Attention Option-Critic

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

Abstract

Temporal abstraction refers to an agent's ability 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. 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 high-level actions that 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 option-critic architecture (Bacon et al., 2017) enables end-to-end 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 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. The deliberation cost model (Harb et al., 2018) combats this by assigning a penalty to option termination. This approach leads to extended options, but is susceptive to a hard-to-interpret cost parameter.

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 and popular assumption that all options are available everywhere, and acts as a proxy towards learning the initiation sets (Sutton et al., 1999).

2. Background

A discrete-time finite discounted MDP (Markov Decision Process) $\mathcal{M}$ (Puterman, 1995; Sutton and Barto, 1998) is characterized by the tuple $\{\mathcal{S}, \mathcal{A}, R, P, \gamma\}$. 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: $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s]$ and the discounted action value function is: $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a]$.

2.1. The Options Framework

A Markovian option $\omega \in \Omega$ (Sutton et al., 1999) consists of an initiation set $\mathcal{I}_\omega \subseteq \mathcal{S}$, an intra-option policy $\pi_\omega : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, and a termination condition $\beta_\omega : \mathcal{S} \rightarrow [0, 1]$. $\Omega(s)$ 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 $\Omega(s), \forall s \in \mathcal{S}$. In call-and-return option execution (Bacon et al., 2017), 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 termination

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

Inductive Biases, Invariances and Generalization in RL (BIG) Workshop at International Conference on Machine Learning 2020, Vienna, Austria, 2020.

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 assume universal option availability, i.e., $\forall s \in \mathcal{S}, \forall \omega \in \Omega : s \in \mathcal{I}_\omega$, but later show how our approach relaxes this assumption.

For parameterized intra-option policies $\pi_{\omega,\theta}$ and option terminations $\beta_{\omega,\nu}$, the option-value function is:

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

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

$$\begin{aligned} Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} \big[P(s'|s,a) \\ \times((1-\beta_{\omega,\nu}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\nu}(s')V_\Omega(s'))\big] \end{aligned} \tag{2}$$

Here, $V_\Omega(s) = \sum_\omega \pi_\Omega(\omega|s) Q_\Omega(s,\omega)$ is the option-level state value function. $\pi_{\omega,\theta}$ and $\beta_{\omega,\nu}$ are learned using the intra-option policy gradient theorem and termination gradient theorem respectively (Bacon et al., 2017).

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 is applied as a learnable mask over the state observations.

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 option's value, policy and termination. 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 attend to different features. 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.

The attention mechanism brings an aspect of explainability to the agent. It also provides a highly interpretable knob to tune options since the characteristics of the options can be controlled through the attentions. 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.

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 and executed in a particular state when the corresponding attention of that option in that state is high. This prevents frequent option termination and switching, and also prevents option domination by ensuring that a single option cannot always be followed.

4. Experimental Results

4.1. Learning in the four-rooms environment

In the four-rooms navigation task (Sutton et al., 1999), the agent must reach a specified goal. The observation space consists of one-hot-encoded vectors for every state in the grid. The available actions are up, down, left and right. Environment details are provided in section C.2. The reward is +20 upon reaching the goal, and -1 otherwise. We use 4 options for learning, with $\gamma = 0.99$. 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 policies, 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 policy over options π_Ω.

The option attentions, values, policies and 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 attention diversity. 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.

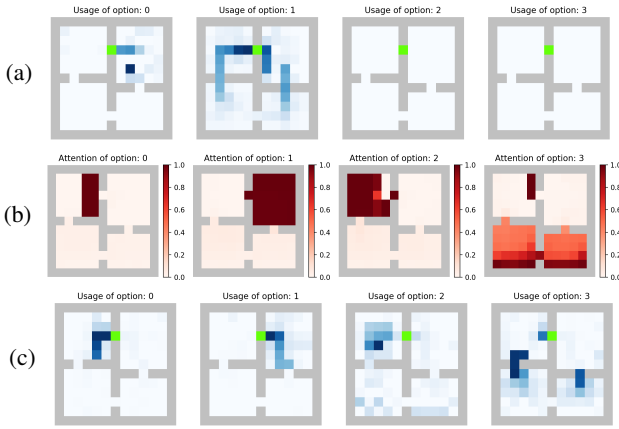


Figure 1. An example of learned options in the four-rooms domain with goal at north hallway (shown in green). **(a)** example of degenerate options learned by OC. Darker color indicates more frequent option execution in that particular state. Option 1 dominates and is used 88.12% while Option 0 is used 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.

This results in smooth attentions and leads to temporally extended options by minimizing switching. 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.6) and qualitatively (Figure 1). Further details regarding hyperparameters and architecture are provided in section C.

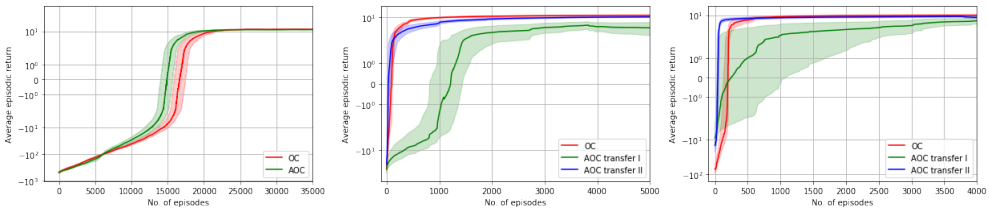
The resulting option usage and their attentions are shown in Figure 1. The learned options are distinct and specialized, and they perform state abstraction by focussing on a subset of the observation. The option usage respects the corresponding area of attention, which indicates that the options are typically limited to this area and that their behavior can be reasonably interpreted from their attentions. AOC also learns stable options whose behaviors and usage do 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 (see 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. Furthermore, a comparison between option domination in AOC and OC (see section A.3) 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: goal transfer (the goal is changed to a new random location) and blocked hallway (same goal 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.4). The slower transfer of AOC transfer I can be explained by the over-preference towards optimizing attention characteristics which AOC transfer II mitigates.

From another perspective, upon transfer, option-critic completely relearns the options. Figure 5 shows a specific instance of transfer. Comparing Figure 1a with Figures 5a and 5b 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, they should exhibit similar behavior upon transfer, and only change as



(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.

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 5c to 5f 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

For the ALE (Bellemare et al., 2013), we use 2 options with $\gamma = 0.99$. Each 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 deep network similar to option-critic. The architecture is shown in section C.

In addition to maximizing the total expected return, the attentions are constrained to exhibit some desired characteristics like diversity (enforced by maximizing the L1 norm between the object attentions of the options), sparsity (by penalizing non-zero attentions for the background) and regularity (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 3b). 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 attentions 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.

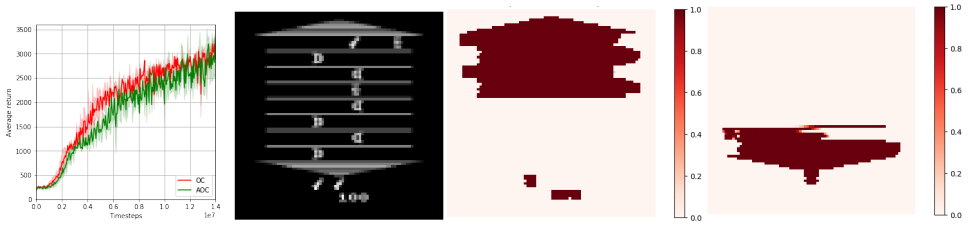
For the Asterix environment, the values 5000, 0.01, 100, and 1 for the weights w_1, w_2, w_3 and w_4 respectively, resulted in diverse attentions and good performance (Figure 3). AOC achieves a similar sample complexity compared to OC, despite also having to learn the state-dependent attention mechanism. Learning the attentions enables options to specialize early on, and hence speed up training, despite having more parameters to learn. Figures 3c and 3d show the resulting option attentions and indicate that the options have respectively specialized to behaviors pertaining to the main sprite's position in the upper and 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

Specific to the options framework (Sutton et al., 1999), 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) learns the initiation sets as differentiable interest functions, but the initialization of the interest functions is biased towards all options being available everywhere. Our AOC approach does not require any special initializations. Deep skill chaining (Bagaria and Konidaris, 2020) 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.

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.



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

Figure 3. Training curves, a game frame, and learned option attentions for Asterix.

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.

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.

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

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.

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.

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.

Appendix

A. Other four-rooms experiments

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

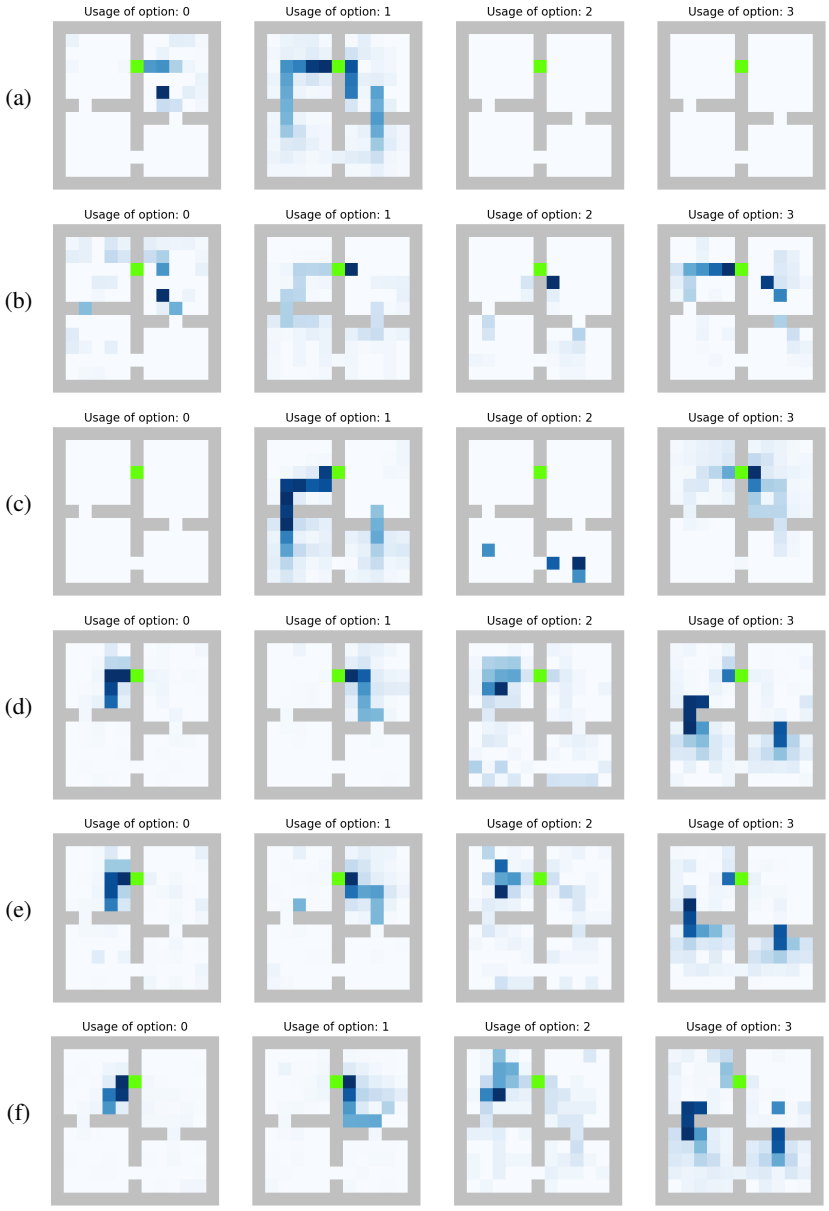


Figure 4. **(a) to (c):** even after convergence, options learned with OC are volatile and continue to change frequently. **(d) to (f):** AOC learns more stable options which continue to exhibit similar behavior. In the snapshots of the options above, for both OC and AOC, 100,000 frames of training has been performed between successive rows. The goal is the north hallway, shown in green.

A.2. Comparison of option transfer between AOC and OC

A comparison between options and attentions before and after transfer with AOC 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.

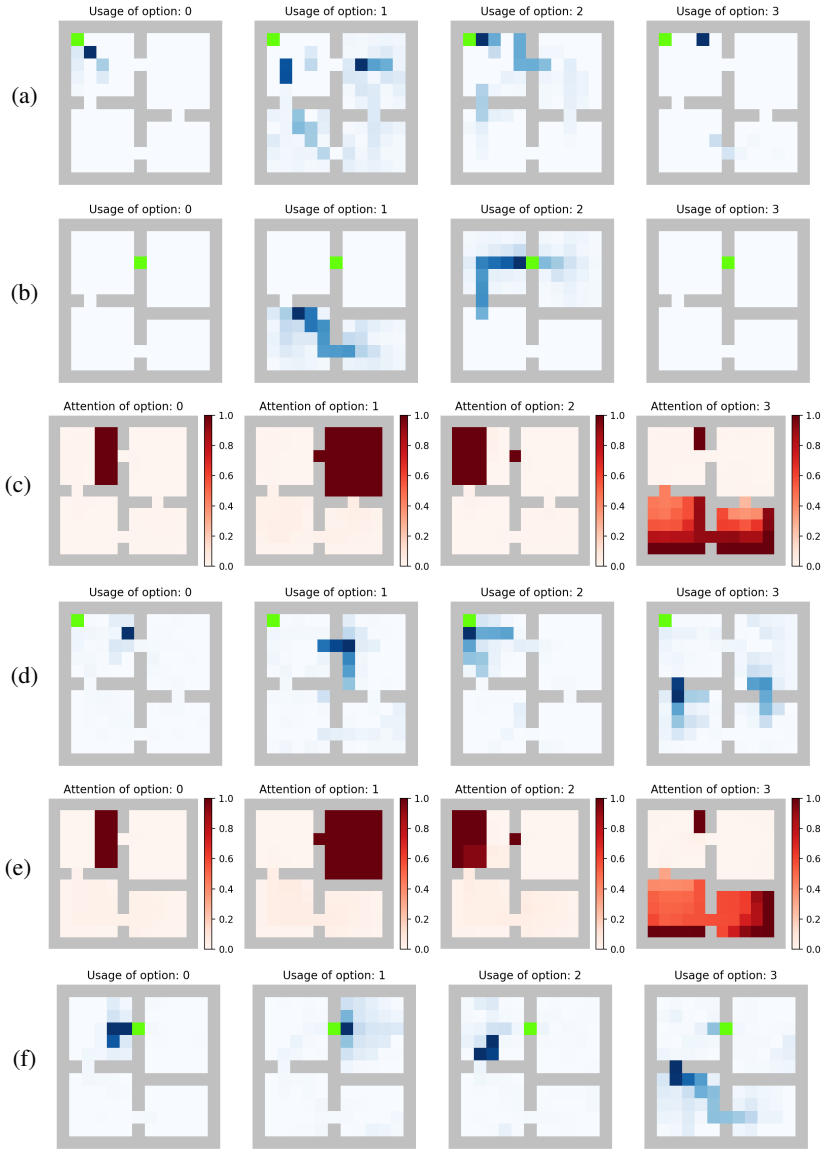


Figure 5. **(a) and (b):** resulting options learned by option-critic upon goal transfer and transfer with blocked hallway respectively. **(c) and (d):** the resulting option attentions and usage upon goal transfer. **(e) and (f):** the resulting option attentions and usage upon transfer with blocked 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.

A.3. 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.

A.4. 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

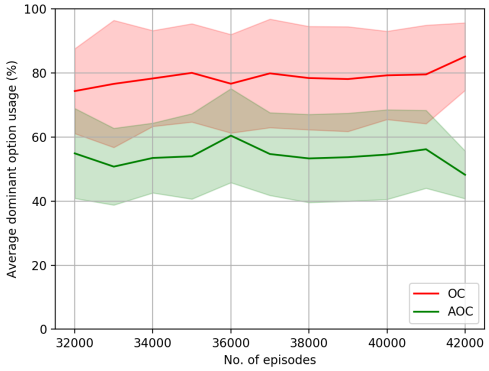
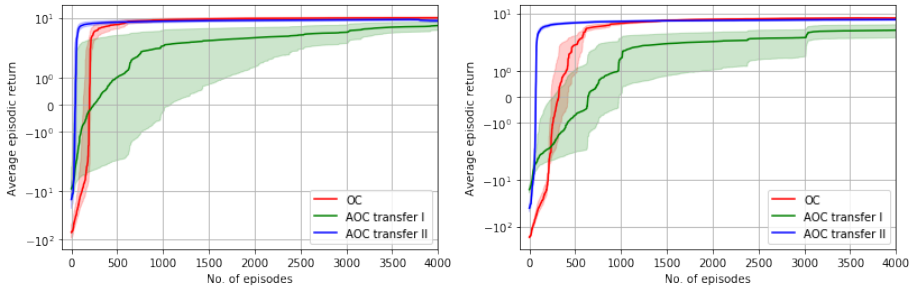


Figure 6. Comparison of average usage of the dominant option in the four-rooms domain.

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

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

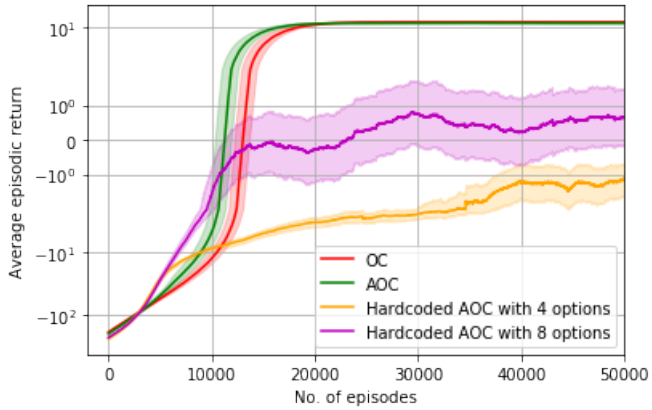


Figure 8. Comparison of OC, AOC, and AOC with hardcoded attentions

A.6.1. 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.6.2. 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.6.3. 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.6.4. 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.

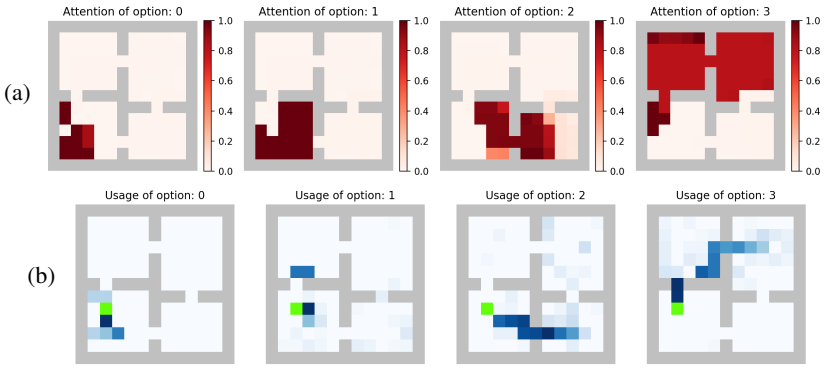


Figure 9. When multiple options have significant overlapping attentions in a state, any of these options may be executed. The goal is shown in green.

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

C.1. Algorithm

Algorithm 1 shows the algorithm for AOC.

C.2. Four-rooms environment

In the four-rooms navigation task, the agent must reach a specified goal. The observation space consists of one-hot-encoded vectors for every 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. We use 4 options for learning, with a discount factor of 0.99. The attention $h_{\omega,\phi}$ for each option ω is

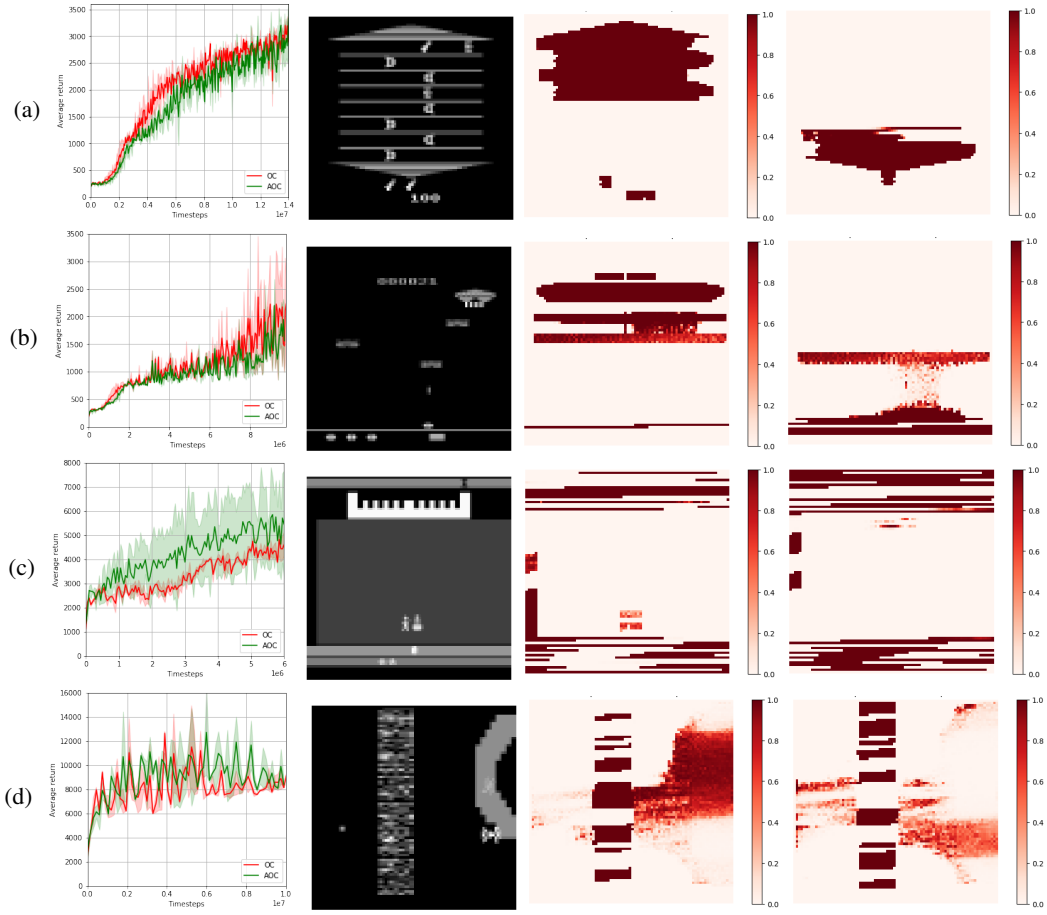


Figure 10. AOC results for the shared-attention model. Column-wise: learning curves, game frame, option 0 attention, option 1 attention. (a) Asterix (b) Assault (c) Krull (d) Yars' Revenge

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 policies, 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 policy over options π_Ω. Intra-option exploration is enforced with entropy regularization.

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 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. All learning curves for the four-rooms domain are averaged over 15 independent runs with randomly chosen goals (before and after transfer) and randomly blocked hallways.

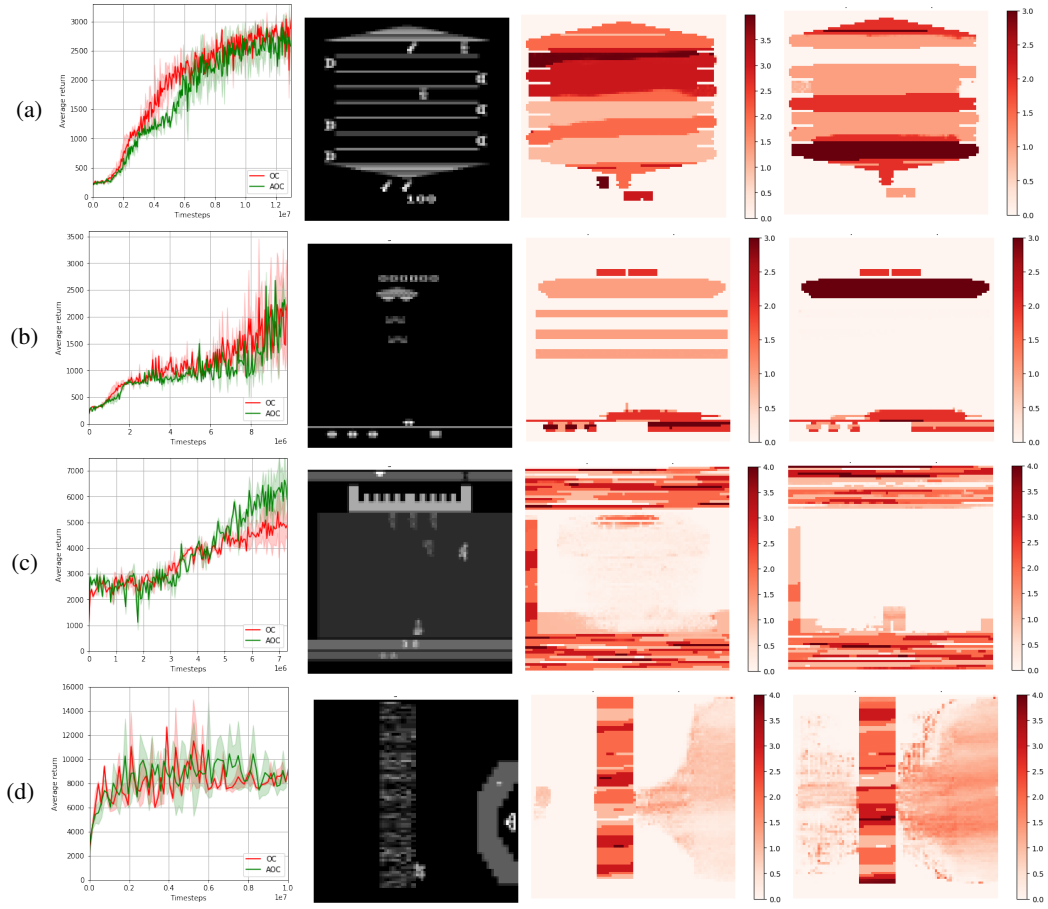
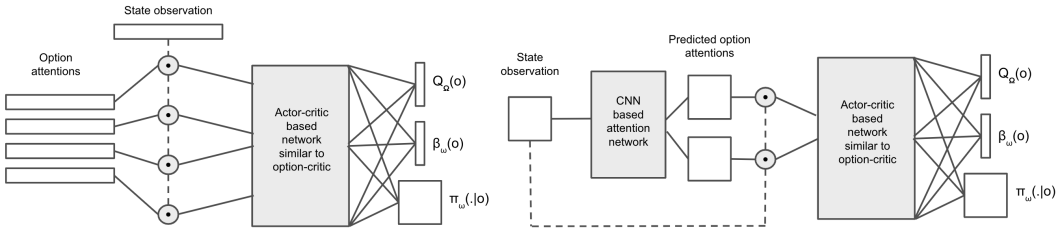


Figure 11. AOC results for the frame-dependent-attention model. Note that the attention maps shown here are the sum of frame-wise attention maps for each option. Framewise attentions are much more distinct and are similar to the attention maps from the shared-attention model. Column-wise: learning curves,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



(a) Architecture for four-rooms (b) Architecture for atari

Figure 12. The shared network models for option learning with AOC. $\odot$ 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.3. 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

Algorithm 1 Attention Option-Critic

Input: $\alpha_\theta, \alpha_\nu, \alpha_\phi$ as learning rates for θ, ν and ϕ 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 ω according to ϵ-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 ω according to ϵ-soft $\pi_\Omega(o')$
  end if
  $s \leftarrow s'$
  $o \leftarrow \{h_{\omega,\phi}(s) \odot s : \omega \in \Omega\}$
until s' is terminal

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

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.