Diversifying Options in Option-Critic Framework of Hierarchical Reinforcement Learning

Ruoyu Yan Wenda Chu

Date: January 10, 2021

1 Introduction

Reinforcement learning has achieved great successes in many different domains recent years. However, it remains a big challenge for these method to address environments with sparse and delayed rewards, which are often encounter in real world scenarios. As an innovative approach to solve this problem, Hierarchical Reinforcement Learning manages to learn knowledge at multiple levels and make plans with temporal abstraction. In addition to its great performance on sparse reward problems, previous researches have also revealed its potential of transfer learning.

Two main approaches have been proposed for designing HRL architectures. The first one is to find and assign subgoals to guide the low level policy[3]. The other one is to learn skills on the low level policy and a policy to utilize these skills on the higher level.

In our research, we focus on the option framwork[1] as a representative of the second approach. We implement the Option-Critic architecture [2] and reproduce its result on maze problems. During experiments, however, we find the natural tendency of the agent to develop only one option for the whole problem, which essentially degrades to vanilla policy gradient method. We are therefore motivated to develop methods to enhance the diversity of options. We consider several possible methods including dropouts on options, giving intrinsic rewards to guide the choice of options and enhancing option specialization on termination probability.

2 Preliminaries

A reinforcement learning problem is typically modeled by a Markov Decision Process, which defines a set of state $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and a reward function R. The options framework, proposed by Sutton et al.[1], extend the basic actions by introducing multiple options $\omega \in \Omega$. Each option can be expressed by a triple $\langle \mathcal{I}, \beta, \pi \rangle$, where $\mathcal{I} \subseteq S$ is the set initial states where o can be called; $\beta : \mathcal{S} \to [0, 1]$ is the termination probability of the option; and $\pi : \mathcal{S} \times A \to [0, 1]$ is the intra-option policy.

While an agent is allowed to develop options on the low level, it should also learns a policy over options on the high level that chooses to execute at each step either a primitive action or an option involving a sequence of actions. A corresponding value function $V_\Omega(s)$ and option value function $Q_\Omega(s, \omega)$ can also be learned using a generalized version of Bellman equation.

Our work relies heavily on the Option-Critic architecture proposed by Bacon et al.[2] Each option $\omega \in \Omega$ has its intra-option policy π_ω and termination probability β_ω parameterized. Thus, policy gradient method can be applied for both the development of options and the optimization of the policy over options π_Ω. The option value function Q_Ω and the value of state action pair of a particular option Q_U can be written as:

$$Q_\Omega(s, \omega) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a) \tag{1}$$

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

where $U(\omega, s')$ is the value of executing option ω upon entering state s', so

$$U(\omega, s') = (1 - \beta_{\omega,\vartheta}(s')) Q_\Omega(s', \omega) + \beta_{\omega,\vartheta}(s') V_\Omega(s') \tag{3}$$

Two important theorems are proved in [2] for updating $\pi_{\omega,\theta}$ and $\beta_{\omega,\vartheta}$.

$$\nabla_\theta ER = \sum_{s,\omega} \mu_\Omega(s, \omega|s_0, \omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a) \tag{4}$$

$$\nabla_\vartheta ER = -\sum_{s',\omega} \mu_\Omega(s', \omega|s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \vartheta) \tag{5}$$

where $\mu_\Omega(s, \omega|s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = t|\omega_t = \omega|s_0, \omega_0)$ is the discounted weighting of state option pairs along trajectory starting at (s_0, ω_0).

3 Method

The following methods are applied to the Option-Critic model, with a view to keeping the diversity among the options.

3.1 Dropout

We first consider the method of simply adding dropouts to options. At each step of the high level policy, each option has an independent probability p of being dropped. In this way, we force the agent to develop different options instead of always using one highly developed option.

However, this naive solution only helps developing more options, but gives no guarantee for keeping the diversity of different options. In the worst case, the agent may develop multiple options that handle exactly the same subtask, which leads to a great redundancy of both computation and memory resource.

3.2 Intrinsic Reward

To deal with the problems above, we add intrinsic reward to the option value function $Q_\Omega(s, \omega)$ for better exploration of different options.

Inspired by [4], we added an intrinsic curiosity module (ICM) in the architecture, which gives a prediction of the features of the termination state of executing an option $\hat{\phi}(s_{t+1})$ given current state feature $\phi(s_t)$ and the chosen option ω. Namely,

$$\hat{\phi}(s_{t+1}) = f(\phi(s_t), \omega) \tag{6}$$

The ICM gives curiosity r_t as an intrinsic reward to $Q_\Omega(s, \omega)$, where

$$r_t^c = \frac{\eta}{2}||\hat{\phi}(s_{t+1} - \phi(s_{t+1}))||_2^2 \tag{7}$$

However, ther option-version curiosity based intrinsic reward does not work well empirically. This is partially due to the hardness of predicting the termination state of stochastic process with randomness and uncertainty.

Therefore, we consider another intrinsic reward with less predicting difficulty. Instead of predicting $\phi(s_{t+1})$, we predict the option used given start state s_t and the termination state of this option s_{t+1}. Specifically, we train a model $\hat{g}$ to estimate:

$$g(\phi(s_t), \phi(s_{t+1})) = \Pr(\omega | s_t, s_{t+1}) \tag{8}$$

We compare $\hat{g}(\phi(s_t), \phi(s_{t+1}))$ with the real option used ω^* (here ω^* denotes a 0/1 vector with size=number of options), and apply

$$r_t^i = \eta||\hat{g}(\phi(s_t), \phi(s_{t+1})) - \omega^*||_2^2 \tag{9}$$

as an intrinsic reward to $Q_\Omega(s, \omega)$. For each given pair of features $(\phi(s_t), \phi(s_{t+1}))$, we first use $\hat{g}^{(t)}$ to generate an intrinsic reward, and then train the network $\hat{g}$ with a minibatch of B containing the new pair $(\phi(s_t), \phi(s_{t+1}); \omega_t)$. The iteration is as follows:

$$\begin{cases} r_t = r(s,a) + r_t^i = r(s,a) + \eta||\hat{g}^{(t)}(\phi(s_t), \phi(s_{t+1})) - \omega_t||_2^2 \\ g^{(t+1)} \leftarrow SGD\left(g^{(t)}, \{\phi(s_k), \phi(s_{k+1}); \omega_k\}_{k=t-B}^{t}\right) \end{cases} \tag{10}$$

The intrinsic reward r_t^i essentially estimates the unpredictability of an option ω, which defines curiosity in another way.

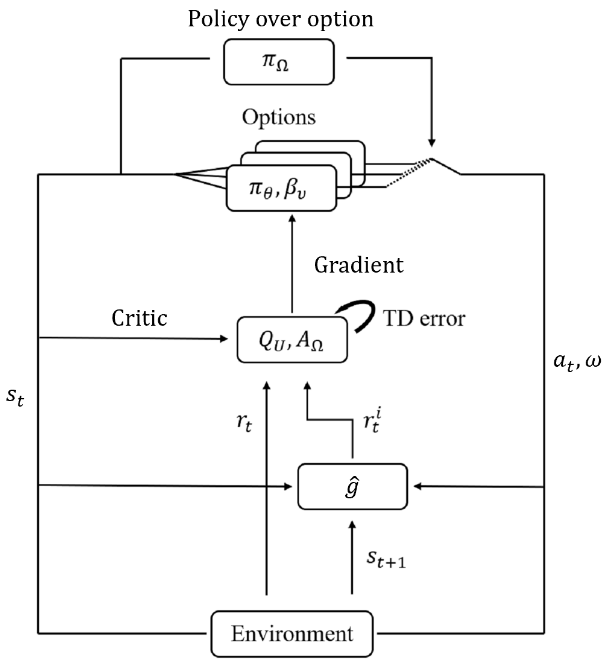


Figure 1: Option-Critic Architecture with Intrinsic Reward

3.3 Specialization by Controlling Termination Probability

Both of the above approaches attempt to keep the diversity of options by learning a better high level policy to guide the development of options, but they do not further consider the training details inside each options on the low level. We follow the idea of [5], which gives the termination probability π_ω of each option ω an alternative objective:

$$I(s_{t+1};\omega|s_t) = H(s_{t+1}|s_t) - H(s_{t+1}|s_t,\omega) \tag{11}$$

where s_t and s_{t+1} are start and terminal states of option ω. H is the entropy function, so $I(s_{t+1};\omega|s_t)$ is essentially the conditional mutual entropy between option-terminating states and options conditioned on the initial state s_t, and can thus measures their correlation.

4 Experiment

We test our model using different number of options (1, 4 or 9) on maze games. The start point is drawn uniformly at random among the maze and the agent is rewarded 1 if it reaches the terminal state. The terminal state is changed randomly every 5000 episodes.

4.1 Performance of Original Option-Critic

4.1.1 Convergence

We first test the performance of the originial Option-Critic architecture on a 20×20 grid world that consists of 9 different rooms.

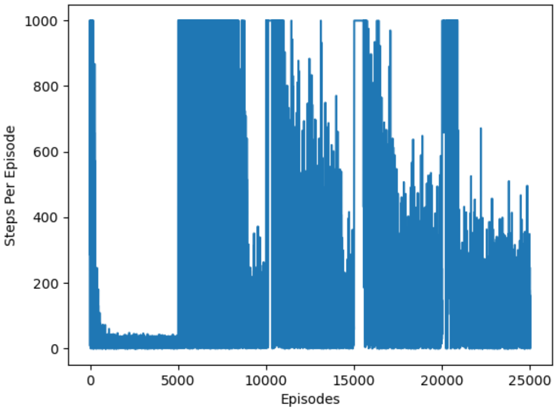


Figure 2: 1 option

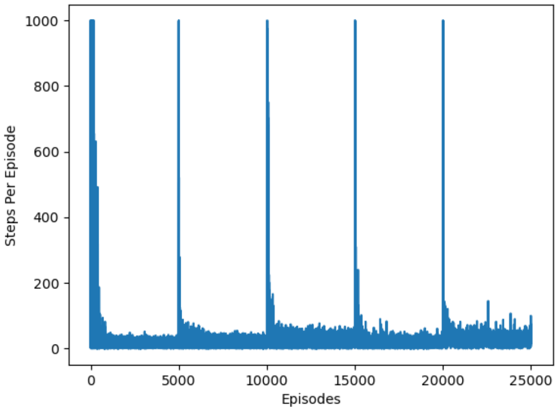


Figure 3: 9 options

Remember that we change the goal state every 5000 iterations. The architecture with only 1 option degrades to vanilla policy gradient method. It faces great trouble of realizing the sudden change of goal state, and cannot converge after 5000 iterations. However, the agent with 9 options recovers rapidly (less than 50 iterations, also much faster than the first 5000 iterations as the goal not changed) when the goal changes.The comparison of the figures above shows the significant advantage on transfer learning of option framework over vanilla policy gradient method.

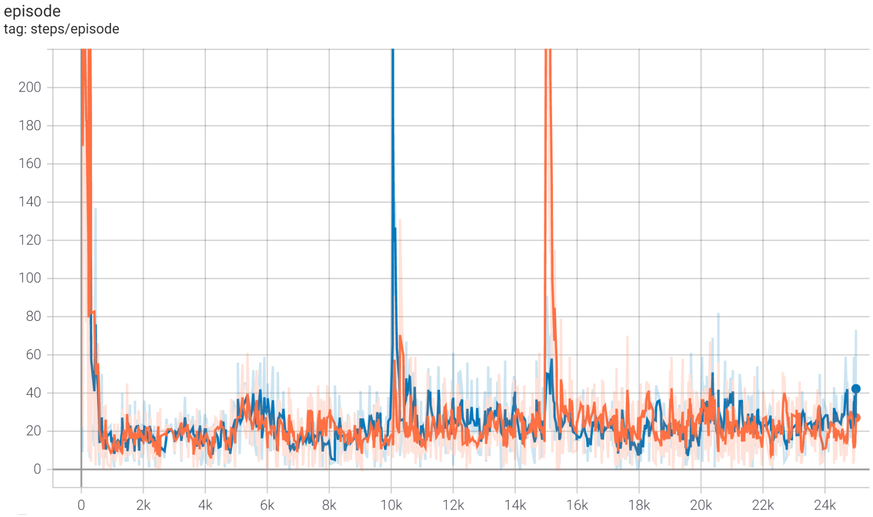


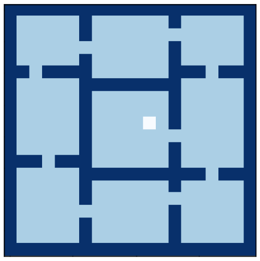
Figure 4: 4 options (Blue) and 9 options (Orange)

4.1.2 Option Visualization

The difference of convergence rate between 4 options and 9 options are too slight to catch for grid world problem. However, the diversity of developed options is another crucial criterion of performance. The intra-option policies and termination probabilities are shown by heatmap as below.

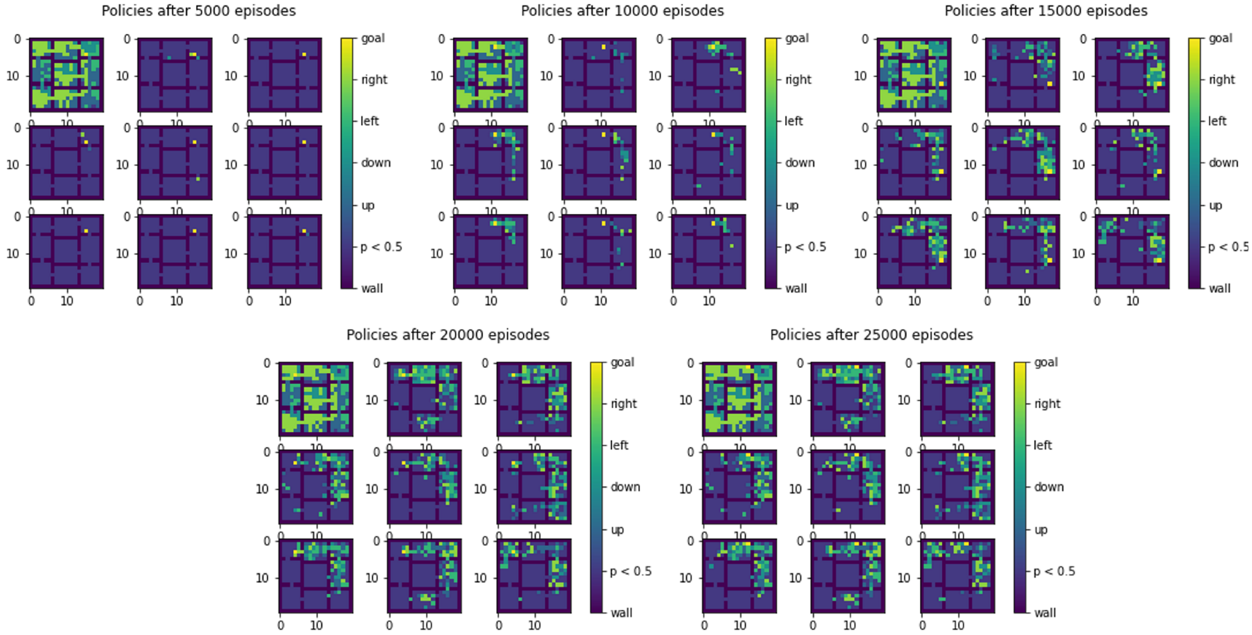
As we have stated above, the agent only develops 1 option in the first 5000 iterations. This is natural since Figure 2 in 4.1.1 has shown that using one options can eventually converge to an optimal strategy when the goal has not been changed. Moreover, when the goal changes, the agent is forced to learn new options because 1 option reacts too slow to this change of environment, as shown by Figure 2 and Figure 3.

However, even though multiple options are activated and developed after 25000 episodes, they are very similar in shape, which means they essentially solve similar subtasks and are thus lacking of diversity.



20×20 nine rooms

Figure 5: The configuration of the 9-room maze



Policies after 5000 episodes

Policies after 10000 episodes

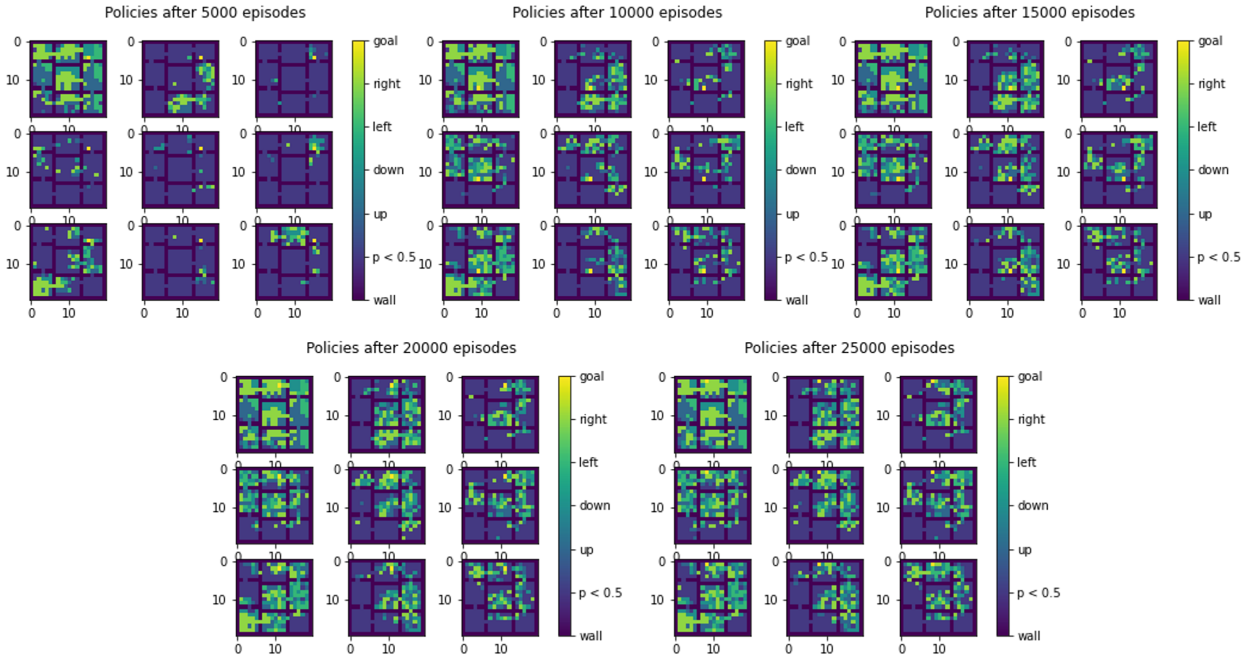
Policies after 15000 episodes

Policies after 20000 episodes

Policies after 25000 episodes

Figure 6: The intra-option policies after each convergence (9 options)

4.2 A naive method: Dropout



Policies after 5000 episodes

Policies after 10000 episodes

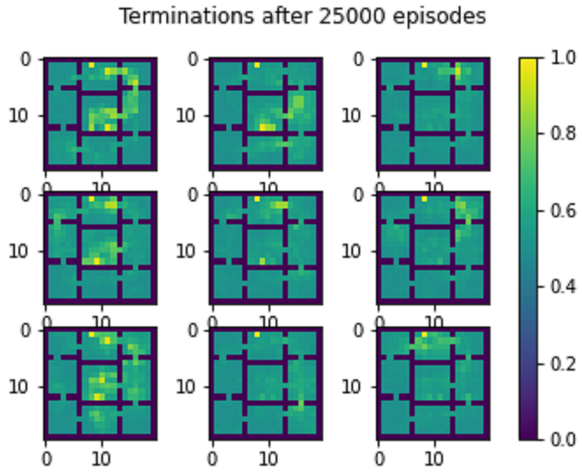
Policies after 15000 episodes

Policies after 20000 episodes

Policies after 25000 episodes

Figure 7: Dropout = 0.2

As shown in Figure 7, the intra-option policies with dropout on options are much more developed than the those without. However, as the former the options do not specialize well. Figure 8 shows the termination probability of each option among the maze. Observe that the termination $\pi_\omega(s)$ are relatively high (greater than 0.5) for any $s \in \mathcal{S}$ and $\omega \in \Omega$. This is a direct evidence to the fact that the options are not specialized: multiple options can be activated at any state and any option can terminate at any state.



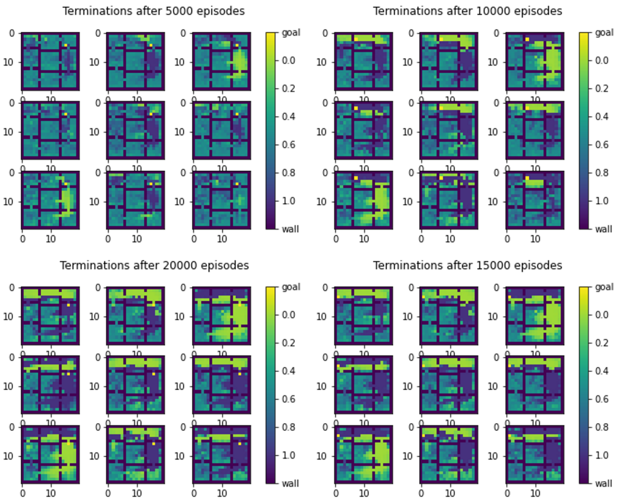
Terminations after 25000 episodes

Figure 8: Termination probability
(Iteration = 25000, Dropout = 0.2)

4.3 Final Performance

For the last experiment, we combine the methods introduced in section 3.2 and 3.3. We set the intrinsic reward rate $\eta = 0.001$ and train an MLP network with 30 hidden neurons for $\hat{g}(\phi(s_t), \phi(s_{t+1}))$. The learning rate is fixed to 0.005.

We no longer use the gradient of the expected discounted return objective that relies on A_Ω to update $\beta_{\omega,\vartheta}$. Instead, use the new objective $I(s_{t+1}; \omega | s_t)$. The termination probability of each option is shown in Figure 9 and 10.



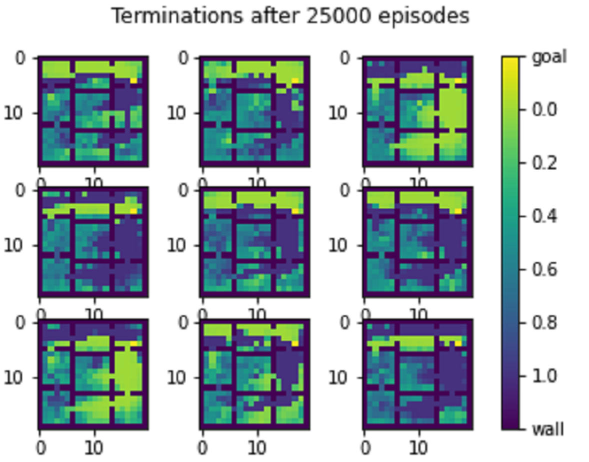
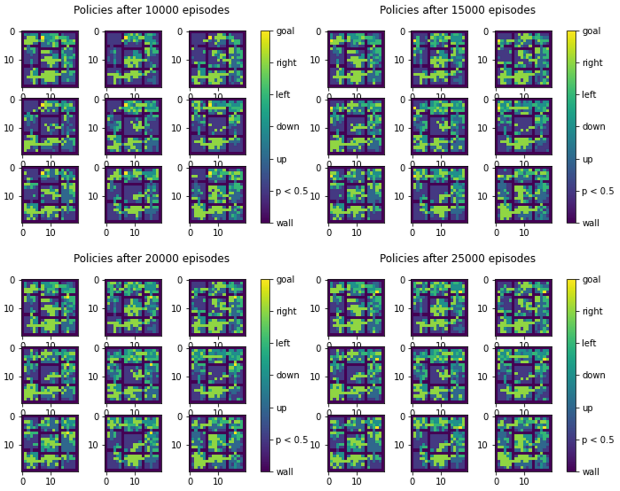


Figure 9: Termination probability

Figure 10: Termination probability (25000 episodes)

It is shown that the area of high termination probability (the states with dark colors) is much diffentiated from the area of low termination probability (the state with bright colors).



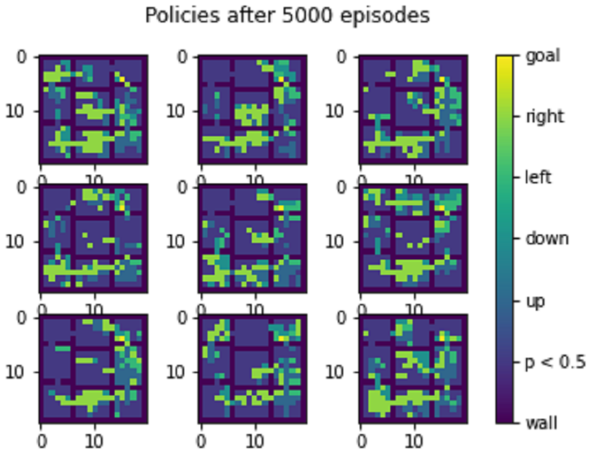


Figure 11: Intra-option policies

Figure 12: Intra-option policies (5000 episodes)

Figure 11 and 12 are the heatmaps of the intra-option policies. Even at the first 5000 episodes, the agent manages to split the whole problem of finding the goal state into subtasks. Each option is evenly developed and the heatmap of each intra-option policy is in different shape, which indicates that the options are well diversified and specialized.

5 Discussion

Empirical and theoretical works have been done to show the promise of hierarchical reinforcement learning to deal with sparse delayed reward environment. Having spotted the tendency of Option-Critic agent to handle the whole problem by only one option or develop multiple homogeneous options, the main incentive of our work is to diversify the development of options. We proposed several approaches to achieve better diversity in options, at the same time enhancing the power of our model in tranfer learning.

Our methods involve both optimization on both policy over options π_Ω and intra-options policies π_ω. We add a predicting network and give intrinsic reward by estimating the unpredictability of an option ω. The option diversity is also promoted by maximizing the correlation between termination probability π_ω and option choosed ω.

In option-critic network, each option $\omega \in \Omega$ has the same initial state $\mathcal{I}_\omega = \mathcal{S}$, namely each option can be initiated on every state $s \in \mathcal{S}$. Making I_ω learnable may utilize the feature of the option framework more completely and improve its ability of temporal abstraction. Another limitation of current Option-Critic architecture is that the number of options is given as a hyperparameter, and each option is generated by parameters (θ, ϑ) by a same function. This homogeneity of all options limits the ability of an agent to handle tasks of different forms and to achieve lifelong learning.

References

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

[2] Bacon, Pierre-Luc, Precup, Doina, and Harb, Jean. The option-critic architecture. In AAAI, 2017.

[3] T.D. Kulkarni, K..R Narasimhan, Saeedi, Ardavan, Tenenbaum, Joshua B. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. NIPS, 2016.

[4] Frank Röder, Manfred Eppe, Phuong D.H. Nguyen, and Stefan Wermter. Curious Hierarchical Actor-Critic Reinforce-mentLearning. arXiv:2005.03420, 2020.

[5] Yuji Kanagawa and Tomoyuki Kaneko. Diverse Exploration via InfoMax Options. arXiv:2010.02756, 2020.