Q(λ) with Off-Policy Corrections

Anna Harutyunyan[1]*, Marc G. Bellemare[2], Tom Stepleton[2], and Rémi Munos[2]

[1] VU Brussel
[2] Google DeepMind
`aharutyu@vub.ac.be`
`{bellemare,stepleton,munos}@google.com`

Abstract. We propose and analyze an alternate approach to off-policy multi-step temporal difference learning, in which off-policy returns are corrected with the current Q-function in terms of rewards, rather than with the target policy in terms of transition probabilities. We prove that such approximate corrections are sufficient for off-policy convergence both in policy evaluation and control, provided certain conditions. These conditions relate the distance between the target and behavior policies, the eligibility trace parameter and the discount factor, and formalize an underlying tradeoff in off-policy TD(λ). We illustrate this theoretical relationship empirically on a continuous-state control task.

1 Introduction

In reinforcement learning (RL), learning is off-policy when samples generated by a *behavior* policy are used to learn about a distinct *target* policy. The usual approach to off-policy learning is to disregard, or altogether discard transitions whose target policy probabilities are low. For example, Watkins's Q(λ) [21] cuts the trajectory backup as soon as a non-greedy action is encountered. Similarly, in policy evaluation, importance sampling methods [8] weight the returns according to the mismatch in the target and behavior probabilities of the corresponding actions. This approach treats transitions conservatively, and hence may unnecessarily terminate backups, or introduce a large amount of variance.

Many off-policy methods, in particular of the Monte Carlo kind, have no other option than to judge off-policy actions in the probability sense. However, *temporal difference* methods [14] in RL maintain an approximation of the value function along the way, with *eligiblity traces* [22] providing a continuous link between one-step and Monte Carlo approaches. The value function assesses actions in terms of the following expected cumulative reward, and thus provides a way to directly correct immediate *rewards*, rather than transitions. We show in this paper that such approximate corrections can be sufficient for off-policy convergence, subject to a tradeoff condition between the eligibility trace parameter and the distance between the target and behavior policies. The two extremes of this tradeoff are one-step Q-learning, and on-policy learning. Formalizing the continuum of the tradeoff is one of the main insights of this paper.

* This work was carried out during an internship at Google DeepMind.

In particular, we propose an off-policy return operator that augments the return with a correction term, based on the current approximation of the Q-function. We then formalize three algorithms stemming from this operator: (1) off-policy $\mathrm{Q}^\pi(\lambda)$, and its special case (2) *on*-policy $\mathrm{Q}^\pi(\lambda)$, for policy evaluation, and (3) $\mathrm{Q}^*(\lambda)$ for off-policy control.

In policy evaluation, both on- and off-policy $\mathrm{Q}^\pi(\lambda)$ are novel, but closely related to several existing algorithms of the $\mathrm{TD}(\lambda)$ family. Section 7 discusses this in detail. We prove convergence of $\mathrm{Q}^\pi(\lambda)$, subject to the $\lambda - \varepsilon$ tradeoff where $\varepsilon \overset{\text{def}}{=} \max_x \|\pi(\cdot|x) - \mu(\cdot|x)\|_1$ is a measure of dissimilarity between the behavior and target policies. More precisely, we prove that for any amount of "off-policy-ness" $\varepsilon \in [0, 2]$ there is an inherent maximum allowed backup length value $\lambda = \frac{1-\gamma}{\gamma\varepsilon}$, and taking λ below this value guarantees convergence to Q^π without involving policy probabilities. This is desirable due to the instabilities and variance introduced by the likelihood ratio products in the importance sampling approach [9].

In control, $\mathrm{Q}^*(\lambda)$ is in fact identical to Watkins's $\mathrm{Q}(\lambda)$, except it does not cut the eligiblity trace at off-policy actions. Sutton and Barto [16] mention such a variation, which they call *naive* $\mathrm{Q}(\lambda)$. We analyze this algorithm for the first time and prove its convergence for small values of λ. Although we were not able to prove a $\lambda - \varepsilon$ tradeoff similar to the policy evaluation case, we provide empirical evidence for the existence of such a tradeoff, confirming the intuition that naive $\mathrm{Q}(\lambda)$ is "not as naive as one might at first suppose" [16].

We first give the technical background, and define our operators. We then specify the incremental versions of our algorithms based on these operators, and state their convergence. We follow by proving convergence: subject to the $\lambda - \varepsilon$ tradeoff in policy evaluation, and more conservatively, for small values of λ in control. We illustrate the tradeoff emerge empirically in the Bicycle domain in the control setting. Finally, we conclude by placing our algorithms in context within existing work in $\mathrm{TD}(\lambda)$.

2 Preliminaries

We consider an environment modelled by the usual discrete-time Markov Decision Process $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ composed of the finite state and action spaces $\mathcal{X}$ and $\mathcal{A}$, a discount factor γ, a transition function P mapping each $(x, a) \in (\mathcal{X}, \mathcal{A})$ to a distribution over $\mathcal{X}$, and a reward function $r : \mathcal{X} \times \mathcal{A} \to [-R_{\text{MAX}}, R_{\text{MAX}}]$. A *policy* π maps a state $x \in \mathcal{X}$ to a distribution over $\mathcal{A}$. A Q-function Q is a mapping $\mathcal{X} \times \mathcal{A} \to \mathbb{R}$. Given a policy π, we define the operator P^π over Q-functions:

$$(P^\pi Q)(x, a) \overset{\text{def}}{=} \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P(x' \mid x, a)\pi(a' \mid x')Q(x', a').$$

To each policy π corresponds a unique Q-function Q^π which describes the expected discounted sum of rewards achieved when following π:

$$Q^\pi \overset{\text{def}}{=} \sum_{t \geq 0} \gamma^t (P^\pi)^t r, \tag{1}$$

where for any operator X, $(X)^t$ denotes t successive applications of X, and where we commonly treat r as one particular Q-function. We write the *Bellman operator* $\mathcal{T}^\pi$, and the *Bellman equation* for Q^π:

$$\begin{aligned} \mathcal{T}^\pi Q &\overset{\text{def}}{=} r + \gamma P^\pi Q, \\ \mathcal{T}^\pi Q^\pi &= Q^\pi = (I - \gamma P^\pi)^{-1} r. \end{aligned} \tag{2}$$

The *Bellman optimality operator* $\mathcal{T}$ is defined as $\mathcal{T}Q \overset{\text{def}}{=} r + \gamma \max_\pi P^\pi Q$, and it is well known [e.g. 1, 10] that the optimal Q-function $Q^* \overset{\text{def}}{=} \sup_\pi Q^\pi$ is the unique solution to the Bellman optimality equation

$$\mathcal{T}Q = Q. \tag{3}$$

We write $\textsc{Greedy}(Q) \overset{\text{def}}{=} \{\pi | \pi(a|x) > 0 \Rightarrow Q(x,a) = \max_{a'} Q(x,a')\}$ to denote the set of greedy policies w.r.t. Q. Thus $\mathcal{T}Q = \mathcal{T}^\pi Q$ for any $\pi \in \textsc{Greedy}(Q)$.

Temporal difference (TD) learning [14] rests on the fact that iterates of both operators $\mathcal{T}^\pi$ and $\mathcal{T}$ are guaranteed to converge to their respective fixed points Q^π and Q^*. Given a sample experience x, a, r, x', a', SARSA(0) [12] updates its Q-function estimate at k^{th} iteration as follows:

$$\begin{aligned} Q_{k+1}(x,a) &\leftarrow Q_k(x,a) + \alpha_k \delta, \\ \delta &= r + \gamma Q_k(x',a') - Q_k(x,a), \end{aligned}$$

where δ is the *TD-error*, and $(\alpha_k)_{k\in\mathbb{N}}$ a sequence of nonnegative stepsizes. One need not only consider short experiences, but may sample trajectories $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$, and accordingly apply $\mathcal{T}^\pi$ (or $\mathcal{T}$) repeatedly. A particularly flexible way of doing this is via a weighted sum A^λ of such *n-step* operators:

$$\begin{aligned} \mathcal{T}^\pi_\lambda Q &\overset{\text{def}}{=} A^\lambda[(\mathcal{T}^\pi)^{n+1} Q] \\ &= Q + (I - \lambda\gamma P^\pi)^{-1}(\mathcal{T}^\pi Q - Q), \\ A^\lambda[f(n)] &\overset{\text{def}}{=} (1-\lambda) \sum_{n\geq 0} \lambda^n f(n). \end{aligned}$$

Naturally, Q^π remains the fixed point of $\mathcal{T}^\pi_\lambda$. Taking $\lambda = 0$ yields the usual Bellman operator $\mathcal{T}^\pi$, and $\lambda = 1$ removes the recursion on the approximate Q-function, and restores Q^π in the *Monte Carlo* sense. It is well-known that λ trades off the bias from *bootstrapping* with an approximate Q-function, with the variance from using a sampled multi-step return [4], with intermediate values of λ usually performing best in practice [15, 13]. The above λ-operator can be efficiently implemented in the online setting via a mechanism called *eligibility traces.* As we will see in Section 7, it in fact corresponds to a number of online algorithms, each subtly different, of which SARSA(λ) [12] is the canonical instance.

Finally, we make an important distinction between the *target policy* π, which we wish to estimate, and the *behavior policy* μ, from which the actions have been generated. If $\mu = \pi$, the learning is said to be *on-policy*, otherwise it is *off-policy.* We will write $\mathbb{E}_\mu$ to denote expectations over sequences $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$, $a_i \sim \mu(\cdot|x_i)$, $x_{i+1} \sim P(\cdot|x_i, a_i)$ and assume conditioning on $x_0 = x$ and $a_0 = a$ wherever appropriate. Throughout, we will write $\|\cdot\|$ for supremum norm.

3 Off-Policy Return Operators

We will now describe the Monte Carlo *off-policy corrected return operator* $\mathcal{R}^{\pi,\mu}$ that is at the heart of our contribution. Given a target π, and a return generated by the behavior μ, the operator $\mathcal{R}^{\pi,\mu}$ attempts to approximate a return that would have been generated by π, by utilizing a correction built from a current approximation Q of Q^π. Its application to Q at a state-action pair (x, a) is defined as follows:

$$(\mathcal{R}^{\pi,\mu}Q)(x,a) \overset{\text{def}}{=} r(x,a) + \mathbb{E}_\mu\big[\sum_{t\geq 1}\gamma^t\big(r_t + \underbrace{\mathbb{E}_\pi Q(x_t,\cdot) - Q(x_t,a_t)}_{\text{off-policy correction}}\big)\big], \tag{4}$$

where we use the shorthand $\mathbb{E}_\pi Q(x,\cdot) \equiv \sum_{a\in\mathcal{A}} \pi(a|x)Q(x,a)$.

That is, $\mathcal{R}^{\pi,\mu}$ gives the usual expected discounted sum of future rewards, but each reward in the trajectory is augmented with an *off-policy correction*, which we define as the difference between the *expected* (with respect to the target policy) Q-value and the Q-value for the taken action. Thus, how much a reward is corrected is determined by both the approximation Q, and the target policy probabilities. Notice that if actions are similarly valued, the correction will have little effect, and learning will be roughly on-policy, but if the Q-function has converged to the correct estimates Q^π, the correction takes the immediate reward r_t to the expected reward with respect to π exactly. Indeed, as we will see later, Q^π is the fixed point of $\mathcal{R}^{\pi,\mu}$ for any behavior policy μ.

We define the n-step and λ-versions of $\mathcal{R}^{\pi,\mu}$ in the usual way:

$$\begin{aligned}\mathcal{R}^{\pi,\mu}_\lambda Q &\overset{\text{def}}{=} A^\lambda[\mathcal{R}^{\pi,\mu}_n], \qquad\qquad (5)\\ (\mathcal{R}^{\pi,\mu}_n Q)(x,a) &\overset{\text{def}}{=} r(x,a) + \mathbb{E}_\mu\big[\sum_{t=1}^{n}\gamma^t\big(r_t + \mathbb{E}_\pi Q(x_t,\cdot) - Q(x_t,a_t)\big)\\ &\qquad +\gamma^{n+1}\mathbb{E}_\pi Q(x_{n+1},\cdot)\big].\end{aligned}$$

Note that the λ parameter here takes us from TD(0) to the Monte Carlo version of our operator $\mathcal{R}^{\pi,\mu}$, rather than the traditional Monte Carlo form (1).

4 Algorithm

We consider the problems of *off-policy policy evaluation* and *off-policy control.* In both problems we are given data generated by a sequence of behavior policies $(\mu_k)_{k\in\mathbb{N}}$. In policy evaluation, we wish to estimate Q^π for a fixed target policy π. In control, we wish to estimate Q^*. Our algorithm constructs a sequence $(Q_k)_{k\in\mathbb{N}}$ of estimates of Q^{π_k} from trajectories sampled from μ_k, by applying the $\mathcal{R}^{\pi_k,\mu_k}_\lambda$-operator:

$$Q_{k+1} = \mathcal{R}^{\pi_k,\mu_k}_\lambda Q_k, \tag{6}$$

where π_k is the k^{th} interim target policy. We distinguish between three algorithms:

Algorithm 1 Q(λ) with off-policy corrections

```
Given: Initial $Q$-function $Q_0$, stepsizes $(\alpha_k)_{k\in\mathbb{N}}$
  for $k = 1 \dots$ do
    Sample a trajectory $x_0, a_0, r_0, \dots, x_{T_k}$ from $\mu_k$
    $Q_{k+1}(x, a) \leftarrow Q_k(x, a) \quad \forall x, a$
    $e(x, a) \leftarrow 0 \quad \forall x, a$
    for $t = 0 \dots T_k - 1$ do
      $\delta_t^{\pi_k} \leftarrow r_t + \gamma \mathbb{E}_{\pi_k} Q_{k+1}(x_{t+1}, \cdot) - Q_{k+1}(x_t, a_t)$
      for all $x \in \mathcal{X}, a \in \mathcal{A}$ do
        $e(x, a) \leftarrow \lambda\gamma e(x, a) + \mathbb{I}\{(x_t, a_t) = (x, a)\}$
        $Q_{k+1}(x, a) \leftarrow Q_{k+1}(x, a) + \alpha_k \delta_t^{\pi_k} e(x, a)$
      end for
    end for
  end for
```

On-policy $\mathbf{Q}^\pi(\lambda)$: $\mu_k = \pi_k = \pi$.
Off-policy $\mathbf{Q}^\pi(\lambda)$: $\mu_k \neq \pi_k = \pi$.
$\mathbf{Q}^*(\lambda)$: $\pi_k \in \textsc{Greedy}(Q_k)$.

Off-policy $\mathbf{Q}^\pi(\lambda)$ for policy evaluation: $\pi_k = \pi$ is the fixed target policy. We write the corresponding operator $\mathcal{R}_\lambda^\pi$.
On-policy $\mathbf{Q}^\pi(\lambda)$ for policy evaluation: for the special case of $\mu_k = \mu = \pi$.
$\mathbf{Q}^*(\lambda)$ for off-policy control: $(\pi_k)_{k\in\mathbb{N}}$ is a sequence of greedy policies with respect to Q_k. We write the corresponding operator $\mathcal{R}_\lambda^*$.

We wish to write the update (6) in terms of a simulated trajectory $x_0, a_0, r_0, \dots, x_{T_k}$ drawn according to μ_k. First, notice that (5) can be rewritten:

$$\mathcal{R}_\lambda^{\pi,\mu} Q(x, a) = Q(x, a) + \mathbb{E}_\mu \big[\sum_{t \geq 0} (\lambda\gamma)^t \delta_t^\pi \big],$$
$$\delta_t^\pi \stackrel{\text{def}}{=} r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t),$$

where δ_t^π is the *expected* TD-error. The *offline* forward view[1] is then

$$Q_{k+1}(x, a) \leftarrow Q_k(x, a) + \alpha_k \sum_{t=0}^{T_k} (\gamma\lambda)^t \delta_t^{\pi_k}, \tag{7}$$

While (7) resembles many existing TD(λ) algorithms, it subtly differs from all of them, due to $\mathcal{R}_\lambda^{\pi,\mu}$ (rather than $\mathcal{T}_\lambda^\pi$) being at its basis. Section 7 discusses the distinctions in detail. The practical *every-visit* [16] form of (7) is written

$$Q_{k+1}(x, a) \leftarrow Q_k(x, a) + \alpha_k \sum_{t=0}^{T} \delta_t^{\pi_k} \sum_{s=0}^{t} (\gamma\lambda)^{t-s} \mathbb{I}\{(x_s, a_s) = (x, a)\}, \tag{8}$$

[1] The true online version can be derived as given by van Seijen and Sutton [19]

and the corresponding online backward view of all three algorithms is summarized in Algorithm 1.

The following theorem states that when μ and π are sufficiently close, the off-policy Q$^\pi(\lambda)$ algorithm converges to its fixed point Q^π.

Theorem 1. *Consider the sequence of Q-functions computed according to Algorithm 1 with fixed policies μ and π. Let $\varepsilon = \max_x \|\pi(\cdot|x) - \mu(\cdot|x)\|_1$. If $\lambda\varepsilon < \frac{1-\gamma}{\gamma}$, then under the same conditions required for the convergence of TD(λ) (1–3 in Section 5.3) we have, almost surely:*

$$\lim_{k\to\infty} Q_k(x,a) = Q^\pi(x,a).$$

We state a similar, albeit weaker result for Q$^*(\lambda)$.

Theorem 2. *Consider the sequence of Q-functions computed according to Algorithm 1 with π_k the greedy policy with respect to Q_k. If $\lambda < \frac{1-\gamma}{2\gamma}$, then under the same conditions required for the convergence of TD(λ) (1–3 in Section 5.3) we have, almost surely:*

$$\lim_{k\to\infty} Q_k(x,a) = Q^*(x,a).$$

The proofs of these theorems rely on showing that $\mathcal{R}^\pi_\lambda$ and $\mathcal{R}^*_\lambda$ are contractions (under the stated conditions), and invoking classical stochastic approximation convergence to their fixed point (such as Proposition 4.5 from [2]). We will focus on the contraction lemmas, which are the crux of the proofs, then outline the sketch of the online convergence argument.

Discussion Theorem 1 states that for *any* $\lambda \in [0,1]$ there exists some degree of "off-policy-ness" $\varepsilon < \frac{1-\gamma}{\lambda\gamma}$ under which Q_k converges to Q^π. This is the $\lambda - \varepsilon$ tradeoff for the off-policy $Q^\pi(\lambda)$ learning algorithm for policy evaluation. In the control case, the result of Theorem 2 is weaker as it only holds for values of λ smaller than $\frac{1-\gamma}{2\gamma}$. Notice that this threshold corresponds to the policy evaluation case for $\varepsilon = 2$ (arbitrary off-policy-ness). We were not able to prove convergence to Q^* for any $\lambda \in [0,1]$ and some $\varepsilon > 0$. This is left as an open problem for now.

The main technical difficulty lies in the fact that in control, the greedy policy with respect to the current Q_k may change drastically from one step to the next, while Q_k itself changes incrementally (under small learning steps α_k). So the current Q_k may not offer a good off-policy correction to evaluate the new greedy policy. In order to circumvent this problem we may want to use slowly changing target policies π_k. For example we could keep π_k fixed for slowly increasing periods of time. This can be seen as a form of optimistic policy iteration [10] where policy improvement steps alternate with approximate policy evaluation steps (and when the policy is fixed, Theorem 1 guarantees convergence to the value function of that policy). Another option would be to define π_k as the empirical average $\pi_k \overset{\text{def}}{=} \frac{1}{k}\sum_{i=1}^k \pi'_i$ of the previous greedy policies π'_i. We conjecture that defining π_k such that (1) π_k changes slowly with k, and (2) π_k becomes increasingly greedy, then we could extend the $\lambda - \varepsilon$ tradeoff of Theorem 1 to the control case. This is left for future work.

5 Analysis

We begin by verifying that the fixed points of $\mathcal{R}_\lambda^{\pi,\mu}$ in the policy evaluation and control settings are Q^π and Q^*, respectively. We then prove the contractive properties of these operators: $\mathcal{R}_\lambda^\pi$ is always a contraction and will converge to its fixed point, $\mathcal{R}_\lambda^*$ is a contraction for particular choices of λ (given in terms of γ). The contraction coefficients depend on λ, γ, and ε: the distance between policies. Finally, we give a proof sketch for online convergence of Algorithm 1.

Before we begin, it will be convenient to rewrite (4) for all state-action pairs:

$$\mathcal{R}^{\pi,\mu}Q = r + \sum_{t\geq 1}\gamma^t (P^\mu)^{t-1}[P^\mu r + P^\pi Q - P^\mu Q].$$

We can then write $\mathcal{R}_\lambda^\pi$ and $\mathcal{R}_\lambda^*$ from (5) as follows:

$$\mathcal{R}_\lambda^\pi Q \overset{\text{def}}{=} Q + (I - \lambda\gamma P^\mu)^{-1}[\mathcal{T}^\pi Q - Q], \tag{9}$$

$$\mathcal{R}_\lambda^* Q \overset{\text{def}}{=} Q + (I - \lambda\gamma P^\mu)^{-1}[\mathcal{T} Q - Q]. \tag{10}$$

It is not surprising that the above along with the Bellman equations (2) and (3) directly yields that Q^π and Q^* are the fixed points of $\mathcal{R}_\lambda^\pi$ and $\mathcal{R}_\lambda^*$:

$$\mathcal{R}_\lambda^\pi Q^\pi = Q^\pi,$$
$$\mathcal{R}_\lambda^* Q^* = Q^*.$$

It then remains to analyze the behavior of $\mathcal{R}_\lambda^{\pi,\mu}$ as it gets iterated.

5.1 λ-return for policy evaluation: $\mathbf{Q}^\pi(\lambda)$

We first consider the case with a fixed arbitrary policy π. For simplicity, we take μ to be fixed as well, but the same will hold for any sequence $(\mu_k)_{k\in\mathbb{N}}$, as long as each μ_k satisfies the condition imposed on μ.

Lemma 1. *Consider the policy evaluation algorithm $Q_k = (\mathcal{R}_\lambda^\pi)^k Q$. Assume the behavior policy μ is ε-away from the target policy π, in the sense that $\max_x \|\pi(\cdot|x) - \mu(\cdot|x)\|_1 \leq \varepsilon$. Then for $\varepsilon < \frac{1-\gamma}{\lambda\gamma}$, the sequence $(Q_k)_{k\geq 1}$ converges to Q^π exponentially fast: $\|Q_k - Q^\pi\| = O(\eta^k)$, where $\eta = \frac{\gamma}{1-\lambda\gamma}(1-\lambda+\lambda\varepsilon) < 1$.*

Proof. First notice that

$$\begin{aligned}\|P^\pi - P^\mu\| &= \sup_{\|Q\|\leq 1}\|(P^\pi - P^\mu)Q\| \\ &= \sup_{\|Q\|\leq 1}\max_{x,a}\Big|\sum_y P(y|x,a)\sum_b \left((\pi(b|y) - \mu(b|y)\right) Q(y,b)\Big| \\ &\leq \max_{x,a}\sum_y P(y|x,a)\sum_b |\pi(b|y) - \mu(b|y)| \leq \varepsilon.\end{aligned}$$

Let $B = (I - \lambda\gamma P^\mu)^{-1}$ be the resolvent matrix. From (9) we have

$$\begin{aligned}\mathcal{R}^{\pi}_{\lambda}Q - Q^{\pi} &= B\big[\mathcal{T}^{\pi}Q - Q + (I - \lambda\gamma P^{\mu})(Q - Q^{\pi})\big]\\ &= B\big[r + \gamma P^{\pi}Q - Q^{\pi} - \lambda\gamma P^{\mu}(Q - Q^{\pi})\big]\\ &= B\big[\gamma P^{\pi}(Q - Q^{\pi}) - \lambda\gamma P^{\mu}(Q - Q^{\pi})\big]\\ &= \gamma B\big[(1-\lambda)P^{\pi} + \lambda(P^{\pi} - P^{\mu})\big](Q - Q^{\pi}).\end{aligned}$$

Taking the sup norm, since μ is ε-away from π:

$$\|\mathcal{R}^{\pi}_{\lambda}Q - Q^{\pi}\| \leq \eta\|Q - Q^{\pi}\|$$

for $\eta = \frac{\gamma}{1-\lambda\gamma}(1 - \lambda + \lambda\varepsilon) < 1$. Thus $\|Q_k - Q^{\pi}\| = O(\eta^k)$.

5.2 $\boldsymbol{\lambda}$-return for control: $\mathbf{Q^*(\boldsymbol{\lambda})}$

We next consider the case where the k^{th} target policy π_k is greedy with respect to the value estimate Q_k. The following Lemma states that is possible to select a small, but nonzero λ and still guarantee convergence.

Lemma 2. *Consider the off-policy control algorithm $Q_k = (\mathcal{R}^*_{\lambda})^k Q$. Then*

$$\|\mathcal{R}^*_{\lambda}Q_k - Q^*\| \leq \frac{\gamma + \lambda\gamma}{1 - \lambda\gamma}\|Q_k - Q^*\|,$$

and for $\lambda < \frac{1-\gamma}{2\gamma}$ the sequence $(Q_k)_{k\geq 1}$ converges to Q^* exponentially fast.

Proof. Fix μ and let $B = (I - \lambda\gamma P^{\mu})^{-1}$. Using (10), we write

$$\begin{aligned}\mathcal{R}^*_{\lambda}Q - Q^* &= B\left[\mathcal{T}Q - Q + (I - \lambda\gamma P^{\mu})(Q - Q^*)\right]\\ &= B\left[\mathcal{T}Q - Q^* - \lambda\gamma P^{\mu}(Q - Q^*)\right].\end{aligned}$$

Taking the sup-norm, since $\|\mathcal{T}Q - Q^*\| \leq \gamma\|Q - Q^*\|$, we deduce the result:

$$\|\mathcal{R}^*_{\lambda}Q - Q^*\| \leq \frac{\gamma + \lambda\gamma}{1 - \lambda\gamma}\|Q - Q^*\|.$$

5.3 Online Convergence

We are now ready to prove the online convergence of Algorithm 1. Let the following hold for every sample trajectory τ_k and all $x \in \mathcal{X}, a \in \mathcal{A}$:

1. **Minimum visit frequency:** $\sum_{t\geq 0}\mathbb{P}\{x_t, a_t = x, a\} \geq D > 0$.
2. **Finite trajectories:** $\mathbb{E}_{\mu_k}T_k^2 < \infty$, where T_k is the length of τ_k.
3. **Bounded stepsizes:** $\sum_{k\geq 0}\alpha_k(x,a) = \infty$, $\sum_{k\geq 0}\alpha_k^2(x,a) < \infty$.

Assumption 2 requires trajectories to be finite w.p. 1, which is satisfied by *proper* behavior policies. Equivalently, we may require from the MDP that all trajectories eventually reach a zero-value absorbing state. The proof closely follows that of Proposition 5.2 from [2], and requires rewriting the update in the suitable form, and verifying Assumptions (a) through (d) from their Proposition 4.5.

Proof. (Sketch) Let $z_{k,t}(x,a) \overset{\text{def}}{=} \sum_{s=0}^{t} (\gamma\lambda)^{t-s} \mathbb{I}\{(x_s, a_s) = (x,a)\}$ denote the accumulating trace. It follows from Assumptions 1 and 2 that the total update at phase k is bounded, which allows us to write the online version of (8) as

$$Q^o_{k+1}(x,a) \leftarrow (1 - D_k\alpha_k)Q^o_k(x,a) + D_k\alpha_k\big(\mathcal{R}^{\pi_k,\mu_k}_{\lambda} Q^o_k(x,a) + w_k + u_k\big)$$

$$w_k \overset{\text{def}}{=} (D_k)^{-1}\Big[\sum_{t\geq 0} z_{k,t}\delta^{\pi_k}_t - \mathbb{E}_{\mu_k}\big[\sum_{t\geq 0} z_{k,t}\delta^{\pi_k}_t\big]\Big],$$

$$u_k \overset{\text{def}}{=} (D_k\alpha_k)^{-1}\big(Q^o_{k+1}(x,a) - Q_{k+1}(x,a)\big),$$

where $D_k(x,a) \overset{\text{def}}{=} \sum_{t\geq 0} \mathbb{P}\{x_t, a_t = x, a\}$, and we use the shorthand $y_k \equiv y_k(x,a)$ for α_k, D_k, w_k, u_k, and $z_{k,t}$. Combining Assumptions 1 and 2, we have $0 < D \leq D_k(x,a) < \infty$, which, combined in turn with Assumption 3, assures that the new stepsize sequence $\tilde{\alpha}_k(x,a) = (D_k\alpha_k)(x,a)$ satisfies Assumption (a) of Prop. 4.5. Assumptions (b) and (d) require the variance of the noise term $w_k(x,a)$ to be bounded, and the residual $u_k(x,a)$ to converge to zero, both of which can be shown identically to the corresponding results from [2], if Assumption 2 and Assumption (a) are satisfied. Finally, Assumption (c) is satisfied by Lemmas 1 and 2 for the policy evaluation and control cases, respectively.[2] We conclude that the sequence $(Q^o_k)_{k\in\mathbb{N}}$ converges to Q^π or Q^* in the respective settings, w.p. 1.

6 Experimental Results

Although we do not have a proof of the $\lambda - \varepsilon$ tradeoff (see Section 4) in the control case, we wished to investigate whether such a tradeoff can be observed experimentally. To this end, we applied Q$^*(\lambda)$ to the Bicycle domain [11]. Here, the agent must simultaneously balance a bicycle and drive it to a goal position. Six real-valued variables describe the state – angle, velocity, etc. – of the bicycle. The reward function is proportional to the angle to the goal, and gives -1 for falling and +1 for reaching the goal. The discount factor is 0.99. The Q-function was approximated using multilinear interpolation over a uniform grid of size $10 \times \cdots \times 10$, and the stepsize was tuned to 0.1. We are chiefly interested in the interplay between the λ parameter in Q$^*(\lambda)$ and an ε-greedy exploration policy. Our main performance indicator is the frequency at which the goal is reached by the greedy policy after 500,000 episodes of training. We report three findings:

1. Higher values of λ lead to improved learning;
2. Very low values of ε exhibit lower performance; and
3. The Q-function diverges when λ is high relative to ε.

Together, these findings suggest that there is indeed a $\lambda - \varepsilon$ tradeoff in the control case as well, and lead us to conclude that with proper care it can be beneficial to do off-policy control with Q$^*(\lambda)$.

[2] Note that the control case goes through without modifications, for the values of λ prescribed by Lemma 2.

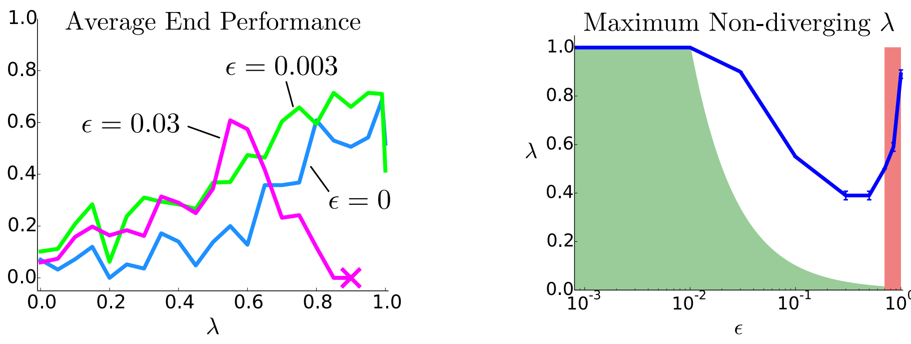

Fig. 1. Left. Performance of Q$^*(\lambda)$ on the Bicycle domain. Each configuration is an average of five trials. The 'X' marks the lowest value of λ for which $\varepsilon = 0.03$ causes divergence. **Right.** Maximum non-diverging λ in function of ε. The left-hand shaded region corresponds to our hypothesized bound. Parameter settings in the right-hand shaded region do not produce meaningful policies.

Learning speed and performance. Figure 1 (left) depicts the performance of Q$^*(\lambda)$, in terms of the goal-reaching frequency, for three values of ε. The agent performs best ($p < 0.05$) for $\varepsilon \in [0.003, 0.03]$ and high (w.r.t. ε) values of λ.[3]
Divergence. For each value of ε, we determined the highest *safe* choice of λ which did not result in divergence. As Figure 1 (right) illustrates, there is a marked decrease in what is a safe value of λ as ε increases. Note the left-hand shaded region corresponding to the *policy evaluation* bound $\frac{1-\gamma}{\gamma\varepsilon}$. Supporting our hypothesis on the true bound on λ (Section 5), it appears clear that the maximum safe value of λ depends on ε. In particular, notice how $\lambda = 1$ stops diverging exactly where predicted by this bound.

7 Related Work

In this section, we place the presented algorithms in context of the existing work in TD(λ) [16], focusing in particular on action-value methods. As usual, let $(x_t, a_t, r_t)_{t\geq 0}$ be a trajectory generated by following a behavior policy μ, i.e. $a_t \sim \mu(\cdot|x_t)$. At time s, SARSA(λ) [12] updates its Q-function as follows:

$$Q_{s+1}(x_s, a_s) \leftarrow Q_s(x_s, a_s) + \alpha_s(\underbrace{A^\lambda R_s^{(n)} - Q(x_s, a_s)}_{\Delta_s}), \tag{11}$$

$$R_s^{(n)} = \sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1}), \tag{12}$$

where Δ_s denotes the update made at time s, and can be rewritten in terms of one-step TD-errors:

[3] Recall that Randløv and Alstrøm's agent was trained using SARSA(λ) with $\lambda = 0.95$.

$$\Delta_s = \sum_{t \geq s} (\lambda\gamma)^{t-s} \delta_t, \tag{13}$$
$$\delta_t = r_t + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t).$$

SARSA(λ) is an on-policy algorithm and converges to the value function Q^μ of the behavior policy. Different algorithms arise by instantiating $R_s^{(n)}$ or Δ_s from (11) differently. Table 1 provides the full details, while in text we will specify the most revealing components of the update.

7.1 Policy Evaluation

One can imagine considering *expectations* over action-values at the corresponding states $\mathbb{E}_\pi Q(x_t, \cdot)$, in place of the value of the sampled action $Q(x_t, a_t)$, i.e.:

$$\delta_t = r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - \mathbb{E}_\pi Q(x_t, \cdot). \tag{14}$$

This is the one-step update for *General Q-Learning* [18], which is a generalization of *Expected SARSA* [20] to arbitrary policies. We refer to the direct eligibility trace extensions of these algorithms formed via Equations (11)-(13) by General Q(λ) and Expected SARSA(λ) (first mentioned by Sutton et al. [17]) Unfortunately, in an off-policy setting, General Q(λ) will not converge to the value function Q^π of the target policy, as stated by the following proposition.

Proposition 1. *The stable point of General Q(λ) is* $Q^{\mu,\pi} = (I - \lambda\gamma(P^\mu - P^\pi) - \gamma P^\pi)^{-1} r$ *which is the fixed point of the operator* $(1-\lambda)\mathcal{T}^\pi + \lambda\mathcal{T}^\mu$.

Proof. Writing the algorithm in operator form, we get

$$\begin{aligned}\mathcal{R}Q &= (1-\lambda) \sum_{n \geq 0} \lambda^n \Big[\sum_{t=0}^{n} \gamma^t (P^\mu)^t r + \gamma^{n+1} (P^\mu)^n P^\pi Q \Big] \\ &= \sum_{t \geq 0} (\lambda\gamma)^t (P^\mu)^t \Big[r + (1-\lambda)\gamma P^\pi Q \Big] = (I - \lambda\gamma P^\mu)^{-1} \Big[r + (1-\lambda)\gamma P^\pi Q \Big].\end{aligned}$$

Thus the fixed point $Q^{\mu,\pi}$ of $\mathcal{R}$ satisfies the following:

$$Q^{\mu,\pi} = (I - \lambda\gamma P^\mu)^{-1} \Big[r + (1-\lambda)\gamma P^\pi Q^{\mu,\pi} \Big] = (1-\lambda)\mathcal{T}^\pi Q^{\mu,\pi} + \lambda \mathcal{T}^\mu Q^{\mu,\pi}.$$

Solving for $Q^{\mu,\pi}$ yields the result.

Alternatively to replacing both terms with an expectation, one may only replace the value at the *next* state x_{t+1} by $\mathbb{E}_\pi Q(x_{t+1}, \cdot)$, obtaining:

$$\delta_t^\pi = r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t). \tag{15}$$

This is exactly our policy evaluation algorithm Q$^\pi$(λ). Specifically, when $\pi = \mu$, we get the on-policy Q$^\pi$(λ). The induced *on-policy* correction may serve as a variance reduction term for Expected SARSA(λ) (it may be helpful to refer to the n-step return in Table 1 to observe this), but we leave variance analysis of this algorithm for future work. When $\pi \neq \mu$, we recover off-policy Q$^\pi$(λ), which (under the stated conditions) converges to Q^π.

Target Policy Probability Methods: The algorithms above directly descend from basic SARSA(λ), but often learning off-policy requires special treatment. For example, a typical off-policy technique is importance sampling (IS) [9]. It is a classical Monte Carlo method that allows one to sample from the available distribution, but obtain (unbiased or consistent) samples of the desired one, by reweighing the samples with their likelihood ratio according to the two distributions. That is, the updates for the ordinary *per-decision* IS algorithm for policy evaluation are made as follows:

$$\begin{aligned}\mathit{\Delta}_s &= \sum_{t\geq s}(\lambda\gamma)^{t-s}\delta_t \prod_{i=s+1}^{t}\frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}\\ \delta_t &= r_t + \gamma\frac{\pi(a_{t+1}|s_{t+1})}{\mu(a_{t+1}|s_{t+1})}Q(x_{t+1},a_{t+1}) - Q(x_t,a_t).\end{aligned}$$

This family of algorithms converges to Q^π with probability 1, under any soft, stationary behavior μ [8]. There are several (recent) off-policy algorithms that reduce the variance of IS methods, at the cost of added bias [5, 6, 3].

However, off-policy Q$^\pi$(λ) is perhaps related closest to the *Tree-Backup (TB) algorithm*, also discussed by Precup et al. [8]. Its one-step TD-error is the same as (15), the algorithms back up the same tree, and neither requires knowledge of the behavior policy μ. The important difference is in the weighting of the updates. As an off-policy precaution, TB(λ) weighs updates along a trajectory with the cumulative target probability of that trajectory up until that point:

$$\mathit{\Delta}_s = \sum_{t\geq s}(\lambda\gamma)^{t-s}\delta_t^\pi \prod_{i=s+1}^{t}\pi(a_i|x_i). \tag{16}$$

The weighting simplifies the convergence argument, allowing TB(λ) to converge to Q^π without further restrictions on the distance between μ and π [8]. The drawback of TB(λ) is that in the case of near on-policy-ness (when μ is close to π) the product of the probabilities cuts the traces unnecessarily (especially when the policies are stochastic). What we show in this paper, is that plain TD-learning *can* converge off-policy with no special treatment, subject to a tradeoff condition on λ and ε. Under that condition, Q$^\pi$(λ) applies both on- and off-policy, without modifications. An ideal algorithm should be able to automatically cut the traces (like TB(λ)) in case of extreme off-policy-ness while reverting to Q$^\pi$(λ) when being near on-policy.

7.2 Control

Perhaps the most popular version of Q(λ) is due to Watkins and Dayan [21]. Off-policy, it truncates the return and bootstraps as soon as the behavior policy takes a non-greedy action, as described by the following update:

$$\mathit{\Delta}_s = \sum_{t=s}^{s+\tau}(\lambda\gamma)^{t-s}\delta_t, \tag{17}$$

where $\tau = \min\{u \geq 1 : a_{s+u} \notin \arg\max_a Q(x_{s+u}, a)\}$. Note that this update is a special case of (16) for deterministic greedy policies, with $\prod_{i=s+1}^{t} \mathbb{I}\{a_i \in \arg\max_a Q(x_i, a)\}$ replacing the probability product. When the policies μ and π are not too similar, and λ is not too small, the truncation may greatly reduce the benefit of complex backups.

Q(λ) of Peng and Williams [7] is meant to remedy this, by being a hybrid between SARSA(λ) and Watkins's Q(λ). Its n-step return $\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$ requires the following form for the TD-error:

$$\delta_t = r(x_t, a_t) + \gamma \max_a Q(x_{t+1}, a) - \max_a Q(x_t, a).$$

This is, in fact, the same update rule as the General Q(λ) defined in (14), where π is the greedy policy. Following the same steps as in the proof of Proposition 1, the limit of this algorithm (if it converges) will be the fixed point of the operator $(1-\lambda)\mathcal{T} + \lambda\mathcal{T}^{\mu}$ which is different from Q^* unless the behavior is always greedy.

Sutton and Barto [16] mention another, *naive* version of Watkins's Q(λ) that does not cut the trace on non-greedy actions. That is exactly the Q*(λ) algorithm described in this paper. Notice that despite the similarity to Watkins's Q(λ), the equivalence representation for Q*(λ) is different from the one that would be derived by setting $\tau = \infty$ in (17), since the n-step return uses the *corrected* immediate reward $r_t + \gamma \max_a Q(x_t, a) - Q(x_t, a_t)$ instead of the immediate reward alone. This correction is invisible in Watkins's Q(λ), since the behavior policy is assumed to be greedy, before the return is cut off.

8 Conclusion

We formulated new algorithms of the TD(λ) family for off-policy policy evaluation and control. Unlike traditional off-policy learning algorithms, these methods do not involve weighting returns by their policy probabilities, yet under the right conditions converge to the correct TD fixed points. In policy evaluation, convergence is subject to a tradeoff between the degree of bootstrapping λ, distance between policies ε, and the discount factor γ. In control, determining the existence of a non-trivial ε-dependent bound for λ remains an open problem. Supported by telling empirical results in the Bicycle domain, we hypothesize that such a bound exists, and closely resembles the $\frac{1-\gamma}{\gamma\varepsilon}$ bound from the policy evaluation case.

Acknowledgements

The authors thank Hado van Hasselt and others at Google DeepMind, as well as the anonymous reviewers for their thoughtful feedback on the paper.

References

1. Richard Bellman. *Dynamic Programming.* Princeton University Press, 1957.

2. Dimitry P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
3. Assaf Hallak, Aviv Tamar, Rémi Munos, and Shie Mannor. Generalized emphatic temporal difference learning: Bias-variance analysis. *arXiv:1509.05172*, 2015.
4. Michael J. Kearns and Satinder P. Singh. Bias-variance error bounds for temporal difference updates. In *Conference on Computational Learning Theory*, pages 142–147, 2000.
5. Ashique R. Mahmood and Richard S. Sutton. Off-policy learning based on weighted importance sampling with linear computational complexity. In *Conference on Uncertainty in Artificial Intelligence*, 2015.
6. Ashique R. Mahmood, Huizhen Yu, Martha White, and Richard S. Sutton. Emphatic temporal-difference learning. *arXiv preprint arXiv:1507.01569*, 2015.
7. Jing Peng and Ronald J. Williams. Incremental multi-step q-learning. *Machine Learning*, 22(1-3):283–290, 1996.
8. Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*, 2000.
9. Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *International Conference on Machine Learning*, 2001.
10. Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.
11. Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *International Conference on Machine Learning*, 1998.
12. Gavin A. Rummery and Mahesan Niranjan. On-line q-learning using connectionist systems. Technical report, Cambridge University Engineering Department., 1994.
13. Satinder Singh and Peter Dayan. Analytical mean squared error curves for temporal difference learning. *Machine Learning*, 32(1):5–40, 1998.
14. Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.
15. Richard S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, 1996.
16. Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.
17. Richard S. Sutton, Ashique R. Mahmood, Doina Precup, and Hado van Hasselt. A new q (λ) with interim forward view and monte carlo equivalence. In *International Conference on Machine Learning*, pages 568–576, 2014.
18. Hado Philip van Hasselt. *Insights in Reinforcement Learning: formal analysis and empirical evaluation of temporal-difference learning algorithms*. PhD thesis, Universiteit Utrecht, January 2011.
19. Harm van Seijen and Richard S. Sutton. True online TD(λ). In *International Conference on Machine Learning*, pages 692–700, 2014.
20. Harm van Seijen, Hado van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected sarsa. In *Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184. IEEE, 2009.
21. Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8 (3):272–292, 1992.
22. Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

Table 1. Comparison of the update rules of several learning algorithms using the λ-return. We show both the n-step return and the resulting update rule for the λ-return from any state x_s when following a behavior policy $a_t \sim \mu(\cdot|x_t)$. **Top part. Policy evaluation algorithms:** SARSA(λ), Expected SARSA(λ), General Q(λ), Per-Decision Importance Sampling (PDIS(λ)), TB(λ), and Q$^\pi$(λ), in both on-policy (i.e. $\pi = \mu$) and off-policy settings (with a target policy $\pi \neq \mu$). Note the same $Q^\pi(\lambda)$ equation applies to both on- and off-policy settings. We abbreviate $\pi_i \equiv \pi(a_i|x_i)$, $\mu_i \equiv \mu(a_i|x_i)$, $\rho_i \equiv \pi_i/\mu_i$, and write $\mathbb{E}_\pi^{a\neq b} Q(x,\cdot) \equiv \sum_{a\in\mathcal{A}\setminus b} \pi(a|x)Q(x,a)$. **Bottom part, control algorithms:** Watkins's $Q(\lambda)$, Peng and Williams's $Q(\lambda)$, and $Q^*(\lambda)$. The **FP** column denotes the stable point of these algorithms (i.e. the fixed point of the expected update), regardless of whether the algorithm converges to it. General Q(λ) may converge to $Q^{\mu,\pi}$ defined as the fixed point of the Bellman operator $(1-\lambda)\mathcal{T}^\pi + \lambda\mathcal{T}^\mu$. The fixed point of Watkins's Q(λ) is Q^* but the case $\lambda > 0$ may not be significantly better than $\lambda = 0$ (regular Q-learning) if the behavior policy is different from the greedy one. The fixed point $Q^{\mu,*}$ of Peng and Williams's $Q(\lambda)$ is the fixed point of $(1-\lambda)\mathcal{T} + \lambda\mathcal{T}^\mu$, which is different from Q^* when $\mu \neq \pi$ (see Proposition 1). The algorithms analyzed in this paper are $Q^\pi(\lambda)$ and $Q^*(\lambda)$, for which convergence to respectively Q^π and Q^* occurs under some conditions (see Lemmas).

Algorithm	***n*-step return**	**Update rule for the λ-return**	**FP**
TD(λ) (on-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} V(x_{s+n+1})$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t$ $\delta_t = r_t + \gamma V(x_{t+1}) - V(x_t)$	V^μ
SARSA(λ) (on-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1})$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t$ $\delta_t = r_t + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)$	Q^μ
$\mathbb{E}$ SARSA(λ) (on-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \mathbb{E}_\mu Q(x_{s+n+1}, \cdot)$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t + \mathbb{E}_\mu Q(x_s, \cdot) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma \mathbb{E}_\mu Q(x_{t+1}, \cdot) - \mathbb{E}_\mu Q(x_t, \cdot)$	Q^μ
General Q(λ) (off-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \mathbb{E}_\pi Q(x_{s+n+1}, \cdot)$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t + \mathbb{E}_\pi Q(x_s, \cdot) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - \mathbb{E}_\pi Q(x_t, \cdot)$	$Q^{\mu,\pi}$
PDIS(λ) (off-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t \prod_{i=s+1}^{t} \rho_i$ $+\gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1}) \prod_{i=s}^{s+n} \rho_i$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t \prod_{i=s+1}^{t} \rho_i$ $\delta_t = r_t + \gamma \rho_{t+1} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)$	Q^π
TB(λ) (off-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} \prod_{i=s+1}^{t} \pi_i \big[r_t + \gamma \mathbb{E}_\pi^{a\neq a_{t+1}} Q(x_{t+1}, \cdot)\big]$ $+\gamma^{n+1} \prod_{i=s+1}^{s+n+1} \pi_i Q(x_{s+n+1}, a_{s+n+1})$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t \prod_{i=s+1}^{t} \pi_i$ $\delta_t = r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t)$	Q^π
Q$^\pi$(λ) (on/off-policy)	$\sum_{t=s}^{s+n} \gamma^{t-s} \big[r_t + \mathbb{E}_\pi Q(x_t, \cdot) - Q(x_t, a_t)\big]$ $+\gamma^{n+1} \mathbb{E}_\pi Q(x_{s+n+1}, \cdot)$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t$ $\delta_t = r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t)$	Q^π
Q(λ) (Watkins's)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$ (for any $n < \tau = \arg\min_{u\geq 1} \mathbb{I}\{\pi_{s+u} \neq \mu_{s+u}\}$)	$\sum_{t=s}^{s+\tau} (\lambda\gamma)^{t-s}\delta_t \prod_{i=s+1}^{t}$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)$	Q^*
Q(λ) (P & W's)	$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$	$\sum_{t=s}^{s+n} (\lambda\gamma)^{t-s}\delta_t + \max_a Q(x_s, a) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - \max_a Q(x_t, a)$	$Q^{\mu,*}$
Q*(λ)	$\sum_{t=s}^{s+n} \gamma^{t-s} \big[r_t + \max_a Q(x_t, a) - Q(x_t, a_t)\big]$ $+\gamma^{n+1} \max_a Q(x_{s+n+1}, a)$	$\sum_{t\geq s} (\lambda\gamma)^{t-s}\delta_t$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)$	Q^*