A Flexible Approach to Deliberation Cost in the Option-Critic Architecture

Augusto Leal
Universidade do Vale do Rio dos Sinos
Porto Alegre, Brazil
aaleal@edu.unisinos.br

Mateus Begnini Melchiades
Universidade do Vale do Rio dos Sinos
Porto Alegre, Brazil
mateusbme@edu.unisinos.br

Gabriel de Oliveira Ramos
Universidade do Vale do Rio dos Sinos
São Leopoldo, Brazil
gdoramos@unisinos.br

ABSTRACT

Temporal abstraction, frequently modeled using the options framework, enables agents to perform temporally extended actions, optimizing intrinsic policies, termination functions, and policies over options without the need for assigning extra rewards. In this context, the deliberation cost emerges as a crucial component, as it penalizes the premature termination of options, promoting more efficient use of computational resources and accelerating the agent's response in dynamic environments. We propose a flexible and adaptable approach to the deliberation cost, dynamically adjusting it based on the termination decisions of the options. Our results indicate that this approach not only improves learning efficiency but also contributes to the specialization and effectiveness of the options, enabling superior performance.

KEYWORDS

Reinforcement Learning, Temporal Abstraction, Options Framework, Flexible Deliberation Cost

1 INTRODUCTION

Reinforcement learning stands as one of the pillars of artificial intelligence, aiming to establish a mapping from states to actions [12]. This is achieved through trial-and-error processes, utilizing the concept of increasing a numerical reward derived from these attempts. When considering that actions may have consequences that are not necessarily immediate and that future situations can be affected and alter the outcome of the desired reward, it becomes crucial for the agent to understand different time scales.

One way to describe different temporal scales involves the concept of temporal abstraction. A challenge associated with reinforcement learning in the context of temporal abstraction is how an agent can autonomously discover these abstractions. Among the possible approaches, the options framework serves as a model for representing and learning temporally extended actions [14]. By using the options model alongside the simultaneous learning of intra-option policies (actions), termination functions, and policies over options, it became possible to achieve results without the need to establish sub-goals, additional rewards, or other incentives and resources.

Related to option-based learning, the deliberation cost [4] represents an opportunity to enhance both computational efficiency and the results obtained. By penalizing the premature termination of options, computational cost is reduced, as the temporal structure of these options allows for a more efficient allocation of computational resources, while also promoting faster decision-making.

In this study, we propose a flexible and adaptive approach to the deliberation cost, adjusting it dynamically based on the duration of an option rather than relying on a fixed penalty value. Unlike previous approaches that either set deliberation costs arbitrarily or required additional neural networks to regulate them, our method introduces a cost that decreases over time as the agent remains within the same option. This encourages stability by discouraging excessive option switching while still allowing the agent to adapt when necessary. To implement this, we modified the option-critic architecture by incorporating a deliberation cost term inversely proportional to the time spent in a given option. These modifications not only eliminate the challenge of tuning a sensitive hyper-parameter but also enhance computational efficiency. Experimental results demonstrate that this approach enables effective training without fixed hyper-parameters, leading to better option specialization and preventing their degeneration into primitive actions. This is evidenced by the superior rewards achieved when comparing the original Option-Critic Architecture to the version with flexible deliberation cost. In the MuJoCo Ant environment, rewards increased significantly. Moreover, our method enhances learning performance, as evidenced by increased rewards during training, while preserving a structured and interpretable decision-making process based on options.

The main contributions of this paper can be summarized as:

- We introduce a flexible deliberation cost operator. In particular, we define the penalty factor as a function of the time spent on a given option. As a consequence, our approach becomes dynamic and prevents option degeneration.
- We modify the option-critic architecture to incorporate the dynamic deliberation cost, adapting the option termination function to account for the new penalty mechanism.
- We conduct an experimental evaluation in both discrete and continuous action spaces, demonstrating that the flexible deliberation cost increases rewards during training, promotes option specialization, prevents degeneration into primitive actions across diverse environments, and in several cases leads to more stable learning dynamics, as reflected by reduced variance across runs.

This paper is organized as follows. Section 2 presents the basic concepts upon which this work was elaborated. Section 3 overviews the related literature. Section 4 introduces our flexible approach to the deliberation cost. The experimental evaluation is described and discussed in Section 5. Concluding remarks are then presented in Section 6.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2025), Avalos, Aydeniz, Müller, Mohammedalamen (eds.), May 19 – 20, 2025, Detroit, Michigan, USA, ala-workshop.github.io. 2025.

2 BACKGROUND

In this chapter, we present fundamental concepts of reinforcement learning, including Markov Decision Processes (MDPs) and policy gradients, as well as the structure of hierarchical learning based on options.

2.1 Markov Decision Processes

When considering a Markov Decision Process (MDP), we have a classical formalization for decision-making in which actions influence not only immediate rewards but also future states and their corresponding rewards [12]. While a Markov Process (MP) can be defined by the tuple $(\mathcal{S}, \mathcal{P})$, and a Markov Reward Process (MRP) by $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$, an MDP is characterized by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, which includes the additional component $\mathcal{A}$, representing a finite set of actions $(a_1, a_2, \ldots)$.[1] [3]

MDPs serve as modeling tools for sequential decision-making problems, where an agent interacts with the system in a sequential manner [15]. Considering a finite MDP, the random variables R_t and S_t are defined by discrete probability distributions that depend only on the previous state and action. Thus, for a given value of these random variables $s' \in \mathcal{S}$ and $r \in \mathcal{R}$, there is a probability that these values will occur at time t, given the preceding state and action:

$$p(s_0, r \mid s, a) \doteq \Pr\{S_t = s_0, R_t = r \mid S_{t-1} = s, A_{t-1} = a\}, \qquad (1)$$

for all $s', s \in \mathcal{S}$, $r \in \mathcal{R}$, and $a \in \mathcal{A}(s)$. In the notation above the equality sign ($\doteq$) serves as a reminder that this is more of a definition of the function p rather than a consequence of previous definitions [12].

Given their abstraction and flexibility, MDPs can be applied to various problems and in different ways, framing goal-directed learning as an interaction process that reduces the problem to three signals exchanged between the agent and the environment: actions, states, and rewards [12].

2.2 Policy Gradients

Methods that use policy gradients do not necessarily depend on action-value methods. While these methods learn the values of actions and make decisions based on these estimates, methods that use policy gradients learn through a parameterized policy and, depending on the algorithm, make decisions without relying on a value function.

According to Sutton and Barto [12], the representation of policy parameter vectors is given by $\theta \in \mathbb{R}^{d'}$, where the probability of a specific action a being taken at time t when a state s is given at time t with parameter θ is represented by the equation:

$$\pi(a|s, \theta) = Pr\{A_t = a|S_t = s, \theta_t = \theta\} \qquad (2)$$

Assuming $J(\theta)$ is a scalar measure of performance on which the learning of the policy parameter is based, the methods aim to maximize performance by updating a gradient ascent in J [12].

Proposed for stochastic policies by Sutton et al. [13], the policy gradient theorem provides a way to compute the gradient of performance $p(\pi)$ with respect to the parameters θ, so that the gradient of the average reward concerning the policy parameters can be calculated by summing the contribution of each state s and action a, weighted by the probability of visiting state s and taking action a.

The policy gradient theorem provides a theoretical foundation for all policy gradient-based methods [12], as it offers an exact formula for how a model's performance is affected by the policy parameter without involving derivatives associated with the state distribution.

2.3 Options Framework

The model based on options formalizes the concept associated with temporally extended actions. By proposing the concept of options as a reinforcement learning model involving temporal abstraction, Sutton et al. [14] emphasize that time is used to generalize the concept of actions, but they are not to be confused with actions themselves, which are formally considered primitive choices.

MDPs, as seen in Subsection 2.1, are based on discrete actions where a single action at time t affects a state and a reward at time $t+1$. This formulation prevents actions from persisting over a variable period of time, making it difficult for MDPs to take advantage of the benefits that would arise from higher levels of temporal abstraction.

To enable the application of temporal abstraction, an option $\omega \in \Omega$ is defined as a tuple consisting of three components $(I_\omega, \pi_\omega, \beta_\omega)$, where $I_\omega \subseteq \mathcal{S}$ is the initiation set of states, $\pi_\omega : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the intra-option policy, and $\beta_\omega : \mathcal{S} \to [0, 1]$ is the termination function with a stochastic terminal condition following a Bernoulli distribution. An option ω is available in a state s if and only if $s \in \mathcal{I}_\omega$. An agent selects an option using a policy over options and observes it until its termination, at which point a new policy over options is selected. Once a specific option ω is selected, actions are chosen according to π_ω until the option terminates [14][3].

The SMDP (*semi-Markov decision process*) is an option-to-option learning method defined by Theorem 1 proposed by Sutton et al. [14]: given an MDP and a set of options, the decision-making process that selects among the options and executes them until completion is an SMDP. An SMDP is an MDP with an additional element $\mathcal{F}$, $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{F}$ [14], where $\mathcal{F}(t|s, a)$ represents[2] the transition probability for time t, given a state s and an action a. It is noteworthy that from an SMDP, one can derive an optimal value function over options $V_\Omega(s)$ and an optimal option-value function $Q_\Omega(s, \omega)$.

In addition to the approach involving a hybrid strategy between options and primitive actions, Hauskrecht et al. [6] proposed the concept of *options*[3] as a hierarchical solution to MDPs. According to the authors, options are treated as local policies that operate within specific regions of the state space, while an abstract MDP considers only the states at the boundaries of these regions. This abstraction significantly reduces the effective size of the state space, simplifying the overall solution process. The authors developed a framework to generate transition models for options in MDPs based on two concepts. First, they introduce the *augmented MDP*, denoted M_a, where the original action space $\mathcal{A}$ is expanded to include a set of available options $O = \{A_1, A_2, \ldots, A_n\}$, resulting in an extended

[1]According to [15], an MDP is called finite when the set of states $\mathcal{S}$ and the set of actions $\mathcal{A}$ are finite. Similarly, [12] express this, although they add the finiteness of $\mathcal{R}$ as a criterion.

[2]The notation follows the formalization presented by Ding et al. [3].

[3]The authors use the term "macro-action," but the terminology of options is retained here, as it refers to temporally extended or abstract actions.

action space $\mathcal{A} \cup O$. This augmented MDP can then be solved using traditional methods such as value iteration, where each option in O is treated as a high-level action with its own transition and reward model. Second, they define a *reduced MDP*, denoted M_r, in which the primitive actions $\mathcal{A}$ are replaced entirely by the set of options O. While M_a prioritizes optimality by retaining all primitive actions and adding options, M_r sacrifices optimality in favor of flexibility.

The model proposed by Hauskrecht et al. [6] considers a hierarchical approach for solving MDPs, abstracting the original problem into a reduced system of options and peripheral states of regions. The goal is to reduce computational complexity without disregarding essential components required for a convergent solution.

3 RELATED WORK

The state of the art in learning with options is marked by the development of the option-critic architecture by Bacon et al. [1]. This study focuses on temporal abstraction in reinforcement learning, allowing the agent to learn at different temporal scales more efficiently regarding scalability, temporality, and data volume than traditional methods that find subgoals and learn policies to reach them. The approach integrates the discovery and learning of options with policy gradient theorems, enabling simultaneous learning of intra-option policies and termination functions alongside an overall policy. It is applicable to both discrete and continuous state and action spaces and supports linear and non-linear function approximators.

A continuous perspective is adopted for learning options, where experiences are continuously integrated into each system component, including the value function, the policy over options, intra-option policies, and termination functions [1]. The Intra-Option Policy Gradient Theorem states that, given a set of Markov options with stochastic intra-option policies differentiable with respect to their parameters θ, the gradient of the expected discounted return with respect to (s_0, ω_0) is defined as:

$$\sum_{s,\omega} \mu_\Omega(s, \omega | s_0, \omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a). \tag{3}$$

Here, $\sum_{s,\omega}$ indicates the summation over all possible states s and options ω, with $\mu_\Omega(s, \omega | s_0, \omega_0)$ representing the discounted weighting of state-option pairs. The gradient $\frac{\partial \pi_{\omega,\phi}(a|s)}{\partial \theta}$ captures the derivative of the policy with respect to parameters, while $Q_U(s, \omega, a)$ is the action-value function estimating the expected return of action a within option ω from state s. This theorem enables adjustments to intra-option policies and termination functions by maximizing returns, facilitating continuous learning, in contrast to subgoal models that do not propagate to the overall objective.

The second theorem introduced by Bacon et al. [1], known as the Termination Gradient Theorem, states that the gradient of the expected discounted return with respect to the termination function parameters ϑ, given an initial state-option pair (s_1, ω_0), is given by:

$$-\sum_{s',\omega} \mu_\Omega(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega), \tag{4}$$

where $\sum_{s',\omega}$ is the sum over all possible states s' and options ω, $\mu_\Omega(s', \omega | s_1, \omega_0)$ is the discounted weighting of state-option pairs, which calculates the importance of the pair s', ω along trajectories starting from (s_1, ω_0), $\frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta}$ represents the gradient of the termination function $\beta_\omega(s')$, and $A_\Omega(s', \omega)$ is the advantage function. Thus, the advantage function is a direct consequence of the derivation and is intuitively interpreted such that if an option is not the best possible, the gradient is adjusted to increase its termination probability.

The second theorem of the option-critic architecture adjusts the termination probability of options based on their advantage over the default policy, leading to their degeneration into primitive actions during training. To address this, Harb et al. [4] introduce the deliberation cost as a regularizer in the advantage function, promoting longer options and reflecting the costs of decision-making and computation.

The deliberation cost enables efficient planning and execution by allowing the agent to amortize this cost over prolonged option use, aiming to maximize return while minimizing frequent option switches. By penalizing the agent for interrupting options, the method encourages more continuous behavior. Additionally, Harb et al. [4] employ Lagrangian optimization to model the MDP with adjusted rewards, where the deliberation cost reduces reward values upon switching options. A margin η is used to adjust the advantage function, helping the agent navigate uncertainty. Increasing η reduces the advantage difference, thus favoring longer option retention over frequent terminations.

Following the formulation proposed by Harb et al. [4], the termination gradient shown in Equation 4, which incorporates a deliberation cost η, is given by:

$$-\sum_{s',\omega} \mu_\Omega(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} \left(A_\Omega(s', \omega) + \eta \right), \tag{5}$$

Here, η represents the deliberation cost and introduces a margin that determines how advantageous an option must be to justify its continuation. A larger value of η reduces the likelihood of premature terminations by promoting greater persistence in the current option. This, in turn, encourages more stable usage of options and helps mitigate the effects of noisy or uncertain value estimates during training.

Building on the study by Harb et al. [4], Klissarov et al. [9] enhance learning with temporally extended actions through the option-critic architecture and deliberation cost, integrating proximal policy optimization. Their tests in continuous environments show that adding the deliberation cost accelerates learning, although its performance is not directly proportional to the parameter η. They suggest exploring adaptive adjustments to the deliberation cost based on environmental contexts. Furthermore, they discuss the importance of option usage, proposing initiation sets to determine when and how many options to employ.

Alongside the previous study, various adaptations to the option-critic architecture have emerged. For instance, Harutyunyan et al. [5] reformulate the neural network architecture to define termination objectives in a reward-independent manner, guided by an informational criterion for option compression. Their framework, termed the termination critic, optimizes termination by connecting the gradient of the option transition model with the termination gradient, preventing uniform behavior across options even when trained on the same reward. Other studies include the simultaneous updating of multiple options via importance sampling [10], the use of interest functions with generalized option initiation sets for interpretable and reusable options in multi-task learning [8], and the safe option-critic architecture, which emphasizes controllability and safe exploration in uncertain environments, facilitating learning about variability within complex state spaces using non-linear function approximation [7].

In the option-critic architecture, as considered in its original model, all options are deemed available in every state. Theoretically, the agent is allowed to select any option, irrespective of the context or how this equation adheres to the current state. In complex environments, this approach becomes a challenge and may lead to infeasibility due to the computational and performance capacities required, as the agent is permitted to waste time evaluating options that would be considered irrelevant. Furthermore, the unrestricted availability of options may lead to interference between options and result in suboptimal behaviors.

The choice of the deliberation cost, which adds a penalty to training when there is excessive switching between options, encourages the agent to maintain options for a longer period and reduce changes that are deemed unnecessary. With the adoption of the deliberation cost, the agent's training becomes significantly dependent on the selected cost hyperparameter. If a very high value is chosen, the agent avoids switching options even when necessary, compromising learning. Conversely, if the parameter is set too low, switches will be frequent, and learning will revert to the problem of the option-critic architecture nullifying the benefits of learning with options.

4 PROPOSED METHOD

In the previously mentioned study [4], the deliberation cost operates with a fixed penalty value, which should be considered since, as addressed by Harutyunyan et al. [5], training the agent presents challenges due to the high sensitivity of the hyperparameter being adopted. Seeking to propose an alternative to fixing a value for the deliberation cost, this study proposes the adoption of a flexible deliberation cost.

In our approach, the penalty for the deliberation cost ceases to depend on an arbitrary definition or an additional neural network and becomes dynamically adjusted according to the time spent in a given option. Thus, the penalty is inversely proportional to the duration of the option, providing an incentive for stability, as the penalty decreases over time due to remaining in the same option. The agent is directed to avoid excessive option changes, even though it can make changes when necessary. Additionally, there is computational efficiency in the algorithm, as no additional neural networks are required.

The changes highlighted in red in Algorithm 1 reflect on the option-critic architecture concerning the deliberation cost. Firstly, by replacing the fixed cost with a dynamic value, the deliberation cost η is adjusted based on the duration of the option κ. This change allows the agent to have greater flexibility and incentive to remain in an option longer, reducing unnecessary penalties. Furthermore, the inclusion of a continuous adjustment mechanism for the deliberation cost based on the agent's decisions over options $\frac{1}{1+\kappa}$ enables a more refined customization of the learning process. This not only

Algorithm 1: Option-Critic Architecture with Flexible Deliberation Cost

```
1  $s \leftarrow s_0$;
2  Choose $\omega$ according to an $\epsilon$-soft policy over options $\pi_\Omega(s)$;
3  repeat
4      Choose $a$ according to $\pi_{\omega,\theta}(a \mid s)$;
5      Take action $a$ in state $s$, observe $s'$, $r$;
6      1. Option Evaluation;
7      $\delta \leftarrow r - Q_U(s, \omega, a)$;
8      if $s'$ is not terminal then
9          $\delta \leftarrow \delta + \gamma(1 - \beta_{\omega,\vartheta}(s'))Q_\Omega(s', \omega) + \gamma\beta_{\omega,\vartheta}(s') \max_{\tilde{\omega}} Q_\Omega(s', \tilde{\omega})$;
10     end
11     $Q_U(s, \omega, a) \leftarrow Q_U(s, \omega, a) + \alpha\delta$;
12     2. Option Improvement;
13     $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a)$
14     if $\beta_{\omega,\vartheta}$ terminates at $s'$ then
15         if $\omega \neq \omega_{previous}$ then
16             $\kappa \leftarrow 0$;
17         else
18             $\kappa \leftarrow \kappa + 1$;
19         end
20         $\eta \leftarrow \frac{1}{1+\kappa}$;
21         Choose a new $\omega$ according to $\epsilon$-soft $(\pi_\Omega(s'))$;
22     end
23     $\omega_{\text{previous}} \leftarrow \omega$ ;
24     $s \leftarrow s'$;
25     $\vartheta \leftarrow \vartheta + \alpha_\vartheta \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} (Q_\Omega(s', \omega) - V_\Omega(s') + \eta)$;
26 until $s'$ is terminal;
```

improves the computational efficiency of the algorithm but also ensures that the agent can adapt more quickly to the environmental conditions. These changes aim to maximize performance and stability in learning over time.

The introduction of the variable κ, which counts the number of times the agent switches options, allows for even more refined adaptation to the environment. If the agent frequently changes between options, the value of η adjusts, increasing the cost of switching and promoting stability. Even in situations where multiple options are available, the agent learns how to identify advantageous strategies over time, which leads to superior overall performance.

It is important to consider that the modifications aim not only to maximize reward outcomes but also to ensure the stability of learning. A primary intuition behind the implementation of the flexible deliberation cost is to promote efficient learning that favors the specialization of options, encouraging exploration of the benefits of decisions while avoiding abrupt changes that could fragment learning or make it ineffective.

For example, consider a robot designed to navigate a room with a wall blocking its path. If the deliberation cost is fixed and set too large, it might lead the robot to stick to a suboptimal option. If the cost is set too small, the robot could frequently switch between options, potentially resulting in inefficient movements and collisions

Table 1: Comparative Results of Option-Critic and Option-Critic with Flexible Deliberation Cost

Environment	*Original*	*Flexible Deliberation Cost*
Cart Pole	328.586	**415.418**
Lunar Lander	191.286	**225.70**
MuJoCo Ant	680.717	**1287.45**
MuJoCo Half Cheetah	1738.108	**2001.45**

with the wall. However, with a dynamic deliberation cost, when the robot opts to go around the wall, the penalty for staying in that option would decrease over time. This might encourage the robot to explore thoroughly for the best route to bypass the wall before deciding to change, ultimately allowing it to find the most efficient path and reducing decision-making time with fewer errors.

In the algorithm, this is reflected when the termination condition is met. If the option $\beta_{\omega,\vartheta}$ terminates at state s', the algorithm checks whether the current option ω is the same as the previous one. If it is not, the variable κ resets to zero; if it is, κ increments, which in turn adjusts the deliberation cost η as $\eta = \frac{1}{1+\kappa}$. This dynamic adjustment allows the robot to stay in a successful option longer, promoting stability in its decision-making process. After evaluating the termination conditions, the robot then chooses a new option according to an ϵ-soft policy, ensuring it balances exploration and exploitation effectively.

5 EXPERIMENTS

The use of a flexible deliberation cost, inversely proportional to the duration of options, was grounded in the pursuit of computational efficiency and improvements in training specialized options, without relying on a fixed and highly sensitive hyperparameter. We aimed to reduce excessive deliberation on subsequent options, stabilize the training process, and provide robust results regarding the permanence of options, as well as to promote an increase in rewards by favoring longer and more specialized options.

The experiments focus on modifying the option-critic architecture to evaluate the impact of flexible deliberation costs. To establish a solid foundation, the approaches of Bacon et al. [1] and Harb et al. [4] were considered as baselines, along with other relevant research and advancements in the field, since they represent a structural component of the algorithm's development for future studies. This choice is also justified as an alternative to the additional neural network proposed by Harutyunyan et al. [5]. Additionally, in the study by Klissarov et al. [9], the use of PPO within the option-critic architecture was based on fixed deliberation costs in continuous action environments. This context provided an opportunity to reconsider this approach, as well as the studies mentioned in [10], [7], and [8], and to explore alternatives to fixed deliberation costs while evaluating their potential benefits and limitations.

Our evaluation focuses on three key aspects: the termination of options, the obtained rewards, and the average option alternation. We aim to demonstrate that the use of a flexible deliberation cost leads to higher rewards by fostering the development of longer options. Additionally, our analysis of option terminations provides

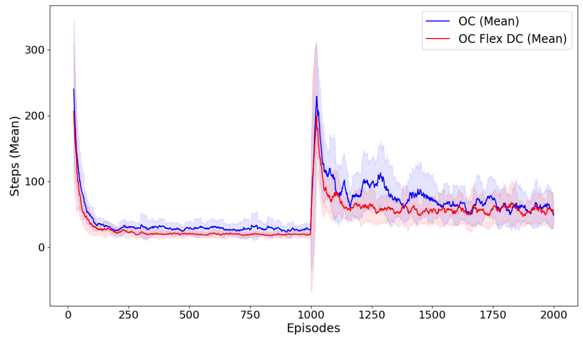

Figure 1: Recovery after goal change in Four-Room

evidence supporting the intuition of specialization, while the reduction in the average frequency of option alternation highlights the stability and efficiency gains achieved with this approach.

5.1 Methodology

Our experiments were conducted in three distinct domains: tabular, discrete actions, and continuous actions. In all cases, the results were satisfactory and promising regarding the adopted approach. The tests in the tabular domain were conducted in the Four-Rooms environment, as proposed by Sutton et al. [14]. However, unlike the authors' approach, the options were independently discovered by the agent through training focused on minimizing the number of steps to reach the goal. Two different learning rates were employed for the option termination function, with 10 executions carried out for each rate. After 1000 steps, the goal was changed to a new random location.

The flexible deliberation cost approach was tested in a discrete action space, involving two environments: Cart Pole and Lunar Lander. The Cart Pole environment, is based on the problem described by [2] and represents a pendulum attached by a joint to a small cart, with the objective of balancing the pendulum by moving left and right. The tests were conducted with different seeds for a total of 10 runs, considering both the original architecture and the modifications with the flexible deliberation cost. The training was performed with a learning rate of 1×10^{-4}, 1700 episodes and a maximum of 500 steps per episode.

In the Lunar Lander environment, which represents a rocket trajectory optimization challenge, the main objective is to safely land the spacecraft on a landing pad. The discrete action space of the environment is based on whether the engine is on or off, with the possibility of activating the left, right, or main engine, as well as the action of doing nothing. The training was conducted 10 times with different seeds, a learning rate of 7×10^{-4}, 2000 episodes and a maximum of 1000 steps per episode.

For continuous action spaces, two environments from MuJoCo (Multi-Joint dynamics with Contact) were used, a physics engine with simulations characterized by precision and aimed at scenarios involving physical contacts between robots and the environment. Since these are continuous actions, the algorithm was adapted to operate with Gaussian distribution policies. In the Ant environment, Based on the problem developed by [11], a quadruped robot consists

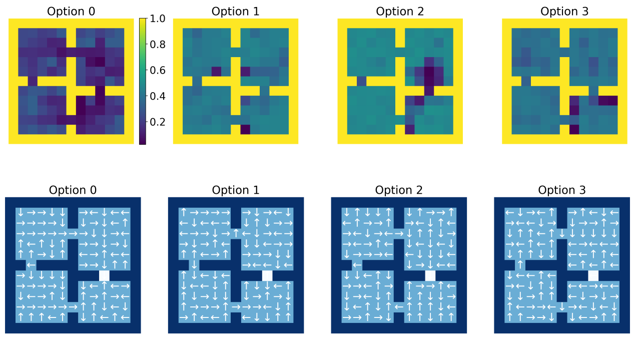

Figure 2: (Top) Termination map for each option in the Four-Room environment, where the walls are highlighted in bright yellow, and lighter colors indicate higher probabilities of termination. (Bottom) Agent's trajectory in the Four-Room environment, illustrating the paths taken for each option and the complementarity between options 0, 1, and 2.

of a torso (free rotational body) with four legs, each divided into two parts. The goal is to coordinate the four legs to move forward (right) by applying torque to the eight joints that connect the two parts of each leg to the torso, totaling nine body parts and eight joints. Tests were conducted over 4000 episodes, repeated 10 times with different seeds, using a learning rate of 1×10^{-4}, and a maximum of 1000 steps per episode.

In another tested environment, Half Cheetah, following the problem developed by [16], the goal is to apply torque to 6 of the 8 connected joints of the robot, making it move, with the reward associated with the distance traveled. Learning tests were performed 10 times with different seeds, considering 1000 episodes, with 1000 steps per episode and a learning rate of 1×10^{-4}.

5.2 Numerical Results

With the tests conducted in three different types of environments, it was possible to verify that the results involving the adoption of the flexible deliberation cost achieved higher rewards throughout the training process. The results presented in Table 1 illustrate a comparative analysis of the original Option-Critic algorithm and the Option-Critic algorithm with a flexible deliberation cost across various environments. The average rewards achieved during training highlight a marked improvement with our flexible deliberation cost approach. In each environment, the flexible deliberation cost not only enhanced the overall performance of the agent but also demonstrated its effectiveness in promoting better decision-making strategies. The data indicate that this approach allows agents to adapt more effectively to the specific challenges of each environment, leading to improved learning efficiency and specialization of options under varying conditions. These findings suggest that employing a flexible deliberation cost can significantly optimize the learning process, resulting in higher rewards and better performance across diverse tasks.

Considering Four-Room domain, in addition to demonstrating better performance in the first 1000 steps, after the goal change, the learning with flexible deliberation cost showed a faster recovery

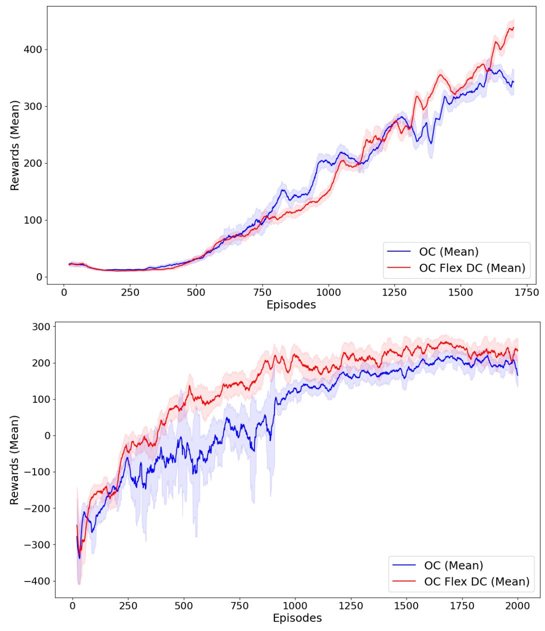

Figure 3: Comparative learning curves for the Cart Pole (top) and Lunar Lander (bottom) environments. The graphs illustrate the performance of the learning algorithms over training episodes, highlighting the differences in efficiency and effectiveness in each environment.

than the original architecture, which can be seen in Figure 1, where the steps needed to reach the goal in each episode are visualized on the vertical axis.

In the evaluation of the termination regions of options in Figure 2, where the goal is located at the rightmost door, represented as a blank quarter, it is interesting to observe how the termination of options in the environment trained with a flexible deliberation cost shows a higher probability of completion when deviating from the specific route defined in option 0. The walls are highlighted in yellow, while lighter colors indicate higher termination probabilities. The white door in the figure represents a goal that has been reached, symbolizing a subgoal the agent needs to achieve to progress toward the final objective. The variation in routes, along with the adjustments made by options 1, 2, or 3 depending on the agent's position, reinforces the intuition that relevant subgoals can emerge in the learning process, allowing the agent to develop more effective strategies for navigating the environment.

In domains with discrete, Figure 3 and continuous action spaces, Figure 4, the application of a flexible deliberation cost resulted in superior performance compared to the original architecture. The highlight is in continuous domains, especially in the MuJoCo Ant environment, where this approach proved advantageous. In this scenario, the agent's need to learn sequences of coordinated and efficient movements is crucial, and maintaining an option for a longer period seems to favors the execution of more stable and natural locomotion patterns.

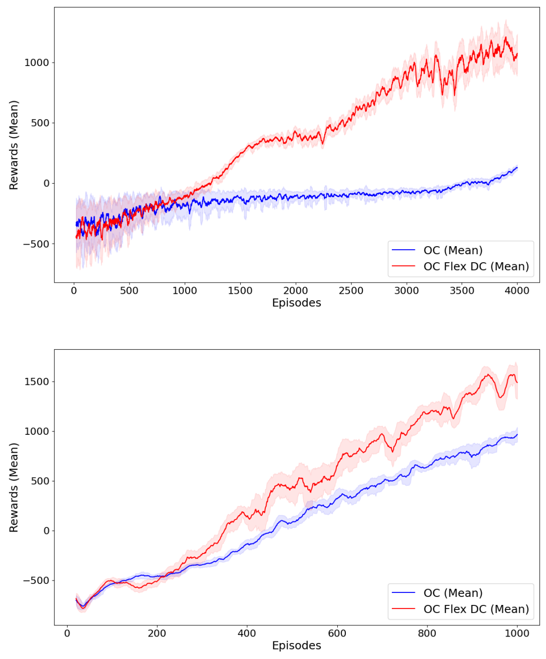

Figure 4: Comparative learning curves for the MuJoCo Ant (top) and MuJoCo Half Cheetah (bottom) environments, illustrating the performance across training episodes. The curves highlight the differences in learning efficiency and effectiveness for each environment.

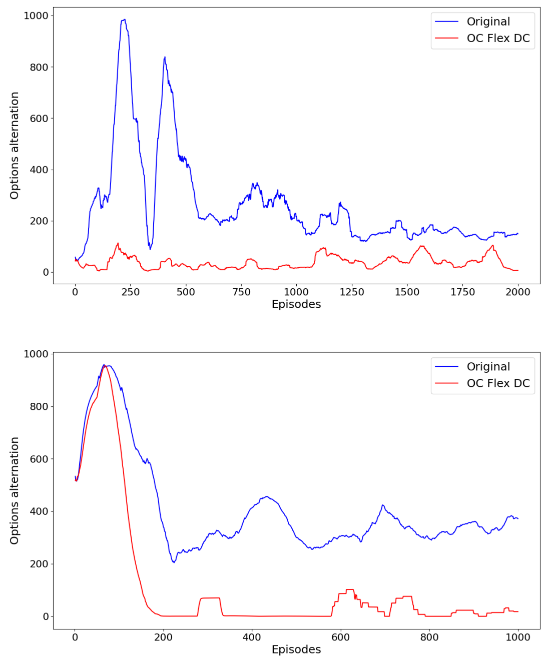

Figure 5: Option changes per episode for the Lunar Lander (top) and MuJoCo Half Cheetah (bottom) environments. The graphs illustrate the frequency of option alternation throughout training, highlighting the decision-making dynamics in each environment.

The reduction in option alternation, as can be seen in Figure 5 when using the flexible deliberation cost, also demonstrates the associated computational efficiency, as it eliminates the need to consume resources for deliberating on the policy over options.

The results obtained demonstrate that it is possible to train agents using options and deliberation cost without the need to apply a fixed hyperparameter. The flexibility of the deliberation cost, incorporated into the original architecture of the algorithm, favored the agent's learning, allowing for the specialization of options and eliminating additional computational costs that could arise from deliberating on the choice of new options. The advantages of this approach are evidenced by the increase in rewards during training and the preservation of options, preventing their degeneration into primitive actions.

A potential problem arising from the inversely proportional condition to the duration of the options relates to abrupt changes that may occur due to a rapid increase in the persistence of a single option, leading the agent to become overly dependent on that option. This condition can be mitigated by smoothing the increment of the value of κ, as defined in Algorithm 1. In this regard, it is possible to test approaches such as moving averages or consider their application in conjunction with return values.

6 CONCLUSIONS

The introduction of a flexible and adaptable deliberation cost, inversely proportional to the duration of options in reinforcement learning based on options, has shown significant and promising results, ensuring both efficiency and performance. By making the deliberation cost dynamic and proportional to the termination decisions of options, limitations associated with fixed hyperparameters are overcome, promoting a more flexible approach. The obtained results demonstrate that this strategy not only improves the stability of the option-critic architecture but also mitigates the degeneration of options into primitive actions throughout training.

Despite the advances provided by the proposed approach, a limitation concerns the possibility of abrupt changes in the deliberation cost due to a sudden increase in the duration of a specific option, making the agent overly dependent on it. To mitigate this issue, smoothing strategies such as applying a moving average or incorporating return values into the adjustment of κ can be explored. Additionally, the generalization of the approach to different types of environments still requires further investigation to ensure its effectiveness in more diverse and complex contexts. It is also essential to assess when the use of the options framework is truly justified, considering scenarios in which temporal abstraction provides concrete benefits in terms of performance and computational efficiency.

Future work includes exploring multiple deliberation cost hyperparameters combined via averaging, and testing alternative formulations beyond inverse proportionality to control how the cost evolves with option duration.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. This research was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (grants 443184/2023-2, 313845/2023-9, and 445238/2024-0).

REFERENCES

[1] Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The Option-Critic Architecture. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (Feb. 2017). https://doi.org/10.1609/aaai.v31i1.10916

[2] A. G. Barto, R. S. Sutton, and C. W. Anderson. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13, 5 (1983), 834–846. https://doi.org/10.1109/tsmc.1983.6313077

[3] Zihan Ding, Yanhua Huang, Hang Yuan, and Hao Dong. 2020. Introduction to Reinforcement Learning. In *Deep Reinforcement Learning: Fundamentals, Research and Applications*, Hao Dong, Zihan Ding, and Shanghang Zhang (Eds.). Springer Singapore, 47–123. https://doi.org/10.1007/978-981-15-4095-0

[4] Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. 2018. When Waiting Is Not an Option: Learning Options With a Deliberation Cost. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://doi.org/10.1609/aaai.v32i1.11831

[5] Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Remi Munos, and Doina Precup. 2019. The Termination Critic. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 2231–2240. https://proceedings.mlr.press/v89/harutyunyan19a.html

[6] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas L. Dean, and Craig Boutilier. 2013. Hierarchical Solution of Markov Decision Processes using Macro-actions. *CoRR* abs/1301.7381 (2013). arXiv:1301.7381 http://arxiv.org/abs/1301.7381

[7] Arushi Jain, Khimya Khetarpal, and Doina Precup. 2018. Safe Option-Critic: Learning Safety in the Option-Critic Architecture. *CoRR* abs/1807.08060 (2018). arXiv:1807.08060 http://arxiv.org/abs/1807.08060

[8] Khimya Khetarpal, Martin Klissarov, Maxime Chevalier-Boisvert, Pierre-Luc Bacon, and Doina Precup. 2020. Options of Interest: Temporal Abstraction with Interest Functions. *CoRR* abs/2001.00271 (2020). arXiv:2001.00271 http://arxiv.org/abs/2001.00271

[9] Martin Klissarov, Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. Learnings Options End-to-End for Continuous Action Tasks. arXiv:1712.00004 [cs.LG] https://arxiv.org/abs/1712.00004

[10] Martin Klissarov and Doina Precup. 2021. Flexible Option Learning. *CoRR* abs/2112.03097 (2021). arXiv:2112.03097 https://arxiv.org/abs/2112.03097

[11] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv:1506.02438 [cs.LG] https://arxiv.org/abs/1506.02438

[12] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). MIT Press, Cambridge, MA.

[13] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems* 12 (1999).

[14] Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112, 1 (1999), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1

[15] Csaba Szepesvári. 2010. *Algorithms for Reinforcement Learning* (1 ed.). Springer Cham. https://doi.org/10.1007/978-3-031-01551-9

[16] Paweł Wawrzyński. 2009. A Cat-Like Robot Real-Time Learning to Run. In *Adaptive and Natural Computing Algorithms*, Mikko Kolehmainen, Pekka Toivanen, and Bartlomiej Beliczynski (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 380–390.