Learning with options that terminate off-policy

Anna Harutyunyan
Vrije Universiteit Brussel
Brussels, Belgium

Peter Vrancx
PROWLER.io
Cambridge, England

Pierre-Luc Bacon
McGill University
Montreal, Canada

Doina Precup
McGill University
Montreal, Canada

Ann Nowé
Vrije Universiteit Brussel
Brussels, Belgium

Abstraction is essential for scaling up learning, and there has been a renewed interest in methods that extract, or leverage it (Vezhnevets et al., 2016; Kulkarni et al., 2016; Tessler et al., 2017). The options framework (Sutton et al., 1999) is the standard for modeling *temporal* abstraction in reinforcement learning. The temporal aspect of an option is determined by its *termination condition* β, which roughly determines its length. Learning and planning with longer options is known to be more efficient (Mann et al., 2015). This is partly due to an option having similar properties to the familiar multi-step λ-returns, which are known to yield faster convergence (Bertsekas and Tsitsiklis, 1996). The key qualitative difference between β and λ, however, is that β directly affects the solution rather than, like λ, just the rate of convergence. If β is not trivial, this couples the quality of the solution with the quality of the options at hand. This can be restrictive especially if the options are not perfect, which of course is likely. Indeed, when a set of options is given, we can show that the more they terminate, the more optimal the resulting policy is at the primitive action level. This poses a challenge: on the one hand, we wish for the options to be long to yield fast convergence and meaningful exploration, but on the other, if these options are not ideal, the more we commit to them, the poorer the quality of our solution. *Interrupting* suboptimal options is one way of addressing this (Sutton et al., 1999), but like "cutting" traces in off-policy learning, it may prevent us from following a coherent policy for more than a couple of steps.

To this end, we propose to terminate options *off-policy*, that is: decouple the *behavior* termination condition that the options execute with, from the *target* termination condition that is to be factored into the solution. The behavior terminations then solely become a means of influencing convergence speed. We describe a new algorithm, Q(β), that achieves this by leveraging connections to several old and new multi-step off-policy temporal difference algorithms.

1 Framework and notation

We assume the standard reinforcement learning setting (Sutton and Barto, 2017) of a discrete time MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$. Below we introduce the options framework. For the remainder of the background and notation, we refer the reader to Appendix A.

An option o is a tuple $(\mathcal{I}^o, \beta^o, \pi^o)$, with $\mathcal{I}^o \subseteq \mathcal{S}$ the initiation set, from which an option may start, $\beta^o : \mathcal{S} \to [0, 1]$, the probabilistic termination condition, and π^o, the option policy with which it navigates through the environment. Just like the MDP reward and transition models r and p, options can be seen to induce *semi*-MDP (Puterman, 1994) models R and P as follows (Sutton et al., 1999):

$$P_{ss'}^o \stackrel{\text{def}}{=} \mathbb{E}_{D:s \to s'|o} \left[\gamma^D \right] = \gamma p_{ss'}^{\pi^o} \beta_{s'} + \gamma \sum_{s'' \in \mathcal{I}^o} p_{ss'}^{\pi^o} (1 - \beta_{s''}) P_{s''s'}^o,$$

$$R_s^o \stackrel{\text{def}}{=} \mathbb{E}_{D:s|o} \left[\sum_{i=1}^{D} \gamma^{i-1} r^{\pi^o}(S_{t+i}) | S_t = s \right] = r_s^{\pi^o} + \gamma \sum_{s' \in \mathcal{I}^o} p_{ss'}^{\pi^o} (1 - \beta_{s'}) R_{s'}^o. \tag{1}$$

where $\mathbb{E}_{D:s|o}[\cdot]$ and $\mathbb{E}_{D:s\to s'|o}[\cdot]$ are the expectations of the option duration D from state s and the travel time between states s and s', respectively, w.r.t. option dynamics p^{π^o} and the termination condition β^o.

In the *call-and-return* model of option execution, an option is run until completion (according to its termination condition), and only then a new option choice is made (Precup et al., 1998). This suggests the following state-*option* analogues of the state-action transition operator from Eq. (11), and the Bellman operator from Eq. (12). For a policy over options μ:

$$\mathcal{P}^\mu_\mathcal{O} q(s,o) \stackrel{\text{def}}{=} \sum_{s'} P^o(s,s') \sum_{o'} \mu(o'|s') q(s',o'), \qquad \mathcal{T}^\mu_\mathcal{O} q(s,o) \stackrel{\text{def}}{=} R^o_s + \mathcal{P}^\mu_\mathcal{O} q(s,o). \qquad (2)$$

It will also be relevant to consider the *marginal* flat policy κ over primitive actions $\kappa(a|s) \stackrel{\text{def}}{=} \sum_o \mu(o|s)\pi^o(a|s)$. For simplicity, we assume that options can initiate anywhere: $\mathcal{I}^o = \mathcal{S}, \forall o \in \mathcal{O}$. Finally, in the main departure from the standard framework, we will distinguish between the *target* termination condition β that is to be estimated, and a *behavior* termination condition ζ (*"zeta"*) that the options actually terminate with.

2 The call-and-return operator

Before proceeding to describe our key idea and algorithm, let us quantify the role of β in the target solution of planning with options. To do this, we plan to derive this solution at the primitive action resolution.

Let ν be an arbitrary policy over options, and $c : \mathcal{S} \times \mathcal{O} \to [0,1]$ a coefficient function. Consider the following transition operator and its corresponding Bellman operator:

$$\mathcal{P}^{c\nu} q(s,o) \stackrel{\text{def}}{=} \sum_{s'\in\mathcal{S}} p^{\pi^o}_{ss'} c(s',o) \sum_{o'} \nu(o'|s') q(s',o'), \qquad \mathcal{T}^{c\nu} q \stackrel{\text{def}}{=} cr^\pi + \gamma \mathcal{P}^{c\nu} q,$$

where r^π is the $|\mathcal{S}| \times |\mathcal{O}|$-vector of r^{π^o} for all options. Note that $\mathcal{T}^{c\nu}$, like $\mathcal{T}^\pi$, is a one-step operator, whereas $\mathcal{T}^\mu_\mathcal{O}$ is an option-level operator. The transition operator $\mathcal{P}^{c\nu}$ in particular defines the following operators corresponding to option *continuation* and *termination*, respectively:

$$\mathcal{P}^{(1-\beta)\iota}(s,o) \stackrel{\text{def}}{=} \sum_{s'\in\mathcal{S}} p^{\pi^o}_{ss'}(1-\beta^o(s')) q(s',o), \qquad \mathcal{P}^{\beta\mu}(s,o) \stackrel{\text{def}}{=} \sum_{s'\in\mathcal{S}} p^{\pi^o}_{ss'}\beta^o(s') \sum_{o'} \mu(o'|s') q(s',o').$$

That is: ι (*"iota"*) is the policy over options that maintains the current (argument) option. Using these operators, we can express $\mathcal{P}^\mu_\mathcal{O}$ from Eq. (2) and the reward model from Eq. (1) concisely for all state-option pairs:

$$\mathcal{P}^\mu_\mathcal{O} q = (I - \gamma \mathcal{P}^{(1-\beta)\iota})^{-1} \gamma \mathcal{P}^{\beta\mu} q, \qquad R = (I - \gamma \mathcal{P}^{(1-\beta)\iota})^{-1} r^\pi,$$

and rewrite the option-level Bellman operator from Eq. (2):

$$\mathcal{T}^\mu_\mathcal{O} q = (I - \gamma \mathcal{P}^{(1-\beta)\iota})^{-1}(r^\pi + \gamma \mathcal{P}^{\beta\mu} q). \qquad (3)$$

The following proposition derives the fixed point of $\mathcal{T}^\mu_\mathcal{O}$ in terms of the one-step operators $\mathcal{P}^{(1-\beta)\iota}$ and $\mathcal{P}^{\beta\mu}$.

Proposition 1. *The fixed point of the operator $\mathcal{T}^\mu_\mathcal{O}$ is the same as the fixed point of the operator $\mathcal{T}^{(1-\beta)\iota} + \mathcal{T}^{\beta\mu}$, and writes:*

$$q^{\mu,\iota}_\beta = (I - \gamma(\mathcal{P}^{\beta\mu} - \mathcal{P}^{\beta\iota}) - \gamma \mathcal{P}^{1\iota})^{-1} r^\pi. \qquad (4)$$

Thus, the termination scheme directly affects the convergence limit: in the extreme, if $\beta = \mathbf{0}$, options never terminate, and we have the fixed point of $\mathcal{T}^{1\iota}$: $q^{\mu,\iota}_0(s,o) = \mathbb{E}_{\pi^o} q^{\pi^o}(s,\cdot)$, the value of the option o. In the other extreme, $\beta = \mathbf{1}$, the options terminate at every step and we have the fixed point of $\mathcal{T}^{1\mu}$, which can be shown (Theorem 1 in (Bacon and Precup, 2016)) to correspond to the value of the marginal policy κ:

$$q^{\mu,\iota}_1(s,o) = \mathbb{E}_\kappa q^\kappa(s,\cdot), \forall o \in \mathcal{O}. \qquad (5)$$

3 Off-policy option termination

We would like to decouple the *target* termination condition β that factors into the solution from the *behavior* termination condition ζ that governs for how long the options are followed. Apart from the theoretical appeal of the freedom that this allows, a key motivation is in the fact that on the one hand just like with multi-step returns, the less options terminate the faster the convergence, but on the other the more options terminate, the better the control solution (as we show formally in the next section). Being able to decouple the two allows one to achieve the best of both worlds, and is exactly what we propose to do in this paper.

The critical insight in our approach is the *off-policy-ness* at the primitive action level that is introduced by the discrepancy between policies μ (that picks a new option) and ι (that maintains the current option) in Eq. (4). The *degree* of this off-policy-ness is modulated exactly by the termination condition β, and is usually implicit. We propose to leverage multi-step off-policy learning and to impose a desired degree *explicitly*, independent of the behavior terminations ζ.[1] In the extreme, we can learn the marginal policy κ directly. In the other extreme, we can learn the value of the current option. The two extremes are traded off via β. The algorithm we propose is analogous to the unifying algorithm $Q(\sigma)$ in which σ modulates the degree of off-policy-ness (Asis et al., 2017).

To highlight the parallels between learning with options and learning off-policy from multi-step returns, this section will incrementally build the technical intuition towards the proposed *off-policy termination* operator, starting from the classical intra-option equations. The next section will analyze the convergence of this operator.

3.1 From one step intra-option learning to General Q(λ)

To begin, let us first re-derive the target from Proposition 1 starting from the familiar intra-option equations (Sutton and Precup, 1998). At this point let us assume that the options execute w.r.t. some behavior condition ζ. Letting Δ denote the update on the estimated Q-function, we have at time t and the current option o:

$$\Delta q(S_t, o) \propto r(S_t, A_t) + \gamma \tilde{q}(S_{t+1}, o) - q(S_t, o), \tag{6}$$
$$\tilde{q}(s, o) = (1 - \beta^o(s))q(s, o) + \beta^o(s)\mathbb{E}_\mu q(s, \cdot),$$

where as before we write $\mathbb{E}_\mu q(s, \cdot) \stackrel{\text{def}}{=} \sum_o \mu(o|s)q(s, o)$. Notice that this is exactly a sample of the one-step update corresponding to $\mathcal{T}^{(1-\zeta)\iota} + \mathcal{T}^{\zeta\mu}$. In fact, if we roll it out over multiple steps, and take an expectation, we obtain Eq. (3) for the behavior ζ exactly:

$$q(s, o) = \mathbb{E}_{\pi^o}\Big[\sum_{t=0}^\infty \gamma^t \Big(\prod_{i=1}^t (1 - \beta^o(S_i))\Big)\big[r(S_t, A_t) + \gamma\beta^o(S_{t+1})\mathbb{E}_\mu q(S_{t+1}, \cdot)\big]\Big], \tag{7}$$

from which some simple algebra yields:

$$\Delta q(s, o) = r^{\pi^o}(s) + \gamma\mathbb{E}_\mu q(S_{t+1}, \cdot) - q(s, o) + \mathbb{E}_{\pi^o}\Big[\sum_{t=1}^\infty \gamma^t \Big(\prod_{i=1}^t (1 - \beta^o(S_i))\Big)\delta_t^{\mu\mu}\Big], \tag{8}$$

$$\delta_t^{\mu\mu} = r(S_t, A_t) + \gamma\mathbb{E}_\mu q(S_{t+1}, \cdot) - \mathbb{E}_\mu q(S_t, \cdot).$$

This is the same (expected) update as Peng's Q(λ) for greedy policies (Peng and Williams, 1996) or General Q(λ) (i.e. off-policy Expected SARSA(λ) (Van Seijen et al., 2009)) for arbitrary policies, with $1 - \beta^o(S_i)$ being the state-option analogue of λ from those algorithms. The fixed point of both of those algorithms is given in Proposition 1 in (Harutyunyan et al., 2016) and is indeed analogous to that given in Proposition 1 in this paper.

3.2 General Q(λ) to Tree-Backup(λ)

The update (8) converges to the fixed point from Prop. 1, which is an on-/off-policy mixture, regulated by the behavior terminations ζ. We wish to reweigh this mixture by the desired target terminations β.

---

[1]Technically this should be referred to as *off-termination* learning, since the difference is indeed in termination conditions, of which the induced policies are a consequence: the "behavior policy" ι takes the current option w.p. 1, while the "target policy" is the actual policy over options μ.

Let us now begin introducing the off-policy machinery that will allow us to do so. It will be useful to cast the multi-step intra-option update (8) in the form of the general off-policy operator (13). This can be done by replacing the second expectation in the TD-error with the point-estimate $q(S_t, o)$, which introduces *off-policy corrections*:

$$\delta_t^\mu = r(S_t, A_t) + \gamma \mathbb{E}_\mu q(S_{t+1}, \cdot) - q(S_t, o).$$

This update will converge to the "off-policy" fixed point of $\mathcal{T}^{1\mu}$ only if μ and ι are close (Harutyunyan et al., 2016), which, in our particular case of the indicator behavior policy ι, is not a very interesting scenario. If we further augment the "trace" $1 - \beta^o(S_i)$ with the policy probability coefficient $\mu(o|S_i)$, we obtain option-level Tree-Backup(λ) (Precup et al., 2000) whose target policy is μ, and behavior policy is ι, and with $1 - \beta^o(S_i)$ being the state-option analogue of λ:

$$\Delta q(s, o) = \mathbb{E}_{\pi^o} \Big[\sum_{t=0}^\infty \gamma^t \Big(\prod_{i=1}^t c_i^o \Big) \delta_t^\mu | S_0 = s \Big], \qquad c_i^o = \mu(o|S_i)(1 - \zeta^o(S_i)). \tag{9}$$

From the convergence guarantees of Tree-Backup, we know that this update converges to the fixed point of $\mathcal{T}^{1\mu}$, which in turn corresponds to q^κ (Eq. (5)).

3.3 The off-policy termination operator

We are now ready to present the operator underlying the Q(β) algorithm, which is the main contribution of this paper. Eq. (9) can be considered a special case where we correct *all* of the off-policy-ness, thus implicitly assuming $\beta = 1$. To obtain the general case, we need to split the target in each TD-error into two off- and on-policy terms:

$$\mathcal{R}_\beta^\mu q(s, o) = q(s, o) + \sum_{t=0}^\infty \gamma^t \mathbb{E}_{\pi^o} \Big[\Big(\prod_{i=1}^t c_i^o \Big) \delta_t^{\beta, \mu} \Big], \quad \delta_t^{\beta, \mu} = R_{t+1} + \gamma \tilde{q}(S_{t+1}, o) - q(S_t, o), \tag{10}$$

$$\tilde{q}(s, o) = (1 - \beta^o(s))q(s, o) + \beta^o(s)\mathbb{E}_\mu q(s, \cdot), \quad c_i^o = (1 - \zeta^o(S_i))(1 - \beta^o(S_i) + \beta^o(S_i)\mu(o|S_i)).$$

Inspecting these updates, it is clear that the parameter β controls the degree of partial "off-policy-ness": in the next state, with a weight $1 - \beta^o(S_{t+1})$ the option continues and the update considers the "on-policy" current option value $q(S_{t+1}, o)$, and with a weight $\beta^o(S_{t+1})$, the option terminates, and the update considers the "off-policy" value w.r.t. the policy over options $\sum_{o'} \mu(o'|s)q(S_{t+1}, o')$. Note that, these coefficients appear in the correction terms c_i^o as well, since in order for the algorithm to converge to the correct on-/off-policy mixed target, the $\beta^o(S_{t+1})$-weighted off-policy portion needs to be corrected. Finally, note that in the learning setting, the second factor in c_i^o is explicit, while $1 - \zeta^o(S_i)$ is sampled from the current option. Algorithm 1 presents the forward view of the algorithm underlying this expected operator, for the general case of an evolving sequence of policies $(\mu_k)_{k \in \mathbb{N}}$.

This algorithm is very similar to the recently formalized Q(σ) (Asis et al., 2017; Sutton and Barto, 2017), with β being a state-option generalization of $1 - \sigma$. In Q(σ), the parameter σ controls the degree of off-policy-ness: $\sigma = 1$ corresponds to (on-policy) SARSA, and $\sigma = 0$ to (off-policy) Tree-Backup. Analogously, Q(β) with $\beta = \mathbf{0}$ learns the "on-policy" value of the current option policy π^o (i.e. the policy ι), and Q(β) with $\beta = \mathbf{1}$ learns the "off-policy" value of the marginal policy κ (i.e. the policy μ). The behavior terminations ζ on the other hand have a role analogous to that of the eligibility trace parameter λ.

4 Analysis

In this section we will analyze the convergence behavior of the off-policy termination operator in both policy evaluation and control settings, and show that learning about shorter target options off-policy is generally asymptotically more efficient than on-policy. We will then consider the relationship of the solution quality in control with option duration, and show that shorter options generally yield better solutions.

4.1 Policy evaluation

We will prove that the evaluation operator $\mathcal{R}_\beta^\mu$ is contractive around the appropriate fixed point, and that its contraction factor is less than that of the respective *on*-policy operator, given the target options

are longer than the behavior ones. The proof of the following theorem is analogous to that of Theorem 1 from (Munos et al., 2016) and is given in appendix.

Theorem 1 (Policy evaluation). *The operator $\mathcal{R}_\beta^\mu$ defined in Eq. (10) has a unique fixed point $q_\beta^{\mu,\iota}$, as defined in Eq. (4). Furthermore, if for each state $S_i \in \mathcal{S}$ and option $o \in \mathcal{O}$ we have $c_i^o \leq (1 - \beta^o(S_i)) + \beta^o(S_i)\mu(o|S_i)$, then for any Q-function q:*

$$|\mathcal{R}_\beta^\mu q(s,o) - q_\beta^{\mu,\iota}(s,o)| \leq \eta(s,o)\|q(s,o) - q_\beta^{\mu,\iota}(s,o)\|,$$

where $\eta(s,o) \stackrel{def}{=} 1 - (1-\gamma)\mathbb{E}_{\pi^o}\left[\sum_{t=0}^\infty \gamma^t \left(\prod_{i=1}^t c_i^o\right)\right] \leq \gamma$.

The contraction coefficient η controls the convergence speed of this operator: the smaller η (and the larger $\prod_{i=1}^t c_i^o$) the fewer iterations are needed to converge, but the larger the computational expense when planning, or the variance when learning (Bertsekas and Ioffe, 1996; Munos et al., 2016). Since $c_i^o \leq 1$, and options terminate eventually, the variance is less significant here, and generally, larger c_i^o will yield faster convergence. In our case, since a behavior option is assumed (i.e. the $1 - \zeta^o(S_i)$ factor in Eq. (10) is fixed), the additional β-term in c_i^o can only *reduce* the existing trace. However, since we are interested in learning about a different target, we ought to compare these traces to the setting in which that same target is learnt *on*-policy. The following corollary derives a sufficient condition for Q(β) to maintain larger traces than its on-policy counterpart.

Corollary 1. *If for all states s and options o we have:*

$$\beta^o(s) \geq 1 - \frac{\mu(o|s)(1 - \zeta^o(s))}{\mu(o|s)(1 - \zeta^o(s)) + \zeta^o(s)},$$

then the iteration (10) converges faster than if β^o was used on-policy, in place of ζ^o in iteration (7).

In particular if $\zeta^o(s) = 0$, any $\beta^o(s) > 0$ satisfies this, irrespective of μ. If μ is deterministic, this holds for all $\beta^o(s) > \zeta^o(s)$ for the chosen o. In general, the intuition here is that it's easier to learn from longer option traces about shorter options than vice versa.

4.2 Control

Let us now formulate the control analogue of Theorem 1. Its proof is a simpler version of that of Theorem 2 from (Munos et al., 2016), and we omit it here.

Theorem 2 (Control). *Consider a sequence of policies over options $(\mu_k)_{k\in\mathbb{N}}$ that are greedy w.r.t. the sequence of estimates $(q_k)_{k\in\mathbb{N}}$, and consider the update: $q_{k+1} = \mathcal{R}_\beta^{\mu_k} q_k$, where the operator $\mathcal{R}_\beta^{\mu_k}$ is defined by Eq. (10) for the k-th policy μ_k. Let $\mathcal{T}_\beta^{\mu,\iota} q = \mathcal{T}^{(1-\beta)\iota}q + \mathcal{T}^{\beta\mu}q$, and let $q^* = \max_\mu q_\beta^{\mu,\iota}$. Suppose that $\mathcal{T}_\beta^{\mu_0,\iota}q_0 \geq q_0$.* [2] *Then for any $k \geq 0$,*

$$\|q_{k+1} - q^*\| \leq \gamma\|q_k - q^*\|.$$

It follows that $q_k \to q^*$ as $t \to \infty$.

We do not give a proof of online convergence at this time, but verify it empirically in our experiments.

4.3 Option duration and solution quality

Our convergence results show that learning about shorter options off-policy is more efficient than on-policy. Here, we motivate why one would want to learn about shorter options in the first place. In particular, we will show that given a set of options, the more they terminate, the better the resulting control solution at the primitive action resolution. Intuitively, this is because upon termination, the learner picks the current best option, whereas during option execution, the target value includes the potentially suboptimal current option (Proposition 1). The following theorem (proof in appendix) formalizes this intuition, and may be of independent interest.

Theorem 3 (The more options terminate, the better the solution.). *Given a set of options $\mathcal{O}$, and a greedy policy over options μ. Let $\beta \geq \zeta$ be two termination conditions for the options in $\mathcal{O}$. Then: $q_\beta^{\mu,\iota} \geq q_\zeta^{\mu,\iota}$.*

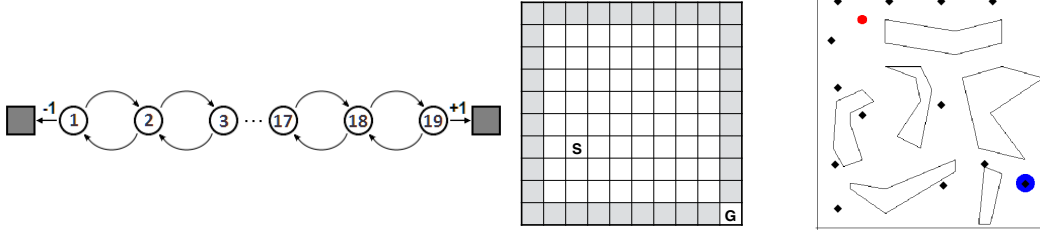[2] This can be attained by pessimistic initialization: $q_0 \leq r_{\max}/(1-\gamma)$.

Figure 1: The domains used in the experiments. **Left:** The 19-state random walk task. The agent starts in the middle. Transitions are deterministic, and the task terminates in each end. **Middle:** The modified cliffwalk task. Shaded regions are cliffs. **Right:** Pinball domain configuration used. The red ball must be moved to the blue hole. Each black diamond indicates an option landmark.

Note that this result refers to the target solution. During learning, the more decisions there is to make, the more potential there is for error. As such, in reasonably complex tasks we expect the performance to obey a tradeoff on β.

5   Experiments

Finally, let us evaluate our algorithm empirically. We aim to illustrate the following claims: (1) The learning speed improves as ζ gets smaller (behavior options get longer); (2) The control performance improves, as β gets larger (target options get shorter); and (3) Q(β) converges with off-policy terminations. For simplicity, we assume that options terminate deterministically in a set of goal states, and hence reduce β and ζ to single parameters that determine the likelihood of terminating *before* reaching the goal. The "plain" variant refers to the on-policy intra-option update from Eq. (7). The complete experimental details are provided in appendix G.

5.1   Policy evaluation

First, we show that Q(β) learns the correct values on the 19-state random walk task (Fig. 1, Left). There are two options: one leads all the way to the left, the other to the right. The policy over options is uniform. The task is to estimate the value function w.r.t. target terminations β. The results are given in Figure 2. Q(β) is able to learn the correct values (up to an irreducible exploration-related error). As expected, Q(β) gets more efficient as behavior options get longer (ζ gets smaller). The opposite is true for the plain algorithm, since without interrupting sufficiently, it does not have a chance to update the intermediate states with anything other than the option policy. There is a small inflection point in the performance of the plain algorithm at the *on*-policy value of ζ.
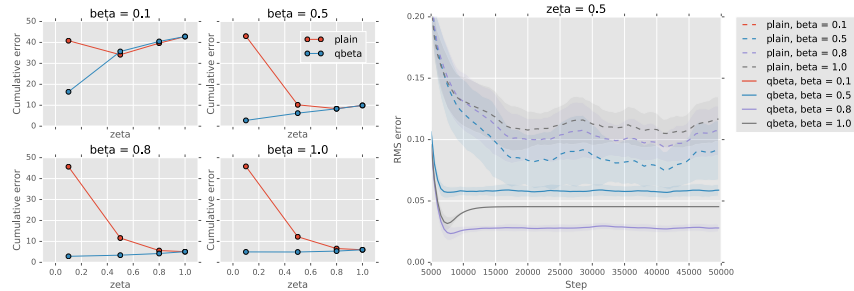

Figure 2: Prediction error on the 19-state chain task. Each variant is an average of 10 seeds. **Left:** Sum error for each ζ-β combination. Q(β) always gets more efficient as ζ decreases (the options get longer). **Right:** Example learning curves. The lines corresponding to $\beta = 0.1$ are outside the axes' bounds. The shaded region covers standard deviation.
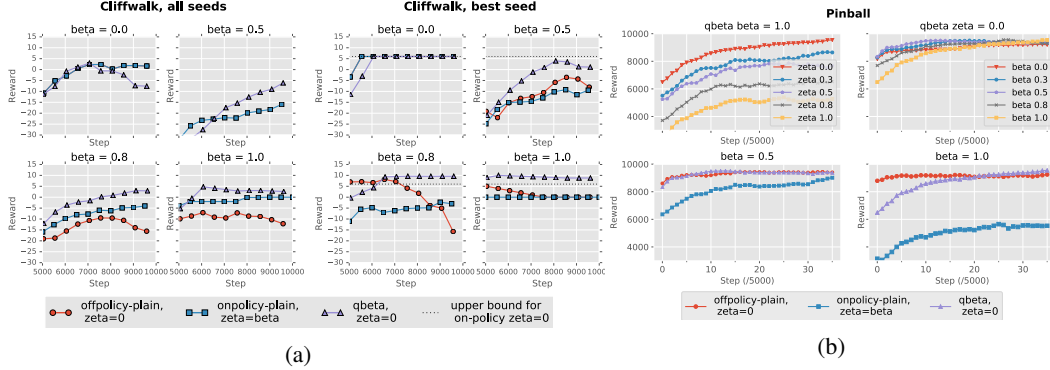
Figure 3: Control performance. **(a)** Cliffwalk. Each variant is evaluated on 5 seeds for 10 runs each. *Left:* Average performance per value of β on all seeds. *Right:* Learning curves for the best seeds per variant. Notice how Q(β) is the only variant that escapes the plateau of the suboptimal policy. **(b)** Pinball. Each variant is evaluated on 20 independent runs. *Top row:* Influence of β and ζ on Q(β): performance improves as ζ gets smaller. Overall, intermediate target β-s are best, but $\beta = 1$ reaches slightly better performance at the end of learning. *Bottom row:* Comparison within the variants for select values of β

5.2 Control

To demonstrate the benefit of decoupled off- and on- policy terminations, we compare our algorithm with the plain on-policy variant (labelled: *onpolicy-plain*) that uses $\zeta = \beta$ during both learning and evaluation. In order to demonstrate that the learning target plays a role, we also compare it with the plain algorithm that uses β at evaluation only (labelled: *offpolicy-plain*). The behavior $\zeta = 0$, unless specified otherwise.

Modified Cliffwalk We illustrate the benefits of off-policy termination on a modified Cliffwalk example (Fig. 1, middle). The agent starts in a position inside a $n \times n$ grid with the goal of getting to a corner where a positive reward is given. The step reward is zero, but there are small cliffs along the border that aren't fatal, but induce a penalty. We have four options, one for each cardinal direction, that take the agent up until the corresponding border (and cliff). Thus, while these options are able to learn to reach the goal in an optimal number of *steps*, they are unable to learn the optimal policy which only moves inside the grid, so as to not encounter cliffs. To ensure adequate exploration we consider ϵ_{opt}-soft option policies, as well as a usual ϵ-greedy policy over options during learning (but not during evaluation). The results are plotted in Fig. 3a. Q(β) outperforms the alternatives in all cases, and its performance improves with larger β. Note that it is the only variant able to surpass the value of the suboptimal upper bound of the naive approach.

Pinball We finally evaluate our algorithm on a variation of the Pinball domain (Konidaris and Barto, 2009). Here, a ball must be maneuvered through a set of obstacles into a hole. Observations consist of four continuous variables describing the ball's x, y positions and velocities. There are five primitive actions: the first four apply a small force to either the x or y velocity, the final action leaves all velocities unchanged. There is a step penalty and a final reward. We define a set of *landmark* options (Mann et al., 2015) that move the ball near a target goal location on the board. The agent can initiate and terminate each option from within some initiation and termination distances from the respective landmark. To illustrate the benefits of terminating suboptimal options, landmarks were placed in such a way that the paths from start to the goal via the landmarks are suboptimal (Fig. 1, right). The results are plotted in Fig. 3b. As expected, the performance of Q(β) drastically improves with longer behavior options. The influence of β is more subtle: intermediate β-s perform best overall, but $\beta = 1$ reaches slightly better eventual performance. In a comparison, Q(β) outperforms the on-policy variant that learns with $\zeta = \beta$. However, in this domain, the off-policy variant (that learns with $\zeta = 0$, but evaluates with β) performs comparably to Q(β). This may in part be due to the use of function approximation, which allows the plain update to generalize meaningfully within the option trajectory, and in part due to the noisy nature of Pinball, in which there are many optimal

7

policies of similar values. Since, as we have seen, Q(β) is the only variant to learn accurate values, we expect it to stand out more in settings where the reward scheme is more intricate.

6   Discussion

We propose decoupling behavior and target termination conditions, like it is done with policies in off-policy learning. We formulate an algorithm for learning target terminations off-policy, analyze its expected convergence, and validate it empirically, confirming the theoretical intuition that learning shorter options from longer options is beneficial both computationally and qualitatively. More generally, we cast learning with options into a common framework with well-studied multi-step off-policy temporal difference learning, which allows us to carry over existing results with ease.

Learning longer options from shorter options.   We have assumed here that the options are given, but may not express the optimal policy well. This scenario applies when the options describe simple rules of thumb, or are transferred from a different task. If the options are not given, but learnt end-to-end, our wish typically is to distill meaningful behavior in them. However, instead, the result often ends up reducing to degenerate options (Bacon et al., 2017; Mann et al., 2014). Being able to impose longer durations on the target off-policy may mitigate this. Though it should be noted that our convergence results suggest that learning may not be as efficient then.

Action-level importance sampling.   The option policy term in the trace is ambivalent to the action choice. Thus if $\mu(o|S_i)$ is small, c_i^o will be small, even if the taken action is consistent with the option policy π^o. It would be interesting to replace this term with the importance sampling ratio at the primitive action level, like $\frac{\kappa(A_i|S_i)}{\pi^o(A_i|S_i)}$, which corresponds to another multi-step off-policy algorithm, Retrace(λ) (Munos et al., 2016). Another direction towards this goal is to incorporate the intra-option correction from Eq. (14).

References

Asis, K. D., Hernandez-Garcia, J. F., Holland, G. Z., and Sutton, R. S. (2017). Multi-step reinforcement learning: A unifying algorithm. *CoRR*, abs/1703.01327.

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *Proceedings of AAAI*, pages 1726–1734.

Bacon, P.-L. and Precup, D. (2016). A matrix splitting perspective on planning with options. *CoRR*, abs/1612.00916.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA.

Bertsekas, D. P. and Ioffe, S. (1996). Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical Report LIDS-P-2349, MIT.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. (2016). Q(λ) with off-policy corrections. In *Proceedings of the Conference on Algorithmic Learning Theory*, pages 305–320.

Kearns, M. J. and Singh, S. P. (2000). Bias-variance error bounds for temporal difference updates. In *COLT*, pages 142–147.

Konidaris, G. and Barto, A. (2009). Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 3675–3683.

Mahmood, A. R., Yu, H., and Sutton, R. S. (2017). Multi-step off-policy learning without importance sampling ratios. *CoRR*, abs/1702.03006.

Mann, T., Mankowitz, D., and Mannor, S. (2014). Time-regularized interrupting options (trio). In *Proceedings of the International Conference on Machine Learning*, pages 1350–1358.

Mann, T. A., Mannor, S., and Precup, D. (2015). Approximate value iteration with temporally extended actions. *JAIR*, 53:375–438.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1046–1054.

Peng, J. and Williams, R. J. (1996). Incremental multi-step q-learning. *Machine Learning*, 22(1-3):283–290.

Precup, D., Sutton, R. S., and Singh, S. (1998). Theoretical results on reinforcement learning with temporally abstract options. In *Proceedings of the European conference on machine learning*, pages 382–393.

Precup, D., Sutton, R. S., and Singh, S. (2000). Eligibility traces for off-policy policy evaluation. In *Proceedings of the International Conference on Machine Learning*, pages 759–766.

Puterman, M. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1st edition.

Ryan, M. R. K. (2002). Using abstract models of behaviours to automatically generate reinforcement learning hierarchies. In *In Proceedings of The International Conference on Machine Learning*, pages 522–529.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *Proceedings of the International Conference on Machine Learning (ICML-15)*, pages 1312–1320.

Sutton, R., Modayil, J., Delp, M., Degris, T., Pilarski, P., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, pages 761–768.

Sutton, R. S. (1995). Td models: Modeling the world at a mixture of time scales. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 531–539.

Sutton, R. S. and Barto, A. G. (2017). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Sutton, R. S. and Precup, D. (1998). Intra-option learning about temporally abstract actions. In *In Proceedings of the International Conference on Machine Learning*, pages 556–564.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Szepesvari, C., Sutton, R. S., Modayil, J., Bhatnagar, S., et al. (2014). Universal option models. In *Advances in Neural Information Processing Systems*, pages 990–998.

Tessler, C., Givony, S., Zahavy, T., Mankowitz, D. J., and Mannor, S. (2017). A deep hierarchical approach to lifelong learning in minecraft. In *Proceedings of AAAI*, pages 1553–1561.

Touati, A., Bacon, P.-L., Precup, D., and Vincent, P. (2017). Convergent tree-backup and retrace with function approximation. *CoRR*, abs/1705.09322.

Van Seijen, H., Van Hasselt, H., Whiteson, S., and Wiering, M. (2009). A theoretical and empirical analysis of expected sarsa. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184.

Vezhnevets, A., Mnih, V., Osindero, S., Graves, A., Vinyals, O., Agapiou, J., and Kavukcuoglu, K. (2016). Strategic attentive writer for learning macro-actions. In *Advances in Neural Information Processing Systems 29*, pages 3486–3494.

White, M. (2017). Unifying task specification in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, volume 70, pages 3742–3750.

Yu, H. and Bertsekas, D. P. (2012). Weighted bellman equations and their applications in approximate dynamic programming. Technical Report LIDS-P-2876, MIT.

A Background

We assume the standard reinforcement learning setting (Sutton and Barto, 2017) of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of discrete actions; $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the transition that specifies the environment dynamics, with $p(s'|s, a)$ (or $p_{ss'}^a$ in short), denoting the probability of transitioning to state s' upon taking action a in s; $r : \mathcal{S} \times \mathcal{A} \to [-r_{\max}, r_{\max}]$ is the reward function, and γ the scalar discount factor. A policy is a probabilistic mapping from states to actions. For a policy π, and the MDP transition matrix p, let the matrix p^π denote the dynamics of the induced Markov chain: $p^\pi(s, s') = \sum_{a \in \mathcal{A}} \pi(a|s) p(s'|s, a)$, and r^π the reward expected for each state under π: $r^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) r(s, a)$.

Consider the Q-function q as a mapping $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and define the one-step transition operator over Q-functions as:

$$(\mathcal{P}^\pi q)(s, a) \stackrel{\text{def}}{=} \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} p(s'|s, a) \pi(a'|s') q(s', a'). \tag{11}$$

Using operator notation, define the Q-function corresponding to the *value* of policy π as:

$$q^\pi \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} \gamma^t (\mathcal{P}^\pi)^t r = r + \gamma \mathcal{P}^\pi q^\pi = (I - \gamma \mathcal{P}^\pi)^{-1} r,$$

the recursive form of which defines the Bellman equation for the policy π (Bellman, 1957). The corresponding one-step Bellman operator can be applied to any Q-function:

$$\mathcal{T}^\pi q(s, a) \stackrel{\text{def}}{=} r(s, a) + \gamma \mathcal{P}^\pi q(s, a), \tag{12}$$

and its repeated applications are guaranteed to produce its fixed point q^π (Puterman, 1994). The policy evaluation setting is concerned with estimating this quantity for a given policy π.

The Bellman *optimality* operator introduces maximization over the set of policies: $\mathcal{T} q \stackrel{\text{def}}{=} r + \gamma \max_\pi \mathcal{P}^\pi q$, and its repeated applications are guaranteed to produce its fixed point, $q^* \stackrel{\text{def}}{=} \max_\pi q^\pi = q^{\pi^*}$, which is the value of the *optimal* policy π^*. The control setting is concerned with finding this value. A policy is *greedy* w.r.t. a Q-function if at each state it picks the action of maximum value. The optimal policy π^* is greedy w.r.t. the optimal Q-function q^*.

Throughout, we will write $\mathbf{1}$ and $\mathbf{0}$ for the vectors with 1- or 0-components. For a policy π^b, we will use $\mathbb{E}_{\pi^b}[\cdot]$ as a shorthand for the expectation $\mathbb{E}_{S_{1:\infty}, A_{1:\infty}|\pi^b}[\cdot]$ w.r.t. the trajectory $S_0 = s, A_0 = a, S_1, A_1, \ldots$, with A_t drawn according to $\pi^b(\cdot|S_t)$ and S_{t+1} drawn according to $p(\cdot|S_t, A_t)$. We will occasionally use R_{t+1} to denote $r(S_t, A_t)$, and write arguments in subscripts (i.e. use $x(s)$ and x_s equivalently). In the learning setting, the algorithms often update with a sample of the residual $\mathcal{T}^\pi q - q$:

$$\delta_t = R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) - q(S_t, A_t),$$

which is referred to as a temporal difference (TD) error.

A.1 Multi-step off-policy TD learning

One need not only consider single-step operators, and may apply $\mathcal{T}^\pi$ and $\mathcal{T}$ repeatedly. The λ-operator is a particularly flexible mixture of such multi-step operators:

$$\mathcal{T}_\lambda^\pi q = (1 - \lambda) \sum_{t=0}^{\infty} \lambda^n (\mathcal{T}^\pi)^n q = q + (I - \lambda \gamma \mathcal{P}^\pi)^{-1} (\mathcal{T}^\pi q - q).$$

In the learning setting, this corresponds to considering multi-step *returns*, rather than one-step samples for updating the estimate, and λ trades off the bias of bootstrapping with an approximate estimate with the variance of sampling multiple steps. A high intermediate value of λ is typically best in practice (Kearns and Singh, 2000).

Off-policy learning is the setting where the behavior and target policies are decoupled. That is: $\pi^b \neq \pi$. Multi-step methods pose a challenge when considered off-policy, and advances have recently

been made in this direction (Munos et al., 2016; Mahmood et al., 2017). To this end, Munos et al. (2016) unified several off-policy return-based algorithms under a common umbrella:

$$\mathcal{R}_c^{\pi,\pi^b} q(s,a) \stackrel{\text{def}}{=} q(s,a) + \mathbb{E}_{\pi^b}\Big[\sum_{t=0}^{\infty} \gamma^t \Big(\prod_{i=1}^{t} c_i \Big) \delta_t^{\pi} \Big], \tag{13}$$

$$\delta_t^{\pi} = R_{t+1} + \gamma \mathbb{E}_{\pi} q(S_{t+1}, \cdot) - q(S_t, A_t).$$

where $\mathbb{E}_{\pi} q(s \cdot) \stackrel{\text{def}}{=} \sum_{a \in \mathcal{A}} \pi(a|s) q(s,a)$, and the *trace* c_i is a state-action coefficient, whose specific shapes correspond to different algorithms. In particular, *Tree-Backup(λ)* sets c_i to $\lambda \pi_b(A_i|S_i)$ (Precup et al., 2000), while *Retrace(λ)* sets c_i to $\lambda \min(1, \frac{\pi(A_i|S_i)}{\pi_b(A_i|S_i)})$ (Munos et al., 2016).

A.1.1 Unifying algorithm

Recently, Asis et al. (2017) formulated an algorithm that encompasses the ones discussed so far even more generally. Instead of the binary taxonomy of on-and off-policy algorithms, the authors introduce a parameter σ that smoothly transitions between the two. This algorithm can also be expressed via Eq. (13) with:

$$\delta_t^{\sigma,\pi} = R_{t+1} + \gamma(\sigma q(S_{t+1}, A_{t+1}) + (1-\sigma)\mathbb{E}_{\pi} q(S_{t+1}, \cdot)) - q(S_t, A_t),$$
$$c_i = \lambda((1-\sigma)\pi_b(A_i|S_i) + \sigma).$$

In particular, $\sigma = 1$ corresponds to the on-policy SARSA(λ) algorithm, while $\sigma = 0$ to Tree-Backup(λ).

B Proof of Proposition 1

Proof. From Eq. (3), and writing q for $q_{\beta}^{\mu,\iota}$ for less clutter:

$$q = (I - \gamma \mathcal{P}^{(1-\beta)\iota})^{-1} (r^{\pi} + \gamma \mathcal{P}^{\beta\mu} q),$$
$$q - \gamma \mathcal{P}^{(1-\beta)\iota} q = r^{\pi} + \gamma \mathcal{P}^{\beta\mu} q,$$
$$q = r^{\pi} + \gamma \mathcal{P}^{\beta\mu} q + \gamma \mathcal{P}^{(1-\beta)\iota} q$$
$$= r^{\pi} + \gamma \mathcal{P}^{\beta\mu} q + \gamma \mathcal{P}^{1\iota} q - \gamma \mathcal{P}^{\beta\iota} q$$
$$= r^{\pi} + \gamma (\mathcal{P}^{\beta\mu} - \mathcal{P}^{\beta\iota}) q + \gamma \mathcal{P}^{1\iota} q.$$

Solving for q yields the result. $\qquad\square$

C Relationship with intra-option learning

The off-policy-ness discussed so far is different than that in the more familiar off-policy intra-option setting. The intra-option learning algorithm suggests applying the update (6) to all options o "consistent with" the experience stream $S_1, A_1, \ldots$ (Sutton and Precup, 1998). When considering multiple steps, this amounts to introducing importance sampling into Eq. (7). That is, given a behavior option b, the trace coefficient $1 - \beta^o(S_i)$ from Eq. (7) now becomes

$$c_i^{ob} = \frac{\pi^o(A_i|S_i)}{\pi^b(A_i|S_i)} (1 - \beta^o(S_i)). \tag{14}$$

And hence, writing β_{t+1}^o for $\beta^o(S_{t+1})$, the value of option o from Eq. (7) writes:

$$q(s,o) = \mathbb{E}_{\pi^b} \sum_{t=0}^{\infty} \gamma^t \Big[\Big(\prod_{i=1}^{t} c_i^{ob} \Big) [R_{t+1} + \gamma \beta_{t+1}^o \mathbb{E}_{\mu} q(S_{t+1}, \cdot)] \Big]$$

Now, notice that there are two sources of off-policy-ness in these formulas. One is π^b vs. π^o, the contrast between option policies, and the other is in the target: ι vs. μ itself, as discussed before.

Algorithm 1 Q(β) algorithm

Given: Option set $\mathcal{O}$, target termination function β, initial Q-function q_0, step-sizes $(\alpha_k)_{k\in\mathbb{N}}$, start state s_0

1: $S_0 \leftarrow s_0$
2: **for** $k = 0, 1, \ldots$ **do**
3:     Sample an option o from $\mu_k(\cdot|S_0)$
4:     Sample the return $R_1, S_1, R_2 \ldots, S_{D_k}$ from π^o. D_k is determined by sampling $1 - \zeta^o(S_i)$.
5:     **for** $t = 0, 1, \ldots D_k - 1$ **do**
6:         $\delta_t^{\beta,\mu_k} = R_{t+1} + \gamma\tilde{q}_{\mu_k}(S_{t+1}, o) - q(S_t, o)$
7:         $\tilde{q}_{\mu_k}(s, o) \stackrel{\text{def}}{=} (1 - \beta^o(s))q(s, o) + \beta^o(s)\mathbb{E}_{\mu_k}q(s, \cdot)$
8:         $c_j^o = 1 - \beta^o(S_j) + \beta^o(S_j)\mu(o|S_j)$
9:         $\Delta_t = \sum_{i=t}^{D_k-1} \gamma^{i-t}\left(\prod_{j=t+1}^{i} c_j^o\right)\delta_t^{\beta,\mu_k}$
10:       $q_{k+1}(S_t, o) \leftarrow q_k(S_t, o) + \alpha_k\Delta_t$
11:     **end for**
12:     $S_0 \leftarrow S_{D_k}$
13: **end for**

Indeed if we write the above in the form from Eq. (8) we get a different correction:

$$\Delta q(s, o) = r^{\pi^o}(s) + \gamma\mathbb{E}_\mu q(S_{t+1}, \cdot) - q(s, o) + \mathbb{E}_{\pi^b}\left[\sum_{t=1}^{\infty}\gamma^t\left(\prod_{i=1}^{t-1} c_i^{ob}\right)(1 - \beta^o(S_t))\delta_t^{ob}\right],$$

$$\delta_t^{ob} = \frac{\pi^o(A_t|S_t)}{\pi^b(A_t|S_t)}[R_{t+1} + \gamma\mathbb{E}_\mu q(S_{t+1}, \cdot)] - \mathbb{E}_\mu q(S_t, \cdot).$$

Since the corrections for the two sources of off-policy-ness are orthogonal, it could be possible to combine them. We leave this for the future.

D   Proof of Theorem 1

Proof. Write $\mathcal{R}$ for $\mathcal{R}_\beta^\mu$. The fact that the fixed point is the desired one is clear from Eq. (10) and Proposition 1. Recall that:

$$c_i^o = (1 - \beta^o(S_i) + \beta^o(S_i)\mu(o|S_i))(1 - \zeta^o(S_i)),$$
$$\tilde{q}(s, o) = (1 - \beta^o(s))q(s, o) + \beta^o(s)\mathbb{E}_\mu q(s, \cdot).$$

Now, let us derive the contraction result. Eq. (10) can be written:

$$\mathcal{R}q(s, o) = \sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\pi^o}\left[\left(\prod_{i=1}^{t} c_i^o\right)T_t^{co}\right]$$

$$T_t^{co} \stackrel{\text{def}}{=} r(S_t, A_t) + \gamma(\tilde{q}(S_{t+1}, o) - c_{t+1}^o q(S_{t+1}, o))$$

Let $\Delta q(s, o) \stackrel{\text{def}}{=} q(s, o) - q_\beta^{\mu,\iota}(s, o)$, and $\Delta q(s, \cdot)$ be defined analogously. Since $\mathcal{R}q_\beta^{\mu,\iota} = q_\beta^{\mu,\iota}$, and after shifting the sum index forward, we have:

$$\mathcal{R}q(s, o) - q_\beta^{\mu,\iota}(s, o) = \sum_{t=1}^{\infty}\gamma^t\mathbb{E}_{\pi^o}\left[\left(\prod_{i=1}^{t-1} c_i^o\right)\Delta T_t^{co}\right],$$

$$\Delta T_t^{co} = (1 - \beta_t^o)\Delta q(S_t, o) + \beta_t^o\mathbb{E}_\mu\Delta q(S_t, \cdot) - c_t^o\Delta q(S_t, o)$$

$$= (1 - \beta_t^o - c_t^o)\Delta q(S_t, o) + \beta_t^o\sum_{b\in\mathcal{O}}\mu(b|S_t)\Delta q(S_t, b)$$

$$= \sum_b \left(\mathbb{I}_{b=o}(1 - \beta_t^b - c_t^b) + \beta_t^o\mu(b|S_t)\right)\Delta q(S_t, b),$$

12

where we write $\mathbb{I}_{b=o}$ for the indicator function. Since $c_t^o \leq (1 - \beta_t^o) + \beta_t^o \mu(o|S_t)$ and $\beta_t^o \mu(b|S_t) \geq 0$, $\forall b \neq o$, we have a linear combination of $\Delta q(S_t, b)$ weighted by non-negative coefficients $w_{y,b}$ defined as: $w_{y,b} = \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_{\pi^o} \left[\left(\prod_{i=1}^{t-1} c_i^o \right) (\mathbb{I}_{b=o} (1 - \beta_t^b - c_t^b) + \beta_t^o \mu(b|S_t)) \mathbb{I}_{S_t=y} \right]$, whose sum is:

$$
\begin{aligned}
\sum_{b \in \mathcal{O}} w_{y,b} &= \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_{\pi^o} \left[\left(\prod_{i=1}^{t-1} c_i^o \right) \sum_{b \in \mathcal{O}} \mathbb{I}_{b=o}(1 - \beta_t^b - c_t^b) + \beta_t^o \mu(b|S_t) \right] \\
&= \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_{\pi^o} \left[\left(\prod_{i=1}^{t-1} c_i^o \right) ((1 - \beta_t^o - c_t^o) + \sum_{b \in \mathcal{O}} \mu(b|S_t)\beta_t^o \right] \\
&= \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_{\pi^o} \left[\left(\prod_{i=1}^{t-1} c_i^o \right) ((1 - \beta_t^o - c_t^o) + \beta_t^o) \right] \\
&= \sum_{t=1}^{\infty} \gamma^t \mathbb{E}_{\pi^o} \left[\left(\prod_{i=1}^{t-1} c_i^o \right) (1 - c_t^o) \right] \\
&= \mathbb{E}_{\pi^o} \left[\sum_{t=1}^{\infty} \gamma^t \left(\prod_{i=1}^{t-1} c_i^o \right) - \sum_{t=1}^{\infty} \gamma^t \left(\prod_{i=1}^{t} c_i^o \right) \right] \\
&= \gamma C - (C - 1) \leq \gamma
\end{aligned}
$$

where $C = \mathbb{E}_{\pi^o} \left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{i=1}^{t} c_i^o \right) \right]$, and the last inequality is due to $C \geq 1$. It follows that $\mathcal{R}$ is a γ-contraction around $q_\beta^{\mu,\iota}$. $\square$

E   Proof of Corollary 1

Proof. We would like for the off-policy trace $c_s^o = (1 - \beta_s^o + \beta_s^o \mu(o|s))(1 - \zeta_s^o)$ to be larger than the equivalent on-policy trace $1 - \beta_s^o$.

$$
\begin{aligned}
1 - \beta_s^o &\leq (1 - \beta_s^o + \beta_s^o \mu(o|s))(1 - \zeta_s^o) \\
(1 - \beta_s^o)\zeta_s^o &\leq \beta_s^o \mu(o|s)(1 - \zeta_s^o) \\
\zeta_s^o &\leq \beta_s^o(\mu(o|s)(1 - \zeta_s^o) + \zeta_s^o) \\
\beta_s^o &\geq \frac{\zeta_s^o}{\mu(o|s)(1 - \zeta_s^o) + \zeta_s^o} \\
&= 1 - \frac{\mu(o|s)(1 - \zeta_s^o)}{\mu(o|s)(1 - \zeta_s^o) + \zeta_s^o}
\end{aligned}
$$

Thus, if β^o obeys this bound, it is more beneficial to learn it off- rather than on- policy, from the point of view of convergence speed. $\square$

F   Proof of Theorem 3

Proof. First notice that Proposition 1 rewrites:

$$
\begin{aligned}
q &= (I - \gamma(\mathcal{P}^{\beta\mu} - \mathcal{P}^{\beta\iota}) - \gamma\mathcal{P}^{1\iota})^{-1} r^\pi \\
&= (I - \gamma\mathcal{P}^{\beta\mu} - \gamma\mathcal{P}^{(1-\beta)\iota})^{-1} r^\pi \\
&= \sum_{t=0}^{\infty} \gamma^t (\mathcal{P}^{\beta\mu} + \mathcal{P}^{(1-\beta)\iota})^t r^\pi. \tag{15}
\end{aligned}
$$

13

Let $\Delta_\beta^\zeta \stackrel{\text{def}}{=} q_\beta^{\mu,\iota} - q_\zeta^{\mu,\iota}$. From the above (15), we have:

$$\Delta_\beta^\zeta = \sum_{t=1}^{\infty} \gamma^t((\mathcal{P}^{\beta\mu} + \mathcal{P}^{(1-\beta)\iota})^t - (\mathcal{P}^{\zeta\mu} + \mathcal{P}^{(1-\zeta)\iota})^t)r^\pi$$

$$= (\mathcal{P}^{\beta\mu} + \mathcal{P}^{(1-\beta)\iota} - \mathcal{P}^{\zeta\mu} - \mathcal{P}^{(1-\zeta)\iota})\sum_{t=1}^{\infty}\gamma^t \mathcal{A}_t r^\pi,$$

where $\mathcal{A}_t = (\mathcal{P}^{\beta\mu} + \mathcal{P}^{(1-\beta)\iota} - \mathcal{P}^{\zeta\mu} - \mathcal{P}^{(1-\zeta)\iota})^{-1}((\mathcal{P}^{\beta\mu} + \mathcal{P}^{(1-\beta)\iota})^t - (\mathcal{P}^{\zeta\mu} - \mathcal{P}^{(1-\zeta)\iota})^t)$. Let $f = \sum_{t=1}^{\infty}\gamma^t \mathcal{A}_t r^\pi$ be a Q-function. Then simple manipulations yield:

$$\Delta_\beta^\zeta = q_\beta^{\mu,\iota} - q_\zeta^{\mu,\iota} = (\mathcal{P}^{(\beta-\zeta)\mu} + \mathcal{P}^{(I-\beta-I+\zeta))\iota})f$$

$$= (\mathcal{P}^{(\beta-\zeta)\mu} - \mathcal{P}^{(\beta-\zeta)\iota})f$$

$$= (\mathcal{P}^{1\mu} - \mathcal{P}^{1\iota})(\beta - \zeta)f$$

$$\geq (\mathcal{P}^{1\mu} - \mathcal{P}^{1\iota})f \geq \mathbf{0},$$

since the operators $\mathcal{P}^{1\mu}$ and $\mathcal{P}^{1\iota}$ are monotone, $\beta \geq \zeta$ and $\mathcal{P}^{1\mu}f = \max_\nu \mathcal{P}^{1\nu}f \geq \mathcal{P}^{1\iota}f$ for any Q-function f. $\qquad\qquad\square$

G Experimental details

The setting is as follows: an option o is picked according to μ, and a trajectory $s, R_1, S_2, R_2, \ldots, R_{D-1}, S_D$ is generated according to π^o and ζ^o. Then for each state S_i in the trajectory, $q(S_i, o)$ is updated according to the considered algorithm.

G.1 19-chain

The termination conditions β and ζ are evaluated in the range of $\{0.1, 0.5, 0.8, 1\}$, with the first value being positive to ensure adequate state visitation. The step-sizes set via a linear search over $\alpha \in \{0.1, 0.2, 0.3, 0.4\}$. The discount factor $\gamma = 0.99$.

G.2 Modified Cliffwalk

The reward scheme is: $r_{goal} = 10$ and $r_{cliff} = -2$, and the grid size $n = 10$. We set $\epsilon = 0.1$, and $\epsilon_{opt} = 0.3$, and determine the step-size for each variant from a linear search over $\alpha \in \{0.1, 0.2, 0.3, 0.4\}$, behavior $\zeta = 0$, and target β is evaluated on the range of $\{0, 0.5, 0.8, 1\}$. The discount factor $\gamma = 0.99$.

G.3 Pinball

The reward is -1 on every step, except the final step which receives a reward of 10000. We use initiation distance of 0.3 and termination distance of 0.03. The state option value function was approximated using tile coding with 16, 10×10 tilings. All algorithms used a learning rate $\alpha = 0.01$, discount $\gamma = 0.99$ and an exploration rate $\epsilon = 0.05$ and $\epsilon_{opt} = 0.01$ during learning. The target β is evaluated on the range of $\{0, 0.3, 0.5, 0.8, 1\}$.

H Related work

Much of the related work has already been discussed throughout. We mention a few more relevant works below.

Mann et al. (2014) propose an algorithm for multi-step option interruption that stems from the same motivation of mitigating poor-quality options. In order to avoid the resulting options being too short, they introduce a time-regularization term. Our approach bypasses the need to do so by interrupting off-policy and the ability to explicitly specify target terminations.

Ryan (2002) also considers the problem of *termination improvement*, or interrupting options when they are no longer relevant, in a hybrid decision-theoretic and classical planning setting. The intuitions

in that work are particularly aligned with the ones here – in particular the tradeoff between efficiency and optimality is hinted at, and it is even suggested that "persistence [with an option] is a kind of off-policy exploration, at the behavior level." In this work, we formalize and exploit this intuition.

Yu and Bertsekas (2012) consider general λ-operators with state-dependent λ. In the case of options, $1 - \zeta^o$ takes the role of a state-option-dependent λ. White (2017) proposes to consider $1 - \zeta^o$ as part of the transition-based *discount* instead.