Attention Option-Critic

Raviteja Chunduru [1 2] **Doina Precup** [1 2 3]

Abstract

Temporal abstraction in reinforcement learning is the ability of an agent to learn and use high-level behaviors, called options. The option-critic architecture provides a gradient-based end-to-end learning method to construct options. We propose an attention-based extension to this framework, which enables the agent to learn to focus different options on different aspects of the observation space. We show that this leads to behaviorally diverse options which are also capable of state abstraction, and prevents the degeneracy problems of option domination and frequent option switching that occur in option-critic, while achieving a similar sample complexity. We also demonstrate the more interpretable and reusable nature of the learned options in comparison with option-critic through different transfer settings. Experimental results in a relatively simple four-rooms environment and the more complex ALE (Arcade Learning Environment) showcase the efficacy of our approach.

1. Introduction

Humans are effortlessly adept at many forms of abstraction. We plan and perform actions at a high-level of decision making, and not at the level of individual muscle movements. Such high-level actions typically last for an extended period of time. This is known as temporal abstraction. When observing our surroundings before making a decision, we rely and focus on only the important aspects of our sensory input, and ignore the unnecessary signals. This is called state abstraction.

Within the options framework (Sutton et al., 1999; Precup, 2000), the end-to-end learning of hierarchical behaviors has recently become possible via the option-critic architecture (Bacon et al., 2017), which enables the learning of intra-option policies, the termination functions and the policy over options, to maximize the expected return. However, if this is the sole objective for option discovery, the benefit that options have over primitive action policies is questionable. Indeed, the option-critic architecture eventually results in option degeneracy i.e. either one option dominates and is the only one that is used, or there is frequent termination of and switching between options. Introduced to combat this problem, the deliberation cost model (Harb et al., 2018) modifies the termination gradient to assign a penalty to option termination. This leads to extended options, but is susceptive to a hard-to-interpret cost parameter. Alternatively, the termination critic (Harutyunyan et al., 2019) employs a predictability objective for option termination to prevent option collapse and improve planning.

We adopt the view that options should be diverse in their behavior by explicitly learning to attend to different parts of the observation. In doing so, we solve the degeneracy problem by ensuring that options are only used when their respective attentions are activated. This lends credibility to the notion of options specializing to achieve specific behaviors. For example, in the four-rooms environment (Sutton et al., 1999), it makes little sense to use the complete observation when deciding how to move out of a particular room. Current option discovery methods in the function approximation setting do just this. Our approach also, in effect, relaxes the strong assumption – made by many option discovery methods – that all options are available everywhere, and acts as a proxy towards learning the initiation sets for options (Sutton et al., 1999), which are otherwise inconvenient to directly learn using a gradient-based learning approach.

The view of bounded rationality (Simon, 1957) can be seen as one of the motivations for temporal abstraction. The added capability of state abstraction takes this one step further, and serves as an additional rationale for our work.

2. Background

A discrete-time finite discounted MDP (Markov Decision Process) $\mathcal{M}$ (Puterman, 1995; Sutton and Barto, 1998) is characterized by the tuple $\{\mathcal{S}, \mathcal{A}, R, P, \gamma\}$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition

[1] McGill University [2] Mila [3] Deepmind. Correspondence to: Raviteja Chunduru <raviteja.chunduru@mail.mcgill.ca>.

LifelongML Workshop at the 37^{th} International Conference on Machine Learning 2020, Vienna, Austria. Copyright 2020 by the author(s).

probability function which specifies the dynamics of the environment, and $\gamma \in [0, 1)$ is the scalar discount factor. A Markovian stationary policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is a probabilistic mapping from the set of states $\mathcal{S}$ to the set of actions $\mathcal{A}$. At each timestep t, the agent observes state $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$ according to policy π, thereby receiving reward $r_{t+1} = R(s_t, a_t)$ and transitioning to state $s_{t+1} \in \mathcal{S}$ with probability $P(s_{t+1}|s_t, a_t)$. For policy π, the discounted state value function is given by: $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1}|s_0 = s]$ and the discounted action value function by: $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1}|s_0 = s, a_0 = a]$.

For a parameterized policy π_θ with $J(\theta) = V_\pi(s_0)$ as the objective function, the policy gradient theorem (Sutton et al., 2000) can be used to learn the optimal policy π_θ^* that maximizes $V_\pi(s_0)$ as:

$$\nabla_\theta J(\theta, s_0) = \sum_s d^\pi(s|s_0) \sum_a \nabla_\theta \pi(s, a) Q^\pi(s, a) \quad (1)$$

where $d^\pi(s|s_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|s_0, \pi)$ is the discounted weighting of states with s_0 as the starting state. In actor-critic methods (Konda and Tsitsiklis, 2000), the action values $Q_\phi^\pi(s, a)$, parameterized by ϕ, are typically estimated by using temporal difference (TD) learning (Sutton, 1988). For instance, the update rule for 1-step TD(0) follows as: $\phi = \phi + \alpha \delta_t \nabla_\phi Q_\phi^\pi(s_t, a_t)$ where the TD(0) error $\delta_t = r_{t+1} + \gamma Q_\phi^\pi(s_{t+1}, a_{t+1}) - Q_\phi^\pi(s_t, a_t)$ and α is the learning rate.

2.1. The Options Framework

A Markovian option $\omega \in \Omega$ (Sutton et al., 1999) is a tuple that consists of an initiation set $\mathcal{I}_\omega \subseteq \mathcal{S}$, which denotes the permissible set of states where the option can be initiated, an intra-option policy $\pi_\omega : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, which specifies a probabilistic mapping from states to actions, and a termination condition $\beta_\omega : \mathcal{S} \rightarrow [0, 1]$, which signifies the probability of option termination in a state. $\Omega(s)$ denotes the set of available options for state s and an option ω is available in state s if $s \in \mathcal{I}_\omega$. Ω is the union of all $\Omega(s), \forall s \in \mathcal{S}$.

Similar to Bacon et al. (2017), we consider call-and-return option execution. In this model, when the agent is in state s_t, it chooses an option $\omega \in \Omega(s_t)$ according to a policy over options π_Ω. The intra-option policy π_ω is then followed until the current option terminates according to β_ω after which a new option that is available at the new state is chosen by π_Ω, and the process repeats. Like many existing option discovery methods, we too make the assumption that all options are available everywhere, i.e., $\forall s \in \mathcal{S}, \forall \omega \in \Omega : s \in \mathcal{I}_\omega$. However, we show that our approach relaxes this assumption, in effect, and provides an elegant way to learn distinct initiation sets for options.

The option-critic architecture (Bacon et al., 2017) provides an end-to-end gradient-based method to learn options. For parameterized intra-option policies $\pi_{\omega,\theta}$ and option terminations $\beta_{\omega,\nu}$, the option-value function is:

$$Q_\Omega(s, \omega) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a) \quad (2)$$

where $Q_U : \mathcal{S} \times \Omega \times \mathcal{A} \rightarrow \mathbb{R}$ is the value of executing action a in the context of state-option (s, ω):

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

and $U : \Omega \times \mathcal{S} \rightarrow \mathbb{R}$ is the option-value on arrival (Sutton et al., 1999) and represents the value of executing option ω in state s':

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

where $V_\Omega(s) = \sum_\omega \pi_\Omega(\omega|s) Q_\Omega(s, \omega)$ is the option-level state value function. The intra-option policies and option terminations can be learned by using the policy gradient theorem to maximize the expected discounted return (Bacon et al., 2017). The gradient of this objective with respect to intra-option policy parameters θ when the initial condition is (s_0, ω_0) is:

$$\begin{aligned}\nabla_\theta J(\theta, s_0, \omega_0) = \sum_{s,\omega} \Big\{ &\mu_\Omega(s, \omega|s_0, \omega_0 \\ &\times \sum_a \big[\nabla_\theta \pi_{\omega,\theta}(a|s)\big] Q_U(s, \omega, a) \Big\}\end{aligned} \quad (5)$$

where $\mu_\Omega(s, \omega|s_0, \omega_0)$ is the discounted weighting of state-option pairs along trajectories that start with (s_0, ω_0): $\mu_\Omega(s, \omega|s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega|s_0, \omega_0)$. Similarly, the gradient with respect to option termination parameters ν with initial condition (s_1, ω_0) is:

$$\begin{aligned}\nabla_\nu J(\nu, s_1, \omega_0) = -\sum_{s',\omega} \Big\{ &\mu_\Omega(s', \omega|s_1, \omega_0) \\ &\times \big[\nabla_\nu \beta_{\omega,\nu}(s')\big] A_\Omega(s', \omega) \Big\}\end{aligned} \quad (6)$$

where $A_\Omega(s, \omega) = Q_\Omega(s, \omega) - V_\Omega(s)$ is the advantage of choosing option ω in state s.

2.2. Attention

The attention mechanism was first proposed in language translation tasks (Bahdanau et al., 2015) but has since been applied in vision (Sorokin et al., 2015) and reinforcement learning (Mnih et al., 2014) as well. It enables the localization of important information before making a prediction. In our approach, soft attention (smoothly varying and differentiable) is applied as a learnable mask over the state observations.

Algorithm 1 Attention Option-Critic

```
Input: $\alpha_\theta, \alpha_\nu, \alpha_\phi$ as learning rates for $\theta, \nu$ and $\phi$ respectively.
Initialize policy over options $\pi_\Omega(o)$, intra-option policies $\pi_{\omega,\theta}$, option terminations $\beta_{\omega,\nu}$, and option attentions $h_{\omega,\phi}$
$s \leftarrow s_0$
$o \leftarrow \{h_{\omega,\phi}(s) \odot s : \omega \in \Omega\}$
Choose $\omega$ according to $\epsilon$-soft $\pi_\Omega(o)$
repeat
    Choose $a$ according to $\pi_{\omega,\theta}(a|o_\omega)$
    Take action $a$ in $s$, observe $s', r$
    $o' \leftarrow \{h_{\omega,\phi}(s') \odot s' : \omega \in \Omega\}$

    1. Options evaluation:
    $\delta \leftarrow r - Q_U(o_\omega, \omega, a)$
    if $s'$ is non-terminal then
        $\delta \leftarrow \delta + \gamma(1 - \beta_{\omega,\nu}(o'_\omega))Q_\Omega(o'_\omega, \omega)$
                $+\gamma\beta_{\omega,\nu}(o'_\omega) \max_{\bar{\omega}} Q_\Omega(o'_\omega, \bar{\omega})$
    end if
    $Q_U(o_\omega, \omega, a) \leftarrow Q_U(o_\omega, \omega, a) + \alpha\delta$

    2. Options improvement:
    $\theta \leftarrow \theta + \alpha_\theta \big[\nabla_\theta log\pi_{\omega,\theta}(a|o_\omega)\big] Q_U(o_\omega, \omega, a)$
    $\nu \leftarrow \nu - \alpha_\nu \big[\nabla_\nu \beta_{\omega,\nu}(o'_\omega)\big] \big[Q_\Omega(o'_\omega, \omega) - V_\Omega(o'_\omega)\big]$
    $\phi \leftarrow \phi + \alpha_\phi \nabla_\phi \big[Q_\Omega(o_\omega, \omega) + L\big]$

    if $\beta_{\omega,\nu}$ terminates in $s'$ then
        choose new $\omega$ according to $\epsilon$-soft $\pi_\Omega(o')$
    end if
    $s \leftarrow s'$
    $o \leftarrow \{h_{\omega,\phi}(s) \odot s : \omega \in \Omega\}$
until $s'$ is terminal
```

3. Attention Option-Critic

We introduce the Attention Option-Critic (AOC) architecture to enable options to learn to be attentive to specific features in the observation space in order to diversify their behavior and prevent degeneracy. An attention mechanism $h_{\omega,\phi}$, parameterized by ϕ, is applied to the observation s for each option ω as: $o_\omega = h_{\omega,\phi}(s) \odot s$ where $\odot$ denotes element-wise multiplication. $h_{\omega,\phi}$ consists of values in $[0, 1]$ and is the same size as the original observation s. The result o_ω is used to determine the value of the option, the intra-option policy and the termination condition. This is done for each option separately, and ensures only the required features from the observation determine the option's behavior. We refer to o as the list of all attention-modified observations for each option $o = \{o_\omega : \omega \in \Omega\}$. The learning of the option terminations and intra-option policies is similar to the option-critic architecture. The complete algorithm is shown in Algorithm 1.

The attention for each option is learned to maximize the expected cumulative return of the agent while simultaneously maximizing a distance measure between the attentions of the options, so that they are attentive to different features. Additionally, some regularization is added to facilitate the emergence of desired option characteristics. The attention parameters ϕ are updated with gradient ascent as $\phi = \phi + \alpha_\phi \nabla_\phi \big[Q_\Omega(o_\omega, \omega) + L\big]$, where L denotes the sum of the distance measure and the regularization, weighted by their respective importance. More details are specified in the next section.

The attention mechanism brings an aspect of explainability to the agent, and allows one to easily understand each option's focus and behavior. Also, it provides a highly interpretable knob to tune options since the characteristics of the resulting options can be controlled by affecting how the attentions of the options are learned during training. For example, constraining attentions to be distinct enables the diversity of options to be set explicitly as a learning objective. Alternatively, penalizing differences in option attention values for states along a trajectory results in temporally extended options, which achieves an effect similar to the deliberation cost model (Harb et al., 2018), but in a more explainable way.

The resulting attention for each option also serves as an indication of the regions of state space where that option is active and can be initialized. Thus, along with the intra-option policies and option terminations, AOC essentially learns the initiation sets of the options in that an option is typically only initiated in a particular state when the corresponding attention of that option in that state is high. It is this result which prevents the options from degenerating. Since every option cannot be executed or initiated everywhere, it prevents frequent option termination and switching, and also prevents option domination (Figure 6) by ensuring that a single option cannot always be followed.

3.1. Optimality of Learned Solution

Since each option receives different information, it is not immediately obvious whether the solution that is learned by AOC is flat, hierarchically or recursively optimal (Dietterich, 2000). However, each option learns to act optimally based on the information that it sees, and apart from some constraints enforced via option attentions, individual option optimality is driven through the policy over options to maximize the total expected return. Since there is no pseudo reward or subtask assigned to each option, their attentions and areas of usage are learned to maximize this objective and we reason that in the absence of attention constraints, a flat optimal policy will be learned in the limit. In the presence of constraints, the optimality of the learned options will depend on the interactions between the multiple objectives. Even in such cases, AOC is capable of achieving a flat

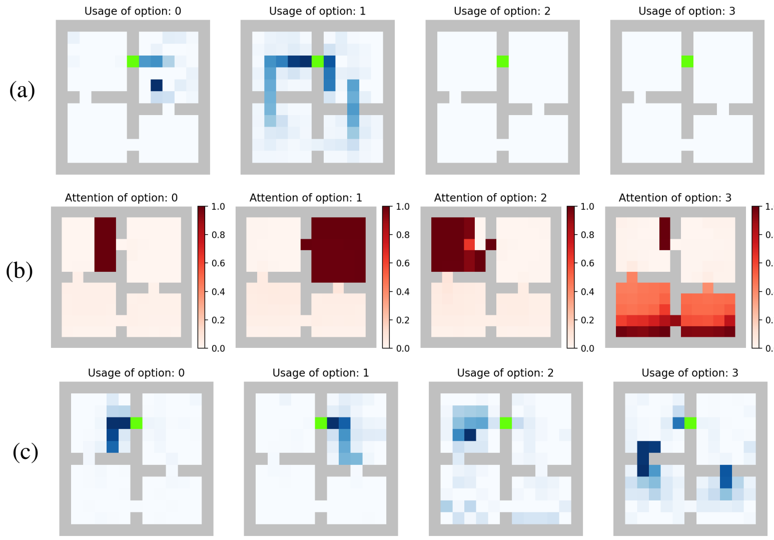


Figure 1. An example of learned options in the four-rooms domain with goal at north hallway (shown in green). **(a)** example of degenerate options learned by OC. Darker color indicates more frequent option execution in that particular state. Option 1 dominates and is used 88.12% while Option 0 is used 11.88%. Options 2 and 3 are unused. **(b)** the resulting attention learned for each option with AOC. **(c)** the options learned using AOC. The options are diverse and respect their attentions. The option usage is relatively balanced at 19.5%, 34.3%, 8.1% and 38.1% respectively.

optimal solution as shown empirically in the next section.

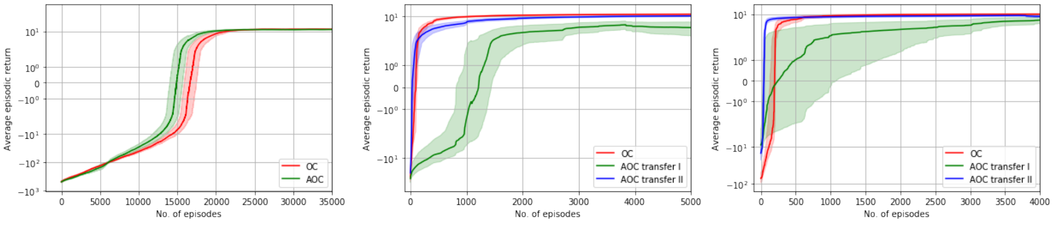
4. Experimental Results

In this section, we show the benefit of attention and empirically demonstrate that it prevents option degeneracy, provides interpretability, and promotes reusability of options in transfer settings.

4.1. Learning in the four-rooms environment

We start off by showing the benefit of attention in the four-rooms navigation task (Sutton et al., 1999) where the agent must reach a specified goal. The observation space consists of one-hot-encoded vectors for every permissible state in the grid. The available actions are up, down, left and right. The chosen action is executed with probability 0.98 and a random action is executed with 0.02 probability. The reward is +20 upon reaching the goal, and -1 otherwise. The agent starts in a uniformly random state and the goal is set randomly for each run.

We use 4 options for learning, with a discount factor of 0.99. The attention $h_{\omega,\phi}$ for each option ω is initialized randomly as a vector of the same length as the input observation s. Thus, in this situation, the option attentions are independent of the state observation. We employ a 2-layer shared-parameter neural network to approximate the intra-option policy, the option termination functions, and the option values. In our implementation of AOC (for all experiments), the network learns the option values Q_Ω to which the ϵ-greedy strategy is applied to determine the pol-



(a) Training curves

(b) Goal transfer

(c) Blocked hallway

Figure 2. Learning and transfer (averaged over 15 runs) in the four-rooms domain with 4 options.

icy over options π_Ω. Intra-option exploration is enforced with entropy regularization. The architecture is shown in section C.

The option attentions, option values, intra-option policies and option terminations are learned in an end-to-end manner to maximize the total expected discounted return. The cosine similarity between the option attentions is added to the loss to ensure that the learned attentions are diverse. Furthermore, a regularization loss – the sum of absolute differences between attentions (for each option) of adjacent states in a trajectory – penalizes irregularities in the option attentions of states in close temporal proximity along trajectories. This results in smooth attentions and leads to temporally extended options by minimizing switching, and achieves a similar effect to deliberation cost (Harb et al., 2018). Thus, for the four-rooms domain, the L term in Algorithm 1 is enforced by adding $w_1 L_1 + w_2 L_2$ to the overall network loss function, where L_1 is the total sum of cosine similarities between the attentions of every pair of options, L_2 is the temporal regularization loss for option attentions, and w_1 and w_2 are the respective weights for these additional objectives. We found that a value of 2.0 for both w_1 and w_2 resulted in the most diverse options, judged quantitatively (see section A.5) and qualitatively (Figure 1). Further details regarding hyperparameters and reproducibility are provided in section C.

The resulting option usage and their attentions are shown in Figure 1. The learned options are distinct and specialized in their behavior, and they perform state abstraction by focussing on a subset of the observation to perform their specific tasks. The option usage respects the corresponding area of attention, which indicates that the options are typically limited to this area and that their behavior can be reasonably interpreted from their attentions. AOC also learns stable options and the behavior and usage of options does not vary significantly during the course of training. This is in contrast to option-critic (OC), which tends to learn degenerate options that are volatile and continuously change behavior. A qualitative comparison that demonstrates AOC option stability is shown in section A.1.

Although AOC additionally needs to learn option attentions, it learns faster than OC, as shown in Figure 2a. One possible reason could be that in AOC, options specialize to different regions and enable quicker learning because of less overlap between their usage. Each learning curve is averaged over 15 independent runs, each with a random goal location. A comparison between option domination in AOC and OC (see A.2) during training indicates that the latter prevents it.

4.2. Transfer in the four-rooms environment

We perform two experiments to assess the transfer capability of AOC in the four-rooms domain, both after 30,000 episodes of training. The first is goal transfer, where the location of the goal is changed to a new random location and the second is blocked hallway, where the goal is the same but a random hallway is blocked. AOC transfer I and transfer II respectively represent the scenarios where the weights w_1 and w_2 are kept unchanged or are set to 0 to give priority to option learning over attention regularization, before learning in the new task. From Figures 2b and 2c, it can be seen that in spite of the option volatility that aids OC transfer, AOC transfer II performs similarly in the goal transfer setting and both variants of AOC show superior initial performance in the blocked hallway setting with transfer II being faster overall. The speed of AOC transfer II is even more apparent when the agent needs to go all the way around the blocked hallway (see section A.3). The slower transfer of AOC transfer I can be explained by the over-preference towards optimizing attention characteristics which AOC transfer II mitigates. Each curve is averaged over 15 independent runs with different blocked hallways and different goals before and after transfer.

From another perspective, upon transfer, option-critic completely relearns the options. Figure 3 shows a specific instance of transfer. Comparing Figure 1a with Figures 3a and 3b shows that there is little similarity between the option behavior before and after transfer with OC. We argue that for options to be beneficial for generalization and lifelong learning, they should exhibit similar behavior upon transfer, and only change as required, so that previously learned behaviors can be leveraged, and so that options can be efficiently composed into even higher levels of behavior. AOC exhibits this quality. A comparison of Figures 1b and 1c with Figures 3c to 3f shows that option attentions remain fixed indicating that each option remains in its assigned space, and that the option behavior remains relatively consistent upon transfer.

4.3. Arcade Learning Environment

We now demonstrate the performance of AOC in the Arcade Learning Environment (Bellemare et al., 2013). We use 2 options with a discount factor of 0.99. The input observation s is a stack of 4 frames. The option attentions $h_{\omega,\phi}$ are state dependent and are learned with a convolutional neural network. Each option's attention has the same dimensions as a single frame, and is shared across all frames in the input stack. We refer to this as the shared-attention model. The option policies, values and terminations are learned with a shared-parameter deep neural network, similar to option-critic. The architecture is shown in section C.

Apart from maximizing the total expected return, the attentions are constrained to exhibit some desired characteristics. Attention diversity is enforced by maximizing the L1 norm between the object attentions of the options and attention sparsity is incentivized by penalizing non-zero attentions for

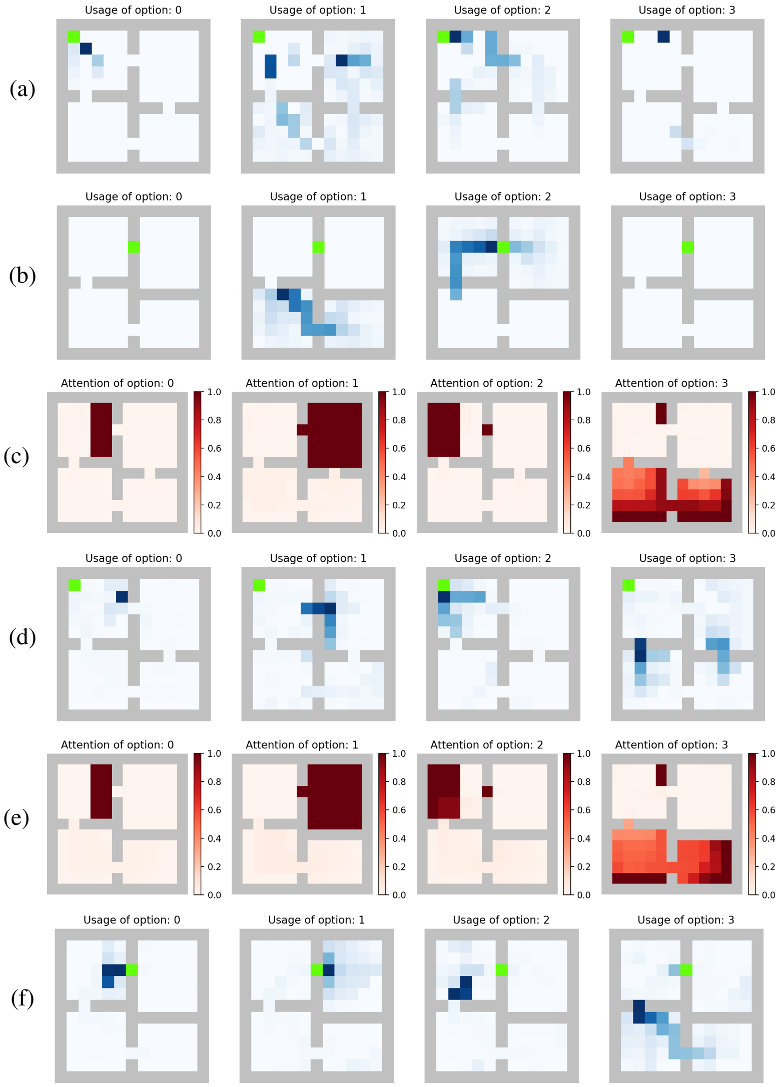
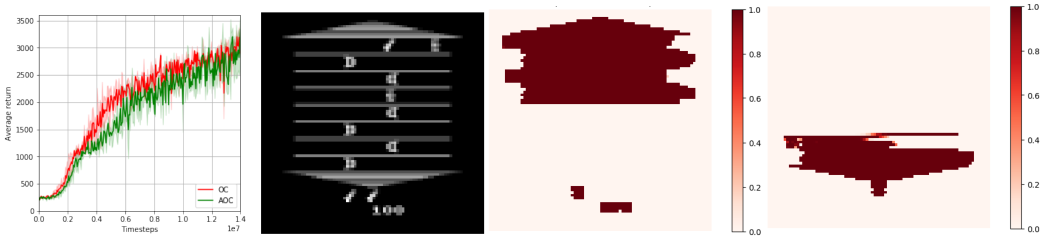


Figure 3. **(a) and (b):** resulting options learned by option-critic upon goal transfer and transfer with blocked hallway respectively. **(c) and (d):** the resulting option attentions and usage upon goal transfer. **(e) and (f):** the resulting option attentions and usage upon transfer with blocked hallway. For the goal transfer plots above, the goal is shifted from north hallway to the top left state in the north west room. For blocked hallway transfer, the goal is kept fixed as the north hallway, but the east hallway is blocked. The transfer results shown here are with OC and AOC transfer I. The goal states are shown in green.

the background. Lastly, attention regularity is promoted between object pixels by penalizing frequent changes in their attention values. The objects and background are identified by finding the connected components in the observation (Figure 4b). Thus, for the atari domain, the L term in Algorithm 1 is enforced by adding $w_1L_1+w_2L_2+w_3L_3+w_4L_4$ to the network loss function, where L_1, L_2, L_3 are the losses for attention diversity, sparsity and regularity respectively. The additional regularizer L_4 is added to prevent an option's attention from collapsing to zeros. w_1, w_2, w_3 and w_4 represent their respective weights. More details regarding hyperparameters are provided in section C.

For training in the Asterix environment, we found that the values 5000, 0.01, 100, and 1 for the weights w_1, w_2, w_3 and w_4 respectively, resulted in diverse attentions and good performance. Figure 4 shows the performance and learned option attentions. Figure 4a shows that AOC achieves a similar sample complexity compared to OC, despite also having to learn the state-dependent attention mechanism. We reason that learning the attentions enable options to specialize early on in the training process, and hence speed up training, despite having more parameters to learn. Each



(a) Training curves (b) Asterix frame (c) Attention of option: 0 (d) Attention of option: 1

Figure 4. Training curves, an example game frame and corresponding learned option attentions for Asterix. The options respectively focus on the top and bottom halves of the frame.

curve is averaged over 3 different random seeds. Figures 4c and 4d show the resulting option attentions and indicate that option 0 and option 1 have respectively specialized to behaviors pertaining to the main sprite's position in the upper or lower half of the frame. Thus, AOC allows for learning diverse and interpretable options in complex environments too. Additional atari results are shown in section B.

5. Related work

There has been extensive research on the benefit of temporal abstraction for reinforcement learning (Parr and Russell, 1998; Dayan and Hinton, 1993; Dietterich, 2000; McGovern and Barto, 2001; Stolle and Precup, 2002; Mann and Mannor, 2014). Specific to the options framework (Sutton et al., 1999; Bacon et al., 2017), there have been many recent approaches to incentivize learned options to be diverse (Eysenbach et al., 2018), temporally extended (Harb et al., 2018), more abstract (Riemer et al., 2018), and easy to plan (Harutyunyan et al., 2019) and explore (Jinnai et al., 2019) with.

The interest option-critic method (Khetarpal et al., 2020) provides a gradient-based approach towards learning where to initialize options by modeling the initiation sets as differentiable interest functions. However, the initialization of the interest functions is biased towards all options being available everywhere. In contrast, our AOC approach is completely end-to-end and does not require any special initializations, and in effect, is able to learn distinct areas where options can be initialized and remain active.

Deep skill chaining (Bagaria and Konidaris, 2020) is another approach that relaxes the assumption of universal option use. This method learns a chain of options by backtracking from the goal and ensuring that the learned initiation set of one option overlaps with the termination of the preceding option. Although each option performs state abstraction, the resulting options are highly dependent on the given task and must be relearned upon transfer. Furthermore, results were mostly confined to navigation-based tasks.

The MAXQ (Dietterich, 2000) approach towards hierarchical reinforcement learning decomposes the value function of the target MDP into value functions of smaller MDPs. Although this decomposition creates an opportunity to perform state abstraction, the overall approach is based on the heavy assumption that the subgoals, and the subtasks necessary to achieve them, are specified beforehand.

6. Conclusion

To the best of our knowledge, our method is the first to combine temporal and state abstraction in a flexible end-to-end gradient based approach and results in learned options that are diverse, stable, interpretable, reusable and transferable. We demonstrate that the addition of an attention mechanism prevents option degeneracy, a major long standing problem in option discovery, and also relaxes the assumption of universal option availability. It also provides a highly intuitive method to control the characteristics of the learned options.

From the lifelong learning perspective, an interesting future direction is to meta-learn the attentions and options across a range of tasks from the same environment. This could lead to faster transfer, while keeping the existing benefits of our approach. From the view of model-based reinforcement learning, predictive approaches with option attentions could allow for efficient long-horizon planning by predicting option activation through predicted attentions. Lastly, the approach we have presented is versatile and can be applied to many existing option discovery methods. We leave such avenues of possible combination as future work.

References

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 1726–1734, 2017.

Akhil Bagaria and George Konidaris. Option discovery using deep skill chaining. In *International Conference* *on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1gqipNYwH.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems 5*, pages 271–278. 1993.

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

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *CoRR*, 2018. URL http://arxiv.org/abs/1802.06070.

Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Rémi Munos, and Doina Precup. The termination critic. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Yuu Jinnai, Jee Won Park, David Abel, and George Konidaris. Discovering options for exploration by minimizing cover time. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Khimya Khetarpal, Martin Klissarov, Maxime Chevalier-Boisvert, Pierre-Luc Bacon, and Doina Precup. Options of interest: Temporal abstraction with interest functions. *CoRR*, 2020. URL https://arxiv.org/abs/2001.00271.

V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. *NIPS 12*, pages 1008–1014, 2000.

Timothy Mann and Shie Mannor. Scaling up approximate value iteration with options: Better policies with fewer iterations. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 127–135. PMLR, 2014.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, page 361–368, 2001.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *CoRR*, 2014. URL http://arxiv.org/abs/1406.6247.

Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, page 1043–1049, Cambridge, MA, USA, 1998. MIT Press. ISBN 0262100762.

D. Precup. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts, Amherst, 2000.

M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. *Journal of the Operational Research Society*, 1995.

Matthew Riemer, Miao Liu, and Gerald Tesauro. Learning abstract options. *CoRR*, 2018. URL http://arxiv.org/abs/1810.11583.

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

Ivan Sorokin, Alexey Seleznev, Mikhail Pavlov, Aleksandr Fedorov, and Anastasiia Ignateva. Deep attention recurrent q-network. *CoRR*, 2015. URL http://arxiv.org/abs/1512.01693.

Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *Proceedings of the 5th International Symposium on Abstraction, Reformulation and Approximation*, page 212–223, 2002.

R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, pages 9–44, 1988.

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

R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, pages 181–211, 1999.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

Appendix

A. Other four-rooms experiments

A.1. Comparison of option stability between AOC and OC

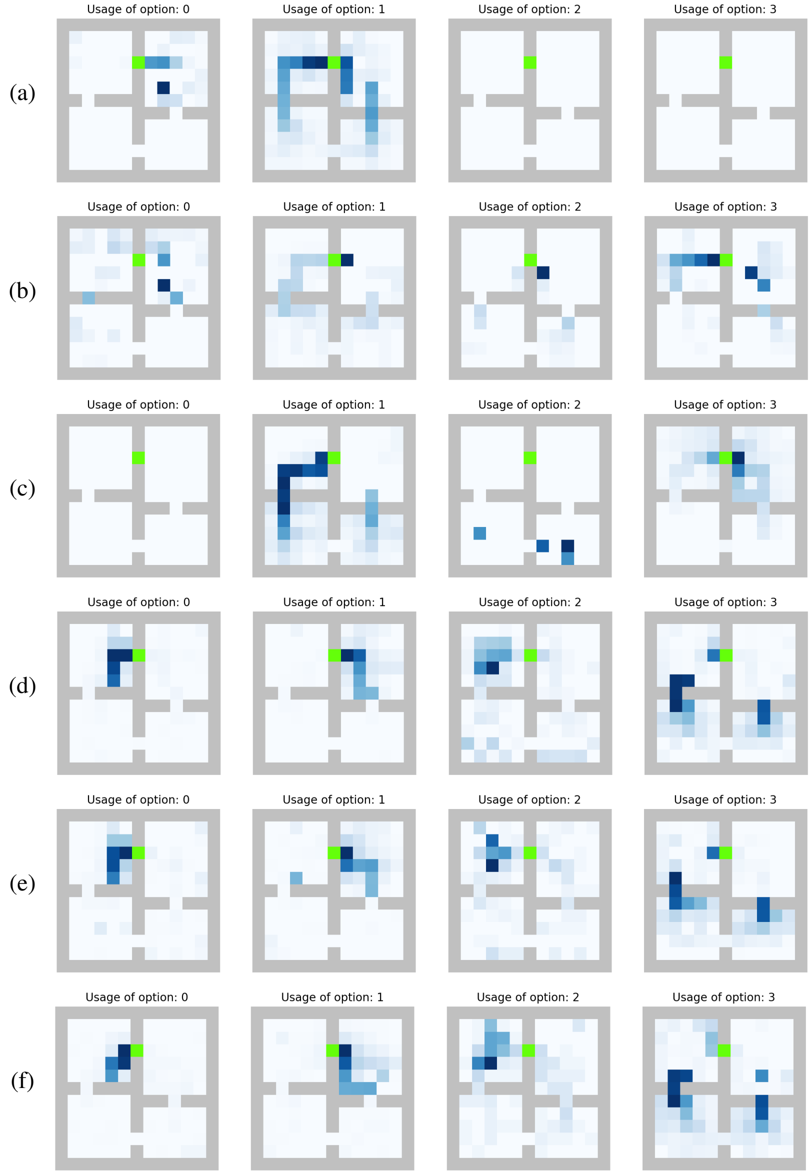


Figure 5. **(a) to (c):** even after convergence, options learned with OC are volatile and continue to change frequently. **(d) to (f):** AOC learns more stable options which continue to exhibit similar behavior. In the snapshots of the options above, for both OC and AOC, 100,000 frames of training has been performed between successive rows. The goal is the north hallway, shown in green.

A.2. Comparison of dominant option usage in AOC and OC

A comparison of the usage of the dominant option in AOC and OC is shown in Figure 6. At each training checkpoint, the dominant option usage is averaged over 50 test episodes for each of the 15 independent training runs. The shaded region represents 1 standard deviation.

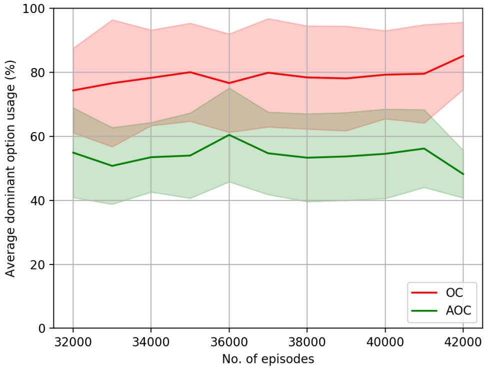
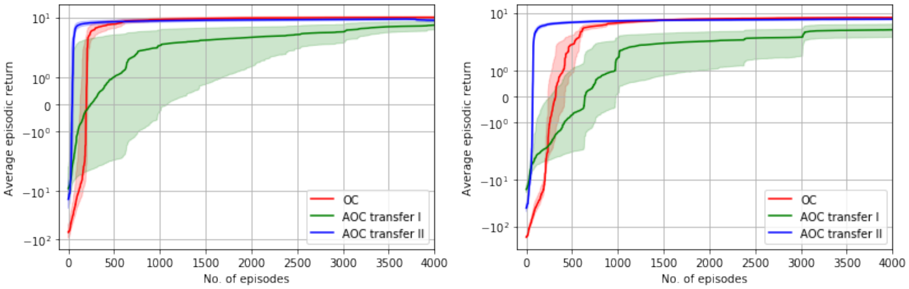


Figure 6. Comparison of average usage of the dominant option in the four-rooms domain.

A.3. Blocked hallway transfer: hard transfers

There are cases where blocking a hallway may mean that the agent has to go all the way around this blockage to reach the goal. For example, if the goal is in the top right room, the east hallway is blocked and the agent starts in the lower right room, then the agent must navigate all the way around the environment, through 3 hallways, to reach the goal. This subset of blocked hallway runs are referred to as hard transfers. A comparison between AOC and OC in handling such hard transfers is shown in Figure 7b and indicates the more apparent benefit of AOC with hard transfers. It was also observed that on some occasions with hard transfer, OC failed to learn altogether unlike AOC which always learned an optimal policy upon transfer. The runs shown in Figure 7b are a subset (approximately half) of the runs shown in Figure 7a. Note that Figure 7a is the same as Figure 2c.



(a) Blocked hallway: all

(b) Blocked hallway: only hard transfer

Figure 7. (a) Transfer comparison for all transfers in the blocked hallway setting. (b) Transfer comparison for hard transfers in the blocked hallway setting.

A.4. Hardcoded option attentions

In the case of hardcoded attention where each option's attention is manually limited to one specific and distinct room (i.e. 1 for all states inside the room and 0 elsewhere), slower learning is observed. This is likely because hardcoding attentions de facto removes option choice from the agent, and requires all options to be optimal to get good performance. When we tried hardcoded attention with 8 options (2 per room), we got better performance, but still significantly slower than AOC and OC. Figure 8 shows the comparison of the learning curves. Each curve is averaged over 15 runs and the shaded region indicates 0.25 standard deviation.

A.5. Quantitative measures for four-rooms options and attentions

All of the following quantitative measures are averaged over 15 independent runs with different goal locations.

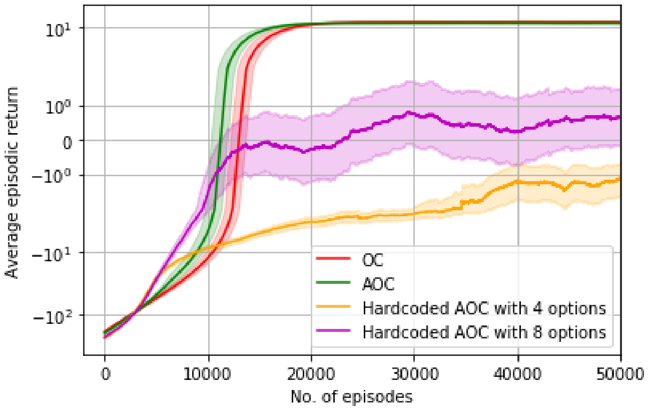


Figure 8. Comparison of OC, AOC, and AOC with hardcoded attentions

A.5.1. Quantitative measure for attention diversity

After training, the argmax operation applied on the option dimension across the attention maps gives the option with most attention for each state in the environment. Let the option which has the highest attention in most states be termed the most attentive option and let the ratio of its number of highest attention states to total states be called most attentive option coverage. Similarly, let the option which has the highest attention in least states be termed the least attentive option and let the ratio of its number of highest attention states to total states be called least attentive option coverage. The closer both the least and most attentive option coverages are to 25% (in the case of 4 options), the more diverse the attentions. When the weights w_1 and w_2 are 2.0 (which we found to be the most optimal), least attentive option coverage = 8.07% and most attentive option coverage = 48.58%. These values indicate that each option has a non-zero area where it is most attentive.

A.5.2. Quantitative measure for attention overlap

After training, let the matrix of maximum attention values for each state (across options) be termed as $max\_attention\_matrix$. Let the matrix of next maximum (2nd highest) attention values for each state (across options) be termed as $second\_max\_attention\_matrix$. Let the difference between these two matrices be called $diff$. Then, a measure of the percentage of state space area where only one option attends to can be calculated as $sum((diff > 0.3)\&\&(second\_max\_attention\_matrix < 0.05)) * 100/total\_states$. Here, $\&\&$ denotes the element-wise logical and operation. This measure calculates the percentage of area where there is no competition among option attentions and there is clearly only one option's attention for each state in this area. The higher this measure is, the better. When the weights w_1 and w_2 are 2.0, this measure was 53.33%. For the remaining 46.66% of the area, it was usually observed to be the case that 2 options' attentions competed for this area (note that this also includes cases where the difference in option attentions is very high i.e. 0.5 or greater but where the second highest option attention was non negligible like 0.15).

A.5.3. Quantitative measures of variance in option usage

The mean option usage for both AOC and OC is near 0.25 for each option (option domination balances out across runs in OC). The standard deviation of option usages for AOC and OC are respectively [0.19, 0.19, 0.22, 0.18] and [0.27, 0.33, 0.37, 0.35] i.e. OC has 3 to 4 times more variance.

A.5.4. Quantitative measure of consistency between option attentions and usage

The probability that an option is executed when its corresponding attention in a state is < 0.05 is only 0.089. This indicates that option usage is largely consistent with the corresponding option attentions.

It should be noted that in the cases where multiple options have significant non-zero attentions in a state, it can be expected that any of these options may be executed. For example, Figure 9 shows the case where multiple options attend to states in the bottom right room. In this case, there is some overlap between the usage of the options that have high attention in these states. Usage in other rooms is still quite distinct.

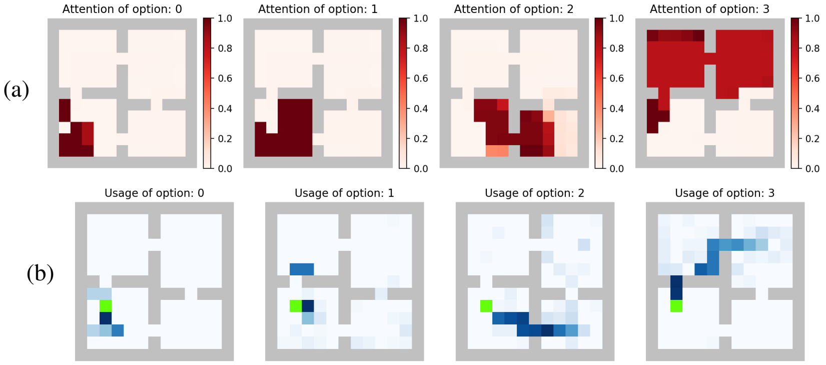


Figure 9. When multiple options have significant overlapping attentions in a state, any of these options may be executed. The goal is shown in green.

B. Other atari experiments

B.1. Atari shared-attention results

As described previously, in the shared-attention model, each option's attention is shared across all the frames of the input stack. The advantage of this approach is that the obtained attentions are much more distinct and the options are more specialized. The disadvantage is that the learning performance and the option diversity is sensitive to the chosen hyperparameters. Figure 10 shows the training curves and the option attentions when trained with hyperparameter values 5000, 0.01, 100, and 1 for the weights w_1, w_2, w_3 and w_4 respectively (these weights were obtained after tuning on the Asterix environment with the frame-dependent attention model). From the figure, it can be observed that the AOC shared-attention model achieves similar performance compared to OC and also results in diverse options with distinct areas of focus.

B.2. Atari frame-dependent attention results

In the frame-dependent attention model, each option's attention is learned individually for each frame on the input stack. Frame stacking implicitly enforces temporal regularization between attentions of successive frames, so we do not specially account for this. The advantage of this approach is the lower sensitivity towards the attention hyperparameters. The disadvantage is the increased overlap between option attentions resulting in decreased option diversity. Figure 11 shows the training curves and the option attentions when trained with hyperparameter values 5000, 0.01, 100, and 1 for the weights w_1, w_2, w_3 and w_4 respectively (these weights were obtained after tuning on the Asterix environment with the frame-dependent attention model). From the figure, it can be observed that the AOC frame-dependent attention model achieves similar performance compared to OC and also results in diverse options with distinct areas of focus. Comparing the learning curves of the shared-attention model and the frame-dependent attention model, it can be seen that the latter has slower initial performance, and this is expected since it must learn more parameters (since option attentions are learned individually for each input frame in the stack).

C. Reproducibility and training details

The models are implemented in PyTorch and experiments were run on an NVIDIA V100 SXM2 with 16GB RAM.

C.1. Four-rooms environment

For all experiments in the four-rooms domain, we use the following option learning model for both AOC and the OC baseline: a 2-layer neural (layerwise with 60 and 200 neurons followed by ReLU activation) with fully-connected branches for option values, intra-option policies (with softmax function) and the option terminations (with sigmoid function). The parameters used for both AOC and baseline OC (after a hyperparameter search) are shown in Table 1.

We performed a grid search across multiple values for w_1 and w_2, the weights for cosine similarity between the attentions

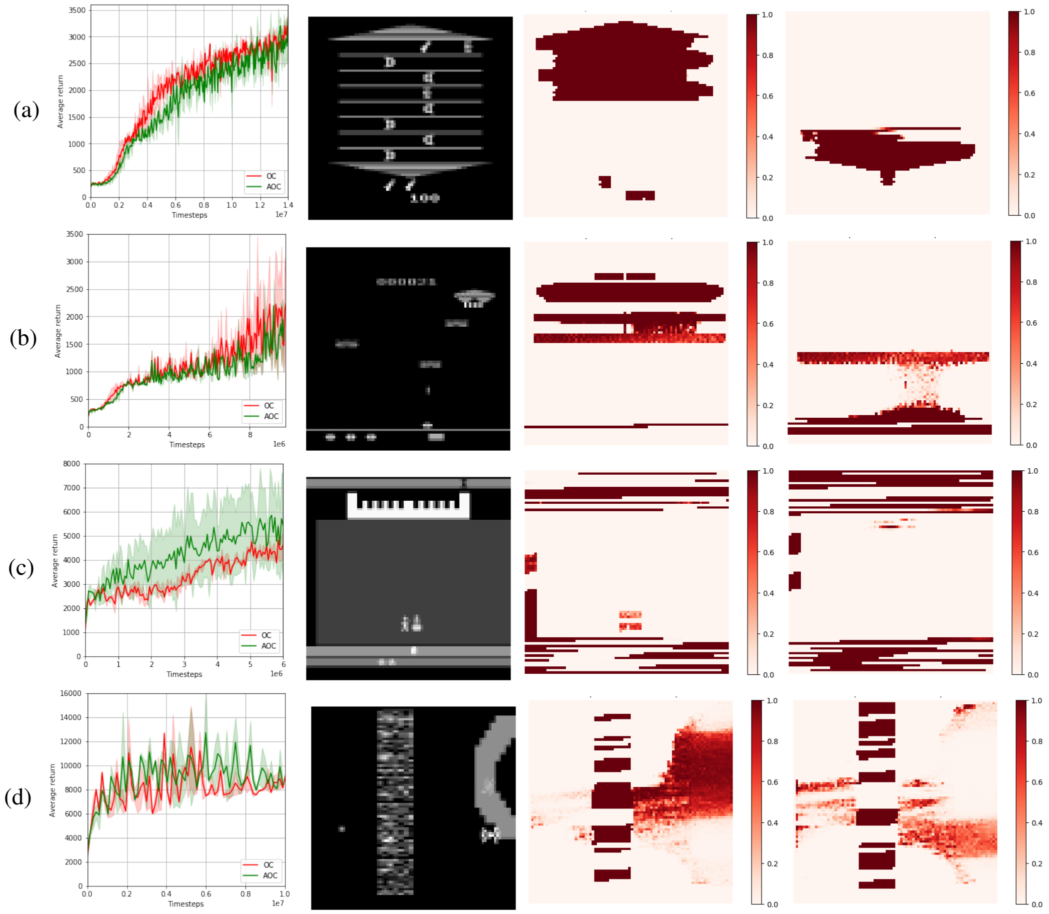


Figure 10. AOC results for the shared-attention model. Column-wise: learning curves, game frame, option 0 attention, option 1 attention. (a) Asterix (b) Assault (c) Krull (d) Yars' Revenge

Parameter	Value
Number of workers	5
Gamma (γ)	0.99
Number of options	4
Optimizer	RMSProp
Learning rate	10^{-3}
Option exploration	Linear(10^0, 10^{-1}, 10^5)
Entropy	Linear(10^2, 10^{-1}, 10^5)
Rollout length	5

Table 1. Hyperparameters for four-rooms

and the temporal regularization loss respectively. The search space for both weights was the range [0, 5.0] in increments of 0.5. The best values (judged according to qualitative attention diversity and quantitative measures explained above) were found to be 2.0 for both w_1 and w_2. The shaded regions in Figure 2a represent 0.5 standard deviation, and 0.25 standard deviation in Figures 2b and 2c.

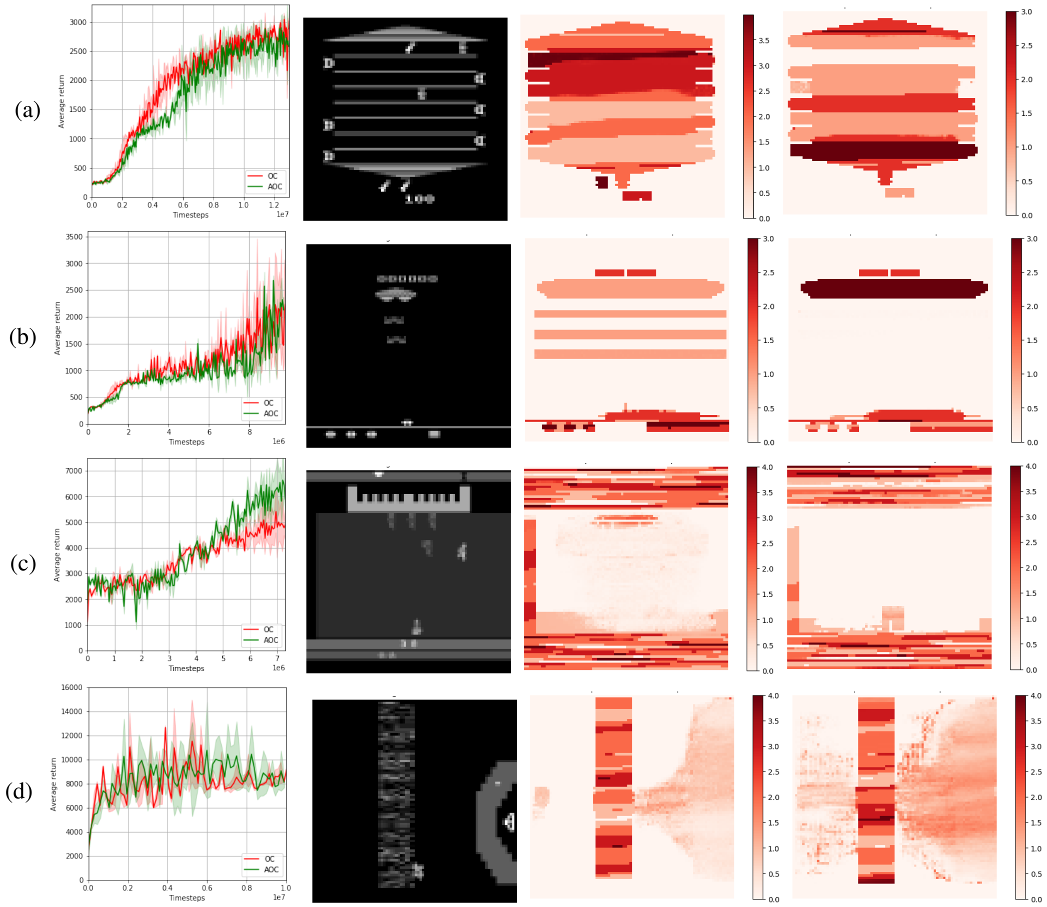
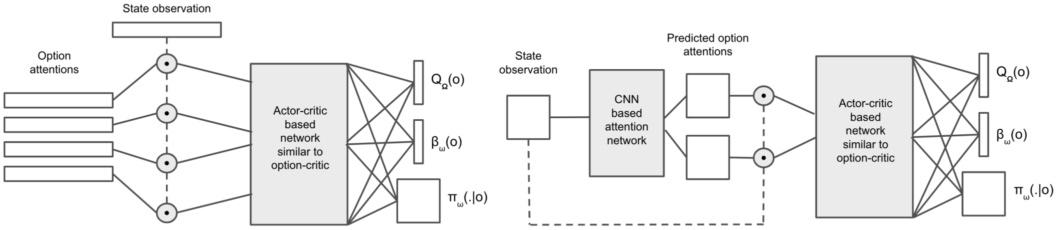


Figure 11. AOC results for the frame-dependent-attention model. Note that the attention maps shown here are the sum of frame-wise attention maps for each option. Framewise attentions are much more distinct and are similar to the attention maps from the shared-attention model. Column-wise: learning curves, game frame, option 0 attention, option 1 attention. (a) Asterix (b) Assault (c) Krull (d) Yars' Revenge


State observation
Option attentions
Actor-critic based network similar to option-critic
$Q_\Omega(o)$
$\beta_\omega(o)$
$\pi_\omega(.|o)$
State observation
Predicted option attentions
CNN based attention network
Actor-critic based network similar to option-critic
$Q_\Omega(o)$
$\beta_\omega(o)$
$\pi_\omega(.|o)$

(a) Architecture for four-rooms

(b) Architecture for atari

Figure 12. The shared network models for option learning with AOC. ⊙ denotes element-wise multiplication. **(a)** In the four-rooms environment, the attentions are independent of the state observation. **(b)** In atari environments, the attentions are observation dependent.

C.2. Arcade Learning Environment

For experiments in the Arcade Learning Environment, the structure of the option learning model for both AOC and the OC baseline is shown in Table 2.

Each convolution layer is followed by ReLU activation. The FC1 layer is followed by fully-connected branches for option values, intra-option policies (with softmax function) and the option terminations (with sigmoid function). For AOC, the

LAYER	IN-CHANNELS	OUT-CHANNELS	KERNEL-SIZE	STRIDE
CONV1	-	32	8	4
CONV2	32	64	4	2
CONV3	64	64	3	1
FC1	$7 \times 7 \times 64$	512	-	-

Table 2. Option learning model for ALE environment

structure of the attention learning model is the same as in Table 2, but another layer FC2 is connected to FC1. In terms of model architecture, the only difference between the shared-attention model and the frame-dependent attention model is the number of neurons in FC2. For the former, it is equal to the number of pixels in a single frame of the input stack and for the latter it is equal to the total number of pixels in the input stack. The parameters used for both models of AOC and baseline OC (after a hyperparameter search) are shown in Table 3. The input observation is a grayscale $84 \times 84 \times 4$ tensor.

PARAMETER	VALUE
NUMBER OF WORKERS	16
GAMMA (γ)	0.99
NUMBER OF OPTIONS	2
OPTIMIZER	RMSPROP
LEARNING RATE	10^{-4}
OPTION EXPLORATION	10^{-1}
ENTROPY	10^{-2}
ROLLOUT LENGTH	5
FRAMESTACK	4

Table 3. Hyperparameters for ALE

We performed a grid search across multiple values for w_1 and w_2 (weights for attention diversity), w_3 (weight for attention sparsity), and w_4 (weight for attention regularity). The search space for all weights was the range $[10^{-1}, 10^5]$ in semi-logarithmic increments. The best weight values were found to be 5000, 1.0, 0.01 and 100 respectively, tuned on the shared-attention model for the Asterix environment.

Each atari learning curve is an average over 3 random seeds and the shaded region represents 1 standard deviation.