Safe Option-Critic: Learning Safety in the Option-Critic Architecture

Arushi Jain
McGill University
Montreal
arushi.jain@mail.mcgill.ca

Khimya Khetarpal
McGill University
Montreal
khimya.khetarpal@mail.mcgill.ca

Doina Precup
McGill University
Montreal
dprecup@cs.mcgill.ca

ABSTRACT

Designing hierarchical reinforcement learning algorithms that induce a notion of *safety* is not only vital for safety-critical applications, but also, brings better understanding of an artificially intelligent agent's decisions. While learning end-to-end options automatically has been fully realized recently, we propose a solution to *learning safe options*. We introduce the idea of controllability of states based on the temporal difference errors [11] in the option-critic framework. We then derive the policy-gradient theorem with controllability and propose a novel framework called *safe option-critic*. We demonstrate the effectiveness of our approach in the four-rooms grid-world, cartpole, and three games in the Arcade Learning Environment (ALE): MsPacman, Amidar and Q*Bert. Learning of end-to-end options with the proposed notion of safety achieves reduction in the variance of return and boosts the performance in environments with intrinsic variability in the reward structure. More importantly, the proposed algorithm outperforms the vanilla options in all the environments and primitive actions in two out of three ALE games.

KEYWORDS

Reinforcement Learning; AI Safety; Temporal abstractions; Safe Option-Critic framework;

1 INTRODUCTION

Safety in Artificial Intelligence (AI) can be viewed from many perspectives. Traditionally, introducing some form of risk-awareness in AI systems has been a prime way of defining safety in the machines. More recently, researchers have broadened the horizon of safety in AI to address different sources of errors and faulty behaviors [3]. The 23 Asilomar AI principles [9] comprise of varied aspects of safety like risk-averseness, transparency, robustness, fairness and also legal and ethical values an agent should hold. In this work, we refer to the following definition of safety: prevent undesirable behavior, in particular, reducing the visits to the undesirable states during the learning process in reinforcement learning (RL).

RL agents primarily learn by optimizing their discounted cumulative rewards [34]. While rewards are a good indicator of how to behave, they do not necessarily always lead to the most desired behavior. Optimal reward design [31] still poses a challenge for the algorithm designers with several issues such as misspecified rewards [2, 13] and corrupted reward channels [7] to name a few. Alternatively, learning with constraints allows us to introduce more clarity in the objective function [1].

During exploration, agents are naturally unaware of the states which may be prone to errors or may lead to catastrophic consequences. Risk-awareness has been introduced in the agents by directing exploration safely [20], optimizing the worst-case performance [39], measuring the probabilities of visiting erroneous states [12] and several other approaches. Garcıa and Fernández (2015) presents a comprehensive survey covering a broad range of techniques to realize safety in RL. In a Markov Decision Process (MDP), majority of the methods seek to minimize the variance of return as a risk mitigation strategy. Many authors [11, 25, 29, 30, 37, 38] have used temporal difference (TD) learning for estimating the variance of return to capture the notion of uncertainty in the value of a state.

While some of the aforementioned approaches leverage TD learning in estimating errors and risks, all of them define notions of safety in the primitive action space. Temporally abstract actions provide an approach to represent the information in a hierarchical format. The concept of learning and planning in a hierarchical fashion is very close to how humans think and approach a problem. Temporal abstractions have been vital to the AI community since 1970s [5, 8, 15, 18, 23, 24]. Prior research has shown that the temporal abstractions improve exploration, reduce complexity of choosing the actions and enhance robustness to the misspecified models. The *options* framework [28, 36] provided an intuitive way to plan, reason, and act in a continual fashion as opposed to learning with the primitive actions. Many authors [6, 16, 17, 19, 22, 32, 41] provide methods for discovering subgoals and then the learning policies to achieve those subgoals.

The option-critic framework [4] enables end-to-end learning of the options. However, defining a *safe* option which does not lead to the erroneous states during the learning process still remains an open question. We introduce the idea of *controllability* [11] in the options framework as an additional condition in the optimality criterion which constrains the variance of the TD error as a measure of uncertainty about the value of a state-option pair. In this work, we propose a new framework called *safe option-critic* for learning the safety in options.

Key Contributions: This work incorporates the notion of safety in the option-critic framework and presents a mechanism to automatically learn safe options. We derive the policy-gradient theorem for the safe option-critic framework using constraint based optimization. We then demonstrate through experiments in the four-rooms grid environment, that learning the options with controllability (term quantifying controllable behavior of an agent) results in the safer policies which avoid states with the high variance in the TD error. Empirically, we show the benefits of learning safe options in the ALE environments with high intrinsic variability in the rewards. Our approach outperforms the vanilla options with

no notion of safety in 3 Atari games namely, MsPacman, Amidar and Q*Bert. In 2 out of 3 games, learning the safe options also outperforms the primitive actions. To this end, we propose a novel **Safe Option-Critic** framework for the future research in the AI Safety paradigm.

2 PRELIMINARIES

In RL, an agent interacts with the environment at discrete time steps $t \in \{1, 2, ...\}$ where it observes a state $s_t \in S$. The agent then chooses an action $a_t \in A$ from a policy which defines a probability distribution of actions over the state space $\pi : S \times A \rightarrow [0, 1]$. After choosing an action, the agent transitions to a new state s_{t+1} according to the transition function $P : S \times A \rightarrow (S \rightarrow [0, 1])$ and receives a reward r_{t+1} where the reward function is defined as $r : S \times A \rightarrow \mathbb{R}$. A MDP is defined by a tuple $< S, A, \gamma, r, P >$ where γ is a discount factor. A discounted state-action value function is defined as $Q(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a]$ with $\gamma \in [0, 1]$. The value of Q can be learned in an incremental fashion using one-step TD learning also written as TD(0) which is a special case of TD(λ) [33]. The state-action value is updated using the equation: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta$. Here α is the step size and δ is TD(0) error which is defined as $\delta = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$.

The policy gradient theorem [35] presents a way of updating the parameterized policy according to the gradient of expected discounted return which is defined as $\rho(\pi, s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \pi]$. The gradient with respect to the policy parameter θ is given as:

$$\frac{\partial \rho(\pi, s_0)}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s, a)}{\partial \theta} Q_\pi(s, a) \tag{1}$$

where $d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s | s_0, \pi)$ is the discounted weighting of the states with the starting state as s_0.

2.1 OPTIONS

The options framework [28, 36] facilitates a way to incorporate the temporally abstract knowledge into RL with no change in the existing setup. An option $\omega \in \Omega$ is defined as a tuple $(I_\omega, \pi_\omega, \beta_\omega)$; where I_ω is the initiation set containing the initial states from which an option ω can start, π_ω is the option policy defining a distribution over actions given a state and β_ω is the termination condition of an option ω defined as the probability of terminating in a state. An example of options could be having high level sub-goals like going to a market, buying vegetables and making the dish wherein the primitive actions for instance could be the muscle twitches.

In case of options being Markov, the intra-option Bellman equation [36] provides an off-policy method for updating the Q value of a state-option pair which can be written as:

$$\begin{aligned} Q(s_t, \omega) = Q(s_t, \omega) + \alpha[r_{t+1} \\ + \gamma(1 - \beta_\omega(s_t))Q(s_{t+1}, \omega) \\ + \gamma\, \beta_\omega(s_t) \max_{\omega' \in \Omega} Q(s_{t+1}, \omega') - Q(s_t, \omega)] \end{aligned} \tag{2}$$

where ω is selected from the policy over options π_Ω.

2.2 LEARNING OPTIONS

The intra-option value learning [36] lays the foundation for learning the options in the option-critic architecture [4]. It is a policy-gradient based method for learning the intra-options policies and the termination conditions of the options. [4] considered the call and return option execution model, where an option ω is chosen according to the policy over options π_Ω, wherein the intra-option policy π_ω is followed until the termination condition β_ω is met. Once the current option terminates, another option to be executed at that state is selected in the same fashion. $\pi_{\omega,\theta}$ denotes the parameterized intra-option policy in terms of θ and $\beta_{\omega,\nu}$ represents the option termination which is parameterized by ν. The value of executing an action a at a particular state-option pair is then given by $Q_U : S \times \Omega \times A \rightarrow \mathbb{R}$ where

$$Q_U(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) U(s', \omega) \tag{3}$$

where U represents the value of executing an option ω at a state s':

$$U(s', \omega) = (1 - \beta_{\omega,\upsilon}(s'))Q_\Omega(s', \omega) \; + \; \beta_{\omega,\upsilon}(s')V_\Omega(s') \tag{4}$$

Here, Q_Ω represents the optimal-value function for a given option ω given by $Q_\Omega(s, \omega) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a)$. V_Ω represents the optimal-value function over Ω given by $V_\Omega(s) = \sum_\omega \pi_\Omega(\omega|s) Q_\Omega(s, \omega)$. [4] derived the gradient of discounted return with respect to θ and the initial condition (s_0, ω_0) as:

$$\begin{aligned} \frac{\partial \rho(\pi, s_o, \omega_0)}{\partial \theta} = \sum_{s,\omega} [\mu(s, \omega | s_0, \omega_0) \\ \times \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a)] \end{aligned} \tag{5}$$

where $\mu(s, \omega | s_0, \omega_0)$ is the discounted weighting of a state-option pair with $\mu(s, \omega | s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega | s_o, \omega_0)$. The gradient of the expected discounted return with respect to the option termination parameter ν and the initial condition (s_1, ω_0) is described as:

$$\frac{\partial \rho(\pi, s_1, \omega_0)}{\partial \upsilon} = -\sum_{s',\omega} \mu(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu} A(s', \omega) \tag{6}$$

where A is an advantage function $A_\Omega(s, \omega) = Q_\Omega(s, \omega) - V_\Omega(s)$.

3 SAFE OPTION-CRITIC MODEL

Taking inspiration from Gehring and Precup's work (2013), we define controllability as a negation of the variance in the TD error of a state-option action pair. We use the aforementioned definition of controllability to introduce the concept of safety in the option-critic architecture which aids in measuring the uncertainty about the value of a state-option pair. Higher the variance in TD error of a state-option pair, higher would be the uncertainty in the value of that state-option pair. In the safety critical applications, the agent should learn to eventually avoid such pairs as they induce variability in the return. We optimize for the expected discounted return along with the controllability value of initial state-option pair. Depending on the nature of the application, one can limit or encourage the agent visiting a state-option pair based on the degree of controllability. Introducing controllability using the TD error facilitates the linear scalability of the method with the increase in the number of state-option pairs.

Continuing with the notations used in [4], we are introducing a parameter vector described by $\Theta = [\theta, \nu]$ where θ is an intra-option policy parameter and ν is an option termination parameter. We assume that an option can be initialized from any state $s \in S$. Given a state-option pair, uncertainty in its value is measured by controllability C, which is given by the negation of the variance in its TD error δ. The expected value of the TD error would converge to zero, hence controllability is written as:

$$C_\Theta(s,\omega) = -\mathbb{E}_{a\sim\pi_{\omega,\theta}(a|s)}\left[\delta^2(s,\omega,a)\right] \tag{7}$$

From now onwards, we would refer $\delta(s,\omega,a)$ as δ whose value is given by:

$$\delta = r(s,a) + \gamma \sum_{s'} P(s'|s,a) U_\Theta(s',\omega) - Q_{U,\Theta}(s,\omega,a) \tag{8}$$

where $Q_{U,\Theta}(s,\omega,a)$ and $U_\Theta(s,\omega)$ are defined in (3) and (4) respectively. The aim here is to maximize the expected discounted return along with the controllability criterion of a state-option pair. We call this objective J, where we want to:

$$\begin{aligned} &\max_\Theta J(\Theta|d), \\ &\text{where } J(\Theta|d) = \mathbb{E}_{(s_0,\omega_0)\sim d}[Q_\Theta(s_0,\omega_0) + \psi\, C_\Theta(s_0,\omega_0)] \end{aligned} \tag{9}$$

where $\psi \in \mathbb{R}$ acts as a regularizer for the controllability and d is the initial state-option pair distribution. The Q value of a state-option pair is defined as $Q_\Theta(s,\omega) = \sum_a \pi_{\omega,\theta}(a|s)\, Q_{U,\Theta}(s,\omega,a)$. The above objective can also be interpreted as a constrained optimization problem with an additional constraint on the controllability function. We will now derive the gradient of the performance evaluator J with respect to the intra-option policy parameter θ assuming they are differentiable. First we will take the gradient of C with θ. Following from (7):

$$\begin{aligned} \frac{\partial C_\Theta(s,\omega)}{\partial\theta} = &-\sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial\theta}\,\delta^2 \\ &- \sum_a 2\delta \frac{\partial\delta}{\partial\theta}\,\pi_{\omega,\theta}(a|s) \end{aligned} \tag{10}$$

where the gradient of TD error δ w.r.t. θ using (8):

$$\frac{\partial\delta}{\partial\theta} = \gamma \sum_{s'} P(s'|s,a)\,\frac{\partial U_\Theta(s',\omega)}{\partial\theta} - \frac{\partial Q_{U,\Theta}(s,\omega,a)}{\partial\theta} \tag{11}$$

Next, the gradient of $Q_\Theta(s,\omega)$ w.r.t. θ is:

$$\begin{aligned} \frac{\partial Q_\Theta(s,\omega)}{\partial\theta} = &\sum_a \frac{\partial\pi_{\omega,\theta}(a|s)}{\partial\theta}\, Q_{U,\Theta}(s,\omega,a) \\ &+ \sum_a \frac{\partial Q_{U,\Theta}(s,\omega,a)}{\partial\theta}\,\pi_{\omega,\theta}(a|s) \end{aligned} \tag{12}$$

The gradient of $J(\Theta|d)$ w.r.t. θ following from (9), (10), (11) and (12) is reduced to:

$$\begin{aligned} \frac{\partial J}{\partial\theta} = &(1+2\delta\psi)\sum_a \pi_{\omega,\theta}(a|s)\,\frac{\partial Q_{U,\Theta}(s,\omega,a)}{\partial\theta} \\ &+ \sum_a \frac{\partial\pi_{\omega,\theta(a|s)}}{\partial\theta}\{\, Q_{U,\Theta}(s,\omega,a) - \psi\delta^2\} \\ &- 2\delta\psi\gamma \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'} P(s'|s,a)\,\frac{\partial U_\Theta(s',\omega)}{\partial\theta} \end{aligned} \tag{13}$$

where the gradient of $Q_{U,\Theta}(s,\omega,a)$ using (3) is:

$$\frac{\partial Q_{U,\Theta}(s,\omega,a)}{\partial\theta} = \gamma \sum_{s'} P(s'|s,a)\frac{\partial U_\Theta(s',\omega)}{\partial\theta} \tag{14}$$

and the gradient of $U_\Theta(s',\omega)$ using (4) is:

$$\begin{aligned} \frac{\partial U_\Theta(s',\omega)}{\partial\theta} = \sum_{\omega'} \Big[&(1-\beta_{\omega,\nu}(s'))\mathbb{1}_{\omega'=\omega} \\ &+ \beta_{\omega,\nu}(s')\,\pi_\Omega(\omega'|s')\Big]\,\frac{\partial Q_\Theta(s',\omega')}{\partial\theta} \end{aligned} \tag{15}$$

Substituting the above gradient value of Q and U from (14) and (15) in (13), the gradient of J w.r.t. θ becomes:

$$\begin{aligned} \frac{\partial J}{\partial\theta} = &\sum_{s',\omega'} P^{(1)}_\gamma(s',\omega'|s,\omega)\frac{\partial Q_\theta(s',\omega')}{\partial\theta} + \sum_a \frac{\partial\pi_{\omega,\theta(a|s)}}{\partial\theta} Q_{U,\Theta}(s,\omega,a) \\ &- \sum_a \frac{\partial\pi_{\omega_0,\theta}(a|s_0)}{\partial\theta}\psi\delta^2(s_0,\omega_0,a) \end{aligned} \tag{16}$$

where $P^{(1)}_\gamma(s',\omega'|s,\omega) = \gamma\sum_a \pi_\omega(a|s)P(s'|s,a)(1-\beta_\omega(s'))\mathbb{1}_{\omega=\omega'} + \beta_\omega(s')\pi_\Omega(\omega'|s')$. [4] derived the gradient of $Q(s,\omega)$ as:

$$\begin{aligned} \frac{\partial Q_\theta(s,\omega)}{\partial\theta} = &\sum_{s',\omega'} P^{(1)}_\gamma(s',\omega'|s,\omega)\frac{\partial Q_\theta(s',\omega')}{\partial\theta} \\ &+ \sum_a \frac{\partial\pi_{\omega,\theta(a|s)}}{\partial\theta} Q_{U,\Theta}(s,\omega,a) \end{aligned} \tag{17}$$

On expanding the gradient of $Q_\Theta(s,\omega)$ as in (17), the gradient of J following (16) becomes:

$$\begin{aligned} \frac{\partial J}{\partial\theta} = &\sum_{k=0}^{\infty}\sum_{s,\omega}\Big\{P^{(k)}_\gamma(s,\omega|s_0,\omega_0) \\ &\times \sum_a \frac{\partial\pi_{\omega,\theta}(a|s)}{\partial\theta} Q_{U,\Theta}(s,\omega,a)\Big\} \\ &- \sum_a \frac{\partial\pi_{\omega_0,\theta}(a|s_0)}{\partial\theta}\psi\delta^2(s_0,\omega_0,a) \end{aligned} \tag{18}$$

Here, (s_0,ω_0) corresponds to the initial state-option pair. The gradient of J here describes that each option aims to maximize its own reward with controllability as a constraint pertaining to that option only. Our interpretation here is that each option learnt with this safety constraint translates to an overall risk-averse behavior.

Now we will compute the gradient of $J(\Theta|d)$ with respect to the option termination function parameter ν. The gradient of controllability C with ν can be written following (7) and (8) as:

$$\frac{\partial C_\Theta(s,\omega)}{\partial\nu} = -2\delta\sum_a \pi_{\omega,\theta}(a|s)\,\frac{\partial\delta}{\partial\nu} \tag{19}$$

where $\frac{\partial\delta}{\partial\nu} = \gamma\sum_{s'} P(s'|s,a)\frac{\partial U_\Theta(s',\omega)}{\partial\nu} - \frac{\partial Q_{U,\Theta}(s,\omega,a)}{\partial\nu}$. The gradient of $Q_\Theta(s,\omega)$ w.r.t ν is written as:

$$\frac{\partial Q_\Theta(s,\omega)}{\partial\nu} = \sum_a \pi_{\omega,\theta}(a|s)\,\gamma\sum_{s'} P(s'|s,a)\frac{\partial U_\Theta(s,\omega)}{\partial\nu} \tag{20}$$

Using (19) and (20) the gradient of J w.r.t. ν is:

$$\begin{aligned}\frac{\partial J}{\partial \nu} &= \sum_{a} \pi_{\omega,\theta}(a|s) \sum_{s'} \gamma\, P(s'|s,a) \frac{\partial U_{\Theta}(s',\omega)}{\partial \nu} \\ &= \frac{\partial Q_{\Theta}(s,\omega)}{\partial \nu}\end{aligned} \tag{21}$$

Therefore, the gradient of J w.r.t. ν becomes equal to that of $Q_{\Theta}(s,\omega)$ which is equal to the Termination Gradient Theorem [4] in (6). The interpretation of the derivation is in accordance with the way the notion of safety has been conceptualized, that is, each option is responsible for making its intra-option policy safe by incorporating the factor of controllability. We are using one-step i.e. *TD(0)* while updating the Q value of a state-option pair. Due to the assumption that each option take care of its own safety through it's intra-option policy, one is only concerned about choosing an option which maximizes the expected discounted return from next state-option pair while terminating an option. Due to this assumption as shown in derivation above, introducing controllability does not impact the termination of an option. Algorithm 1 shows the implementation details of controllability in the option-critic architecture in a tabular setting.

Algorithm 1 Safe Option-Critic with tabular intra-option Q learning

Here $\alpha, \alpha_{\theta}, \alpha_{\nu}$ stands for step size of critic, intra-option policy and termination respectively. ψ is controllability regularization parameter.
$s \leftarrow s_0$
Select ω using softmax policy over options
Let initial ω be ω_0
repeat
  $a \sim \pi_{\omega,\theta}(a|s)$ using softmax intra-option policy
  Let initial a taken at (s_0, ω_0) be a_0
  Maintain (s_0, ω_0, a_0) at the beginning of the episode
  Observe $\{r, s'\}$
  if s' is non-terminal state **then**
    $\delta \leftarrow r + \gamma \Big[(1-\beta_{\omega,\nu}(s'))Q_{\Theta}(s',\omega) + \beta_{\omega,\nu}(s') \max_{\omega' \sim \Omega} Q_{\Theta}(s',\omega')\Big] - Q_{U,\Theta}(s,\omega,a)$
  else
    $\delta \leftarrow r - Q_{U,\Theta}(s,\omega,a)$
  end if
  if $(s_0,\omega_0) == (s,\omega)$ **then**
    Update $(s_0,\omega_0,a_0) \leftarrow (s_0,\omega_0,a)$
  end if
  $Q_{U,\Theta}(s,\omega,a) \leftarrow Q_{U,\Theta}(s,\omega,a) + \alpha\delta$
  $\theta \leftarrow \theta + \alpha_{\theta} \frac{\partial \log(\pi_{\omega,\theta}(a|s))}{\partial \theta} Q_{U,\Theta}(s,\omega,a) - \alpha_{\theta} \frac{\partial \log(\pi_{\omega_0,\theta}(a_0|s_0))}{\partial \theta} \psi \delta^2(s_0,\omega_0,a_0)$
  $\nu \leftarrow \nu - \alpha_{\nu} \frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu} (Q_{\Theta}(s',\omega) - V_{\Omega}(s'))$
  if $\beta_{\omega,\nu}(s')$ terminates **then**
    Choose new $\omega \sim \pi_{\Omega}(\omega|s')$
  end if
  $s \leftarrow s'$
until s' is a terminal state

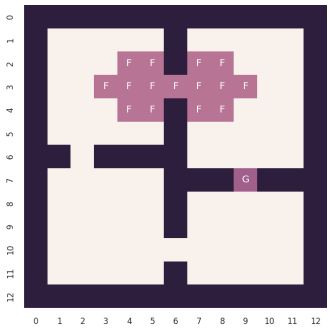


Figure 1: Four Room Environment: F and G depicts the unsafe frozen and goal states respectively. The lightest color represents the normal states whereas the darkest color shows the wall.

4 EXPERIMENTS

4.1 Grid World

First, we consider a simple navigation task in a two dimensional grid environment using a variant of the four-rooms domain as described in [36]. As seen in the Fig. 1, similar to [11], we define some *slippery* frozen states in the environment which are unsafe to visit. We accomplish this by introducing variability in their rewards. States labeled F and G indicate the frozen and goal states respectively.

An agent can be initialized with any random start state in the environment apart from the goal state. The action space consists of four stochastic actions namely, *up*, *down*, *left*, and *right*. The random actions are taken with 0.2 probability in the environment. The task is to navigate through the rooms to a fixed goal state as depicted in Fig. 1. The dark states in Fig. 1 depict the walls. The agent remains in the same state with a reward of 0 if the agent hits the wall. A reward of 0 and 50 is given to the agent while transitioning into the normal and the goal state respectively. The rewards for the unsafe states are drawn uniformly from $[-15, 15]$ while the agent transitions to a slippery state. The expected value of the reward for the normal and the slippery states is kept same.

In the safe option-critic framework, we learn both the policy over options and the intra-option policies with the Boltzmann distribution. We ran the experiments with varying controllability factor ψ for learning 4 options. We optimize for the hyperparameters: temperature and α for both Option-Critic (OC) with $\psi = 0$ and safe OC. The discount factor γ is set to 0.99. The step size of the intra-option policy is set to 0.01. The best performance is achieved for $\psi = 0$ with the step size of termination and critic as 0.01 and 0.1 respectively. The optimal value of controllability was achieved at $\psi = 0.05$ with the step size of termination and critic at 0.1 and 0.5 respectively. The temperature for the Boltzmann distribution is set to 0.001. The results are achieved with total of 600 episodes averaged over 200 trials where training in each trial starts from the scratch. In each episode, the agent is allowed to take only 500 steps, wherein if the agent fails to reach the goal state within those steps then the episode terminates.

To evaluate these experiments, we consider the following metrics: the learned policy, average cumulative discounted return of episodes and the density of the state visits. It can be observed from

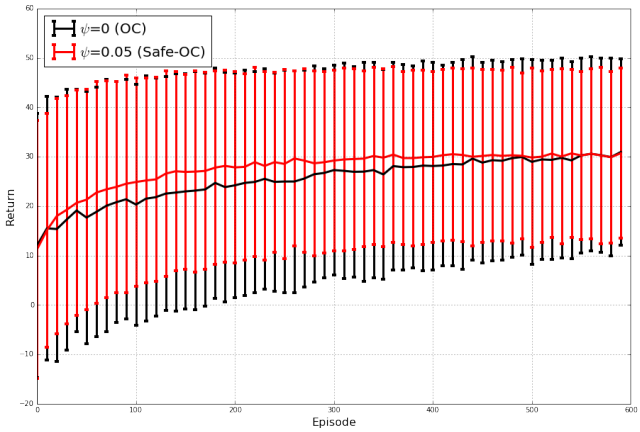
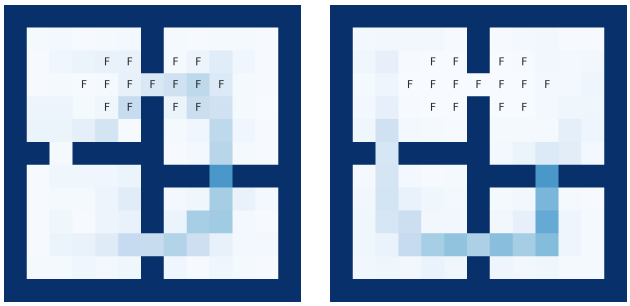


Figure 2: Learning curve with 4 options in Four Room Environment: Graph depicts the averaged return over 200 trials in the four room environment with 4 options. The bands around solid lines represent the standard deviation of the return. The experiment with controllability has lesser standard deviation in the observed return value as compared to the one without controllability.



(a) OC ($\psi = 0$) **(b) Safe-OC ($\psi = 0.05$)**

Figure 3: State frequency in Four Room Environment: Density graph represents the number of times a state was visited during testing over 80 trials. Darker shade represents the higher density. a) Model without safety has equally likely density for both the hallways. b) Model with safety shows higher density for the path without the frozen states.

Fig. 2 that the options with the controllability (Safe-OC) have lower variance in the return of an episode as compared to the options without the controllability (OC). This highlights the fact that the controllability helps the agent in avoiding the unsafe states (inducing variability in the return value). To validate that learning with the controllability causes fewer visits to the unsafe state, we visualize the state frequency graph depicted in the Fig. 3. It is observed that the options with the controllability have lower frequency of visit to the unsafe states as opposed to the vanilla options.

The learning of safe options induces transparency to the behavior of an agent. This is most explicitly demonstrated through the path taken by the agent in case of both controllability and no controllability in the options as shown in Fig. 4. Regardless of the start state, Safe-OC agent navigates to the goal state avoiding the states with a high variance in the reward as opposed to the OC agent which finds a shortest route being unaware of the error prone states.

4.2 CartPole Environment

We consider linear function approximation with the options. In the Cartpole[1] environment a pole is attached to the cart which can move along the horizontal axis. The environment has four continuous features: *position*, *velocity*, *pole angle* and *angular velocity* of the pole. There are two discrete actions that can be taken, namely *left* or *right*. In the environment, a reward of +1 is achieved as long as the pole is maintained upright between a certain angle and a position. The discount factor γ is set to 0.99.

The experiment is conducted with 4 options. We use an intra-option Q-learning in the critic for learning the policy over options. The Boltzmann distribution was used for learning both the intra-option policies and the policy over options. The linear-sigmoid function was used for the termination of options. The hyperparameters were fine-tuned using the grid search over the parameter space. The optimal performance was obtained with the step size being set to 0.1 for termination, intra-option and critic. The temperature for the Boltzmann distribution was set to 0.001. Sutton and Barto's (1998) open source tile coding implementation[2] is used for discretization of the state space. Ten dimensional features (joint space of 4 continuous features) are used to represent the state space. The continuous features: position, velocity and pole angle were discretized into 3 bins and the angular velocity was discretized into 6 bins.

Fig. 5 shows the averaged return over 50 trials with the different degrees of controllability ψ. The best performance is achieved with $\psi = 0.25$. The figure shows that with the right degree of controllability, the variance in the return reduces and leads to the faster learning in terms of the mean return score. The controllability helps in the identification of the features which lead to the consistent behavior of the agent, thus learning to avoid state-action pairs which might lead the cart pole to topple. The code for the experiments in the grid world and the cartpole environment is available on the Github[3].

4.3 Arcade Learning Environment

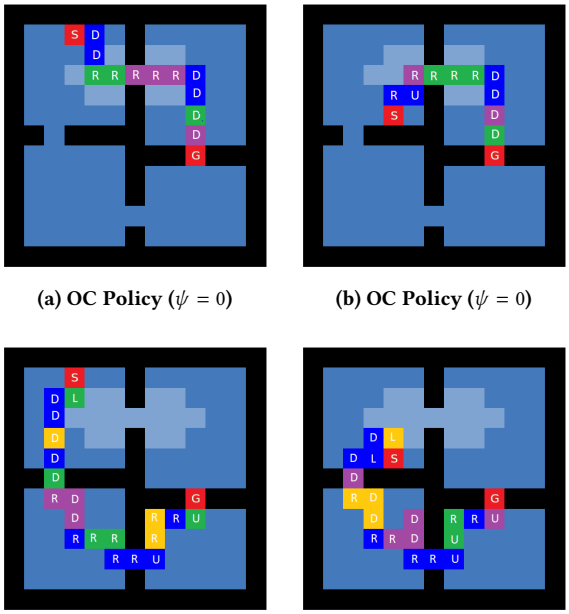
In this section, we discuss our experiments in the ALE domain. Recent work in learning options introduced a deliberation cost [14] in the option-critic framework [4]. The deliberation cost could be interpreted as a penalty for terminating an option, thereby leading to temporally extended options. We use the asynchronous advantage option-critic (A2OC) [14] algorithm as our baseline for learning the 'safe' options with the non-linear function approximation. Within the option-critic architecture, A2OC works in a similar fashion as the asynchronous advantage actor-critic (A3C) algorithm [26].

Introducing controllability in the A2OC algorithm results in an additional term to the intra-option policy gradient alone as shown in

[1] https://gym.openai.com/envs/CartPole-v0/

[2] http://incompleteideas.net/tiles/tiles3.py-remove

[3] The source code is available at https://github.com/arushi12130/SafeOptionCritic



(a) OC Policy ($\psi = 0$)

(b) OC Policy ($\psi = 0$)

(c) Safe-OC Policy ($\psi = 0.05$)

(d) Safe-OC Policy ($\psi = 0.05$)

Figure 4: Policy in Four Room Environment: Policy learned with 4 options where S and G represents the start & goal state. $\{R, L, U, D\}$ denotes 4 actions agent takes according to the learned policy; might take different actions due to environment stochasticity. Change in color represents the option switching. Same color represents the same option. The S and G states are depicted with red color. Light blue patch represents the frozen states. a) & b) shows the policy with $\psi = 0$ passing through the frozen area. c) & d) depicts policy learnt with $\psi = 0.05$ avoiding the frozen area due to the inbuilt safety constraint.

Equation (18). Our update rule for the intra-option policy gradient in the A2OC with controllability setting thus becomes:

$$\begin{aligned}\theta_\pi \leftarrow \theta_\pi + \alpha_{\theta_\pi} \frac{\partial \log(\pi_{\omega,\theta}(a|s)}{\partial \theta}\{G - Q_\Theta(s,\omega)\} \\ - \alpha_{\theta_\pi} \frac{\partial \log(\pi_{\omega_0,\theta}(a_0|s_0))}{\partial \theta}\psi\delta^2(s_0,\omega_0,a_0)\end{aligned} \tag{22}$$

Here G is a mixture of n-step returns similar to the A2OC with the difference that here we consider this return only for the duration an option persisted in continuation. Without any loss in generality, the 1-step TD error in the definition of controllability can be substituted with n-step TD error only if the same option has continued up until the n^{th} step. Similarly, as discussed in the Equation (21), there is no change in the termination gradient and we use the same update rule as derived in the A2OC algorithm. η here is the deliberation cost.

$$\nu \leftarrow \nu - \alpha_\nu \frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu}(Q_\Omega(s',\omega) - V_\Omega(s') + \eta) \tag{23}$$

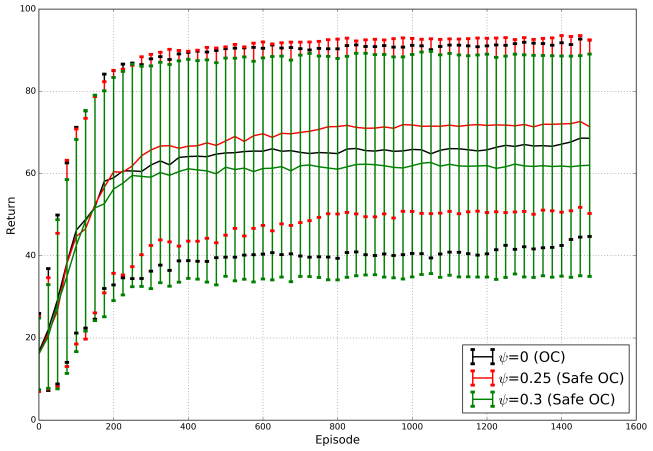


Figure 5: Learning curve for 4 options in Cart Pole Environment: Results are averaged over 50 trials. The band around the solid horizontal lines represents the standard deviation of the return. $\psi = 0.25$ performs the best in case of 4 options.

We use primarily three games; MsPacman, Amidar and Q*Bert from the ATARI 2600 suite to test our Safe-A2OC algorithm and analyze the performances. We introduce Safe-A2OC [4] built using the same deep network architecture as A2OC, wherein the policy over options is ϵ-greedy, the intra-option policies are linear softmax functions, the termination functions use sigmoid activation functions along will the linear function approximation for the Q values. For hyperparameters, we learn 4 options, with a fixed deliberation cost of 0.02, margin cost of 0.99, step size of 0.0007, and entropy regularization of 0.01 for varying degrees of controllability (ψ) and ϵ. The training used 16 parallel threads for all our experiments. We optimized the ϵ parameter for no controllability ($\psi = 0$). For a fair analysis, we compare the best performance of A2OC against different degrees of controllability parameter with the Safe-A2OC.

Results and Evaluation: To evaluate the performances, we use two metrics namely the learning curves [21] and the average performance over k games. Figures 6, 7 and 8 show the learning curves over 80M frames with varying controllability parameter. It is observed that for specific degrees of controllability, options learned with our notion of safety (Safe-A2OC) outperforms the vanilla options (A2OC). It is important to note that the different values of ψ control the degree to which an agent would be risk averse. A grid search over the different degrees of the controllability hyperparameter ψ resulted in a narrow range of 0 to 0.15. For a very high value of $\psi > 0.3$, we observe that the agents become extremely risk-averse resulting in a poor performance. An optimum value of ψ for all the three games is obtained around $0.05 - 0.10$. We present the videos of some of these trained agents as qualitative results in the supplementary material[5]. Upon visual inspection of the trained Safe-A2OC agent, we observe that explicitly optimizing for the variance in TD error results in the agent learning to avoid states with higher variance in TD error. For instance, in MsPacman, the

[4]The source code is available at https://github.com/kkhetarpal/safe_a2oc_delib

[5]Supplementary material is available at https://sites.google.com/view/safe-option-critic

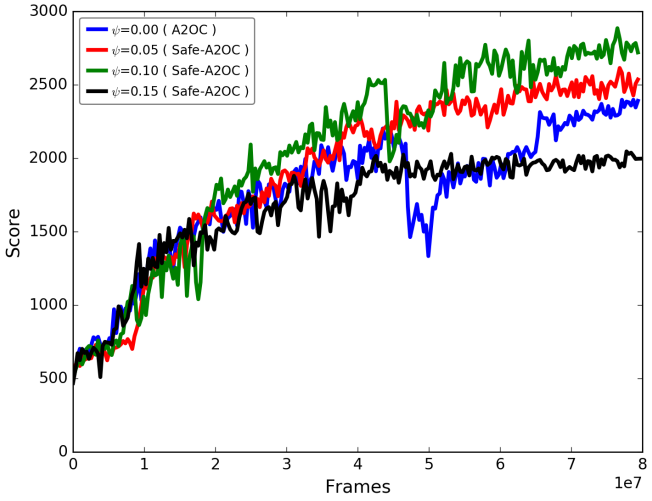


Figure 6: Learning curve for 4 options in MsPacman: Options with a controllability factor of $\psi = 0.10$ learn better than the best performing scenario of no controllability ($\psi = 0, \epsilon = 0.2$). Higher degrees of ψ results in poor performance.

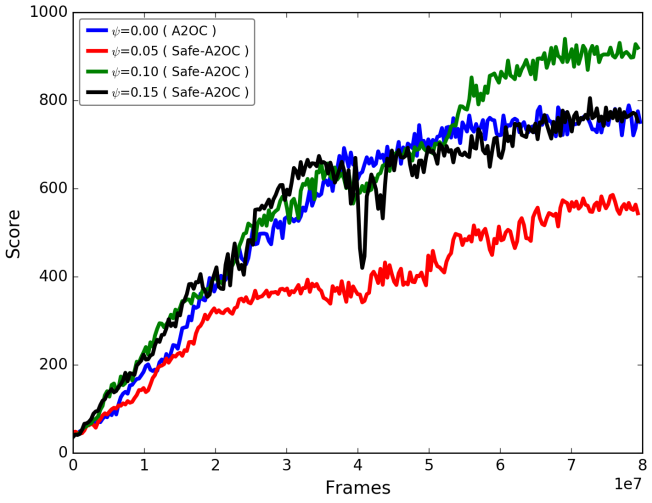


Figure 7: Learning curve for 4 options in Amidar: Options with a controllability factor of $\psi = 0.10$ outperform vanilla options ($\psi = 0$, $\epsilon = 0.2$). Higher degrees of controllability ($\psi > 0.15$) results in reduced exploration and adversely effects the performance.

acquisition of the corner diamonds provide the intrinsic variability in the reward structure. Our objective function helps the agent understand such an intrinsic variability in reward, thus boosting the overall performance.

The trained agents are then tested for their averaged performance across $k = 100$ games as shown in the Table 1. Safe-A2OC with a controllability value of $\psi = 0.05$ in Q*Bert and $\psi = 0.10$ in

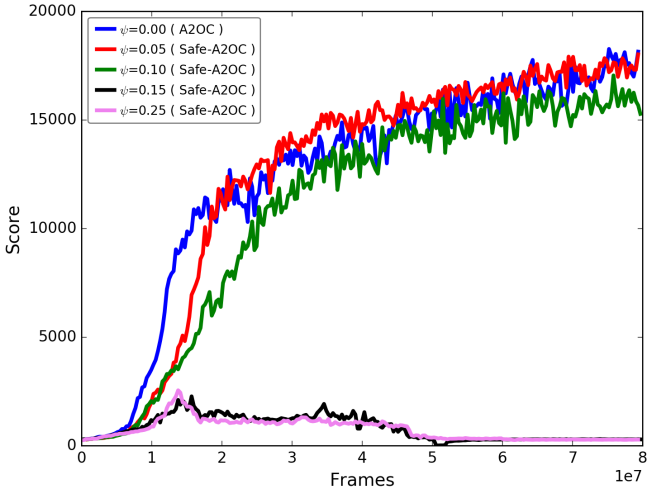


Figure 8: Learning curve for 4 options in Q*bert: Options with a controllability factor of $\psi = 0.05$ see better average performance in the learning than the one with no controllability ($\psi = 0, \epsilon = 0.2$).

MsPacman and Amidar outperforms the score achieved by A2OC. In MsPacman and Amidar, Safe-A2OC also outperforms the other state-of-the-art approaches [26, 27, 40, 42] using the primitive actions. Empirical effects of introducing the right degree of controllability in options demonstrates that an agent which additionally optimizes for low variance in the TD errors learns better than the one optimizing only for the cumulative reward. The intuition here is that using variance in the TD error as a measure of safety in hierarchical RL helps the agents avoid states with high intrinsic variability. Depending on the nature of the game itself, we observe different degrees of response to different levels of controllability in Q*bert, Amidar, and MsPacman.

5 DISCUSSION

In this work, we introduce a new framework called *Safe Option-Critic* wherein we define the safety in learning end-to-end options. We extend the idea of controllability from the primitive action space using the temporal difference error to the option-critic architecture for incorporating safety. The underlying idea of this learning process is to discourage the agent from visiting the harmful or the undesirable state-option pair by constraining the variance in the TD error. Recent work by [30] proposed a direct method to calculate the variance of the λ return instead of the traditional indirect approaches which use the second order moment. The authors proposed a Bellman operator which uses the square of the TD error to measure the variance of return. This work further supports our approach of estimating the risk through the square of TD error.

Our experiments in the tabular methods empirically demonstrate the reduced variance in the return. Moreover, we observe a boost in the overall performance in both the tabular and the linear approximation methods. Furthermore, experiments in the ALE domain demonstrate that an RL agent was able to learn about the intrinsic

Table 1: ALE Final Scores: Average Performance over 100 games once training is completed after $80M$ frames. Scores in boxes highlight the performance with no controllability whereas aqua highlighted cells indicate the benefits of introducing our notion of safety in learning end-to-end options. Introducing controllability in options outperforms best performances of primitive actions in 2 out of 3 games analyzed here. Learning options with our notion of safety outperforms vanilla A2OC in all 3 games. A3C scores have been taken from [26], DQN from [27], Double DQN from [40], and Dueling from [42]. ψ represents the degree of controllability. Values in brackets indicate standard deviation across 100 games.

Algorithm	MsPacman	Amidar	Q*Bert
A3C	850.7	283.9	21307.5
DQN	763.5	133.4	4589.8
Double DQN	1241.3	169.1	11020.8
Dueling	2250.6	172.7	14175.8
$\psi = 0, \epsilon = 0.2$	2285.4 (756.64)	760.71 (204.08)	16881.25 (6107.04)
$\psi = 0.05, \epsilon = 0.2/0.3$	2481.2(909.48)	569(158.77)	**17642.0 (3346.85)**
$\psi = 0.10, \epsilon = 0.2$	**2710.9 (598.69)**	**925.43 (211.52)**	14490.0(5962)
$\psi = 0.15, \epsilon = 0.2$	2055.8(468.09)	781.31(168.79)	1477.5(961.85)
$\psi = 0.25, \epsilon = 0.2$	2290.4(855.00)	458.82(107.77)	298.25(133.71)

variability in a large and complicated state-space such as images with non-linear function approximation. Results from ALE also demonstrated that the options with the notion of safety outperform the algorithms using the primitive actions.

Limitations and Future Work: In this work, we limit the return calculation until an option terminates. Using the n-step returns during the intermediate switching of the options at the SMDP level is of potential interest for the future work. Additionally, it is currently assumed that all the options are available in all the states. In the context of safety, it might be of interest to understand what happens if the options initiation sets were limited to subset of the entire state space. One could also work with varying the degree of controllability regularizer ψ, where ψ could start from 0 to support the exploration in the beginning and gradually increase the value of ψ to limit the exploration to the unsafe states.

A potential direction of future work is the extension of controllability to more than just the initial state-option pair. One could extend the definition of controllability to all the state-option pairs in the trajectory which could potentially expedite the effects of the risk mitigation. The proposed notion of safety could also be extended to different levels of hierarchy in the framework. For instance, a mixture of options with varying degrees of controllability can be learned, wherein at policy over the options level, one could select an option based on how much controllability is desirable for a subset of an environment. The intra-option policy could still retain the current formalizations.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues Herke van Hoof, Pierre-Luc Bacon, Jean Harb, Ayush Jain and Martin Klissarov for their useful comments and discussions throughout the duration of this work. The authors would also like to thank Open Philanthropy for funding this work, and Compute Canada for the computing resources.

REFERENCES

[1] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.

[2] Dario Amodei and Jack Clark. 2016. Faulty Reward Functions in the Wild. (2016). https://blog.openai.com/faulty-reward-functions/

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *CoRR* (2016). arXiv:1606.06565

[4] Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The Option-Critic Architecture.. In *AAAI*. 1726–1734.

[5] Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems* 13, 4 (2003), 341–379.

[6] Christian Daniel, Herke Van Hoof, Jan Peters, and Gerhard Neumann. 2016. Probabilistic inference for determining options in reinforcement learning. *Machine Learning* 104, 2-3 (2016), 337–357.

[7] Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement Learning with a Corrupted Reward Channel. *arXiv preprint arXiv:1705.08417* (2017).

[8] Richard E Fikes, Peter E Hart, and Nils J Nilsson. 1981. Learning and executing generalized robot plans. In *Readings in Artificial Intelligence*. Elsevier, 231–249.

[9] Future of Life Institute. 2017. Asilomar AI Principles. (2017). https://futureoflife.org/2017/01/17/principled-ai-discussion-asilomar/

[10] Javier Garcıa and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.

[11] Clement Gehring and Doina Precup. 2013. Smart Exploration in Reinforcement Learning Using Absolute Temporal Difference Errors. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '13)*. 1037–1044.

[12] Peter Geibel and Fritz Wysotzki. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.(JAIR)* 24 (2005), 81–108.

[13] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. 2017. Inverse reward design. In *Advances in Neural Information Processing Systems*. 6768–6777.

[14] Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. 2017. When waiting is not an option: Learning options with a deliberation cost. *arXiv preprint arXiv:1709.04571* (2017).

[15] Glenn A Iba. 1989. A heuristic approach to the discovery of macro-operators. *Machine Learning* 3, 4 (1989), 285–317.

[16] George Konidaris and Andrew G Barto. 2007. Building Portable Options: Skill Transfer in Reinforcement Learning.. In *IJCAI*, Vol. 7. 895–900.

[17] George Konidaris, Scott Kuindersma, Roderic A Grupen, and Andrew G Barto. 2011. Autonomous Skill Acquisition on a Mobile Manipulator.. In *AAAI*.

[18] Richard E Korf. 1983. *Learning to Solve Problems by Searching for Macro-operators*. Ph.D. Dissertation. Pittsburgh, PA, USA. AAI8425820.

[19] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*. 3675–3683.

[20] Edith LM Law, Melanie Coggan, Doina Precup, and Bohdana Ratitch. 2005. Risk-directed Exploration in Reinforcement Learning. *Planning and Learning in A Priori Unknown or Dynamic Domains* (2005), 97.

[21] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling. 2017. Revisiting the Arcade Learning Environment: Evaluation

Protocols and Open Problems for General Agents. *ArXiv e-prints* (Sept. 2017). arXiv:cs.LG/1709.06009

[22] Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. 2016. Adaptive Skills Adaptive Partitions (ASAP). In *Advances in Neural Information Processing Systems*. 1588–1596.

[23] Amy McGovern and Andrew G Barto. 2001. Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML*, Vol. 1. 361–368.

[24] Ishai Menache, Shie Mannor, and Nahum Shimkin. 2002. Q-cut - dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*. Springer, 295–306.

[25] Oliver Mihatsch and Ralph Neuneier. 2002. Risk-sensitive reinforcement learning. *Machine learning* 49, 2-3 (2002), 267–290.

[26] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*. 1928–1937.

[27] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, Shane Legg, Volodymyr Mnih, Koray Kavukcuoglu, and David Silver. 2015. Massively Parallel Methods for Deep Reinforcement Learning. *CoRR* (2015). arXiv:1507.04296

[28] Doina Precup. 2000. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst.

[29] Makoto Sato, Hajime Kimura, and Shibenobu Kobayashi. 2001. TD algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence* 16, 3 (2001), 353–362.

[30] C. Sherstan, B. Bennett, K. Young, D. R. Ashley, A. White, M. White, and R. S. Sutton. 2018. Directly Estimating the Variance of the λ-Return Using Temporal-Difference Methods. *ArXiv e-prints* (Jan. 2018). arXiv:cs.AI/1801.08287

[31] Jonathan Sorg, Richard L Lewis, and Satinder P Singh. 2010. Reward design via online gradient ascent. In *Advances in Neural Information Processing Systems*. 2190–2198.

[32] Martin Stolle and Doina Precup. 2002. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*. Springer, 212–223.

[33] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.

[34] Richard S. Sutton and Andrew G. Barto. 1998. *Introduction to Reinforcement Learning* (1st ed.). MIT Press, Cambridge, MA, USA.

[35] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*. 1057–1063.

[36] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[37] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 2012. Policy gradients with variance related risk criteria. In *Proceedings of the twenty-ninth international conference on machine learning*. 387–396.

[38] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 2016. Learning the variance of the reward-to-go. *Journal of Machine Learning Research* 17, 13 (2016), 1–36.

[39] Aviv Tamar, Huan Xu, and Shie Mannor. 2013. Scaling up robust MDPs by reinforcement learning. *arXiv preprint arXiv:1306.6189* (2013).

[40] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning.. In *AAAI*, Vol. 16. 2094–2100.

[41] Alexander Vezhnevets, Volodymyr Mnih, Simon Osindero, Alex Graves, Oriol Vinyals, John Agapiou, et al. 2016. Strategic attentive writer for learning macro-actions. In *Advances in neural information processing systems*. 3486–3494.

[42] Ziyu Wang, Nando de Freitas, and Marc Lanctot. 2015. Dueling Network Architectures for Deep Reinforcement Learning. *CoRR* (2015). arXiv:1511.06581