The Knowledge Engineering Review, Vol. 36, e4, 1 of 20.
doi:10.1017/S0269888921000035

Safe option-critic: learning safety in the option-critic architecture

ARUSHI JAIN[†], KHIMYA KHETARPAL[†] and DOINA PRECUP

School of Computer Science, Mila - McGill University, Montreal, Quebec
E-mails: arushi.jain@mail.mcgill.ca, khimya.khetarpal@mail.mcgill.ca, dprecup@cs.mcgill.ca

Abstract

Designing hierarchical reinforcement learning algorithms that exhibit safe behaviour is not only vital for practical applications but also facilitates a better understanding of an agent's decisions. We tackle this problem in the options framework (Sutton, Precup & Singh, 1999), a particular way to specify temporally abstract actions which allow an agent to use sub-policies with start and end conditions. We consider a behaviour as *safe* that avoids regions of state space with high uncertainty in the outcomes of actions. We propose an optimization objective that learns *safe* options by encouraging the agent to visit states with higher behavioural consistency. The proposed objective results in a trade-off between maximizing the standard expected return and minimizing the effect of model uncertainty in the return. We propose a policy gradient algorithm to optimize the constrained objective function. We examine the quantitative and qualitative behaviours of the proposed approach in a tabular grid world, continuous-state puddle world, and three games from the Arcade Learning Environment: Ms. Pacman, Amidar, and Q*Bert. Our approach achieves a reduction in the variance of return, boosts performance in environments with intrinsic variability in the reward structure, and compares favourably both with primitive actions and with risk-neutral options.

1 Introduction

Safety in Artificial Intelligence (AI) has become an important focus of research as AI methods make rapid advancements (Amodei *et al*., 2016). The 23 Asilomar AI principles (Future of Life Institute, 2017) discuss a variety of safety aspects such as risk aversion, transparency, robustness, fairness, and also legal and ethical values that an agent must hold. In this work, we mainly focus on safety from the perspective of preventing undesirable behaviour, in particular, reducing visits to undesirable or risky states during the learning process of a reinforcement learning (RL) system.

RL algorithms optimize the expected value of cumulative rewards (return) obtained over time (Sutton & Barto, 1998). For safety-critical applications such as finance, medicine, or industrial automation, optimizing the expected return alone is often not sufficient. An agent might exhibit undesirable behaviour over certain trajectories or for a certain period. In general, safety requires measuring some notion of *risk* or *uncertainty* in addition to maximizing the mean return thus ensuring that the learning algorithm also minimizes risk.

An additional challenge in RL is that agents need to explore during which an agent might be unaware of the states prone to noise or potentially leading to catastrophic consequences. Risk awareness in RL has been formulated in several ways, for example, by directed exploration that avoids the high entropy states (Law *et al*., 2005), optimizing the worst-case performance instead of the expected performance

† These authors contributed equally to this work.

of an agent (Tamar *et al*., 2013), measuring and bounding the probabilities of visiting the erroneous states (Geibel & Wysotzki, 2005), and several other approaches. Garca and Fernández (2015) present a comprehensive survey covering a broad range of techniques aimed at achieving safety in RL. In the context of Markov Decision Processes (MDPs), the source of uncertainty or variability in a sequential decision-making task can arise mainly from two sources—inherent stochastic uncertainty in the reward and the transition model (aleatoric uncertainty), and imperfect knowledge regarding the model (epistemic uncertainty). The former uncertainty is usually clustered under *risk-sensitive MDPs* (Howard & Matheson, 1972; Heger, 1994; White, 1994; Borkar & Meyn, 2002), whereas the latter is covered under *robust MDPs* (Iyengar, 2005; Nilim & El Ghaoui, 2005; Lim *et al*., 2013). In this work, we focus on risk-sensitive MDPs and address the former source of variability via mean-variance optimization.

The natural question that follows is how can we measure variance in RL? Several works focus on estimating the variance in return using temporal-difference (TD) style learning methods (Tamar *et al*., 2012; Tamar *et al*., 2016; Gehring & Precup, 2013; Sherstan *et al*., 2018; Jain *et al*., 2021). Our work is based on the specific approach proposed by Gehring and Precup (2013), which suggests using the absolute value of the TD error to define *controllability* of a state. Intuitively, for an agent to exhibit safe behaviour, it will prefer taking the actions whose effects are more predictable on a state. Such states are referred as *controllable*. We redefine controllability slightly to provide an estimator which calculates the uncertainty in the value function and then show how to 'encourage' an agent to focus on controllable states.

All approaches discussed so far define safety in the primitive action space (one-step action). Inspired by how humans think and plan, temporally abstract actions also provide a path to learn and plan efficiently. Temporal abstractions have been a crucial part of AI research since the 1970s (Fikes *et al*., 1972, 1981; Iba, 1989; Korf, 1983; Parr & Russell, 1998; Sutton *et al*., 1999; Precup, 2000; Dietterich, 2000; McGovern & Barto, 2001; Menache *et al*., 2002; Barto & Mahadevan, 2003; Konidaris & Barto, 2007; Bacon *et al*., 2017; Barreto *et al*., 2019). Prior research has shown that the temporal abstractions can improve exploration, reduce the complexity of deliberation, and enhance robustness. More importantly, from the safety perspective, temporal abstractions lead to composition of behaviour and predictions. To simplify, if every subpart of the abstraction hierarchy can be derived respecting the safety conditions, the entire hierarchy would automatically be safe in behaviour. In this paper, we take the first step in this direction by showing how to construct temporally safe extended actions by reducing the variability or uncertainty as discussed above.

We approach this problem in the *options* framework (Sutton *et al*., 1999; Precup, 2000), which provides an intuitive way to plan, reason, and act at multiple timescales. We are especially interested in learning options end-to-end from the data obtained by the agent's interaction with the environment. The options framework has received a lot of attention over time, for example, Stolle and Precup (2002), Konidaris and Barto (2007), Konidaris *et al*. (2011); Daniel *et al*. (2016); Kulkarni *et al*. (2016); Vezhnevets *et al*. (2016); Mankowitz *et al*. (2016), Jain and Precup (2018), Riemer *et al*. (2018); Barreto *et al*. (2019).

Bacon *et al*. (2017) introduced the option-critic framework by adapting to the actor-critic model. The option-critic model facilitates an end-to-end learning of options without the need to specify sub-goals. We modify the option-critic objective to include the negation of controllability (Gehring & Precup, 2013) (estimates the variability in the performance) as a regularizer in the optimization objective. Consequently, the **key contributions** of this work can be summarized as follows:

1. We propose a new objective function which uses the effect of the uncertainty in the return as a regularizer for the classical optimization problem of maximizing the expected return. We then use this objective to derive a policy gradient algorithm for automatically learning safe options.
2. We empirically demonstrate the effectiveness of our approach in the tabular setup.
3. We also show that our method is scalable to both linear and non-linear function approximation settings using the puddle world and ALE games: Ms. Pacman, Amidar, and Q*Bert.

The paper is structured as follows. After presenting definitions and notations in Section 2, we introduce the safe option-critic algorithm in Section 3. Next, we present tabular, continuous, and ALE experiments in Section 4. Finally, we conclude with discussion and future work in Section 5.

2 Preliminaries

In RL, an agent interacts with the environment at discrete time steps $t \in \{1, 2, ...\}$. At each step, the agent observes a state $s \in \mathcal{S}$ and chooses an action $a \in \mathcal{A}$ according to a policy which defines a probability distribution of actions over the state space $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Afterwards, the agent transitions from S_t to S_{t+1} state according to the transition probability distribution $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. After transitioning to a next state, an agent receives a reward $R_{t+1} \in \mathbb{R}$, where the expected reward function is $r(s, a) = \sum_{r \in \mathbb{R}} r \sum_{s'} P(s', r|s, a)$, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The MDP is defined as a tuple of $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$, where $\gamma \in [0, 1]$ is the discount factor, which de-values rewards received farther into the future. A MDP with the defined optimality criteria is known as a *Markov Decision Problem*. In this work, we focus on MDP with optimality criteria as the *expected total discounted reward*. According to these optimality criteria, we define the state-action value function as $Q(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_t = s, A_t = a]$. An estimate of Q can be learned in an incremental fashion using TD learning methods (Sutton, 1988). For example, in $TD(0)$, the agent computes a TD error, which captures the difference in the agent's value estimate at two consecutive time steps:

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

The state-action value function is then updated as:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \delta_t,$$

where $\alpha \in [0, 1]$ is the step size.

The policy gradient theorem (Sutton *et al.*, 2000) provides a way to update a parameterized policy in the direction of the gradient of the expected return. Here, return is the total discounted rewards received in a trajectory. The performance of a policy π is measured by the expected return as

$$\rho(\pi, s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | \pi, s_0],$$

where the initial state is denoted by s_0. The gradient of the performance with respect to the policy parameter θ is given by:

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

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

2.1 Options

The options framework (Sutton *et al.*, 1999; Precup, 2000) facilitates the incorporation of temporally abstract knowledge into RL. An option $w \in \mathcal{W}$ is defined as a tuple of (I_w, π_w, β_w). I_w is an initiation set containing initial states from which an option w can start. π_w denotes an option policy that defines a distribution over actions given a state during the execution of option w. The termination condition of w option, β_w, defines a probability of termination in each state.

If the option policies are Markovian, the intra-option Bellman equation (Sutton *et al.*, 1999) provides an off-policy method for updating the value of a state-option pair as

$$\begin{aligned} Q(S_t, W_t) \leftarrow Q(S_t, W_t) + \alpha \Big(& R_{t+1} + \gamma (1 - \beta_{W_t}(S_{t+1})) Q(S_{t+1}, W_t) \\ & + \gamma \beta_{W_t}(S_{t+1}) \max_{w' \in \mathcal{W}} Q(S_{t+1}, w') - Q(S_t, W_t) \Big), \end{aligned}$$

where the same option w continues with $1 - \beta_w$ probability, or terminates with β_w probability and selects the next best option.

2.2 *Option-critic architecture*

The above described intra-option value learning (Sutton *et al.*, 1999) also lays the foundation for how the options are learnt in the *Option-Critic* architecture (Bacon *et al.*, 2017). The option-critic method is a policy gradient-based method for learning intra-options policies and option's termination conditions. Bacon *et al.* (2017) consider the call-and-return option execution model, where an option w is chosen according to a policy over options π_W, and an intra-option policy π_w is followed until the termination condition β_w is met. Once the current option terminates, another option is selected using π_W and the process continues. Let $\pi_{w,\theta}$ denotes intra-option policy parameterized by θ, and $\beta_{w,\nu}$ represents the option termination parameterized by ν. The value of executing an action a at a particular state-option pair is given by $Q_U : \mathcal{S} \times \mathcal{W} \times \mathcal{A} \to \mathbb{R}$ which is defined as

$$Q_U(s, w, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a)U(s', w), \tag{2}$$

where $U(s',w)$ represents the value of arriving through an option w at a state s'. Here, either the option w could terminate with $\beta_{w,\nu}$ probability and select the other options, or could continue with the same option with $(1 - \beta_{w,\nu})$ probability,

$$U(s', w) = (1 - \beta_{w,\nu}(s'))Q_W(s', w) \; + \; \beta_{w,\nu}(s')V_W(s'). \tag{3}$$

Here, Q_W represents the value function over options, which is given by-

$$Q_W(s, w) = \sum_a \pi_{w,\theta}(a|s)Q_U(s, w, a).$$

V_W represents the value function of executing policy over options π_W, given by-

$$V_W(s) = \sum_w \pi_W(w|s)Q_W(s, w).$$

Bacon *et al.* (2017) derived the gradient of the discounted return from an initial condition $(s_0, w_0) \in (\mathcal{S} \times \mathcal{W})$ with respect to θ (intra-option policy parameter) as:

$$\frac{\partial \rho(\pi, s_0, w_0)}{\partial \theta} = \sum_{s,w} \mu(s, w|s_0, w_0) \sum_a \frac{\partial \pi_{w,\theta}(a|s)}{\partial \theta} Q_U(s, w, a), \tag{4}$$

where $\mu(s, w|s_0, w_0) = \sum_{t=0}^{\infty} \gamma^t P(S_t = s, W_t = w|s_0, w_0)$ is the discounted weighting of a state-option pair. The gradient of the expected discounted return with respect to the option termination parameter ν and the initial condition (s_1, w_0) is given by:

$$\frac{\partial \rho(\pi, s_1, w_0)}{\partial \nu} = -\sum_{s',w} \mu(s', w|s_1, w_0) \frac{\partial \beta_{w,\nu}(s')}{\partial \nu} A_W(s', w), \tag{5}$$

where A_W is the advantage function $A_W(s, w) = Q_W(s, w) - V_W(s)$. In (5), the initial condition changed from (s_0, w_0) to (s_1, w_0) because given an initial state-option (s_0, w_0), the first action is selected according to π_{w_0} policy, and then the agent transitions to the next state s_1. The updates to the termination parameter ν are made thereafter based on the advantage of switching to the next option in state s'.

3 Safe option-critic model

In this work, we introduce a notion of learning safe hierarchical policies using a penalty based on the variance in TD error. Previous works in the literature had focused on the measures derived from the TD-based methods for quantifying uncertainty in the value of a state or a state-action pair.

As mentioned earlier, here, we focus on addressing uncertainty arising due to inherent stochasticity present in the model (aleatoric uncertainty). The uncertainty present in the model is reflected in the form of variability in the value function. One such feasible approach to capture the uncertainty/variability in the value function would be to estimate the variance in TD error. After the policy has converged, the expected TD error would approach zero value. The variance in the TD error results from the stochasticity in the environment's reward function and the transition dynamics. We define 'safety' as constraining the uncertainty in the value function. Therefore, we modify the objective function of maximizing the mean return by adding a regularizer on uncertainty in the value function.

Inspired by Gehring and Precup (2013), we define *controllability* as a negation of the variance in 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 in the value of a state-option pair. The key idea is—the higher the variance in TD error of a state-option pair, the higher would be the uncertainty in the value function.

In safety-critical applications, the agent must learn to avoid such state-option pairs as they induce variability in return eventually. We optimize for obtaining the maximum expected discounted return and controllability value of the *initial state-option pair*. Depending on the nature of the application, one can limit or encourage the agent's visit to a state-option pair based on the degree of controllability. Introducing controllability via the TD error facilitates the linear scalability of the proposed approach with the number of state-option pairs.

Analogous to the notations used in Bacon *et al*. (2017), we introduce 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 all options can be initialized from any state such that $I_w = \mathcal{S},\ \forall w \in \mathcal{W}$. Uncertainty in the value of a state-option pair is measured using controllability C, which is given by the negation of the variance in the TD error (δ). The TD error is

$$\delta(S_t, W_t, A_t) = R_{t+1} + \gamma \sum_{s'} P(s'|S_t, A_t) U_\Theta(s', W_t) - Q_{U,\Theta}(S_t, W_t, A_t), \tag{6}$$

where $Q_{U,\Theta}(s, w, a)$ and $U_\Theta(s, w)$ are defined in (2) and (3), respectively. For the ease of notation, we will use δ_t to denote $\delta(S_t, W_t, A_t)$. Whenever it is not clear from the context, we will use the full notation for the TD error. The expected value of the TD error would converge to zero. Therefore, the variance in TD error can be simplified to denote controllability as:

$$\begin{aligned} C_\Theta(s, w) &= -Var_{\pi_{w,\theta}}[\delta_t | S_t = s, W_t = w] \\ &= -\left(\mathbb{E}_{\pi_{w,\theta}}\left[\delta_t^2 | S_t = s, W_t = w\right] - \mathbb{E}_{\pi_{w,\theta}}\left[\delta_t | S_t = s, W_t = w\right]^2\right) \\ &= -\mathbb{E}_{\pi_{w,\theta}}\left[\delta_t^2 | S_t = s, W_t = w\right]. \end{aligned} \tag{7}$$

The aim here is to maximize both the expected discounted return and the controllability value from an initial state-option pair. We want to maximize the objective function J,

$$\begin{aligned} &\max_\Theta J(\Theta|\kappa), \\ &\text{where } J(\Theta|\kappa) = \mathbb{E}_{(s_0,w_0)\sim\kappa}\Big[Q_\Theta(s_0, w_0) + \psi\, C_\Theta(s_0, w_0)\Big], \end{aligned} \tag{8}$$

$\psi \in \mathbb{R}$ is a regularizer on the controllability value, and κ denotes the initial state-option pair distribution. The value of a state-option pair is defined as $Q_\Theta(s, w) = \sum_a \pi_{w,\theta}(a|s)\, Q_{U,\Theta}(s, w, a)$. Another interpretation is to view the above objective as a constrained optimization problem, where the constraints are

imposed on the controllability function. One could bound the controllability value, but it would require specific assumptions about the environment or one needs to rely on an input value from the user. In this paper, we opted to keep the method general without the need to rely on certain assumptions regarding the environment. We will now derive the gradient of the performance evaluator J with respect to the intra-option policy parameter θ, assuming that they are differentiable.

First, we calculate the gradient of the δ_t with θ using (6):

$$\begin{aligned}\frac{\partial \delta_t}{\partial \theta} &= \gamma \sum_{s'} P(s'|S_t, A_t)\frac{\partial U_\Theta(s', W_t)}{\partial \theta} - \frac{\partial Q_{U,\Theta}(S_t, W_t, A_t)}{\partial \theta} \\ &= \gamma \sum_{s'} P(s'|S_t, A_t)\frac{\partial U_\Theta(s', W_t)}{\partial \theta} - \Big(\gamma \sum_{s'} P(s'|S_t, A_t)\frac{\partial U_\Theta(s', W_t)}{\partial \theta}\Big) \\ &= 0 \end{aligned} \tag{9}$$

The above equation follows through (2). Next, we take the gradient of C with respect to θ. Following from (7),

$$\begin{aligned}\frac{\partial C_\Theta(s_0, w_0)}{\partial \theta} &= -\frac{\partial\{\sum_a \pi_{w_0,\theta}(a|s_0)\delta(s_0, w_0, a)^2\}}{\partial \theta} \\ &= -\sum_a \frac{\partial \pi_{w_0,\theta}(a|s_0)}{\partial \theta}\delta(s_0, w_0, a)^2 - 2\sum_a \pi_{w_0,\theta}(a|s_0)\delta(s_0, w_0, a)\frac{\partial \delta(s_0, w_0, a)}{\partial \theta} \\ &= -\sum_a \frac{\partial \pi_{w_0,\theta}(a|s_0)}{\partial \theta}\delta(s_0, w_0, a)^2 \qquad \text{[Using (9)]} \end{aligned} \tag{10}$$

The last term in the above equation vanishes because the gradient of TD error with respect to the policy parameter θ is zero (using (9)).

The one-step state-option transition is denoted by

$$P_\gamma^{(1)}(s', w'|s, w) = \gamma \sum_a \pi_{w,\theta}(a|s)P(s'|s, a)\big[(1 - \beta_{w,v}(s'))\mathbb{1}_{w=w'} + \beta_{w,v}(s')\pi_W(w'|s')\big].$$

Using the above one-step transition, similarly, the k-step state-option transition is expressed as

$$P_\gamma^{(k)}(s', w'|s_0, w_0) = P_\gamma^{(1)}(s_1, w_1|s_0, w_0) \times P_\gamma^{(k-1)}(s', w'|s_1, w_1).$$

Using the *Intra-Option Policy Gradient Theorem* (Bacon *et al.*, 2017), the gradient of $Q_\Theta(s, w)$ with θ is:

$$\frac{\partial Q_\Theta(s_0, w_0)}{\partial \theta} = \sum_{k=0}^{\infty}\sum_{s',w'} P_\gamma^{(k)}(s', w'|s_0, w_0)\sum_a \frac{\partial \pi_{w',\theta}(a|s')}{\partial \theta}Q_{U,\Theta}(s', w', a). \tag{11}$$

Following the objective function (8), combining the gradient of $Q_\Theta(s_0, w_0)$ (11) and $C_\Theta(s_0, w_0)$ (10), the gradient of J with θ is:

$$\begin{aligned}\frac{\partial J(\Theta|\kappa)}{\partial \theta} = \sum_{k=0}^{\infty}\sum_{s,w}\Big\{P_\gamma^{(k)}(s, w|s_0, w_0)\sum_a \frac{\partial \pi_{w,\theta}(a|s)}{\partial \theta}Q_{U,\Theta}(s, w, a)\Big\} \\ - \psi\sum_a \frac{\partial \pi_{w_0,\theta}(a|s_0)}{\partial \theta}\delta(s_0, w_0, a)^2. \end{aligned} \tag{12}$$

In (12), (s_0, w_0) corresponds to the initial state-option pair. In the above equation, one can simply replace $Q_{U,\Theta}(s, w, a)$ with the advantage term $A_\Theta(s, w, a) = Q_{U,\Theta}(s, w, a) - Q_\Theta(s, w)$ because,

$$\sum_a \frac{\partial \pi_{w,\theta}(a|s)}{\partial \theta}Q_\Theta(s, w) = Q_\Theta(s, w)\frac{\partial \sum_a \pi_{w,\theta}(a|s)}{\partial \theta} = Q_\Theta(s, w)\frac{\partial 1}{\partial \theta} = 0. \tag{13}$$

The gradient update of J here describes that each option aims to maximize its own reward, along with, maintaining a high controllability value pertaining to that option policy only.

Now, we will compute the gradient of $J(\Theta|\kappa)$ with respect to the option termination function parameter ν. First, we would calculate the gradient of TD error with ν.

$$\begin{aligned}\frac{\partial \delta_t}{\partial \nu} &= \gamma \sum_{s'} P(s'|S_t, A_t) \frac{\partial U_\Theta(s', W_t)}{\partial \nu} - \frac{\partial Q_{U,\Theta}(S_t, W_t, A_t)}{\partial \nu} \\ &= \gamma \sum_{s'} P(s'|S_t, A_t) \frac{\partial U_\Theta(s', W_t)}{\partial \nu} - \left(\gamma \sum_{s'} P(s'|S_t, A_t) \frac{\partial U_\Theta(s', W_t)}{\partial \nu}\right) \\ &= 0 \end{aligned} \tag{14}$$

The last equation follows through (2). The gradient of controllability C with ν using (7) and (14) can be expressed as:

$$\begin{aligned}\frac{\partial C_\Theta(s_0, w_0)}{\partial \nu} &= -\sum_a \pi_{w_0,\theta}(a|s_0) \frac{\partial \delta(s_0, w_0, a)^2}{\partial \nu} \\ &= -2 \sum_a \pi_{w_0,\theta}(a|s_0)\, \delta(s_0, w_0, a) \frac{\partial \delta(s_0, w_0, a)}{\partial \nu} \\ &= 0 \qquad \text{[Using (14)]} \end{aligned} \tag{15}$$

Therefore, the gradient of J with ν remains same as the gradient of the state-option value. On using the *Termination Gradient Theorem* (Bacon *et al*., 2017), the gradient of J with ν results as follows:

$$\begin{aligned}\frac{\partial J(\Theta|\kappa)}{\partial \nu} &= \frac{\partial Q_\Theta(s_1, w_0)}{\partial \nu} \\ &= -\sum_{k=0}^{\infty} \sum_{s,w} P_\gamma^{(k)}(s, w|s_1, w_0) \frac{\partial \beta_{w,\nu}(s)}{\partial \theta} A_W(s, w). \end{aligned} \tag{16}$$

Our interpretation of the above derivation follows the notion of safety; each option is responsible for making its intra-option policy safe by incorporating the controllability function. Due to the assumption that each option takes care of its own safety through the intra-option policy (the objective function implicitly does not bring changes at the policy over options level), one is only concerned about choosing an option that maximizes the expected discounted return from the next state-option pair while terminating an option. As a result, introduction of controllability does not impact the termination function update. Algorithm 1 shows a prototype implementation details of controllability in the option-critic architecture in the look-up table setting.

We now provide some intuition behind using the variance of TD error to learn the controllable states. The variance in the TD error of the initial state-option-action $\delta(s_0, w_0, a_0)$ uses the difference in the current state-option value $Q_{U,\Theta}(S_t, W_t, A_t)$ and the next expected state-option value $\mathbb{E}[U_\Theta(s', W_t)|S_t = s]$. With the increase in the time steps, the estimate of the Q value function changes (on seeing the new reward samples along a trajectory, the Q function estimate changes). Therefore, the square of the TD error would capture the variations in the return (sum of rewards) from the initial state-option pair onwards, thus yielding the uncertainty in the value function estimates.

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-room domain as described in Sutton *et al*. (1999). As shown in Figure 1, similar to Gehring and Precup (2013), we define some *slippery* frozen states in the environment which are unsafe to visit. We accomplish this by introducing variability in the rewards of these frozen states. The states labelled as F and G indicate the frozen and the goal states, respectively.

Algorithm 1 Safe option-critic with look-up table intra-option Q learning

Here, α_c, α_θ, α_ν stands for the step size of critic, intra-option policy, and termination, respectively. ψ is controllability regularization parameter.

$s \leftarrow s_0$
Take $w \sim \pi_W(w|s_0))$
Let initial w be w_0
repeat
  Take $a \sim \pi_{w,\theta}(.|s)$
  Let initial a taken at (s_0, w_0) be a_0
  Maintain (s_0, w_0, a_0) at the beginning of the episode
  Observe $\{r, s'\}$
  if s' is a non-terminal state **then**
    $\delta \leftarrow r + \gamma\Big[(1 - \beta_{w,\nu}(s'))Q_\Theta(s', w) + \beta_{w,\nu}(s') \max_{w' \sim \mathcal{W}} Q_\Theta(s', w')\Big] - Q_{U,\Theta}(s, w, a)$
  else
    $\delta \leftarrow r - Q_{U,\Theta}(s, w, a)$
  end if
  Maintain $\hat{\delta} \leftarrow \delta^2$ at the beginning of episode with (s_0, w_0, a_0)
  if $(s_0, w_0) == (s, w)$ **then**
    Update $(s_0, w_0, a_0) \leftarrow (s_0, w_0, a)$
    Update $\hat{\delta} \leftarrow \delta^2$ with new (s_0, w_0, a_0)
  end if
  $Q_{U,\Theta}(s, w, a) \leftarrow Q_{U,\Theta}(s, w, a) + \alpha\delta$
  $\theta \leftarrow \theta + \alpha_\theta \Big\{ \frac{\partial \log(\pi_{w,\theta}(a|s))}{\partial\theta} \big(Q_{U,\Theta}(s, w, a) - Q_\Theta(s, w)\big) - \psi \frac{\partial \log(\pi_{w_0,\theta}(a_0|s_0))}{\partial\theta} \hat{\delta} \Big\}$
  $\nu \leftarrow \nu - \alpha_\nu \frac{\partial \beta_{w,\nu}(s')}{\partial \nu} (Q_\Theta(s', w) - V_W(s'))$
  if $\beta_{w,\nu}(s')$ terminates **then**
    Choose new $w \sim \pi_W(w|s')$
  end if
  $s \leftarrow s'$
until s' is a terminal state

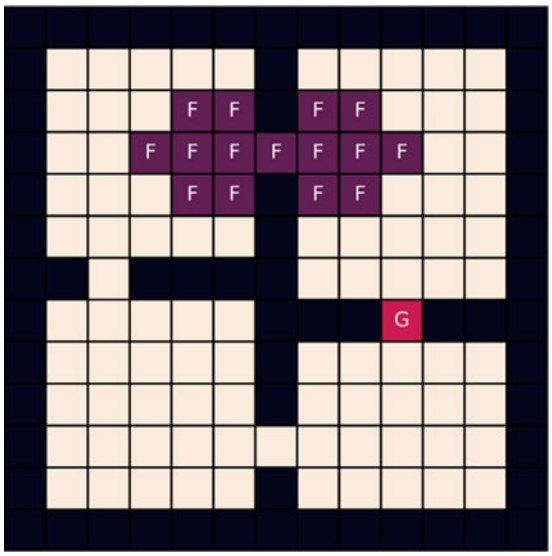

Figure 1 Four-Room (FR) environment: *F* and *G* depict the unsafe frozen and goal states, respectively. The lightest colour represents the normal states, whereas the darkest colour represents the wall.

Table 1 Parameters for discrete and continuous environment: Optimized parameters for both four-room and puddle-world environments.

Type	ψ	α_c	α_θ	α_v	temp	option
Four Room	**0.0**	0.1	0.01	0.01	0.001	4
	0.1	0.5	0.01	0.1	0.001	4
Puddle World	**0.0**	0.1	0.01	0.01	0.1	2
	0.015	0.5	0.05	0.05	0.1	2

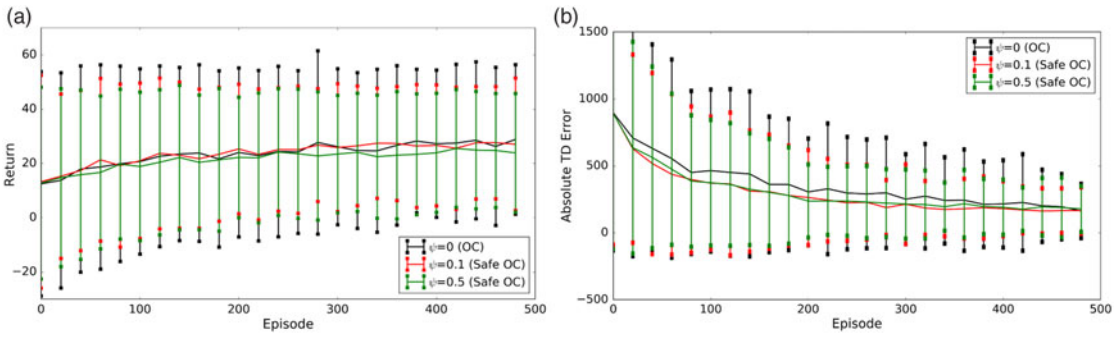

Figure 2 Performance in the FR domain. The graphs depict the performance averaged over 50 independent trials, where the vertical bands depict the standard deviation. The plots show (a) the return and (b) the sum of absolute TD error. Safe policy $\psi = 0.1$ (red) has a smaller standard deviation as compared to the baseline (black) signifying that safety helps the agent to avoid the variance inducing regions.

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*. Upon choosing an action, with 0.2 probability, the agent can transition into any of the four directions (irrespective of the selected action), whereas, with 0.8 probability, an agent takes the intended action deterministically. The task is to navigate through the rooms to a fixed goal state as depicted in Figure 1. The dark states in Figure 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 on transitioning into the normal and the goal state, respectively. Rewards for the unsafe states are drawn from $\mathcal{N}(\mu = 0, \sigma = 15)$ distribution when the agent transitions to a slippery state. The expected value of the reward for the normal and the slippery states is the same.

Here, we are using a look-up table representation to learn the policy. In the safe option-critic framework, we represent both policy over options and intra-option policies with the Boltzmann distribution. Sigmoid activation is used for learning the termination function. We ran the experiments with different controllability factor ψ for learning four options. One could choose a different number of options as well. We optimize for the hyperparameters: temperature and step size for both the baseline Option-Critic (OC) with $\psi = 0$ and Safe-OC. The discount factor γ is set to 0.99. Hyperparameters for the experiment are mentioned in Table 1. We ran the experiments for a total of 500 episodes averaged over 50 trials, where the training in each trial starts from scratch. In each episode, the agent is allowed to take a maximum of 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: compared the expectation and the variance in the performance over multiple trials (to comment on the stability of the introduced algorithm); sum of the absolute TD error over episodes; and density of the state visits. We also show sampled trajectories from both the baseline and the proposed method to qualitatively express the differences between the two after learning has been completed.

It can be observed from Figure 2a that the options with the controllability (Safe-OC) have a lower standard deviation in the return of an episode as compared to the baseline, that is, options without any

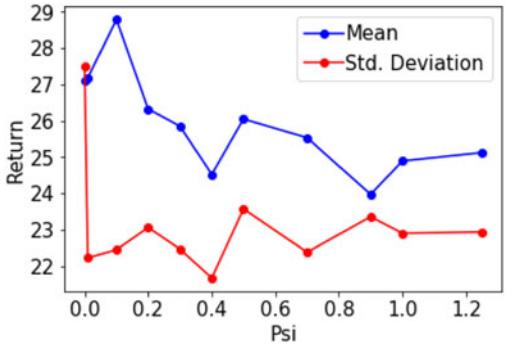

Figure 3 Performance with varying variance regularizer in Four-Room (FR) domain. The plot shows the mean and standard deviation of the score in the FR domain with varying controllability regularizer value ψ, while keeping all other parameters fixed. The best hyperparameters are chosen such that they not only minimize the standard deviation but also maximize the mean performance, which in this case is achieved by $\psi = 0.1$.

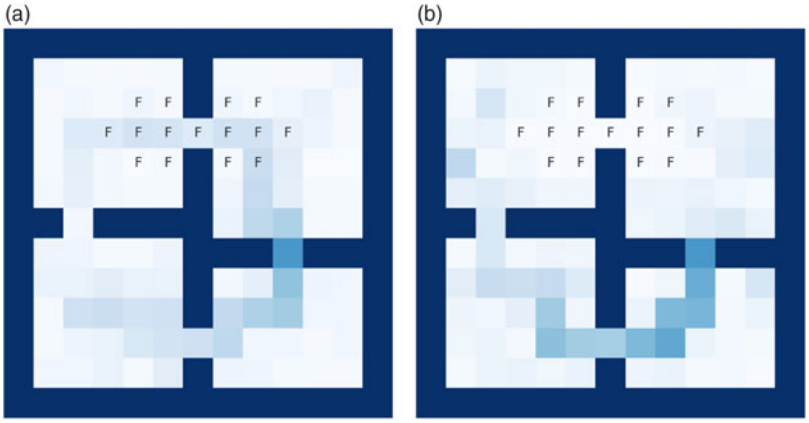

Figure 4 State visitation frequency in Four-Room environment. Density graph represents the number of times a state was visited during testing over 80 independent trials. Darker shades of blue represent a higher density. (a) Model without safety has equally likely density for both the hallways. (b) Model with safety shows higher density for the path without the frozen states.

notion of safety (OC). Figure 2b represents a lower standard deviation in the absolute sum of the TD error with safety. This highlights the fact that the controllability helps the agent to avoid the unsafe states (variable reward), thus resulting into a smaller variance in the sum of absolute TD error over multiple trials. Figure 3 shows how the performance (expectation and variance in the return) varies with the change in the controllability regularizer ψ. We chose the optimal value of ψ which not only maximizes the mean but also minimizes the variance in return. From the figure, it can be easily seen that $\psi = 0.1$ leads to not only maximum expectation but also significant reduction in the variance of return. We also visualize the state frequency graph depicted in Figure 4. It is observed that the options with the controllability function have lower frequency of visit to the frozen states (risky) as opposed to the vanilla options.

Learning of safe options induces transparency in the behaviour of an agent. This is most explicitly demonstrated through the trajectory taken by the agent without controllability regularizer (baseline OC) and with controllability regularizer (Safe-OC) as shown in Figure 5. Regardless of the start state, Safe-OC agent navigates to the goal state by avoiding the states with a high variance in the reward, as opposed to the OC agent, which finds a shortest route to the goal state being unaware of the risky states.

To capture how each option is behaving, we plotted the converged policy for all the options in Figure 6. It can be observed that each option does not capture the safety for the entire state space, but rather a subset of state space. The switches among the options for the safety are similar to the baseline (last image in both

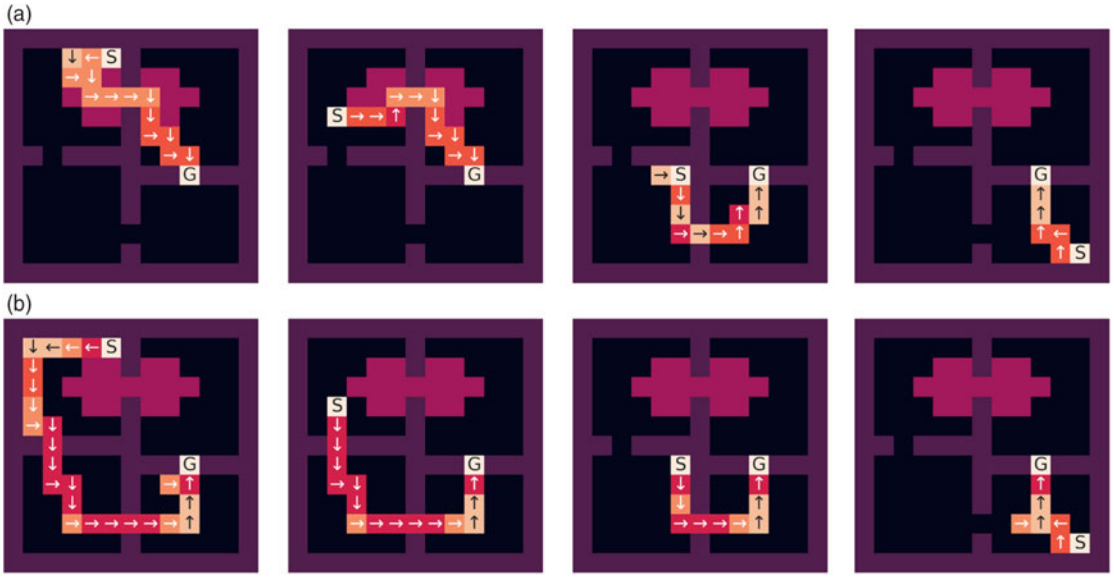

Figure 5 Policy in Four-Room (FR) environment. Sampled trajectories learned with 4 options where S and G represent the start & goal state, respectively. Arrows denote the 4 actions. Agent might take different actions due to environment stochasticity. Change in colour represents the option switching. Same colour represents the same option choice. Purple patch represents the frozen states. (a) depicts the policy with $\psi = 0$ passing through the frozen area. (b) depicts the policy learned with $\psi = 0.1$ that avoids the frozen area due to the in-built safety constraint.

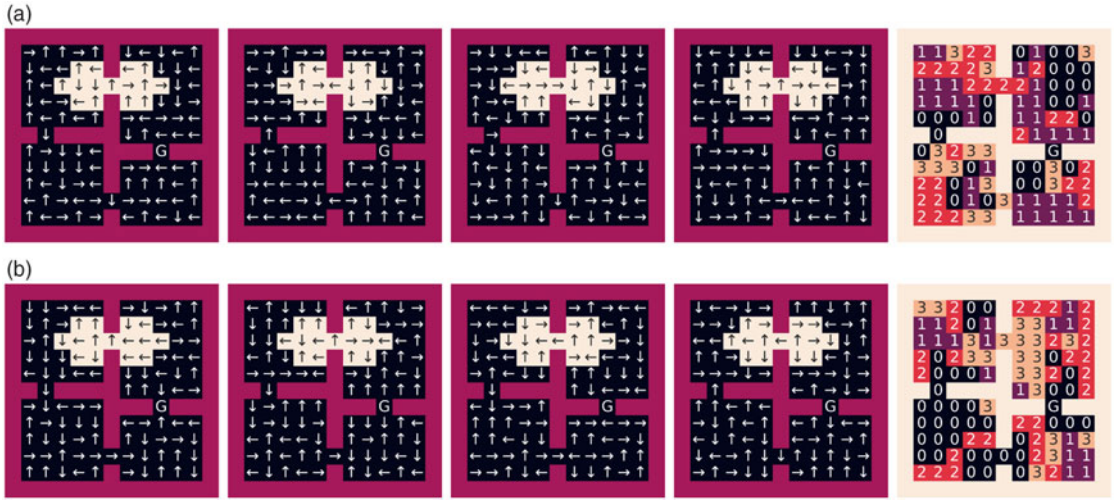

Figure 6 Converged policy for four options in Four-Room environment. The first four figures in each row depict the intra-option policy for all the four options. The last figure in both the rows shows the policy over the options where the number depicts the different options (numbering of options increases from left to right image in a row). Both the baseline and the Safe OC exhibit that the overall hierarchical policy is more reasonable than the individual option policy, though the overall trend of an individual safe option policy is to avoid the frozen hallway.

the rows). The overall trend observed from the individual option policy of safe algorithm is to avoid the unsafe frozen patch, whereas the vanilla option policies do not differentiate between the two hallways.

4.2 Continuous state-space environment

In this section, we examine the performance in the puddle-world Open AI gym environment with the linear function approximation. It is a continuous two-dimensional state-space environment in $[0, 1]^2$. The

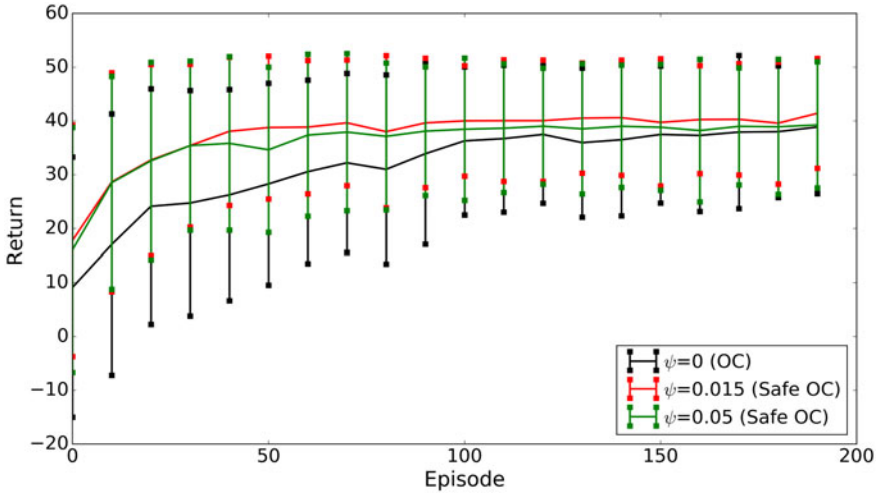

Figure 7 Puddle-world environment: The graphs depict the return averaged over 50 independent trials with 2 options. The vertical bands show the standard deviation in the return over multiple trials. The experiment with safe policy $\psi = 0.015$ (red) has a smaller standard deviation as compared to the baseline (black), signifying safety helps an agent to avoid variance inducing regions.

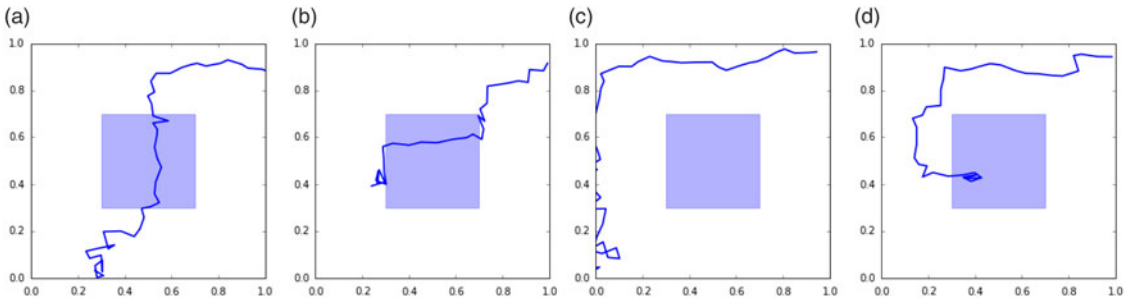

Figure 8 Trajectories in Puddle World: The graph shows sampled trajectories from OC policy in (a) & (b), and Safe-OC policy in (c) & (d). With the designed safety constraint, the agent learns to bypass the unsafe central puddle.

action space consists of four actions that change the position by a value of 0.05 in any one of the coordinates. In each transition, noise is drawn from the *Uniform*$[-0.025, 0.025]$ distribution, which is added to both the coordinates. For introducing the notion of unsafe regions, similar to the four-room environment, we added a square puddle region with a centre at (0.5, 0.5) and 0.4 height. An agent can be randomly initialized from any state, and the goal is located at (1,1). When the agent is within 0.1 L1 distance from the goal state, the agent receives a reward of 50. Whenever the agent transitions from a puddle region, it receives a reward from a normal distribution of $\mathcal{N}(\mu = 0, \sigma = 8)$, and 0 reward otherwise. On expectation, the reward received from both regions is the same. The penalty makes the puddle locations 'unsafe' due to the variability in the rewards.

For these experiments, we use two options. One could choose to have more number of options, but given the complexity of the task, we decide to use only two options. We use intra-option Q-learning in the critic to learn policy over options. Boltzmann distribution is used to represent both intra-option policies and policy over options with a temperature value of 0.1. To learn the termination function, we use sigmoid activation. The agent can take a maximum of 5000 steps in an episode. The standard tile coding (Sutton & Barto, 1998) is used for discretizing the state-space. We use 5 tilings, each of 5×5 over the joint space of two features which is hashed to a vector of 1024 dimension. The discount factor γ is set to 0.99. The optimized parameters are shown in Table 1. The code for both the discrete four-rooms and the continuous puddle world is available at Github[1].

[1] https://github.com/arushi12130/SafeOptionCritic

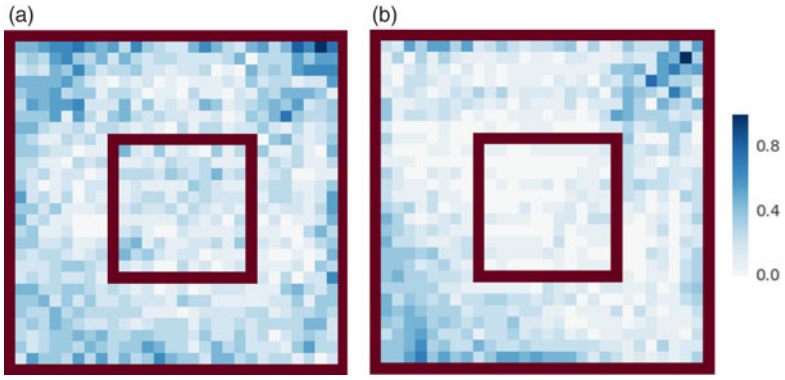

Figure 9 Visitation frequency in Puddle-World environment depicts the state visitation frequency averaged over 50 independent trials for a particular episode in the (a) baseline OC and (b) Safe-OC. The graph highlights that the Safe-OC agent's visit to the unsafe central puddle region decreases in comparison to the OC agent. Our notion of safety enforces the agent to avoid the puddle region.

Figure 7 shows the return observed in the puddle-world domain with different values of ψ averaged over 50 trials. The best performance in terms of stability was observed by $\psi = 0.015$ Safe-OC which exhibits a reduction in the standard deviation of the return across multiple runs as compared to the baseline ($\psi = 0$). This highlights that the agent following the safe policy learns to avoid the variance inducing puddle region. Figure 8 shows the sampled trajectories from both the baseline OC and the Safe-OC. To further validate, the frequency of visits is shown in Figure 9a and 9b. They describe the average number of visits across multiple runs for an episode, depicting a decrease in the number of visits to the unsafe puddle region in the safe policy learning option-critic framework as compared to the baseline. It demonstrates that the safe policy learning agent avoids the erratically behaving unsafe puddle region placed at the centre of the state space.

4.3 Arcade learning environment

We now analyze our method in the ALE domain (Bellemare *et al*., 2013). Harb *et al*. (2018) introduced deliberation cost in the options framework to learn temporally extended options by adding a penalty cost on switching options too frequently. We use the asynchronous advantage option-critic (A2OC) (Harb *et al*., 2018) as our baseline for learning 'safe' options with non-linear function approximation. A2OC provides an extension of the asynchronous advantage actor-critic (A3C) algorithm (Mnih *et al*., 2016) in the option-critic architecture. Introducing the controllability function in the A2OC algorithm results in an additional regularizer term in the intra-option policy gradient (see Equation (12)). The update rule for the intra-option policy gradient in the A2OC with controllability is given as follows:

$$\theta_\pi \leftarrow \theta_\pi + \alpha_{\theta_\pi} \Big\{ \frac{\partial \log \pi_{w,\theta}(a|s)}{\partial \theta} \big(G - Q_\Theta(s, w)\big) - \underbrace{\psi \frac{\partial \log \pi_{w_0,\theta}(a_0|s_0)}{\partial \theta} \delta^2(s_0, w_0, a_0)}_{\text{controllability}} \Big\}. \tag{17}$$

Here, similar to the A2OC algorithm, G is a mixture of n-step return, with a slight modification that the n-step return is only considered until the current option terminates or number of steps are more than n. Without any loss in generality, the one-step TD error in the definition of controllability can be extended to the n-step TD error if the current option persists until the n^{th} step. Similarly, as discussed in Equation (16), there is no change in the termination gradient and we use the same update rule as derived in the A2OC algorithm. In the below equation, η represents the deliberation cost (Harb *et al*., 2018) which penalizes the frequent switching of the options,

$$\nu \leftarrow \nu - \alpha_\nu \frac{\partial \beta_{w,\nu}(s')}{\partial \nu} \big(Q_\Theta(s', w) - V_W(s') + \eta\big). \tag{18}$$

Table 2. ALE final scores: The performance is averaged over 100 games once the training is completed. The scores are averaged over five independent seeds. Scores in the boxes highlight the performance without controllability function (baseline), 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 (grey) in two out of three games analyzed here. Learning options with our notion of safety outperforms the baseline A2OC in all three games. A3C scores have been taken from Mnih *et al*. (2016), DQN from Nair *et al*. (2015), Double DQN from Van Hasselt *et al*. (2016), and Duelling from Wang *et al*. (2015). ψ represents the degree of controllability. Values in the brackets indicate standard deviation across 100 games for 5 independent runs.

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
Duelling	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)

We evaluate the performance in three games, namely, MsPacman, Amidar, and Q*Bert from the ATARI 2600 suite. We introduce Safe-A2OC[2] which uses a similar deep network architecture as A2OC but with additional safety criteria added to it. We use ϵ-greedy for learning policy over options. Intra-option policies follow the linear Boltzmann distribution. Termination functions use sigmoid activation along with linear function approximation for the Q values. For hyperparameters, we learn four 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 also optimize for ϵ parameter for the baseline case ($\psi = 0$). For fair analysis, we compare the best performance of A2OC with varying ψ regularizer for controllability function in the Safe-A2OC.

Results and Evaluation: To evaluate the agents, we investigate the averaged performance over 100 games after the training has been completed over 80M frames (Machado *et al*., 2017). Table 2 depicts the averaged performance over five different runs, where each run contains the averaged score obtained over 100 testing games. It is observed that the Safe-A2OC with a controllability value of $\psi = 0.10$ in MsPacman not only outperforms the score achieved by the baseline A2OC ($\psi = 0$) but also results in a smaller variance in the scores (values are shown in the braces). We observe a similar pattern in Q*Bert, wherein the Safe-A2OC ($\psi = 0.05$) shows an improved robust performance over the A2OC ($\psi = 0$), achieving almost half of the variance of A2OC across multiple games.

In the game of Amidar, we note that Safe-A2OC outperforms the score achieved by A2OC, and the other state-of-the-art approaches (Nair *et al*., 2015; Wang *et al*., 2015; Mnih *et al*., 2016; Van Hasselt *et al*., 2016) using the primitive actions. Empirically, we observe that on adding an optimal amount of controllability value in options, an agent optimizing for low variance in the TD error learns better than the one optimizing only for the cumulative reward. Intuitively, minimizing the variance in the TD error as a measure of safety helps the agents avoid states with high intrinsic variability in the reward. 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. Based on the amount of variability in the reward structure, each game benefits differently in the performance. Overall, a consistent boost in the performance is demonstrated with the proposed approach once the training is completed.

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

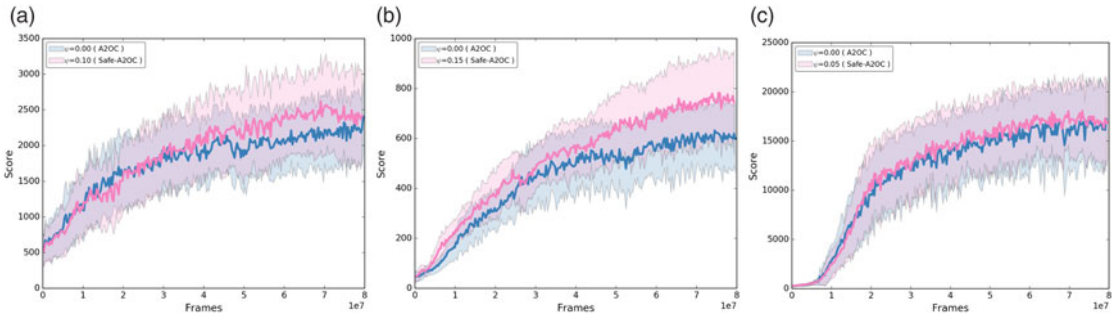

Figure 10 Learning curves in Atari games during training demonstrate that options with a controllability factor of $\psi = 0.10$ learn better than the best-performing A2OC baseline ($\psi = 0, \epsilon = 0.2$). Higher value of ψ results in a poor performance. Performance is averaged across 5 independent seeds for each game.

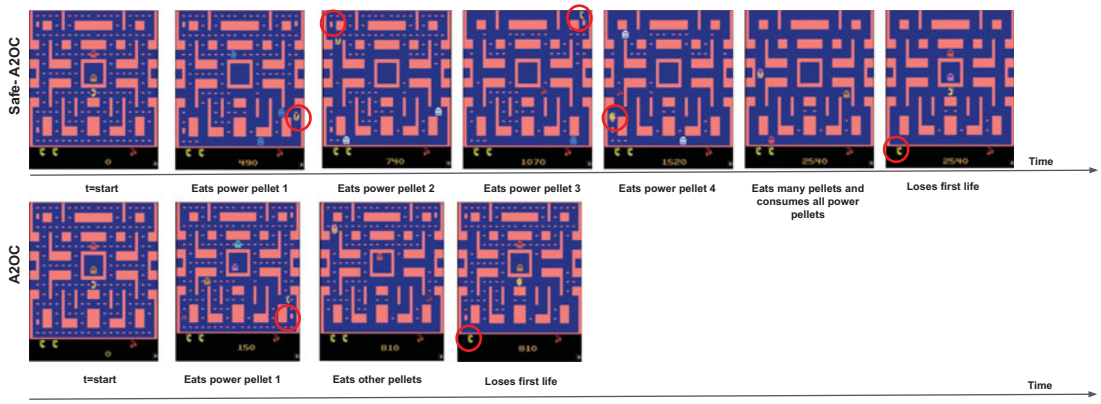

Figure 11 Qualitative analysis in MsPacman: Our interpretation is that the Safe-A2OC agent understands the source of intrinsic variability in the reward due to explicitly optimizing for the variance in TD error. This results in Safe-A2OC agent to adopt the strategy of eating the large flashing *power pellets*, causing the ghosts to flee. The Safe-A2OC agent is able to survive longer and ends up accumulating more cumulative reward as opposed to the baseline A2OC agent. Here, we depict a sampled trajectory of the Safe-A2OC agent in comparison with the baseline A2OC agent.

We also inspect the learning curves during the training phase for all three Atari games. Figure 10 depicts the learning curves over 80M frames with varying controllability parameter. The results are averaged across five runs per game. In Amidar, we observe that with a specific degrees of controllability ($\psi = 0.15$), options learned with our notion of safety (Safe-A2OC) outperform the baseline option-critic (A2OC). An improvement in the training performance also translates well to the testing phase as seen in Table 2. Interestingly, we observe that the performance of Safe-A2OC improves only marginally during the training phase in Ms-Pacman and is at-par with A2OC in Q*Bert. However, it is noteworthy that while we do not see much improvement in the training curves for both MsPacman and Q*Bert, during the testing phase, both these games show robust and improved performance. According to our interpretation, explicitly optimizing for the variance in TD error might not always result in a performance boost. Nevertheless, the proposed objective function makes the agent aware of intrinsic variability in the rewards. In the subsequent section, we will qualitatively analyze the behaviour of the trained agents.

Qualitative Observations: We are now interested in better understanding the behaviour of the trained agents. We observe that different values of ψ control the degree to which an agent tends to be risk averse. A grid search over the different degrees of the controllability hyper parameter 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 three games is obtained around $0.05 - 0.15$. For a qualitative analysis, we also present the videos of the trained agents[3].

[3] Videos of trained agents in Atari games are available at https://sites.google.com/view/safe-option-critic.

Upon visual inspection, we observe improvements in game playing strategies for Safe-OC agent as compared to its counterpart OC. The Safe-A2OC agents demonstrate relatively better understanding of the source of intrinsic variability in the reward structure. For instance, consider the game of MsPacman. The agent is tasked with eating all of the pellets in an enclosed maze while avoiding the four coloured ghosts. On consuming the power pellets in the environment, the ghosts turns blue and flee, resulting in bonus points. Frame-by-frame analysis shows that the Safe-A2OC agent is able to escape from the ghosts and stay alive longer when these ghosts are harmful in the context of a terminal state. Figure 11 depicts this observation, where the A2OC agent loses its first life relatively quickly as compared to the Safe-A2OC agent. These insights demonstrate that explicitly optimizing for the variance in TD error results in avoiding the visits to states with intrinsic variability in the rewards. Particularly, in MsPacman, the additional cost in the objective function helps the agent to understand the intrinsic variability in reward structure such as the behaviour observed upon the acquisition of the power pellets.

Furthermore, on close analysis of the behaviour of the Safe-OC agents, we observe that agents with extremely high value of ψ are relatively more risk-averse but less exploratory in nature, thereby resulting in a poor performance (see Appendix 1). This corroborates with the findings in the learning curves as shown in Figure 10. For an optimal value of controllability regularizer (ψ), we observe a balanced amount of risk aversion and, therefore, the agent adopts a behaviour which demonstrates a better understanding of intrinsic variability in the rewards structure. We perform a similar analysis for the game of Amidar and report similar insights in Appendix 2.

5 Discussion

In this work, we introduced *Safe Option-Critic*, a general framework that extends the idea of controllability from the primitive action space to the option-critic architecture. The key idea is to discourage the agent from visiting the state space with high uncertainty in their behavioural outcomes by constraining the variance in TD error. Recently, Sherstan *et al*. (2018) proposed a direct method to estimate the variance of λ return. The authors proposed a Bellman operator for the variance that uses the square of the TD error. This work further supports our approach to estimating the risk using the squared TD error.

Our experiments in the tabular case empirically demonstrate the reduced variance in the return. Moreover, we observe a boost in the overall performance for both the tabular and the linear function approximation. Experiments in the ALE domain demonstrate that the agent can learn about the intrinsic variability in a large and complicated state-space using non-linear function approximation. Qualitative insights from ALE also demonstrate that the options with a notion of safety are more cautious in their behaviour and, therefore, result in improved overall performance.

Limitations and Future Work: In this work, we limit the return calculation until an option terminates. Using the n-step returns at the SMDP level is of potential interest for future work. Additionally, in this work, we assume that all the options are available in every state. In the context of safety, it would be of interest to understand the setting where options initiation sets are limited to a subset of the entire state space (Khetarpal *et al*., 2020). One could also add a scheduler on the controllability regularizer ψ over time. For instance, ψ could be initiated with a small value to encourage exploration in the initial stages of learning, and gradually the value of ψ could be increased to limit the exploration in the unsafe states.

Besides, a more formal analysis of the agent behaviour in ALE can provide a rigorous understanding of the interplay between learning the dynamics and the controllability parameter. For instance, studying the training and policy traces by human subjects is scope for future work.

In the proposed algorithm, we consider the notion of safety by accounting for the controllability of only the initial state-option pair. However, one can instead account for the variability due to the entire state-space by means of bootstrapping the target value function, aliasing due to the function approximation, and random resets to starting state. A potential direction for future work is to extend the controllability from the initial state-option pair to all the state-option pairs in the trajectory. This extension could potentially enhance the effects of the risk mitigation and speed up learning. The proposed notion of safety can also be extended to different levels of hierarchy, for example, a mixture of options with varying degrees of controllability can be learned.

Acknowledgements

The authors would like to thank Open Philanthropy for funding this work, Compute Canada for the computing resources, Herke van Hoof, Ayush Jain, Pierre-Luc Bacon, Gandharv Patil, Jean Harb, Martin Klissarov, Kushal Arora, for constructive discussions throughout the duration of this work, and the anonymous reviewers for the feedback on earlier drafts of this manuscript.

Competing interests

The authors declare none.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J. & Mané, D. 2016. Concrete problems in AI safety. *CoRR*.

Bacon, P.-L., Harb, J. & Precup, D. 2017. The option-critic architecture. In *AAAI*, 1726–1734.

Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Mourad, S., Silver, D., Precup, D., *et al.* 2019. The option keyboard: combining skills in reinforcement learning. *In Advances in Neural Information Processing Systems*, 13052–13062.

Barto, A. G. & Mahadevan, S. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems* **13**(4), 341–379.

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

Borkar, V. S. & Meyn, S. P. 2002. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research* **27**(1), 192–209.

Daniel, C., Van Hoof, H., Peters, J. & Neumann, G. 2016. Probabilistic inference for determining options in reinforcement learning. *Machine Learning* **104**(2–3), 337–357.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* **13**, 227–303.

Fikes, R. E., Hart, P. E. & Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence* **3**, 251–288.

Fikes, R. E., Hart, P. E. & Nilsson, N. J. 1981. Learning and executing generalized robot plans. In *Readings in Artificial Intelligence*. Elsevier, 231–249.

Future of Life Institute 2017. Asilomar AI principles.

Garcıa, J. & Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* **16**(1), 1437–1480.

Gehring, C. & Precup, D. 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 2013, 1037–1044.

Geibel, P. & Wysotzki, F. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research (JAIR)* **24**, 81–108.

Harb, J., Bacon, P.-L., Klissarov, M. & Precup, D. 2018. When waiting is not an option: learning options with a deliberation cost. In *AAAI*.

Heger, M. 1994. Consideration of risk in reinforcement learning. In *Machine Learning Proceedings 1994*. Elsevier, 105–111.

Howard, R. A. & Matheson, J. E. 1972. Risk-sensitive Markov decision processes. *Management Science* **18**(7), 356–369.

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

Iyengar, G. N. 2005. Robust dynamic programming. *Mathematics of Operations Research* **30**(2), 257–280.

Jain, A., Patil, G., Jain, A., Khetarpal, K. & Precup, D. 2021. Variance penalized on-policy and off-policy actor-critic. arXiv preprint *arXiv2102.01985*.

Jain, A. & Precup, D. 2018. Eligibility traces for options. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 1008–1016.

Khetarpal, K., Klissarov, M., Chevalier-Boisvert, M., Bacon, P.-L. & Precup, D. 2020. Options of interest: Temporal abstraction with interest functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 4444–4451.

Konidaris, G. & Barto, A. G. 2007. Building portable options: Skill transfer in reinforcement learning. *In IJCAI*, **7**, 895–900.

Konidaris, G., Kuindersma, S., Grupen, R. A. & Barto, A. G. 2011. Autonomous skill acquisition on a mobile manipulator. In *AAAI*.

Korf, R. E. 1983. *Learning to Solve Problems by Searching for Macro-operators*. PhD thesis, Pittsburgh, PA, USA. AAI8425820.

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

Law, E. L., Coggan, M., Precup, D. & Ratitch, B. 2005. Risk-directed exploration in reinforcement learning. In *Planning and Learning in A Priori Unknown or Dynamic Domains*, 97.

Lim, S. H., Xu, H. & Mannor, S. 2013. Reinforcement learning in robust Markov decision processes. *Advances in Neural Information Processing Systems* **26**, 701–709.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M. & Bowling, M. 2017. Revisiting the arcade learning environment: evaluation protocols and open problems for general agents. *ArXiv e-prints*.

Mankowitz, D. J., Mann, T. A. & Mannor, S. 2016. Adaptive skills adaptive partitions (ASAP). *In Advances in Neural Information Processing Systems*, 1588–1596.

McGovern, A. & Barto, A. G. 2001. Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML*, **1**, 361–368.

Menache, I., Mannor, S. & Shimkin, N. 2002. Q-cut - dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*. Springer, 295–306.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. & Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 1928–1937.

Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., Maria, A. D., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., Legg, S., Mnih, V., Kavukcuoglu, K. & Silver, D. 2015. Massively parallel methods for deep reinforcement learning. *CoRR*.

Nilim, A. & El Ghaoui, L. 2005. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research* **53**(5), 780–798.

Parr, R. & Russell, S. J. 1998. Reinforcement learning with hierarchies of machines. *In Advances in Neural Information Processing Systems*, 1043–1049.

Precup, D. 2000. *Temporal abstraction in reinforcement learning* (University of Massachusetts Amherst).

Riemer, M., Liu, M. & Tesauro, G. 2018. Learning abstract options. *In Advances in Neural Information Processing Systems*, 10424–10434.

Sherstan, C., Ashley, D. R., Bennett, B., Young, K., White, A., White, M. & Sutton, R. S. 2018. Comparing direct and indirect temporal-difference methods for estimating the variance of the return. In *Proceedings of Uncertainty in Artificial Intelligence*, 63–72.

Stolle, M. & Precup, D. 2002. Learning options in reinforcement learning. In *International Symposium on Abstraction, Reformulation & Approximation*. Springer, 212–223.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* **3**(1), 9–44.

Sutton, R. S. & Barto, A. G. 1998. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition.

Sutton, R. S., McAllester, D. A., Singh, S. P. & Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. *In Advances in Neural Information Processing Systems*, 1057–1063.

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

Tamar, A., Di Castro, D. & Mannor, S. 2012. Policy gradients with variance related risk criteria. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning*, 387–396.

Tamar, A., Di Castro, D. & Mannor, S. 2016. Learning the variance of the reward-to-go. *Journal of Machine Learning Research* **17**(13), 1–36.

Tamar, A., Xu, H. & Mannor, S. 2013. Scaling up robust MDPs by reinforcement learning. *arXiv preprint arXiv:1306.6189*.

Van Hasselt, H., Guez, A. & Silver, D. 2016. Deep reinforcement learning with double Q-learning. In *AAAI*, **16**, 2094–2100.

Vezhnevets, A., Mnih, V., Osindero, S., Graves, A., Vinyals, O., Agapiou, J., *et al.* 2016. Strategic attentive writer for learning macro-actions. *In Advances in Neural Information Processing Systems*, 3486–3494.

Wang, Z., de Freitas, N. & Lanctot, M. 2015. Dueling network architectures for deep reinforcement learning. *CoRR*.

White, D. 1994. A mathematical programming approach to a problem in variance penalised Markov decision processes. *Operations-Research-Spektrum* **15**(4), 225–230.

Appendix A

Appendix A.1 Experiments in the ALE domain

In this section, we show the results of training the agent in ALE games with multiple values of controllability parameter (ψ). Figure A1 shows the averaged results across five runs per game.

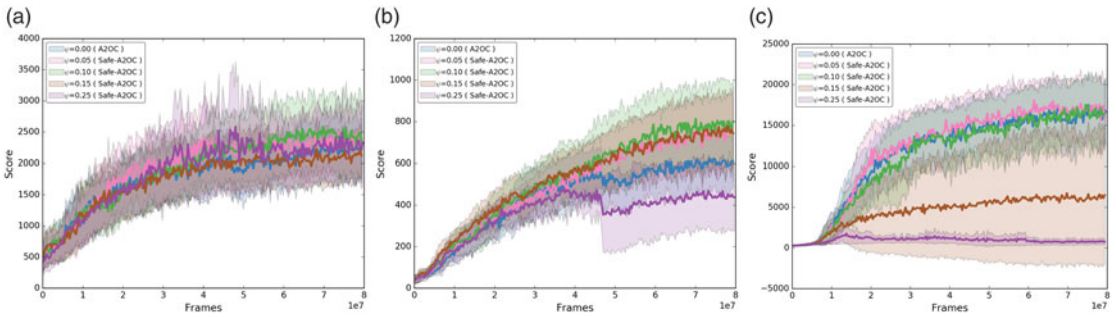

Figure A1 Learning curves during training for ALE games: Higher degrees of controllability ($\psi > 0.15$) result in reduced exploration and adversely affect the performance. We observe that the agents trained with an extremely high value of ψ are relatively more risk averse as compared to smaller values of controllability.

Appendix A.2 Qualitative observations in Amidar

We also analyze the behaviour of the trained agents in the game of Amidar as shown in Figure A2. Just as MsPacman, in Amidar, the agent is faced with enemies, who upon contact can kill the agent. The agent's goal is to paint the rectangular boards by traversing all the sides of the board. Upon painting all the four corners of a rectangle, the agent is briefly equipped with the ability to make the enemies ineffective for a short duration, analogous to consuming the power-pellets in MsPacman.

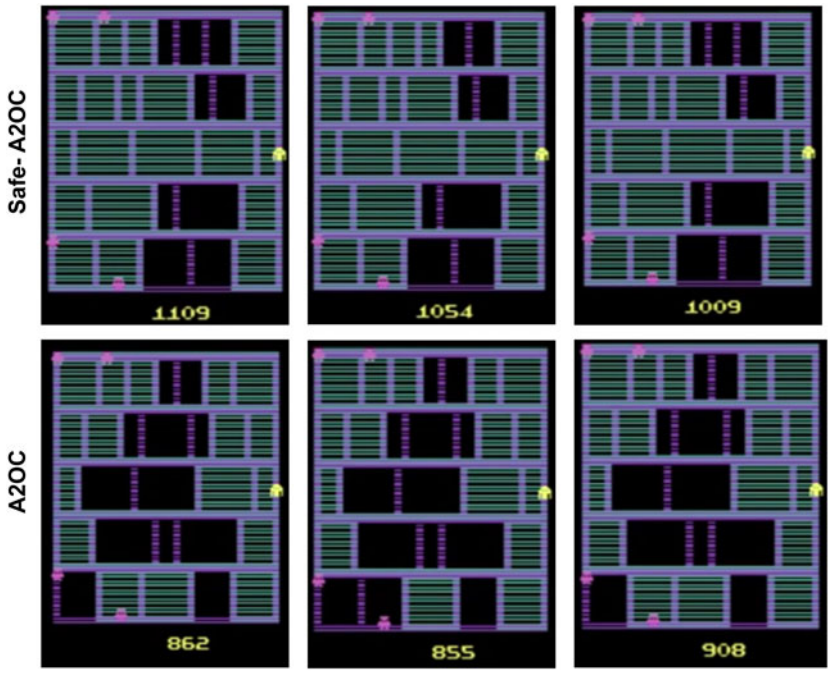

Figure A2 Qualitative analysis in Amidar: We here demonstrate the final frames from randomly sampled 3 games played by the trained agents. Safe-A2OC agent learns to paint relatively more rectangles as compared to the A2OC agent as painting all sides of a rectangle results in making the ghosts ineffective, and thus increases the cumulative return. Amidar offers intrinsic variability in the reward structure, and our approach empowers the agents to understand it better.

Figure A2 depicts the final frame in three games played by the trained agents. Safe-A2OC agents end up painting more rectangles than the A2OC agent. There is inherent variability in the reward structure of this environment. Since painting all four sides of the rectangle results in the ability to make the enemies ineffective, Safe-A2OC agents are able to understand this structure relatively better by directly optimizing for the variance in TD error. Therefore, Safe-A2OC agents result in painting more rectangles in the board as depicted, leading to a higher cumulative reward than the baseline agent.