}^2 \gamma^n \|q^* - q_{\tilde{\omega}^n}\|_\infty .$$

We can conclude the result taking algebraic operations, using $\lambda = \sigma_{\max}\phi_{\max}^2$. □

[1] $\lceil \cdot \rceil : \mathbb{R}^+ \to \mathbb{N}$ is the ceiling function, giving the smallest natural that is larger or equal to its argument.

C Multi Q-learning

Theorem C.1. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$, there exists an equilibrium $\hat{w}^{n,m}$ for the dynamical system*

$$\dot{\omega} = \mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - q_\omega(x,a)\right)\right]$$

Proof. We want to solve

$$\dot{\omega} = 0$$

or, equivalently,

$$\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - q_\omega(x,a)\right)\right] = 0.$$

Taking linear operations, we have equivalently

$$\omega = \mathbb{E}\left[\phi(x,a)\phi^T(x,a)\right]^{-1}\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega)\right)\right]$$

We show existence and uniqueness of solution to the fixed-point equation by showing that the right-hand side is contractive. For starters, we have that

$$\left\|\mathbb{E}\left[\phi(x,a)\phi^T(x,a)\right]^{-1}\right\|_2 = \sigma_{\max}.$$

Then, for any $\delta \in \mathbb{R}^k$, using Jensen's and Cauchy-Schwarz inequality, we have that

$$\begin{aligned}
&\left\|\mathbb{E}\left[\phi(x,a)\phi^T(x,a)\right]^{-1}\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - \tau^{n,m}(\omega+\delta)\right)\right]\right\|_2 \\
&\quad \leq \sigma_{\max}\mathbb{E}\left[\|\phi(x,a)\|_2 \left|\mathbb{E}\left[\tau^{n,m}(\omega) - \tau^{n,m}(\omega+\delta)\right]\right|\right] \\
&\quad \leq \sigma_{\max}\phi_{\max}\gamma^n \|q_\omega - q_{\omega+\delta}\|_\infty \\
&\quad \leq \sigma_{\max}\phi_{\max}^2\gamma^n \|\delta\|_2
\end{aligned}$$

Thus, if n is large enough, the operator is contractive and the fixed-point equation has a unique solution. □

C.1 Convergence

Theorem C.2. *There exists $N \in \mathbb{N}$ such that, for all $n \geq N$ and $m \in \mathbb{N}$, the sequence of $\{\omega_t\}_{t\in\mathbb{N}}$ is such that*

$$\lim_{t\to\infty} \|\hat{\omega}^{n,m} - \omega_t\|_2 = 0.$$

Proof. To establish the result, we verify that we are in the conditions where Theorem A.1, which gives conditions for the convergence of stochastic approximation algorithms, holds. We do so by proving the following lemmas, one for each of the assumptions of the theorem. □

Lemma C.3. *The map $g : \mathbb{R}^k \to \mathbb{R}^k$ such that*

$$g(\omega) = \mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - q_\omega(x,a)\right)\right]$$

is Lipschitz.

Proof. We start by writing

$$\begin{aligned}
\|g(\omega) - g(\theta)\|_2 &= \left\|\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - q_\omega(x,a)\right)\right] - \mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\theta) - q_\theta(x,a)\right)\right]\right\|_2 \\
&\leq \left\|\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega) - \tau^{n,m}(\theta)\right)\right]\right\|_2 + \left\|\mathbb{E}\left[\phi(x,a)\left(q_\theta(x,a) - q_\omega(x,a)\right)\right]\right\|_2.
\end{aligned}$$

We can take care of the second term by making use of Jensen's and Cauchy-Schwarz' inequality.

$$\begin{aligned}\left\|\mathbb{E}\left[\phi(x,a)\left(q_\theta(x,a)-q_\omega(x,a)\right)\right]\right\|_2 &\leq \mathbb{E}\left[\left\|\phi(x,a)\right\|_2\left|q_\theta(x,a)-q_\omega(x,a)\right|\right]\\ &\leq \phi_{\max}\left\|q_\theta-q_\omega\right\|_\infty\\ &\leq \phi_{\max}^2\left\|\omega-\theta\right\|_2.\end{aligned}$$

Finally, by considering the sum of the coefficients from the first (obtained in the proof of C.1) and second terms, we have

$$\left\|g(\omega)-g(\theta)\right\|_2 \leq \phi_{\max}^2\left(1+\gamma^n\right)\left\|\omega-\theta\right\|_2,$$

or that g is Lipschitz continuous. □

Lemma C.4. *The sequence of m_t such that*

$$m_{t+1}=\phi(x_t,a_t)\left(\tau_t^{n,m}-g_{\omega_t}(x_t,a_t)\right)-g(\omega_t)$$

is a martingale difference sequence with respect to $\{(\tau_s^{n,m},\omega_s):s\leq t\}$ and is such that

$$\mathbb{E}\left[\left\|m_{t+1}\right\|^2 \mid \{(\tau_s^{n,m},\omega_s):s\leq t\}\right]\leq c_m\left(1+\left\|\omega_t\right\|^2\right).$$

Proof. A martingale difference sequence has zero expectation conditioned on the past. Such condition is evident once we consider Assumption 3.5. Then, a martingale difference sequence also has finite first moment. That becomes evident when we observe that every term on the definition is bounded. Finally, since, again, every term on the definition is also bounded, the second moment is bounded. □

Lemma C.5. *The o.d.e*

$$\dot{\omega}=g(\omega)$$

has a unique and globally asymptotically stable equilibrium $\tilde{\omega}$ such that

$$\tilde{\omega}=\mathbb{E}\left[\phi\left(x,a\right)\phi^T\left(x,a\right)\right]^{-1}\mathbb{E}\left[\phi\left(x,a\right)\tau^{n,m}(\tilde{\omega})\right].$$

Proof. We already know from Proposition 5 that the equilibrium $\tilde{\omega}$ exists and is unique. Let us consider a Lyapunov function $l:\mathbb{R}^k\to\mathbb{R}$ such that $l(\omega)=\frac{1}{2}\left\|\tilde{\omega}-\omega\right\|^2$. Existence of $\tilde{\omega}$ is guaranteed from Banach's fixed-point theorem and Proposition 3.3. It is clear that $l(\omega)=0$ if $\omega=\tilde{\omega}$ and that $l(\omega)>0$ if $\omega\neq\tilde{\omega}$. Now, we show that $\dot{l}(\omega)<0$ if $\omega\neq\tilde{\omega}$.

We start by noticing that $\dot{l}(\omega)=\nabla l\cdot\dot{\omega}$. Since $\nabla l=-\left(\tilde{\omega}-\omega\right)$ and $\dot{\omega}=g(\omega)$, we have that

$$\dot{l}(\omega)=-\left(\tilde{\omega}-\omega\right)g(\omega).$$

Since $g(\tilde{\omega})=0$, it is also true that

$$\dot{l}(\omega)=-\left(\tilde{\omega}-\omega\right)\left(g\left(\omega\right)-g\left(\tilde{\omega}\right)\right)$$

For now, let us focus on the second term of the right hand side and write

$$g(\omega)-g(\tilde{\omega})=\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega)-\tau^{n,m}(\tilde{\omega})\right)\right]-\mathbb{E}\left[\phi(x,a)\left(q_{\tilde{\omega}}(x,a)-q_\omega(x,a)\right)\right]$$

Let us make two assertions, one for each of the terms on the right-hand side of the equation. First,

$$\left\|\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\tilde{\omega})-\tau^{n,m}(\omega)\right)\right]\right\|_2=\left\|\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\tilde{\omega})-\tau^{n,m}(\omega)\right)\right]\right\|_2.$$

Using Jensen's and Cauchy-Schwarz' inequality just as in the previous proof, we get that

$$\left\|\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega)-\tau^{n,m}(\tilde{\omega})\right)\right]\right\|_2\leq\phi_{\max}^2\gamma^n\left\|\tilde{\omega}-\omega\right\|$$

Second, we have that

$$\mathbb{E}\left[\phi(x,a)\left(q_{\tilde{\omega}}(x,a)-q_{\omega}(x,a)\right)\right]=\mathbb{E}\left[\phi(x,a)\phi^{T}(x,a)\right](\tilde{\omega}-\omega)\,.$$

Having established the two assertions mentioned, we make use of them in the following.

$$\begin{aligned}\dot{l}(\omega) &= -(\tilde{\omega}-\omega)\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega)-\tau^{n,m}(\tilde{\omega})\right)\right]- \\ &\qquad -(\tilde{\omega}-\omega)\mathbb{E}\left[\phi(x,a)\phi^{T}(x,a)\right](\tilde{\omega}-\omega) \\ &\leq \left\|(\tilde{\omega}-\omega)\,\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}(\omega)-\tau^{n,m}(\tilde{\omega})\right)\right]\right\|_2- \\ &\qquad -(\tilde{\omega}-\omega)\mathbb{E}\left[\phi(x,a)\phi^{T}(x,a)\right](\tilde{\omega}-\omega) \\ &\leq \phi_{\max}^{2}\gamma^{n}\left\|\tilde{\omega}-\omega\right\|_2^2-\lambda_{\min}\left\|\tilde{\omega}-\omega\right\|_2^2 \\ &= \left(\phi_{\max}^{2}\gamma^{n}-\lambda_{\min}\right)\left\|\tilde{\omega}-\omega\right\|_2^2,\end{aligned}$$

where $\lambda_{\min}$is the smallest eigenvalue of the auto-correlation matrix $\mathbb{E}\left[\phi(x,a)\phi^{T}(x,a)\right]$. The expression is negative if n is large enough. That concludes the proof. □

Lemma C.6. *The map $g_c:\mathbb{R}^k\to\mathbb{R}^k$ such that*

$$g_c(\omega)=\frac{g(c\omega)}{c}$$

is such that $\lim_{c\to\infty}\|g_c(\omega)-g_\infty(\omega)\|=0$ uniformly on compacts for some $h_\infty:\mathbb{R}^k\to\mathbb{R}^k$ and that the origin is the unique and globally asymptotically stable equilibrium of

$$\dot{\omega}=g_\infty(\omega).$$

Proof. Let us expand the definition and obtain

$$\begin{aligned}g_c(\omega) &= \frac{\mathbb{E}\left[\phi(x,a)\left(\tau^{n,m}-q_{c\omega}(x,a)\right)\right]}{c} \\ &= \frac{\mathbb{E}\left[\phi(x,a)\tau^{n,m}\right]}{c}-\mathbb{E}\left[\phi(x,a)q_\omega(x,a)\right].\end{aligned}$$

As $c\to\infty$, we have that $g_c(\omega)\to g_\infty(\omega)$ uniformly, using $g_\infty(\omega)=-\mathbb{E}\left[\phi(x,a)q_\omega(x,a)\right]$.

We then have that the ordinary differential equation

$$\begin{aligned}\dot{\omega} &= g_\infty(\omega) \\ &= -\mathbb{E}\left[\phi(x,a)q_\omega(x,a)\right] \\ &= -\mathbb{E}\left[\phi(x,a)\phi^{T}(x,a)\right]\omega\end{aligned}$$

is linear and time-invariant. Since the matrix above is invertible, we have that the origin is its unique and globally asymptotically stable equilibrium. We conclude the result. □

C.2 Performance

Lemma C.7. *We establish the following lemma, quantifying the bias between the multi Q-learning target and the target from the multi-Bellman operator. We will use the lemma in the subsequent theorem, which will include function approximation.*

$$\mathbb{E}\left[\tau^{n,m}(x,a)-\left(\mathrm{H}^{n}q\right)(x,a)\right]\leq\sqrt{(|\mathcal{A}|-1)}\sum_{i=1}^{n}\gamma^{n-i}\frac{\vartheta^{i}}{\sqrt{m}},$$

where $\vartheta^n=\max\sqrt{\mathrm{var}}\left(\tau^{n-1,m}\right)$ and $\vartheta^1=0$

Proof. We prove by induction. The basis is true because the bias of the 1-step update is zero. We move on to the step.

From the definition and then taking algebraic operations, we have,

$$\begin{aligned}\mathbb{E}\left[\tau^{n+1,m}(x,a) - \left(\mathrm{H}^{n+1}q\right)(x,a)\right] &= \mathbb{E}\Bigg[\left(r(x,a) + \gamma \max_{a'}\left(\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right)\right)\right) - \\ &\quad - \left(r(x,a) + \gamma \max_{a'}\left(\left(\mathrm{H}^{\mathrm{n}}\mathrm{q}\right)(x',a')\right)\right)\Bigg] \\ &= \gamma\mathbb{E}\left[\max_{a'}\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right) - \max_{a'}\left(\left(\mathrm{H}^{n}q\right)(x',a')\right)\right]\end{aligned}$$

Now, we use a result from Aven that says that, for given set of random variables $\{\zeta_i\}_{i\leq I}$

$$\mathbb{E}\left[\max_i \zeta_i\right] \leq \max_i \mathbb{E}\left[\zeta_i\right] + \sqrt{\frac{I-1}{I}\sum_i \mathrm{var}\left(\zeta_i\right)},$$

and have, after Jensen's inequality exchanges the order of the max and the expectation, that

$$\begin{aligned}\mathbb{E}\left[\tau^{n+1,m}(x,a) - \left(\mathrm{H}^{n+1}q\right)(x,a)\right] &\leq \gamma\mathbb{E}\left[\max_{a'}\mathbb{E}\left[\tau^{n,m}\left(x',a'\right)\right] - \max_{a'}\left(\mathrm{H}^{n}q\right)(x',a')\right] \\ &\quad + \gamma\sqrt{\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\sum_{a'}\mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right)\right)}\end{aligned}$$

Then, we add the absolute value and see that

$$\begin{aligned}\left|\mathbb{E}\left[\tau^{n+1,m}(x,a) - \left(\mathrm{H}^{n+1}q\right)(x,a)\right]\right| &\leq \gamma\mathbb{E}\left[\left|\max_{a'}\mathbb{E}\left[\tau^{n,m}\left(x',a'\right)\right] - \max_{a'}\left(\mathrm{H}^{n}q\right)(x',a')\right|\right] \\ &\quad + \gamma\sqrt{\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\sum_{a'}\mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right)\right)}\end{aligned}$$

$$\begin{aligned}&\leq \gamma\mathbb{E}\left[\max_{a'}\mathbb{E}\left[\tau^{n,m}\left(x',a'\right) - \left(\mathrm{H}^{n}q\right)(x',a')\right]\right] \\ &\quad + \gamma\sqrt{\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\sum_{a'}\mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right)\right)} \\ &\leq \gamma\mathbb{E}\left[\max_{a'}\mathbb{E}\left[\tau^{n,m}\left(x',a'\right) - \left(\mathrm{H}^{n}q\right)(x',a')\right]\right] \\ &\quad + \gamma\sqrt{\frac{|\mathcal{A}|-1}{|\mathcal{A}|}|\mathcal{A}|\max\mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m}\tau^{n,m}\left(x',a'\right)\right)}\end{aligned}$$

Now, we use that for a given random variable ζ,

$$\mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m}\zeta\right) = \frac{1}{m}\mathrm{var}\left(\zeta\right)$$

and proceed to see that

$$\begin{aligned}\left|\mathbb{E}\left[\tau^{n+1,m}(x,a) - \left(\mathrm{H}^{n+1}q\right)(x,a)\right]\right| &\leq \gamma\mathbb{E}\left[\tau^{n,m}(x,a) - \left(\mathrm{H}^{n}q\right)(x',a')\right] \\ &\quad + \gamma\sqrt{\frac{\left(|\mathcal{A}|-1\right)\max\mathrm{var}\left(\tau^{n,m}\left(x',a'\right)\right)}{m}}\end{aligned}$$

Next, we use that to conclude

$$\sqrt{\max(\cdot)} = \max \sqrt{\cdot}$$

$$\begin{aligned}\mathbb{E}\left[\tau^{n+1,m}(x,a) - \left(\mathrm{H}^{n+1}q\right)(x,a)\right] | &\leq \gamma\sqrt{|\mathcal{A}|-1}\sum_{i=1}^{n}\gamma^{n-i}\frac{\vartheta^i}{\sqrt{m}} + \gamma\sqrt{|\mathcal{A}|-1}\frac{\vartheta^{n+1}}{\sqrt{m}} \\ &\leq \sqrt{|\mathcal{A}|-1}\sum_{i=1}^{n+1}\gamma^{n+1-i}\frac{\vartheta^i}{\sqrt{m}}.\end{aligned}$$

□

Theorem C.8. *For all $n \geq N$ and $m \in \mathbb{N}$, we have that*

$$\|q_{\tilde{\omega}^n} - q_{\hat{\omega}^{n,m}}\| \leq \frac{\xi}{1-\lambda\gamma^n}\frac{\vartheta^n}{\sqrt{m}},$$

$\vartheta^n = \max\sqrt{\mathrm{var}\left(\tau^{n-1}\right)}$, $\xi = \phi_{\max}^2\sigma_{\max}\sqrt{|\mathcal{A}|-1}$.

Proof.

$$\begin{aligned}\|q_{\tilde{\omega}^n} - q_{\hat{\omega}^{n,m}}\|_\infty &\leq \|\operatorname{Proj}\mathrm{H}^n\tilde{q} - \operatorname{Proj}\mathrm{H}^n\hat{q}\|_\infty + \|\operatorname{Proj}\mathrm{H}^n\hat{q} - \hat{q}\|_\infty \\ &\leq \lambda\gamma^n\|\tilde{q} - \hat{q}\|_\infty + \|\operatorname{Proj}\mathrm{H}^n\hat{q} - \hat{q}\|_\infty.\end{aligned}$$

$$\|\operatorname{Proj}\mathrm{H}^n\hat{q} - \hat{q}\|_\infty \leq \phi_{\max}^2\sigma_{\max}|\mathbb{E}\left[\left(\mathrm{H}^n\hat{q}\right)(x,a) - \left(\tau^n\hat{q}\right)(x,a)\right]|$$

Using the previous lemma,

$$\begin{aligned}|\mathbb{E}\left[\left(\mathrm{H}^n\hat{q}\right)(x,a) - \left(\tau^n\hat{q}\right)(x,a)\right]| &= \mathbb{E}\left[\left(\tau^n\hat{q}\right)(x,a) - \left(\mathrm{H}^n\hat{q}\right)(x,a)\right] \\ &\leq \sqrt{(|\mathcal{A}|-1)}\sum_{i=1}^{n}\gamma^{n-i}\frac{\vartheta^i}{\sqrt{m}} \\ &\leq \sqrt{(|\mathcal{A}|-1)}\frac{\vartheta^n}{\sqrt{m}}\sum_{i=1}^{n}\gamma^{n-i} \\ &\leq \frac{\sqrt{(|\mathcal{A}|-1)}}{1-\gamma}\frac{\vartheta^n}{\sqrt{m}}\end{aligned}$$

where we used that, for all $i \leq n$

$$\vartheta^i \leq \vartheta^n$$

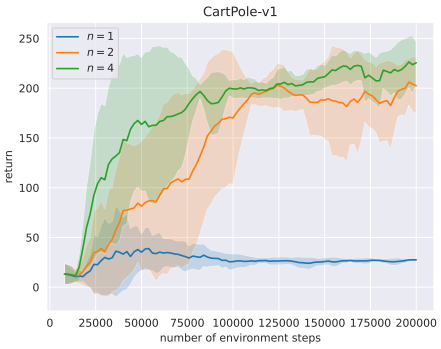
and

$$\sum_{i=1}^{n}\gamma^{n-i} = \frac{1-\gamma^{n-1}}{1-\gamma} \leq \frac{1}{1-\gamma}.$$

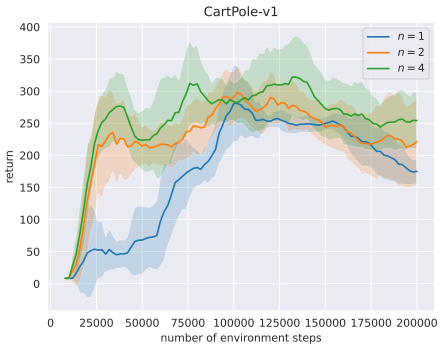
We conclude the result. □

D Additional experiments

In this appendix we provide additional results regarding the Carpole environment. We show results for features resulting from a finer discretization of the state space. Specifically, the each dimension of the state space is discretized in four, whereas in the results in the main text each dimension is discretized in two. We recall the results from the main text for comparison. We see that the formance of Q-learning ($n = 1$) improved the most. The performance of multi Q-learning ($n = 2$ and $n = 4$) also improves. Overall the performance of multi Q-learning is superior to Q-learning.



(a) Cartpole, with each dimension of the state space discretized in two.



(b) Cartpole, with each dimension of the state space discretized in four.

Figure 7: Cartpole results. The y-axis shows the average return. The performance increases with n.