When Waiting Is Not an Option:
Learning Options with a Deliberation Cost

Jean Harb,[*] **Pierre-Luc Bacon,**[*] **Martin Klissarov, Doina Precup**
Reasoning and Learning Lab, McGill University
{jharb,pbacon,mklissa,dprecup}@cs.mcgill.ca

Abstract

Recent work has shown that temporally extended actions (options) can be learned fully end-to-end as opposed to being specified in advance. While the problem of *how* to learn options is increasingly well understood, the question of *what* good options should be has remained elusive. We formulate our answer to what *good* options should be in the bounded rationality framework (Simon, 1957) through the notion of *deliberation cost*. We then derive practical gradient-based learning algorithms to implement this objective. Our results in the Arcade Learning Environment (ALE) show increased performance and interpretability.

Introduction

Temporal abstraction has a rich history in AI (Minsky 1961; Fikes *et al.* 1972; Kuipers 1979; Korf 1983; Iba 1989; Drescher 1991; Dayan and Hinton 1992; Kaelbling 1993; Thrun and Schwartz 1995; Parr and Russell 1998; Dietterich 1998) and has been presented as a useful mechanism for a variety of problems that affect AI systems in may settings, including to: generate shorter plans, speed up planning, improve generalization, yield better exploration, increase robustness against model mis-specification or partial observability. In reinforcement learning, *options* (Sutton *et al.* 1999b) provide a framework to represent, learn and plan with temporally extended actions. Interest in temporal abstraction in reinforcement learning has increased substantially in the last couple of years, due to a series of success in constructing such abstractions automatically from data, e.g. (Bacon *et al.* 2017; Kulkarni *et al.* 2016; Daniel *et al.* 2016; Mankowitz *et al.* 2016; Machado *et al.* 2017). However, defining what constitutes a good set of options remains an open problem.

In this paper, we aim to leverage the bounded rationality framework (Simon 1957) in order to explain what would make good temporal abstractions for an RL system. A lot of existing reinforcement learning work has focused on Markov Decision Processes, where optimal policies can be obtained under certain assumptions. However, optimality does not take into account possible resource limitations

of the agent, which is assumed to have access to a lot of data and computation time. Indeed, options help agents overcome such limitations, by allowing policies to be computed faster (Dietterich 1998; Precup 2000). However, from the point of view of absolute optimality, temporal abstractions are not necessary: the optimal policy is achieved by primitive actions. Therefore, it has been difficult to formalize in what precise theoretical sense temporally abstract actions are helpful.

Bounded rationality is an important framework for understanding rationality in both natural and artificial systems. In this paper, we propose bounded rationality as a lens through which we can describe the desiderata for constructing temporal abstractions, as their goal is mainly to help agents which are restricted in terms of computation time. This perspective helps us to formulate more precisely what objective criteria should be fulfilled during option construction. We propose that good options are those which allow an agent to learn and plan *faster*, and provide an optimization objective for learning options based on this idea. We implement the optimization using the option-critic framework (Bacon *et al.* 2017) and illustrate its usefulness with experiments in Atari games.

Preliminaries

A finite discounted Markov Decision Process $\mathcal{M}$ is a tuple $\mathcal{M} \doteq (\mathcal{S}, \mathcal{A}, \gamma, r, P)$ where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action set respectively, and $\gamma \in [0, 1)$ is a discount factor. The reward function r is often assumed to be a deterministic function of the state and actions, but can also map to a distribution, $r : \mathcal{S} \times \mathcal{A} \to Dist(\mathbb{R})$ (a perspective which we use in our formulation). The transition matrix $P : \mathcal{S} \times \mathcal{A} \to Dist(\mathcal{S})$ is a conditional distribution over next states given that an action $a \in \mathcal{A}$ is taken under a certain state $s \in \mathcal{S}$. The interaction of a randomized stationary policy $\pi : \mathcal{S} \to Dist(\mathcal{A})$ or a deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ with an MDP $\mathcal{M}$ induces a Markov process over states, actions and rewards over which is defined the expected discounted return $V_\pi(s) \doteq \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) | S_0 = s \right]$. The value function V_π of a policy π satisfies the Bellman equations :

$$V_\pi(s) = \sum_a \pi(a|s) \left(r(s,a) + \gamma \sum_{s'} P(s'|s,a) V_\pi(s') \right).$$

---

[*] Indicates equal contribution.

In the control problem, we are interested in finding an *optimal policy* for a given MDP. A policy $\pi^\star$ is said to be optimal if $V_{\pi^\star}(s) \doteq \max_\pi V_\pi(s)$ for all s.

An important class of control methods in reinforcement learning is based on the actor-critic architecture (Sutton 1984). In the same way that function approximation can be used for value functions, policies can also be approximated within a parameterized family. In the policy gradient theorem, (Sutton *et al.* 1999a) shows that the gradient of the expected discounted return with respect to the parameters of a policy is of the form $\mathbb{E}_{\alpha,\theta}\left[\sum_a \frac{\partial \pi_\theta(a|s)}{\partial \theta} Q_{\pi_\theta}(s,a)\right]$, where α is an initial state distribution. A locally optimal policy can then be found by stochastic gradient ascent over the policy parameters while simultaneously learning the action-value function $Q_{\pi_\theta}(s,a)$ (usually by TD).

Options

Options (Sutton *et al.* 1999b) provide a framework for representing, planning and learning with temporally abstraction actions. The option frameworks assumes the existence of a base MDP which is overlaid with temporally abstract actions called *options*. An option is defined as a triple $(\mathcal{I}_o, \pi_o, \beta_o)$ where $\mathcal{I} \subseteq \mathcal{S}$ is an initiation set, $\pi_o : \mathcal{S} \to Dist(\mathcal{A})$ is the policy of an option (which can also be deterministic) and $\beta_o : \mathcal{S} \to [0,1]$ is a termination condition. In the call-and-return execution model, a policy over options $\mu : \mathcal{S} \to Dist(\mathcal{O})$ (deterministic if wanted) chooses an option among those which can be initiated in a given state and executes the policy of that option until termination. Once the chosen option has terminated, the policy over options chooses a new option and the process is repeated until the end of the episode.

The combination of a set of options and base MDP leads to a semi-Markov decision process (SMDP) (Howard 1963; Puterman 1994) in which the transition time between two decision points is a random variable. When considering the induced process only at the level of state-option pairs, usual dynamic programming results can be reused after a transformation to an equivalent MDP (Puterman 1994). To see this, we need to define two kinds of *models* for every option : a reward model $b_o : \mathcal{S} \to \mathbb{R}$ and a transition model $F_o : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$. If an option does not depend on the history since initiation, we can write its models either in closed form or as the solution to Bellman-like equations (Sutton *et al.* 1999b). The expected discounted return associated with a set of options $\mathcal{O}$ and a policy over them is the solution Q_θ to a set of Bellman equations :

$$Q_\theta(s,o) = b_\theta(s,o) + \sum_{s'} F_\theta(s'\,;\,s,o)V_\theta(s')$$

$$\doteq \mathbb{E}_\theta\left[\sum_{t=0} \gamma^t r(S_t, A_t)\middle| S_0 = s, O_0 = o\right] \ .$$

where θ is a concatenation of the policy over options μ, options policies and termination conditions.

Intra-Option Bellman Equations

In the case of Markov options, there exists another form for the Bellman equations, called the intra-option Bellman equations (Sutton *et al.* 1999b), which are key for deriving gradient-based algorithms for learning options.

Let $Z_t \doteq (S_t, O_t)$ be a random variable over state-option tuples. We call the space of state-option pairs the *augmented state space*. This augmentation is sufficient to provide the Markov property, which would otherwise be lost when considering the process at the *flat* level of state-action pairs (Sutton *et al.* 1999b). The transition matrix of the Markov process over the augmented state space (Bacon *et al.* 2017) is given by :

$$\widetilde{P}_\theta(z'|z,a) =$$
$$P(s'|s,a)\left((1 - \beta_\theta(s',o))\mathbb{1}_{o'=o} + \beta_\theta(s',o)\mu_\theta(o'|s')\right) \ .$$

Using this chain structure, we can define the MDP $\widetilde{\mathcal{M}} \doteq (\widetilde{P}_\theta, \widetilde{r}, \gamma)$ whose associated value function $\widetilde{V}_\theta : (\mathcal{S} \times \mathcal{O}) \to \mathbb{R}$ is:

$$\widetilde{V}_\theta(z) = \mathbb{E}_\theta\left[\sum_{t=0}^\infty \gamma^t \widetilde{r}(Z_t, A_t, Z_{t+1})\middle| Z_0 = z\right]$$
$$= \sum_{a,z'} \pi_\theta(a|z)\, \widetilde{P}_\theta(z'|z,a)\left(\widetilde{r}(z,a,z') + \gamma\widetilde{V}_\theta(z')\right) \ . \quad (1)$$

Since the rewards come from the base (primitive) MDP, we can simply write $\widetilde{r}(z,a,z') = r(s,a)$ and because $\sum_{z'} \widetilde{P}_\theta(z'|z,a) = 1$, we get:

$$\sum_{z'} \widetilde{P}_\theta(z'|z,a)\,\widetilde{r}(z,a,z') = r(s,a) \ .$$

Hence, when taking the expectation in (1) over the next values, we obtain :

$$\widetilde{V}_\theta(z) \doteq Q_\theta(s,o) = \sum_a \pi_\theta(a|s,o)\left(r(s,a) + \right.$$
$$\left. \gamma \sum_{s'} P(s'|s,a)\left[Q_\theta(s',o) - \beta_\theta(s',o)A_\theta(s',o)\right]\right) \ , \quad (2)$$

where $A_\theta(s,o) \doteq Q_\theta(s,o) - V_\theta(s)$ is the advantage function (Baird 1993). The equations in (2) correspond exactly to the intra-option Bellman equations (Sutton *et al.* 1999b). However, we chose to present them under an alternate – but more convenient – form highlighting a connection to the advantage function:

$$U_\theta(s',o) \doteq (1 - \beta_\theta(s',o))Q_\theta(s',o) + \beta_\theta(s')V_\theta(s')$$
$$= Q_\theta(s',o) - \beta_\theta(s',o)A_\theta(s',o) \ ,$$

where $U_\theta(s',o)$ represents the *utility* of continuing with the same option or switching to a better one.

Optimization

The option-critic architecture (Bacon *et al.* 2017) is a gradient-based actor-critic architecture for learning options end-to-end. As in actor-critic methods, the idea is to

parametrize the options policies and termination conditions and learn their parameters jointly by stochastic gradient ascent on the expected discounted return. (Bacon *et al.* 2017) provided the form of the gradients for both the option policies and termination *functions* under the assumption that options are available everywhere. Without loss of generality, we further assume that the parameter vector $\theta = [\theta_\mu; \theta_\pi; \theta_\beta]$ is partitioned into disjoint sets of parameters for the policy over option, the option policies and the termination functions.

In the gradient theorem for options policies (Bacon *et al.* 2017), the result maintains the same form as that of original policy gradient theorem for MDP (Sutton *et al.* 1999a) but over the augmented state space. If $J_\alpha(\theta)$ is the expected discount return for the set of options and the policy over them, then the gradient of the option policies (whose parameters are independent from the terminations) is :

$$\frac{\partial J_\alpha(\theta)}{\partial \theta_\pi} = \gamma \mathbb{E}_{\alpha,\theta} \left[\sum_a \frac{\partial \pi_\theta(a|z)}{\partial \theta_\pi} \widetilde{Q}_\theta(z,a) \right] \, ,$$

where α is an initial state distribution over state and options.

To obtain the gradient for the termination functions, let's first take the derivative of the intra-option Bellman equations:

$$\frac{\partial Q_\theta(s,o)}{\partial \theta_\beta} = \gamma \sum_a \pi_\theta(a|s,o) \sum_{s'} P(s'|s,a) \left[\frac{\partial Q_\theta(s',o)}{\partial \theta_\beta} \right.$$
$$\left. - \frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} A_\theta(s',o) - \beta_\theta(s',o) \frac{\partial A_\theta(s',o)}{\partial \theta_\beta} \right] . \quad (3)$$

By noticing the similarity between (3) and (1), we can easily solve for the recursive form of the derivative. Indeed, it suffices to see that $-\frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} A_\theta(s',o)$ plays the role of the "reward" term in the usual Bellman equations (see (Bacon *et al.* 2017) for a detailed proof) and conclude that:

$$\frac{\partial J_\alpha(\theta)}{\partial \theta_\beta} = \mathbb{E}_{\alpha,\theta} \left[-\frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} A_\theta(s',o) \right] . \quad (4)$$

Hence the termination gradient shows that if an option is advantageous, the probability of termination should be lowered, making that option longer. Conversely, if the value of an option is less than what could be achieved through a different choice of option at a given state, the termination gradient will make it more likely to terminate at this state. The termination gradient has the same structure as the interruption operator (Mann *et al.* 2014) in the *interruption execution* model (Sutton *et al.* 1999b). Rather than executing the policy of an option irrevocably until termination, interruption execution consists in choosing a new option whenever $Q_\theta(s,o) < V_\theta(s)$. Moving the the value function V_θ to the left side, interruption execution can also be understood in terms of the advantage function: $Q_\theta(s,o) < V_\theta(s) \Leftrightarrow A_\theta(s,o) < 0$. As for the termination gradient, interruption execution leads to the termination of an option whenever there is no advantage (negative advantage) in maintaining it. Interestingly, (Mann *et al.* 2014) also considered adding a scalar *regularizer* to the advantage function to favor longer

options. From the more general perspective of bounded rationality, we also recover this regularizer but within a larger family which follows from the notion of deliberation cost.

Deliberation Cost Model

From a representation learning perspective, good options ought to allow an agent to learn and plan *faster* (Minsky 1961). Due to their temporal structure, options offer a mechanism through which an agent can make better use of its limited computational resources and act faster. Once an option has been chosen, we assume that the computational cost of executing that option is negligible or constant until termination. After deliberating on the choice of option, an agent can *relax* thanks to the fast – but perhaps imperfect – knowledge compiled within its policy.

This perspective on options is similar to *fast and frugal* heuristics (Gigerenzer and Selten 2001) which form a decision repertoire for efficient decision making under limited resources. Our assumption on the cost structure is also consistent with models of the prefrontal areas (Botvinick *et al.* 2009; Solway *et al.* 2014) presenting decision making over options as a slower model-based planning process as opposed to the fast and habitual learning taking place within an option. When planning with options (in computers), there is also a cost for deciding which option to choose next by making predictions based on their models. For example, options models could be represented by deep networks, necessitating back-and-forth to the GPU, or using a simulator with costly rollouts (Guo *et al.* 2014; Mann *et al.* 2015).

Bounded rationality can also be useful to understand how efficient communication can take place between two agents over a limited channel (Neyman 1985). Options offer a mechanism for communicating intents to and from an agent (Branavan *et al.* 2012; Andreas *et al.* 2017) more efficiently, by compressing the information into a simpler form, sending only the identifier of the options and not the details themselves. Having longer options is a way to provide better interpretability and simplifies communication by compressing information.

Consider the cost model (fig. 1) in which executing an option within an option is free but switching to an option upon arriving in a new state incurs a cost η . To build some intuition, let's further assume that the termination function of an option is constant over all states. If κ is the continuation probability of that option, its expected discounted duration is $d = \frac{1}{1-\gamma\kappa}$. When a fixed cost η is incurred upon termination, the average cost per step for that option is then $\eta/d = (1 - \gamma\kappa)\eta$. Hence, as the probability of continuation increases and options get longer, the cost rate decreases. Conversely, if an option only terminates after one step – a primitive option – κ is 0 and the cost rate is η. The fact that longer options lead to a better amortization of the deliberation cost is key to understanding their benefit in comparison to using only primitive actions.
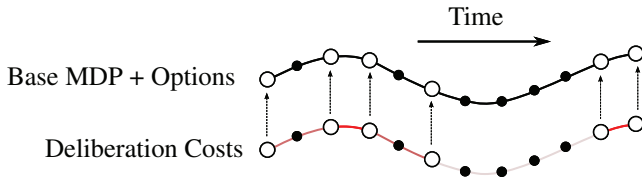
Figure 1: A deliberation cost is incurred upon switching to a new option and is subtracted from the reward of the base MDP. Open circles represent SMDP decision points while filled circles are primitive steps within an option. The cost rate for each option is represented by the intensity of the subtrajectory.

Formulation

In addition to the value function $\widetilde{V}_\theta(z) \doteq Q_\theta(s, o)$ for the base MDP and options over them, we define an immediate cost function $\widetilde{c}(z, a, z') \doteq c(s, o, a, s', o')$ and a corresponding deliberation cost function $\widetilde{D}_\theta(z) \doteq D_\theta(s, o)$. The expected sum of discounted costs associated with a set of options and the policy over them is given by the function $\widetilde{D}_\theta : (\mathcal{S} \times \mathcal{O}) \to \mathbb{R}$:

$$\widetilde{D}_\theta(z) = \mathbb{E}_\theta \left[\sum_{t=0}^\infty \gamma^t \widetilde{c}(Z_t, A_t, Z_{t+1}) \middle| Z_0 = z \right] .$$

We first formulate our goal of maximizing the expected return while keeping the deliberation cost low as a constrained optimization problem:

$$\max_\theta \sum_{s,o} \alpha(s, o) Q_\theta(s, o)$$

$$\text{subject to} : \sum_{s,o} \alpha(s, o) D_\theta(s, o) \le k ,$$

where α is an initial state distribution over state-option pairs. But in general, solving a problem of this form (Altman 1999) requires a Linear Programming (LP) formulation which is both expensive and incompatible with the model-free learning methods adopted in this work. Instead, we consider the unconstrained optimization problem arising from the Lagrangian formulation (Sennott 1991; Altman 1999):

$$\max_\theta J_\alpha(\theta) ,$$

$$\text{where } J_\alpha(\theta) \doteq \sum_{s,o} \alpha(s, o) \left(Q_\theta(s, o) - \eta D_\theta(s, o) \right), \quad (5)$$

and $\eta \in \mathbb{R}$ is a regularization coefficient. While (5) shows the option-value function and the deliberation cost function as separate entities, they can in fact be seen as a single MDP whose reward function is the difference of the base MDP reward and the cost function:

$$\mathbb{E}_{\alpha,\theta} \left[\sum_{t=0}^\infty \gamma^t \widetilde{r}(Z_t, A_t, Z_{t+1}) - \gamma^t \eta \widetilde{c}(Z_t, A_t, Z_{t+1}) \right] .$$

Therefore, there is a set of Bellman equations which the value function $\widetilde{V}_\theta^c(z) \doteq Q_\theta^c(s, o)$ over the transformed reward function satisfies:

$$\widetilde{V}_\theta^c(z) \doteq \sum_{a,z'} \pi_\theta(a|s, z) \widetilde{P}_\theta(z'|z, a,) \Big($$

$$\widetilde{r}(s, a, z') - \eta \widetilde{c}(z, a, z') + \gamma \widetilde{V}_\theta^c(z') \Big) . \quad (6)$$

Furthermore, there exist optimality equations in the sense of (Sutton *et al.* 1999b) for the parameters of the policy over options θ_μ:

$$Q_{\mathcal{O}}^\star(s, o) \doteq \max_{\theta_\mu \in \Pi(\mathcal{O})} \left(Q_{\theta_\mu}(s, o) - \eta D_{\theta_\mu}(s, o) \right) . \quad (7)$$

where the notation Q_{θ_μ} here indicates that the parameters for the options are kept fixed and only θ_μ is allowed to change. A policy over option $\mu^\star$ is η-optimal with respect to a set of options if it reaches the maximum in (7) for a given η. Clearly, when $\eta = 0$, the corresponding $\mu^\star$ is also optimal in the base MDP and there is no loss of optimality in this regard.

Switching Cost and its Interpretation as a Margin

One way to favor long options is by a cost function which penalizes for frequent options switches. In the same way that the MDP formulation allows for randomized reward functions (Puterman 1994), we can also capture the random event of switching through the immediate cost function c. Since $\beta_\theta(s', o)$ is the mean of a Bernoulli random variable over the two possible outcomes, switching or continuing (1 or 0), the cost function corresponding to the switching event is $c_\theta(s', o) = \gamma \beta_\theta(s', o)$ (where γ was added for mathematical convenience).

When expanding the value function over the transformed reward (6) for this choice of $c_\theta(s', o)$, we get:

$$Q_\theta^c(s, o) = \sum_a \pi_\theta(a|s, o) \left(r(s, a) + \gamma \sum_{s'} P(s'|s, a) \Big[\right.$$

$$\left. Q_\theta^c(s', o) - \beta_\theta(s', o) \left(A_\theta^c(s, o) + \eta \right) \Big] \right). \quad (8)$$

with η appearing along with the advantage function : a term which would otherwise be absent from the intra-option Bellman equations over the base MDP (2). Therefore, adding the switching cost function to the base MDP reward contributes a scalar *margin* η to the advantage function A_θ^c over the transformed reward. When learning termination functions in option-critic, the termination gradient for the unconstrained problem (5) is then of the form:

$$\frac{\partial J_\alpha(\theta)}{\partial \theta_\beta} = \mathbb{E}_{\alpha,\theta} \left[-\frac{\partial \beta_\theta(S_{t+1}, O_t)}{\partial \theta_\beta} \left(A_\theta^c(S_{t+1}, O_t) + \eta \right) \right] .$$
$$(9)$$

Hence, η sets a *margin* or a *baseline* for how good an option ought to be : a correction which might be due to approximation error or to reflect some form of uncertainty in the value estimates. By increasing its value, we can reduce the gap in the advantage function, tilting the balance in favor of maintaining an option rather than terminating it.

Computational Horizon

Due to the generality of our formulation, the discount factor of the deliberation cost function can be different from that of the value function over the base MDP reward. The unconstrained formulation of (5) then becomes a function of two discount factors: γ for base MDP and τ for the deliberation cost function:

$$J_\alpha^{\gamma,\tau}(\theta) = \sum_{s,o} \alpha(s,o) \left(Q_\theta^\gamma(s,o) - \eta D_\theta^\tau(s,o) \right)$$

Since the derivative of the deliberation cost function with respect to the termination parameters is:

$$\frac{\partial D_\theta^\tau(s,o)}{\partial \theta_\beta} = \frac{\partial}{\partial \theta_\beta} \sum_a \pi_\theta(a|s,o) \sum_{s'} P(s'|s,a) \Big(c_\theta(s',o) +$$

$$\tau \Big[(1-\beta_\theta(s',o)) D_\theta^\tau(s',o) +$$

$$\beta_\theta(s',o) \sum_{o'} \mu_\theta(o'|s') D_\theta^\tau(s',o') \Big] \Big),$$

setting $\tau = 0$ when the cost function is $c_\theta(s',o) \doteq \gamma \beta_\theta(s',o)$ leaves only one term : $\frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} \eta$. Hence, by linearity with (4), the derivative over the mixed objective is:

$$\frac{\partial J_\alpha^{\gamma,\tau=0}(\theta)}{\partial \theta_\beta} = \gamma \mathbb{E}_{\alpha,\theta} \left[-\frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} \left(A_\theta(s',o) + \eta \right) \right] . \tag{10}$$

While similar to (9) in the sense that the margin η also enters the advantage function, (10) differs fundamentally in the fact that it depends on A_θ and not A_θ^c, the advantage function over the transformed reward. We can also see that when $\gamma = \tau$, we recover the same form for the derivative of the expected return in the transformed MDP from (9):

$$\frac{\partial J_\alpha^{\gamma=\tau}(\theta)}{\partial \theta_\beta} = \gamma \mathbb{E}_{\alpha,\theta} \left[-\frac{\partial \beta_\theta(s',o)}{\partial \theta_\beta} \left(A_\theta^c(s',o) + \eta \right) \right]$$

$$= \frac{\partial J_\alpha(\theta)}{\partial \theta_\beta} .$$

The discount factor τ for the deliberation cost function provides a mechanism for truncating the sum of costs. Therefore, it plays a distinct role from the regularization coefficient η which merely scales the deliberation cost function but does not affect the *computational horizon*. As opposed to the random horizon set by the discount factor γ in the environment, τ pertains to the *internal* environment of the agent about the cost of its own *cognitive* or computational processes. It is a parameter about an introspective process of self-prediction of how likely a sequence of internal costs will be accumulated as a result of deliberating about courses of action in the *outside* environment. In accordance with more general results on discounting (Petrik and Scherrer 2008; Jiang *et al.* 2015), τ should be aligned with the representational capacity of the system since $\tau \to 1$ involves an increasingly more difficult prediction problem.

In that sense, $\tau = 0$ indicates that only the immediate computational cost should be considered when learning options to maximize the reward. When learning termination functions, the resulting shallow evaluation under small values of τ might not take into account the possibility that the overall expected cost could be lowered in exchange of a less favorable immediate cost : it lacks foresight. Despite the fact that the full effect of a change in the options or the policy over them cannot be captured with $\tau = 0$, the corresponding gradient (9) is still useful when $\eta > 0$. It leads to both the regularization strategy proposed in (Bacon *et al.* 2017) for gradient-based learning and (Mann *et al.* 2014) in the dynamic programming case. Furthermore, since (9) does not depend on A_θ^c, values can be learned only for the original reward function and does not require mixed or separate estimates.

Experiments

Previous results (Bacon *et al.* 2017) in the Arcade Learning Environment (Bellemare *et al.* 2013) have shown that while learning options end-to-end is possible, frequent terminations can become an issue unless regularization is used. Hence, we chose to apply the idea of deliberation cost in combination with a novel option-critic implementation based on the Advantage Asynchronous Actor-Critic (A3C) architecture of (Mnih *et al.* 2016). More specifically, our experiments are meant to assess : the interpretability of the resulting options, whether degeneracies (frequent terminations) to single-step options can be controlled, and if the deliberation cost can provide an inductive bias for learning faster.

Asynchronous Advantage Option-Critic (A2OC)

The option-critic architecture (Bacon *et al.* 2017) also introduced a deep RL algorithm to learn options in an end-to-end fashion directly from pixels. However, it is built on top of the DQN algorithm (Mnih *et al.* 2015), which is an off-line algorithm using samples from an experience replay buffer. Option-critic, on the other hand, is an on-line algorithm which uses every new sampled transition for its updates.

The asynchronous advantage actor-critic (A3C) algorithm (Mnih *et al.* 2016) provides a stable and on-line learning algorithm using multiple parallel agents. The parallel agents allows the deep networks to see samples from very different states, which greatly stabilizes learning. This algorithm is also much more consistent with the spirit of option-critic, as they both use on-line policy gradients to train. Hence, we developed [1] the asynchronous advantage option-critic (A2OC), an algorithm (alg. 1) that learns options in a similar way to A3C but within the option-critic architecture.

The architecture used for A2OC was kept as consistent with A3C as possible. We use a convolutional neural network of the same size, which outputs a feature vector that is shared among 3 heads as in (Bacon *et al.* 2017): the option policies, the termination functions and the Q-value networks. The option policies are linear softmax functions, the termination functions use sigmoid activation functions to represent probabilities of terminating and the Q-values

---

[1]The source code is available at https://github.com/jeanharb/a2oc_delib

Algorithm 1: Asynchronous Advantage Option-Critic

Initialize global counter $T = 1$
Initialize thread counter $t = 1$
$c = 0$
repeat
   $t_{start} = t$
   $s_t = s_0$
   Reset gradients: $dw = 0$, $d\theta_\beta = 0$ and $d\theta_\pi = 0$
   Choose o_t with an ϵ-soft policy over options $\mu(s_t)$
   repeat
      Choose a_t according to $\pi_\theta(\cdot|s_t)$
      Take action a_t in s_t, observe r_t, s_{t+1}
      $\widetilde{r}_t = r_t + c_t$
      if *the current option o_t terminates in s_{t+1}* **then**
         choose new o_{t+1} with ϵ-soft$(\mu(s_{t+1}))$
         $c_t = \eta$
      else
         $c_t = 0$
      end
      $t = t + 1$
      $T = T + 1$
   until *episode ends or $t - t_{start} == t_{max}$ or*
   $(t - t_{start} > t_{min}$ and o_t terminated)
   $G = V_\theta(s_t)$
   for $k \in t - 1, ..., t_{start}$ **do**
      $G = \widetilde{r}_k + \gamma G$
      Accumulate thread specific gradients:
      $dw \mathrel{-}= \alpha_w \frac{\partial (G - Q_\theta(s_k, o_k))^2}{\partial w}$
      $d\theta_\pi \mathrel{+}= \alpha_{\theta_\pi} \frac{\partial \log \pi_\theta(a_k|s_k)}{\partial \theta_\pi}(G - Q_\theta(s_k, o_k))$
      $d\theta_\beta \mathrel{-}=$
        $\alpha_{\theta_\beta} \frac{\partial \beta_\theta(s_{k+1}, o_k)}{\partial \theta_\beta}(Q_\theta(s_{k+1}, o_k) - V_\theta(s_{k+1}) + \eta)$
   end
   Update global parameters with thread gradients
until $t > t_{\max}$

---

are simply linear layers. During training, all gradients are summed together, and updating is performed in a single thread instance. A3C only needs to learn a value function for its policy, as opposed to Q-values for every action. Similarly, A2OC gets away with the action dimension through sampling (Bacon *et al.* 2017) but needs to maintain state-option value because of the underlying augmented state space. ???

As for the hyperparameters, we use an ϵ-greedy policy over options, with $\epsilon = 0.1$. The preprocessing are the same as the A3C, with RGB pixels scaled to 84×84 grayscale images. The agent repeats actions for 4 consecutive moves and receives stacks of 4 frames as inputs. We used entropy regularization of 0.01, which pushes option policies not to collapse to deterministic policies. A learning rate of 0.0007 was used in all experiments. We usually trained the agent with 16 parallel threads.

Empirical Effects of Deliberation Cost

We used Amidar, a game of the Atari 2600 suite, to analyze the option policies and terminations qualitatively. The game is grid-like and the task is to cover as much ground as possi-

ble without running into enemies.

Without a deliberation cost, the options eventually learn to terminate at every step, as seen in figure 2a, where every color in the figure represents a different option being executed at the time the agent was in the location. In contrast, figure 2b shows the effect of training an agent with a deliberation cost, which persists with an option over a long period of time. The temporally extended structure of the options shown by color does not result from simply terminating and re-picking the same option at every step but truly represents a contiguous segment of the trajectory where that option was maintained in a call-and-return fashion. Only at certain intersections the options terminate, allowing the agent to select an option which will lead it in a different direction. As opposed to the agent which was trained without a deliberation cost (fig. 2a), figure 2a shows that the options learned with the regularizer were specialized and only selected in specific scenarios. Figure 2c shows us where the agent terminated options on its trajectory. The options are clearly terminating at intersections, which represent key decision points.

Algorithm	Amidar	Asterix	Breakout	Hero
(Mnih *et al.* 2015)	739.5	6012.0	**401.0**	19950.0
(Mnih *et al.* 2016)	283.9	6723.0	551.6	**28765.8**
No deliberation cost	502.9	**7542.7**	365.7	12253.6
$\tau = 0., \eta = 0.010$	829.8	6169.5	398.5	25310.2
$\tau = 0., \eta = 0.020$	799.2	7798.8	389.8	22691.7
$\tau = 0., \eta = 0.030$	808.1	4986.3	381.5	20765.7
$\tau = \gamma, \eta = 0.005$	775.6	5326.6	386.2	10752.8
$\tau = \gamma, \eta = 0.010$	808.6	6850.6	395.8	26910.6
$\tau = \gamma, \eta = 0.015$	**809.5**	5830.3	373.0	24737.4
$\tau = \gamma, \eta = 0.020$	741.7	6335.3	384.2	28843.7
$\tau = \gamma, \eta = 0.025$	780.3	4798.1	377.8	24973.7
$\tau = \gamma, \eta = 0.030$	764.6	4840.0	376.6	19849.3

Table 1: Final performance for different levels of regularization, averaged over 5 runs. Note that the A3C scores in (Mnih *et al.* 2016) use a nonpublic human starts evaluation which may not be directly comparable to our random start initialization.

We also trained agents with multiple levels of deliberation cost, from $\eta = 0$ to $\eta = 0.03$, with increments of 0.005. The range of values was chosen according to the general scale of the values proper to these environments. An increase in deliberation cost η quickly decreases the average termination probabilities as expected by the formulation (5) (see appendix). When no deliberation cost is used, termination raises up to 100% very quickly, meaning each option only lasts a single time-step. The decrease in probability is not the same in every environment, this is due to the difference in returns. The deliberation cost has an effect proportional to its ratio with the state values. Intuitively, environments with many high rewards would indeed require a larger deliberation cost to have substantial effects.

Conclusion and Future Work

We proposed using a deliberation cost to incentivize the creation of options which persist longer. Using this approach in the option-critic architecture yields both good performance
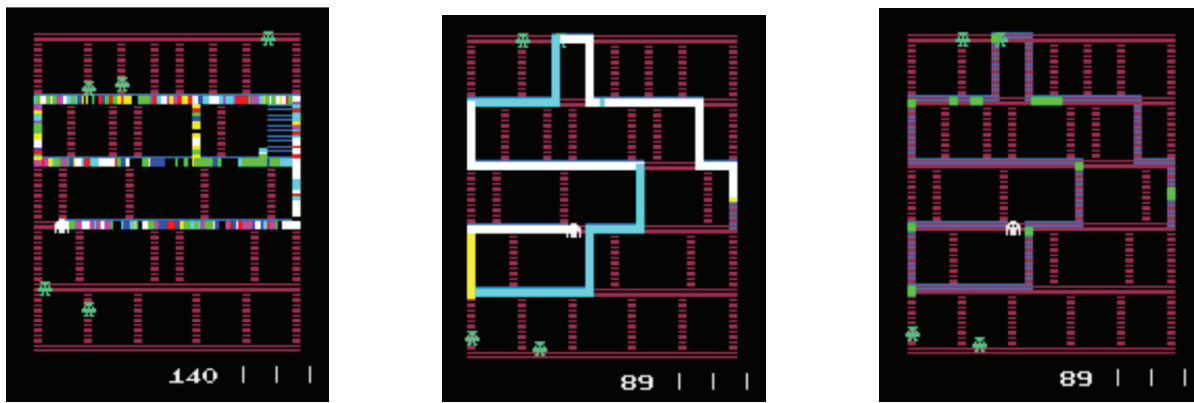
(a) Without a deliberation cost, options terminate instantly and are used in any scenario without specialization.

(b) Options are used for extended periods and in specific scenarios through a trajectory, when using a deliberation cost.

(c) Termination is sparse when using the deliberation cost. The agent terminates options at intersections requiring high level decisions.

Figure 2: We show the effects of using deliberation costs on both the option termination and policies. In figures (a) and (b), every color in the agent trajectory represents a different option being executed. This environment is the game Amidar, of the Atari 2600 suite.

as well as options which are intuitive and do not shrink over time. In doing so, we also outlined a connection from our more general notion of deliberation cost with previous notions of regularization from (Mann *et al.* 2014) and (Bacon *et al.* 2017).

The deliberation cost goes beyond only the idea of penalizing for lengthy computation. It can also be used to incorporate other forms of bounds intrinsic to an agent in its environment. One interesting direction for future work is to also think of deliberation cost in terms of *missed opportunity* and opening the way for an implicit form of regularization when interacting asynchronously with an environment. Another interesting form of limitation inherent to reinforcement learning agents has to do with their representational capacities when estimating action values. Preliminary work seems to indicate that the error decomposition for the action values could be also be expressed in the form of a deliberation cost.

Acknowledgments

References

E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *ICML*, pages 166–175, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, pages 1726–1734, 2017.

Leemon C. Baird. Advantage updating. Technical Report WL–TR-93-1146, Wright Laboratory, 1993.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 06 2013.

Matthew M. Botvinick, Yael Niv, and Andrew C. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3):262 – 280, 2009.

S. R. K. Branavan, Nate Kushman, Tao Lei, and Regina Barzilay. Learning high-level planning from text. In *ACL*, pages 126–135, 2012.

C. Daniel, H. van Hoof, J. Peters, and G. Neumann. Probabilistic inference for determining options in reinforcement learning. *Machine Learning, Special Issue*, 104(2):337–357, 2016.

Peter Dayan and Geoffrey E. Hinton. Feudal reinforcement learning. In *NIPS*, pages 271–278, 1992.

Thomas G. Dietterich. The MAXQ method for hierarchical reinforcement learning. In *ICML*, pages 118–126, 1998.

Gary L. Drescher. *Made-up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1991.

Richard Fikes, Peter E. Hart, and Nils J. Nilsson. Learning and executing generalized robot plans. *Artif. Intell.*, 3(1-3):251–288, 1972.

Gerd Gigerenzer and R. Selten. *Bounded Rationality: The adaptive toolbox*. Cambridge: The MIT Press, 2001.

Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard L Lewis, and Xiaoshi Wang. Deep learning for real-time atari game play using offline monte-carlo tree search planning. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence,

and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3338–3346. Curran Associates, Inc., 2014.

Ronald A. Howard. Semi-markovian decision processes. In *Proceedings 34th Session International Statistical Institute*, pages 625–652, 1963.

Glenn A. Iba. A heuristic approach to the discovery of macro-operators. *Machine Learning*, 3:285–317, 1989.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard L. Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1181–1189, 2015.

Leslie Pack Kaelbling. Hierarchical learning in stochastic domains: Preliminary results. In *ICML*, pages 167–173, 1993.

Richard Earl Korf. *Learning to Solve Problems by Searching for Macro-operators*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1983.

Benjamin Kuipers. Commonsense knowledge of space: Learning from experience. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'79, pages 499–501, San Francisco, CA, USA, 1979. Morgan Kaufmann Publishers Inc.

Tejas Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Joshua Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems 29*, 2016.

Marlos C. Machado, Marc G. Bellemare, and Michael H. Bowling. A laplacian framework for option discovery in reinforcement learning. In *ICML*, pages 2295–2304, 2017.

Daniel J. Mankowitz, Timothy Arthur Mann, and Shie Mannor. Adaptive skills, adaptive partitions (ASAP). In *Advances in Neural Information Processing Systems 29*, 2016.

Timothy Arthur Mann, Daniel J. Mankowitz, and Shie Mannor. Time-regularized interrupting options (TRIO). In *ICML*, pages 1350–1358, 2014.

Timothy Arthur Mann, Shie Mannor, and Doina Precup. Approximate value iteration with temporally extended actions. *J. Artif. Intell. Res.*, 53:375–438, 2015.

Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, January 1961.

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

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, pages 1928–1937, 2016.

Abraham Neyman. Bounded complexity justifies cooperation in the finitely repeated prisoners dilemma. *Economics Letters*, 19(3):227–229, jan 1985.

Ronald Parr and Stuart J. Russell. Reinforcement learning with hierarchies of machines. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 1043–1049. MIT Press, 1998.

Marek Petrik and Bruno Scherrer. Biasing approximate dynamic programming with a lower discount factor. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1265–1272, 2008.

Doina Precup. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amhersts, 2000.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

Linn I. Sennott. Constrained discounted markov decision chains. *Probability in the Engineering and Informational Sciences*, 5(4):463475, 1991.

Herbert A. Simon. *Models of man: social and rational; mathematical essays on rational human behavior in society setting*. Wiley, 1957.

Alec Solway, Carlos Diuk, Natalia Crdova, Debbie Yee, Andrew G. Barto, Yael Niv, and Matthew M. Botvinick. Optimal behavioral hierarchy. *PLOS Computational Biology*, 10(8):1–10, 08 2014.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 1999.

Richard S. Sutton, Doina Precup, and Satinder P. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, 1999.

Richard S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 1984.

Sebastian Thrun and Anton Schwartz. Finding structure in reinforcement learning. In *NIPS*, 1995.