Learning Temporally Extended Options Using Option-Critic

Vedant Gupta
Department of Computer Science
Brown University
vedant_gupta@brown.edu

Yash Gotmare
Department of Computer Science
Brown University
shreeyash_gotmare@brown.edu

Abstract

Learning temporally extended skills or options has been a long-standing goal of reinforcement learning to enable autonomous agents to transfer expertise across domains. A hallmark approach used in HRL is the option-critic algorithm [1], which learns options and a policy over options in an end-to-end fashion. However, when applied out-of-the-box, option-critic suffers from *option-collapse*, i.e. the learned skills only last a single timestep. We offer an analysis of a series of preexisting and novel methods to prevent option collapse, seeking to alleviate this problem while still matching vanilla option-critic performance. We then apply our methods to the OpenAI Gym Taxi-V3 domain, making this the first analysis of option-critic in a multi-task setting to our knowledge.

1 Introduction

Learning hierarchies of abstract actions, or options, is fundamental to how humans solve diverse and complex tasks [4]. For instance, when learning to cook for the first time, individuals acquire generalizable skills such as chopping vegetables, turning on the stove, and cleaning dishes. These foundational skills can then be reused in various contexts, allowing one to adapt quickly to new environments, like different kitchens, or to execute new recipes efficiently.

In an attempt to enable robots to master new tasks with the flexibility and speed of humans, substantial research efforts have been invested in learning temporally extended skills autonomously, collectively defining the field of Hierarchical Reinforcement Learning (HRL). In short, HRL methods aim to decompose long-horizon tasks over primitive actions into shorter-horizon tasks over temporally extended options (or skills). By finding such a structured representation of a domain's action space, one hopes that agents will be able to master new domains faster by reusing learned options and therefore reducing the problem's horizon. While multiple algorithms exist to learn options autonomously, the option-critic algorithm [1] stands out as a policy gradient method that learns options and a policy over options simultaneously. However, when applied out-of-the-box, option-critic suffers from option-collapse, i.e. the learned options often last for a single timestep before terminating. This issue defeats the primary objective of learning options - reducing the horizon of a problem.

We present a comparison of a series of pre-existing and novel approaches to prevent option collapse, testing their performance in the OpenAI Gym Taxi-V3 domain. To our knowledge, option-critic has not been evaluated on a multi-task domain like Taxi before. Hence, we adapt option-critic to the multi-task setting by learning a *task-aware* policy over *task-agnostic* options (more details in 5). We first present a slight modification to option-critic that learns slightly longer skills without affecting performance. Then, we compare three other approaches: (i) pessimistic initialization strategies, (ii) termination costs, and (iii) diversity-enriched option-critic [8]. Our results show a tradeoff between temporal extension and performance—methods that learn longer-lasting options tend to collect less reward. Our results highlight the need for the development of more robust algorithms that are less susceptible to local minima, a common issue with existing HRL approaches. [1]

2 Background

2.1 Markov Decision Processes and Options

Sequential decision-making problems are often abstracted as Markov Decision Process (MDP). An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, r, \mathcal{P}, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $P : \mathcal{S} \times \mathcal{A} \to (\mathcal{S} \to [0, 1])$ is the transition function, and $\gamma \in [0, 1)$ is a discount factor to prioritize immediate rewards over future rewards. The goal of a RL agent is to learn a policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which maps each state to a distribution over actions to take, with the objective to maximizing expected discounted rewards, $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid S_0\right]$, where S_0 is the starting state of the agent.

To formalize the notion of temporally extended skills, we adopt the *options framework* [12]. An option [2], $o \in \Omega$, is tuple $(\mathcal{I}_o, \pi_o, \beta_o)$, where $\mathcal{I}_o \subset \mathcal{S}$ is the *initiation set* of the options, π_o is the *intra-option policy*, and $\beta_o : \mathcal{S} \to [0, 1]$ is the termination function. Put together, these describe a skill, by telling the agent where it should begin the skill ($\mathcal{I}_o$), how it should execute the skill (π_o), and when it should terminate the skill (β_o). Finally, to learn how to complete a task, one also learns a *inter-option policy*, π_Ω. At the start state, the agent selects an option according to this policy and then follows the corresponding intra-option policy to termination. Then, a new option is chosen from π_Ω and so forth till the task is completed.

2.2 Option-Critic

Option-critic [1] is an algorithm to lean options and a policy over options simultaneously. Specifically, given a task, option-critic learns three components: (i) an inter-option policy, (ii) option policies, and (iii) option termination functions, all by maximizing a single objective, the expected sum of discounted rewards. To do this, the authors provide "gradient theorems" that estimate the gradient of the expected sum of discounted rewards in terms of the three components outlined above. We go into the relevant components in the sections below.

3 Options Collapse

To calculate the termination probabilities of options, option-critic proposes the termination gradient theorem, which states that the gradient of the expected discounted return $L(\theta)$ while following the current policy with respect to some parameter of the termination policy θ_β is

$$\frac{\partial L(\theta)}{\partial \theta_\beta} = \mathbb{E}\left[-\frac{\partial \beta(S_t, O_t)}{\partial \theta_\beta}(Q_\pi(S_t, O_t) - V_\pi(S_t))\right] \tag{1}$$

Here, $\beta(S_t, O_t)$ is the probability of terminating option O_t in state S_t. $V_\pi(S_t)$ is the expected discounted return from state S_t under current policy π, i.e., $V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right]$. Similarly $Q_\pi(S_t, O_t)$ is the expected discounted reward given the current state and option, i.e., $Q_\pi(s, o) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, o_0 = o\right]$. Note that from these definitions, it follows that:

$$V_\pi(s) = \max_o Q_\pi(s, o) \tag{2}$$

Intuitively, the option termination theorem increases the termination probability of a state in proportion to the current option's suboptimality. However, notice that by definition, $Q_\pi(S_t, O_t) - V_\pi(S_t)$ is always negative, driving $\beta(S_t, O_t)$ to 1, i.e., options terminate in a single time step. This issue, also

[1] Our implementation is available at https://github.com/Mr-vedant-gupta/Hierarchical-policy-learning/

[2] Also called a "skill" in some literature

called "options collapse", is consistently observed when using the option-critic algorithm, defeating a primary motivation of learning *temporally-extended* options.

In this work, we will explore various preexisting and novel approaches to prevent options-collapse. Most existing approaches ([8], [5]) add auxiliary objectives that discourage options collapse. While successful in some cases, such approaches can be hard to tune and modify the optimal policy. Therefore, in addition to surveying such methods, we propose the following modification to the termination gradient theorem:

$$\frac{\partial L(\theta)}{\partial \theta_\beta} = \mathbb{E}\left[-\frac{\partial \beta(S_t, O_t)}{\partial \theta_\beta}\left(Q_\pi(S_t, O_t) - Q_\pi(S_t, \neg O_t)\right)\right] \tag{3}$$

Here, $Q_\pi(S_t, \neg O_t) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, o_0 \neq o\right]$. This enforces the additional condition that on terminating an option, the next option chosen has to differ from it. As $Q_\pi(S_t, O_t) - Q_\pi(S_t, \neg O_t)$ can be positive, this revised gradient will drive termination probabilities down when O_t is the optimal option at S_t, potentially helping prevent options collapse while leaving the optimal policy intact (see 9.1.1 for more details).

4 Approaches to Prevent Options Collapse

In this section, we will describe the approaches we tried to prevent options collapse. A detailed analysis of the results of each approach with its advantages and disadvantages will be presented in the results section.

4.1 Diversity-Enriched Option-Critic

As described earlier, the vanilla option-critic architecture suffers from options collapse. Diversity-Enriched Option-Critic [8] proposes to alleviate this problem in two ways: (i) by using an additive auxiliary pseudo reward $\mathcal{R}_{bonus}$ to augment the task-specific reward, and (ii) modifying the termination objective $L(\theta)$ to instead account for the relative diversity of options.

4.1.1 Encouraging Diversity While Learning (DEOC)

In contrast to prior literature on learning diverse options, which uses states to specialize options, DEOC uses an option's policy itself to assess its diversity. Intuitively, one can measure how diverse two policies are by calculating the entropy of their action distributions. For two options π_{O_1} and π_{O_2}, their entropy, $\mathcal{H}(A^{\pi_{\mathrm{O}_1}} \mid S)$ and $\mathcal{H}(A^{\pi_{\mathrm{O}_2}} \mid S)$ respectively, can be measured as the Shannon entropy with base e, where A represents the action distribution for an option. DEOC maximizes the entropy of the options themselves, the divergence between their action distributions $\mathcal{H}(A^{\pi_{\mathrm{O}_1}}; A^{\pi_{\mathrm{O}_2}} \mid S)$, and finally, the stochasticity of the policy over options itself $\mathcal{H}(O^{\pi_\Omega} \mid S)$, leading to the following pseudo reward:

$$\begin{aligned}\mathcal{R}_{\text{bonus}} &= \mathcal{H}\left(A^{\pi_{\mathrm{O}_1}} \mid S\right) + \mathcal{H}\left(A^{\pi_{\mathrm{O}_2}} \mid S\right) \\ &\quad + \mathcal{H}\left(O^{\pi_\Omega} \mid S\right) + \mathcal{H}\left(A^{\pi_{\mathrm{O}_1}}; A^{\pi_{\mathrm{O}_2}} \mid S\right)\end{aligned}$$

The pseudo reward bonus above is then used to augment the reward function for the task by introducing a tradeoff parameter τ which controls the importance of the diversity term versus the reward:

$$\mathcal{R}_{\text{aug}}\left(S_t, A_t\right) = (1-\tau) R\left(S_t, A_t\right) + \tau \mathcal{R}_{\text{bonus}}\left(S_t\right)$$

4.1.2 Encouraging Diversity In Termination (TDEOC)

The authors modified the original objective function to satisfy the following two criteria:

1. One should terminate options in states where the available options are diverse. Intuitively, this makes sense since often a temporal decomposition of a task leads to states where there are multiple actions possible, each equally likely to improve the objective. In the Taxi domain, these would be the colored locations.

2. If a diversity metric is used in the termination objective, it should capture diversity relative to other states in the sampled trajectories.

Taking only the first criteria into account, intuitively the right termination function would account for how diverse a state is. Earlier, we measured this using $\mathcal{R}_{bonus}$ since the pseudo reward sum measures how diverse the options at a given state are. Further, to account for the second criteria, one can use a buffer of recent transitions to measure the diversity of the current state relative to the diversity of transitions in that buffer:

$$\mathcal{D}\left(S_t\right)=\frac{\mathcal{R}_{\text{bonus }}\left(S_t\right)-\mu_{\mathcal{R}_{\text{bonus}}}}{\sigma_{\mathcal{R}_{\text{bonus}}}}$$

The termination objective then becomes

$$L\left(\theta_\beta\right)=\mathbb{E}\left[\beta\left(S_t, O_t\right) \mathcal{D}\left(S_t\right)\right]$$

4.2 Learning Pessimistic Q-Values

When running option-critic with a large number of options, one often observes option collapse kicking in almost instantly. This issue can be attributed to the fact that the estimates of Q_π are noisy, given they are trying to match a changing policy. Now, consider the termination gradient in 1, substituting V_π with its definition in 2:

$$\frac{\partial L(\theta)}{\partial \theta_\beta}=\mathbb{E}\left[-\frac{\partial \beta(S_t, O_t)}{\partial \theta_\beta}(Q_\pi(S_t, O_t)-\max_O Q_\pi(S_t, O))\right] \tag{4}$$

Given a large number of options with noisy value estimates, it becomes highly likely that $Q_\pi(S_t, O_t) < \max_O Q_\pi(S_t, O)$, driving termination probabilities to 1. To counter this issue, we simultaneously learn two estimates of value, Q_π and Q_π^{pess}. Both estimates are updated identically, but Q_π^{pess} is initialized at lower values than Q_π. This difference in initializations makes Q_π^{pess} behave as a pessimistic estimate of value. With this modification, the termination gradient can be modified as follows (note that the same change can be applied analogously to 3):

$$\frac{\partial L(\theta)}{\partial \theta_\beta}=\mathbb{E}\left[-\frac{\partial \beta(S_t, O_t)}{\partial \theta_\beta}(Q_\pi(S_t, O_t)-\max_O Q_\pi^{pess}(S_t, O))\right] \tag{5}$$

This modification encodes the intuition that the termination probabilities should only be driven up if there is high confidence that a better option exists. Refer 9.1.3 for a more mathematical treatment.

4.3 Termination Costs

The issue of options collapse can be attributed to the fact that the termination gradient theorem only maximizes the expected discounted reward, which can be done with primitive actions and does not require temporally extended options. Hence, to encode the auxiliary objective of learning maximally-extended options, we add a termination cost, c, to the reward model of the environment every time the agent chooses to switch out of its current option. This encodes the goal of minimizing the number of terminations while still maximizing the expected discounted reward. Implementing this requires adding the cost c to the reward when updating $Q_\pi(S_t, O_t)$ and the following revision to the termination gradient theorem [3] [4]:

$$\frac{\partial L(\theta)}{\partial \theta_\beta}=\mathbb{E}\left[-\frac{\partial \beta(S_t, O_t)}{\partial \theta_\beta}(Q_\pi(S_t, O_t)-V_\pi(S_t)+c)\right]$$

More details can be found in 9.1.2.

[3] While we developed and tested this method independently, we later found out it is identical to the main contribution of [5]

[4] The same change is applied analogously to the modified termination gradient theorem

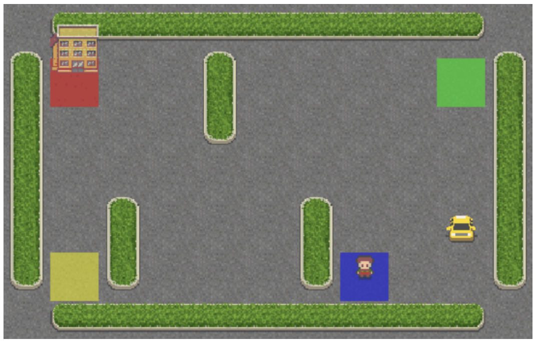


Figure 1: The Taxi Domain

5 Environment

We test the methodologies outlined above in the OpenAI Gym Taxi-V3 environment. Taxi-V3 simulates a taxi navigating a 5x5 grid to pick up and drop off passengers at designated locations. The agent can perform six actions: move south, north, east, and west, pick up the passenger, and drop off the passenger. The agent receives a small negative reward for each navigational action, while the pick-up and drop-off actions receive a large positive or negative reward depending on whether the pickup/dropoff was necessary for the completion of the current task.

As alluded to in our introduction, we chose the Taxi domain for two reasons. Firstly, Taxi is a multi-task environment (as the pick-up and drop-off locations change between episodes), and to our knowledge, option-critic has not been evaluated on such domains. Secondly, the Taxi domain presents a difficult exploration-exploitation problem, since the pick-up and drop-off action incur a large negative action from most states, but must be executed from the right state to successfully complete the task.

To adapt option-critic to the multi-task setting, we split the agent's state into a state of the world (e.g. taxi at $(4, 3)$ and the passenger is in the taxi), and a task description (e.g. pick up the passenger from A and drop them off at D). While the *inter-option* policies receive both these pieces of information, each option's policy does not receive the task description, forcing options to learn policies that are common across tasks, leading to improved reusability in downstream evaluation on unseen tasks. This augmentation makes it extremely suboptimal for the agent to solve all tasks using a single option, adding additional nuance to the option collapse problem, which is another reason we believe the Taxi domain is a good evaluation environment.

6 Experiments

In the subsequent sections, we present results obtained when using the approaches above. We train each approach for $10,000$ episodes, each of which lasts for a maximum of 200 time steps. We average our results over two seeds (more information on implementation details can be found in 9.2). For each run, we present a graph showing discounted rewards and another showing the average option length before it is terminated. The curves in these plots are smoothed using a rolling mean with a window size of 50 episodes.

A near-optimal policy in the taxi domain will receive a reward of around 10, while a reward of around -200 typically indicates that the agent is randomly moving in the domain without completing the task. To put our results into perspective, note that it is possible to solve the task by learning 8 options, 4 of which take the empty taxi to the 4 pickup locations, while the other 4 take a taxi with a passenger to one of the four drop-off locations. Using this decomposition, the average option would last around 6 timesteps, representing the average distance between destinations.

6.1 Diversity-Enriched Option-Critic

Figure 2 shows a comparison between vanilla option-critic, and the two approaches proposed by [8], as outlined in 4.1.1 and 4.1.2. The plots of vanilla option-critic show that while the agent learns to

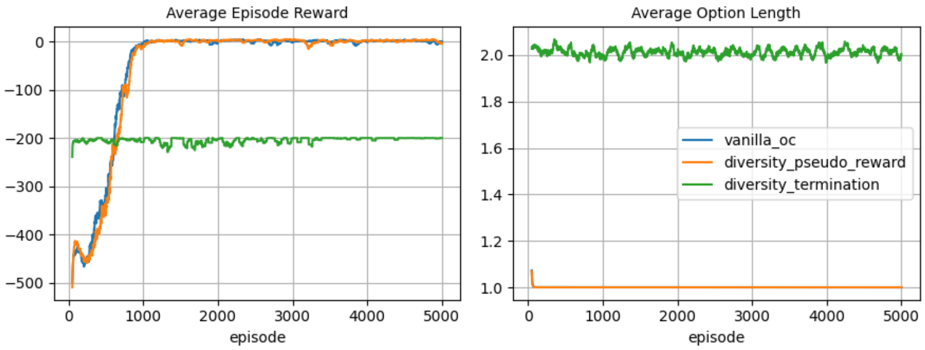


Figure 2: Comparison of mean episodic reward and mean option duration for DEOC, TDEOC, and vanilla option-critic. Both vanilla option-critic and DEOC perform identically.

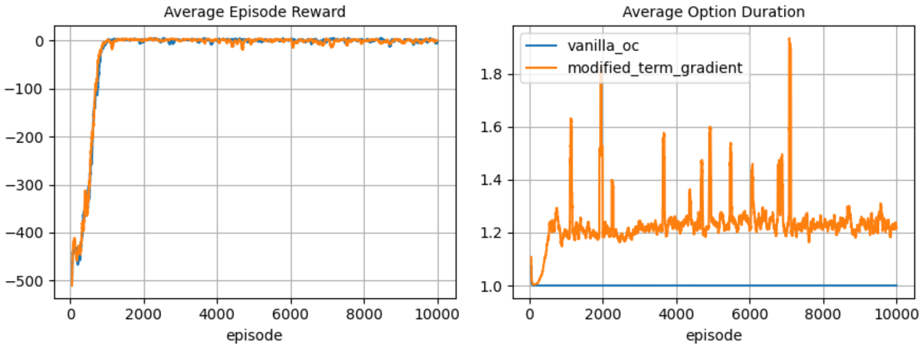


Figure 3: Comparison of mean episodic reward and mean option duration between the modified termination gradient described in 9.1.1 and vanilla option-critic.

complete the task after around 1000 episodes, the options collapse to a single time step right from the start. We analyze the effects of both approaches suggested in [8].

1. The first approach in [8] leads to no noticeable differences in either option duration or average episodic reward. This result is not surprising since even though the intrinsic task reward now has a slight bias towards encouraging states that lead to options that are diverse (we used $\tau = 0.0001$, similar to the authors' implementation), the termination function still validates the best option at each iteration, leading to a result no different than vanilla option-critic.
2. The second approach fails to converge to a policy that successfully learns to solve Taxi. However, the options do last longer than a single timestep. The fact that the agent fails to learn could be attributed to the fact that the termination function no longer optimizes for the task reward. It solely focuses on the relative diversity, which is measured by standardizing $\mathcal{R}_{bonus}$ to mean $\mu = 0$ and standard deviation $\sigma = 1$ using samples in the buffer. The episodic reward hovers around 200, indicating that the agent performs similarly to one that acts randomly.

6.2 Modified Termination Gradient Theorem

Figure 3 shows a comparison between vanilla option-critic and option-critic with the modified termination gradient proposed in 3. While we observe no statistically significant differences in the rewards obtained by both approaches, the modified termination gradient achieves marginally better temporal extension with the average option lasting around 1.2 timesteps. We believe the fact that this method could not achieve better temporal extension can be attributed to two principal reasons:

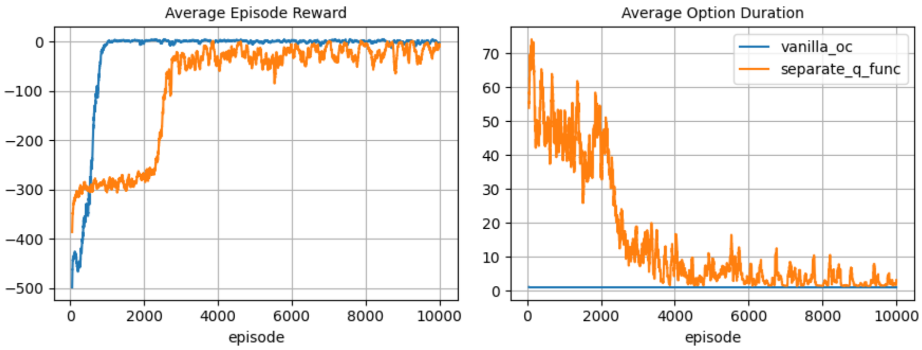


Figure 4: Comparison of mean episodic reward and mean option duration between using the pessimistic initialization approach and vanilla option-critic.

1. There is a large space of options that can be learned to solve the taxi domain, and we believe options that support temporal extension on the other hand are relatively sparse. For example, one could learn 6 options, that respectively execute the 6 primitive actions in the domain. While this option set is sufficient to solve Taxi, it does not support temporal extension. Our modification to the termination gradient does nothing to prevent such solutions. This is consistent with our analysis of the policies obtained, wherein it was extremely unlikely for the optimal option to remain the same between consecutive timesteps.

2. The modified termination condition only offers temporal extension when the current option is optimal. In this scenario, to estimate $Q_\pi(S_t, \neg O_t)$ as required in 3, we use the value of the second best option. Since we trained with a large number of options (10), we observed that the difference in value between the best and second best options is often small, offering only a small decrease in termination probabilities. Note that this issue could be alleviated by introducing additional hyperparameters.

6.3 Pessimistic Q-Values

Figure 4 shows results obtained when using the approach outlined in 4.2. Along with the modified termination gradient described in the previous section, using a pessimistic initialization value of -5, we observe no improvement in the temporal extension achieved compared to using just the modified termination gradient. Moreover, the agent takes longer to learn to complete the task and still performs slightly suboptimally after $10,000$ episodes.

We observe that the use of pessimistic values in 5 initially drives termination probabilities to 0, and in the plots above the agent tried to solve the task with a single option for the first 2000 episodes. However, since our options are task-agnostic, solving the Taxi domain with a single option is extremely difficult, and therefore it looks like the agent is not able to discover much useful information in the first 2000 episodes. As a result, the agent only begins to solve the task when Q_π^{pess} becomes comparable to Q_π, defeating the objective of learning a pessimistic estimate.

6.4 Adding Termination Costs

Figure 5 shows results obtained when using a termination cost as outlined in 4.3. We compare results obtained with a cost of 3, 3.5, and 4 and use the modified termination gradient for all runs. We observe a clear tradeoff between the reward collected and the average option length. Specifically, with a cost value of 4, the agent is discouraged from switching options very strongly, and as a result, does not learn to solve the task. The runs with values 3 and 3.5 are more successful but display the same tradeoff.

However, as alluded to previously, the termination value can be hard to tune, and it took some trial and error to converge upon 3, 3.5, and 4 as good candidate values. We include a visualization of learned policies in 9.3

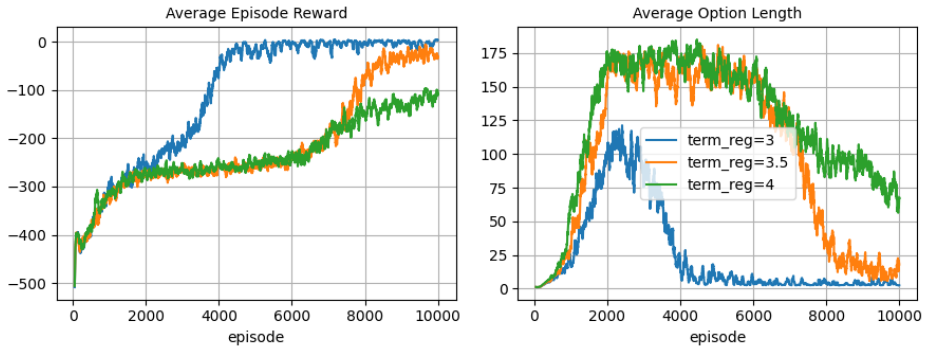


Figure 5: Comparison of mean episodic reward and mean option duration between the three chosen values of termination cost: 3, 3.5, 4, as described in 4.3

7 Conclusion

We compared the performance of four approaches: a modified termination gradient, diversity-enriched option-critic, pessimistic initialization, and termination penalties in the Taxi domain. We believe that our modification to the termination gradient in 3 is a better update rule than the original termination gradient in option-critic, both because of its theoretical validity and the experimental results. However, the modified termination rule in itself is not sufficient to learn temporally extended options.

To achieve this, we tried pessimistic initialization as an approach to learn temporally extended options without needing auxiliary objectives, but we did not observe promising results. We still believe this approach might hold some promise in single-task domains, where it is yet to be evaluated.

Fundamentally, maximizing expected return, as done by option critic, does not encourage temporal extension in itself, and we believe some additional objective is necessary to encode this. While diversity-enriched option-critic did not perform well in our application, adding termination costs, a surprisingly simple approach, held the most promise. The largest downside of this approach is that the termination cost was hard to tune, and small changes in the cost can significantly alter the observed results. Future work could further investigate this problem by trying to find heuristics for calculating good termination costs or changing termination costs on the fly based on performance.

Since option-critic learns options and inter-option policies simultaneously, each of which relies on the other for updates, we believe it is especially susceptible to local minima. Therefore, another line for future work may be to reason that methods such as option-critic are not well suited for multi-task environments.

While the answers to these questions are unclear to us, we believe them to be promising pursuits that can help greatly accelerate robot learning.

8 Related Work

Our work follows a line of work exploring auxiliary objectives to prevent option collapse. Notably, we cover methods developed in [8] and [5]. Other methods we did not explore include [2], which uses attention-based mechanisms to learn diverse and temporally extended options, and [6], which proposes a revised information-theoretic termination objective that is independent of task reward.

Option-critic itself is only one of multiple HRL approaches, a survey of which can be found in [10]. Other notable HRL approaches include [9], which learns a "chain" of skills such that the termination of a previous skill corresponds to the initiation of the next skill. However, this approach only works in goal-directed tasks ([10]). [3] learns to segment action trajectories into options. However, as the learned skills rely on successful trajectories, they can not aid in the agent's initial exploration of the environment.

The approach we outline in 4.2 has some similarity to a preexisting set of literature that learns multiple Q-value estimates to solve a task. Notably, Double Q-Learning ([7]) attempts to resolve the tendency of Q-learning to overestimate value by learning two Q-value estimates: one for action selection and the other for action evaluation.

References

[1] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31. Association for the Advancement of Artificial Intelligence (AAAI), 2017.

[2] Raviteja Chunduru and Doina Precup. Attention option-critic. *arXiv preprint arXiv:2201.02628*, 2022.

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

[4] Maria K. Eckstein and Anne G.E. Collins. Computational evidence for hierarchically structured reinforcement learning in humans. *Proceedings of the National Academy of Sciences*.

[5] Jean Harb, Pierre-Luc Bacon, and Doina Precup. Asynchronous advantage option-critic with deliberation cost.

[6] Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Remi Munos, and Doina Precup. The termination critic. *arXiv preprint arXiv:1902.09996*, 2019.

[7] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

[8] Anand Kamat and Doina Precup. Diversity-enriched option-critic. *arXiv preprint arXiv:2011.02565*, 2020.

[9] George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in neural information processing systems*, 22, 2009.

[10] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

[11] Richard S Sutton, Doina Precup, and Satinder Singh. Intra-option learning about temporally abstract actions. In *ICML*, volume 98, pages 556–564, 1998.

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

9 Appendix

9.1 Mathematical Derivations

Below we provide derivations for some of the results used in the sections above.

9.1.1 Disallowing consecutive options from being identical

The modified gradient theorem provided in 3 adds the additional constraint that consecutive options can not be the same. Using the same notation as in 3, we know that the expected reward from state S_t and option O_t is

$$(1 - \beta(S_t, O_t)) * Q_\pi(S_t, O_t) + \beta(S_t, O_t) * Q_\pi(S_t, \neg O_t)$$

Taking the gradient with respect to parameter θ_β gives the modified termination gradient theorem.

It is also straightforward to see that this modification will not affect the optimal policy, assuming options are chosen greedily, i.e., on terminating an ongoing option at state S_t, the next option

$O_t = \arg\max_O Q_\pi(S_t, O)$. Let the previous option be O_{t-1}. If $O_{t-1} \neq O_t$, $Q_\pi(S_t, \neg O_t) = V_\pi(S_t)$ and so the modified termination gradient is identical to the original gradient theorem. If $O_{t-1} = O_t$, then the expected discounted reward is invariant under $\beta(S_t, O_{t-1})$, so there is no harm done in driving the termination probabilities down to 0.

9.1.2 Adding Termination Costs

In similar fashion to 9.1.1, on adding a termination penalty c, the expected reward from state S_t and option O_t becomes,

$$(1 - \beta(S_t, O_t)) * Q_\pi(S_t, O_t) + \beta(S_t, O_t) * (V_\pi(S_t) - c)$$

Taking the gradient with respect to parameter θ_β gives the modified gradient.

9.1.3 Pessimistic Initialization

When running option-critic, the state-option value estimates $Q_\pi(S, O)$ are updated using the following rule:

$$Q'_\pi(S_t, O_t) \leftarrow (1 - \alpha)Q_\pi(S_t, O_t) + \alpha(r_{t+1} + \gamma * U(S_{t+1}, O_t)) \tag{6}$$

where,

$$U(S_{t+1}, O_t) = (1 - \beta(S_{t+1}, O_t))Q_\pi(S_{t+1}, O_t) + \beta(S_{t+1}, O_t) \max_O Q_\pi(S_{t+1}, O)$$

and α is the learning rate. Theorem 1 in [11] shows that under certain assumptions, this defines a contraction mapping, and therefore Q_π converges to the true values, Q^*_π, regardless of the initialization. Suppose we initialize Q_π with lower bounds of the true value (in the Taxi domain, there are at most 200 time steps per episode, each with a worst-case reward of -10, making -2000 a lower bound).

Assuming that the policy π and the environment is fixed and deterministic, we show that if Q_π is a lower bound on Q^*_π, on applying the update in 6, the new estimate Q'_π is also a lower bound. This serves as a theoretical intuition for the approach in 4.2, as it shows that if Q^{pess}_π is initialized pessimistically, it will remain a lower bound on value after multiple updates from 6, while also becoming an increasingly better estimate. [5]

Substituting the defintion of U into 6, we get:

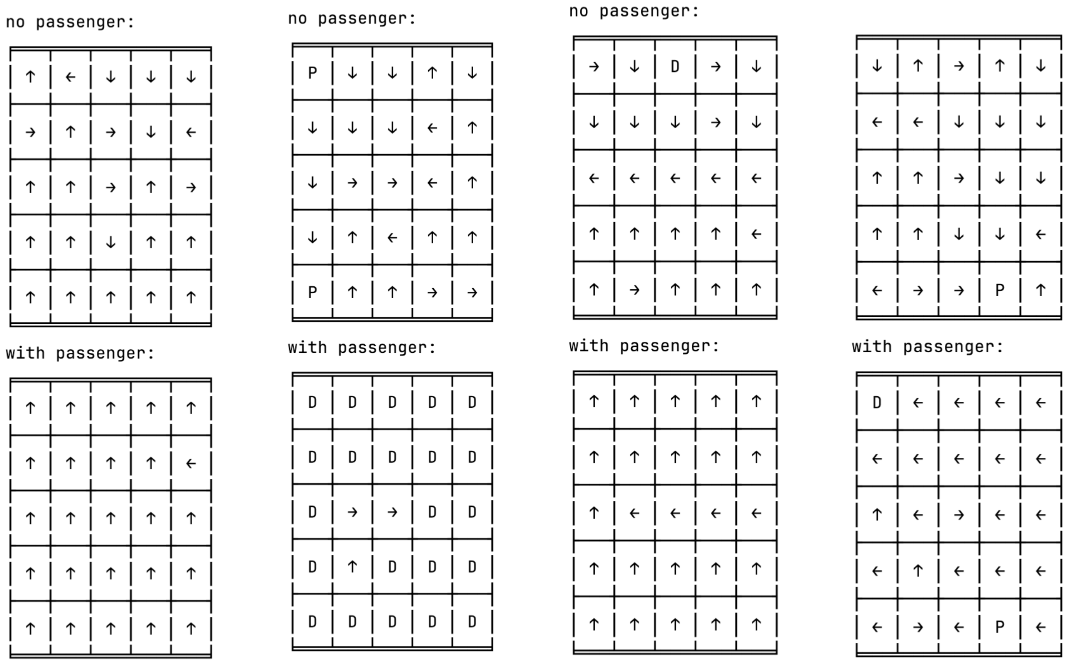
$$\begin{aligned}
Q'_\pi(S_t, O_t) \leftarrow &(1 - \alpha)Q_\pi(S_t, O_t) + \\
&\quad \alpha(r_{t+1} + \gamma * [(1 - \beta(S_{t+1}, O_t))Q_\pi(S_{t+1}, O_t) + \beta(S_{t+1}, O_t) \max_O Q_\pi(S_{t+1}, O)] \\
\leq &(1 - \alpha)Q_\pi(S_t, O_t) + \\
&\quad \alpha(r_{t+1} + \gamma * [(1 - \beta(S_{t+1}, O_t))Q^*_\pi(S_{t+1}, O_t) + \beta(S_{t+1}, O_t) \max_O Q^*_\pi(S_{t+1}, O)] \\
= &(1 - \alpha)Q_\pi(S_t, O_t) + \alpha Q^*_\pi(S_t, O_t) \\
\leq &Q^*_\pi(S_t, O_t)
\end{aligned}$$

Which completes the proof.

9.2 Implementation Details

We implement each component of option-critic with a single linear layer in Pytorch without any bias, which is equivalent to tabular learning. The task-aware state and task-agnostic state are both encoded as one hot vectors. For all our runs we use the following main hyperparameters:

[5] This only serves as an intuition, as the assumptions we make of a fixed and deterministic policy don't hold in practice. Furthermore, we don't initialize Q^{pess}_π to -2000 in our tests, as such a low initialization would take a long time to converge to a reasonable estimate.



```
no passenger:
| ↑ | ← | ↓ | ↓ | ↓ |
| → | ↑ | → | ↓ | ← |
| ↑ | ↑ | → | ↑ | → |
| ↑ | ↑ | ↓ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ↑ |

with passenger:
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ← |
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ↑ |
```

(a) Vanilla OC Option 1

```
no passenger:
| P | ↓ | ↓ | ↑ | ↓ |
| ↓ | ↓ | ↓ | ← | ↑ |
| ↓ | → | → | ← | ↑ |
| ↓ | ↑ | ← | ↑ | ↑ |
| P | ↑ | ↑ | → | → |

with passenger:
| D | D | D | D | D |
| D | D | D | D | D |
| D | → | → | D | D |
| D | ↑ | D | D | D |
| D | D | D | D | D |
```

(b) Vanilla OC Option 2

```
no passenger:
| → | ↓ | D | → | ↓ |
| ↓ | ↓ | ↓ | → | ↓ |
| ← | ← | ← | ← | ← |
| ↑ | ↑ | ↑ | ↑ | ← |
| ↑ | → | ↑ | ↑ | ↑ |

with passenger:
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ← | ← | ← | ← |
| ↑ | ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ | ↑ |
```

(c) OC-termination 3.5 Option 1

```
| ↓ | ↑ | → | ↑ | ↓ |
| ← | ← | ↓ | ↓ | ↓ |
| ↑ | ↑ | → | ↓ | ↓ |
| ↑ | ↑ | ↓ | ↓ | ← |
| ← | → | → | P | ↑ |

with passenger:
| D | ← | ← | ← | ← |
| ← | ← | ← | ← | ← |
| ↑ | ← | → | ← | ← |
| ← | ↑ | ← | ← | ← |
| ← | → | ← | P | ← |
```

(d) OC-termination 3.5 Option 2

Figure 6: Comparison of options learned using vanilla option-critic [(a), (b)] and a termination cost of 3.5 [(c), (d)]. Each policy is divided into two subpolicies depending on whether the passenger is in the taxi, each of which consists of 25 grid cells corresponding to the 25 positions in the Taxi environment. In each cell, the agent can either move in one of four cardinal directions or perform the pick-up (P) or drop-off (D) actions.

- Learning rate: 0.05
- Number of options: 10
- Seed: 0 and 10
- Number of episodes: 10,000
- Maximum episode length: 200

More details can be found in our codebase.

9.3 Visualization of Learned Options

Figure 6 shows a comparison of policies learned with vanilla option-critic, and option-critic along with termination penalties. As alluded to in 6.2, we believe option-critic often learns policies that do not support temporal extension. This can easily be seen in sections of (a) and (b), wherein the same action is executed in every cell. Even in sections of these policies where this is not the case, one is not able to follow the policy for long without getting stuck in a loop.

We observe much better temporal abstraction in (c) and (d). For example, when option (c) has a passenger, it can be interpreted as taking the Taxi to the top of the environment. Similarly, (d) seems to first take the cab to a pick-up location in the bottom right, picks the passenger up, and then takes them to the drop-off location in the top left (although not perfectly). This observation is consistent with the conclusion that termination costs enable temporal extension.