# Human-like Performance on a Foraging Task with Deep Hierarchical Reinforcement Learning using Human Priors

**Anonymous***

**Abstract:** Recent successes of Reinforcement Learning (RL) have managed to accomplish human-level performance in complex sequential decision making tasks with the downside of increasing the required computational effort. This is hindering the use of RL in many applications where humans remain more efficient while maintaining adequate performance. In this work, we develop a method which exploits cognitive human priors and human demonstrations and combines them with deep RL in order to train an agent capable of efficiently matching human performance. We start by collecting data from human trials on a foraging task in a virtual environment and compare humans with a deep Q-learning agent, which is not able to match human performance. To close this gap, we leverage hierarchical imitation learning and incorporate human priors at different levels of the hierarchy. The human-inspired hierarchical policy is then refined with a novel Deep Option-Critic algorithm and eventually manages to reach human-like performance on the foraging task.

## 1 Introduction

Deep Q-learning has led to human-level performance on difficult tasks by combining reinforcement learning with deep neural networks [1]. However, this breakthrough exponentially increases the number of samples necessary for successfully training the agent, limiting its applicability to real-world problems where samples are generally expensive. In the last decade, there has been growing interest in RL among the cognitive science community as a tool to understand the underlying mechanisms of human learning [2]. Motivated by both cognitive studies [3, 4, 5] and the need for more efficient algorithms, classical RL has been further developed to foster hierarchical structures where decisions at different levels of the hierarchy are made based on different temporal abstractions. Recent studies in natural language processing [6, 7] have shown how cognitive and computer science can benefit from each other: the first in using new mathematical models to describe human and animal decision processes, and the second by using neural information to enhance algorithmic efficiency and performance. Motivated by these recent successes, this paper uses a similar approach and combines human behavioral studies with Hierarchical RL (HRL) in an attempt to design algorithms optimized for performance and efficiency.

Here we utilize data from a larger study on human foraging behavior in which human subjects explore a virtual open field environment collecting coin rewards. The locations of rewards vary throughout the environment, and several dense clusters of coins appear in some areas. We use human subjects' trajectories, i.e. the virtual paths they traveled while searching for rewards, to model the decision making process of the subjects with the following questions in mind: How long does each subject stay within a cluster? Do the subjects learn the cluster locations? [8, 9, 10] We exploit then both human demonstrations and cognitive analysis to develop a method based on Hierarchical Imitation Learning (HIL) and deep HRL, which, by incorporating human priors at different levels of the hierarchy, can perform at a level comparable to human agents.

**Related work and contributions:** The idea of incorporating human cognitive priors in the RL training loop with the goal of improving sample efficiency is recent; however, there are a few papers which have shown encouraging results. In particular, the role of human priors in order to improve the

*

performance of deep RL agents on video games is investigated in [11]. Novel exploration techniques which exploit human and animal cognitive data are formalized in [12, 13]. Du and Narasimhan [14] emphasize how incorporating physics priors, in a manner similar to humans, improves Deep RL algorithms. Bourgin and colleagues [15] describe how pre-training neural networks to construct cognitive model priors can provide state of art performance for decision-making predictions, even with a small data set. On the other hand, the literature on HRL and imitation learning (IL) is quite rich and presents a good number of remarkable results [16, 17, 18, 19, 20, 21]. We synthesize this work in the following manner: first, we study the behavior of humans performing the foraging task and extract relevant cognitive priors. Then, using supervised learning, we initialize the highest level of a hierarchical policy with these priors and then we train the pre-initialized policy using an end-to-end HIL approach such as [22, 23]. Note that, end-to-end HIL is usually based on approximately maximizing a log-likelihood function which is non-linear in the policy parameters. The supervised initialization is therefore crucial to enforce a human-inspired structure of the hierarchy which, as we are going to show, improves also the performance. Finally, we formulate a deep hierarchical neural TD(0) algorithm for hierarchical policy evaluation which, together with HIL, returns a hierarchical actor and a critic that are used as initialization of an opportunely modified version of the Option-Critic algorithm in [18] that we rename as Deep Option-Critic (DOC). The DOC then requires a limited amount of iterations to train a policy that can reach human performance. In summary, the main contribution of this paper lies in combining cognitive priors with HIL and HRL in order to obtain human-level performance on a challenging RL task. We test this method on the foraging task and demonstrate its ability to perform similarly to a human expert using a parsimonious amount of training samples.

**Outline:** In Section 2 we formulate the problem, define notation, and present some preliminary material. Section 3 focuses primarily on the deep option-critic algorithm and Section 4 describes the steps we followed to design, test, and experimentally evaluate the various policies. Finally, discussions, limitations of the approach, and conclusions are reported in Section 5.

## 2 Preliminaries

We use uppercase letters (e.g., $S_t$) for random variables, lowercase letters (e.g., $s_t$) for values of random variables, script letters (e.g., $\mathcal{S}$) for sets, and bold lowercase letters (e.g., $\boldsymbol{\theta}$) for vectors. Let $[t_1 : t_2]$ be the set of integers $t$ such that $t_1 \leq t \leq t_2$; we write $S_t$ such that $t_1 \leq t \leq t_2$ as $S_{t_1:t_2}$. We denote by $\mathcal{N}(\mu, \sigma^2)$ the normal distribution, where $\mu$ is the mean and $\sigma$ the standard deviation. Finally, $\mathbb{E}[\cdot]$ represents expectation, $\mathbb{P}(\cdot)$ probability, and $|\mathcal{S}|$ the cardinality of a set.

**Foraging Task:** Foraging is defined as the act of searching for resources in the environment. When resources are organized in clusters, the forager is forced to make decisions about when to search ("exploit") a cluster and when to abandon that cluster to "explore" other cluster options in order to maximize reward intake. Here, we conduct an experiment in which five subjects explore a virtual open field with four differently colored and textured walls while collecting coin rewards (Figure 2a and Figure 2b). Overall, 325 coins are distributed throughout the environment, and 225 of these coins are distributed according to four different Gaussian distributions of varying sizes, specifically: 75 according to $\mathcal{N}((6, 7.5), 0.5^2)$, 40 according to $\mathcal{N}((-1.5, -5.0), 1.1^2)$, 60 according to $\mathcal{N}((-5, 3), 1.8^2)$ and 50 according to $\mathcal{N}((4.9, -4), 1.3^2)$. The remaining 100 coins are randomly distributed (Fig. 2c). Subjects are unaware of the distribution and are only instructed to freely explore and collect coins. The experiment is conducted over two days: on the first day, the participants are familiarized with the environment. On the second day, the five subjects complete ten 8-minute runs of the task, with randomized initial position, collecting on average 243.98 coins. The complete description on how the subjects have been selected, on the experimental procedure and on the task are reported in the supplementary material.

**Reinforcement learning:** We consider an infinite-horizon discounted Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, D, \gamma)$ where $\mathcal{S}$ is the set of states and $\mathcal{A}$ is the set of actions, both possibly infinite. $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta_{\mathcal{S}}$ is the transition probability function and $\Delta_{\mathcal{S}}$ denotes the space of probability distributions over $\mathcal{S}$. The function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ maps rewards to state-action pairs. $D \in \Delta_{\mathcal{S}}$ is the initial state distribution and $\gamma \in [0, 1)$ the discount factor. In classical "flat" RL, the decision agent is modeled as a stationary policy $\pi : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$, where

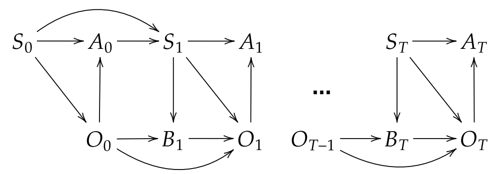


Figure 1: Graphical model for the options framework.

$\pi(a|s)$ is the probability of taking action $a$ in state $s$. We parameterize $\pi$ using a neural network with parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k$ and we write $\pi_{\boldsymbol{\theta}}$. The goal is to find $\boldsymbol{\theta}$ such that the expected total discounted reward $J(\boldsymbol{\theta}) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$ is maximized, where $\tau = (s_0, a_0, s_1, a_1, \dots)$ is sampled according to $s_0 \sim D$, $a_t \sim \pi_{\boldsymbol{\theta}}(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. Note that the index $t$ represents the time. Throughout the paper we use the standard RL definitions: the state-action value function of $\pi_{\boldsymbol{\theta}}$ is denoted as $Q_{\boldsymbol{\theta}}(s,a) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|S_0 = s, A_0 = a]$ and the state value function $V_{\boldsymbol{\theta}}(s) = \mathbb{E}_{a\sim\pi_{\boldsymbol{\theta}}(\cdot|s_t)}[Q_{\boldsymbol{\theta}}(s,a)]$. Finally, we denote the normalized discounted visitation frequency induced by $\pi_{\boldsymbol{\theta}}$ and with $S_0 = s_0$ as $\rho_{\boldsymbol{\theta},s_0}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s; D, \pi_{\boldsymbol{\theta}}, P)$ with $\rho_{\boldsymbol{\theta},s_0} \in \Delta_{\mathcal{S}}$.

**Options framework:** Our HRL algorithm is formulated within the options framework first introduced in [24] and using the probabilistic graphical model of Fig. 1. For brevity, we refer to it as the options probabilistic graphical model (OPGM). Similarly to flat RL, we consider an infinite-horizon discounted MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{B}, P, r, D, \gamma)$ where, in addition to the flat RL MDP we have $\mathcal{O}$ as the finite set of options, and $\mathcal{B} = \{0, 1\}$ as the termination set. The hierarchical policy is based on the triplet of stationary policies $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_b^{\boldsymbol{\theta}_b}\}$; we define $\boldsymbol{\theta} := (\boldsymbol{\theta}_{hi} \in \Theta_{hi}, \boldsymbol{\theta}_{lo} \in \Theta_{lo}, \boldsymbol{\theta}_b \in \Theta_b)$ with $\Theta := (\Theta_{hi} \times \Theta_{lo} \times \Theta_b) \subset \mathbb{R}^k$ and $\pi_{hi}^{\boldsymbol{\theta}_{hi}} : \mathcal{S} \to \Delta_{\mathcal{O}}$ as the high level policy parameterized by $\boldsymbol{\theta}_{hi}$, $\pi_{lo}^{\boldsymbol{\theta}_{lo}} : \mathcal{S} \times \mathcal{O} \to \Delta_{\mathcal{A}}$ the low level policy parameterized by $\boldsymbol{\theta}_{lo}$ and $\pi_b^{\boldsymbol{\theta}_b} : \mathcal{S} \times \mathcal{O} \to \Delta_{\mathcal{B}}$ the termination policy parameterized by $\boldsymbol{\theta}_b$. The hierarchical decision process starts at $t = 0$, where $s_0 \sim D$, $o_0 \sim \pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_0)$, $a_0 \sim \pi_{lo}^{\boldsymbol{\theta}_{lo}}(\cdot|s_0, o_0)$ and $s_1 \sim P(\cdot|s_0, a_0)$. The agent then decides whether to terminate or not the current option $o_0$. This decision is encoded in the termination indicator $b_1 \sim \pi_b^{\boldsymbol{\theta}_b}(\cdot|s_1, o_0)$: if $b_1 = 1$, the option $o_0$ terminates and $o_1$ is sampled according to $\pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_1)$; otherwise, if $b_1 = 0$, the option $o_0$ continues and $o_1 = o_0$. Then, the action $a_1 \sim \pi_{lo}^{\boldsymbol{\theta}_{lo}}(\cdot|s_1, o_1)$, and the next state is selected, $s_2 \sim P(\cdot|s_1, a_1)$ and so on so forth. For simplicity we define

$$\tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o_t|o_{t-1}, s_t, b_t) := \begin{cases} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o_t|s_t), & \text{if } b_t = 1, \\ 1, & \text{if } b_t = 0, o_t = o_{t-1}, \\ 0, & \text{if } b_t = 0, o_t \neq o_{t-1}. \end{cases} \tag{1}$$

The probability of the trajectory $\tau_h = \{s_0, o_0, a_0, s_1, b_1, o_1, \dots, o_T, a_T\}$ is given by

$$\begin{aligned} &\mathbb{P}_D^{\boldsymbol{\theta}}(\tau_h) = D(s_0)\pi_{hi}^{\boldsymbol{\theta}_{hi}}(o_0|s_0)\pi_{lo}^{\boldsymbol{\theta}_{lo}}(a_0|s_0, o_0)\times \\ &\left[\prod_{t=1}^{T} \pi_b^{\boldsymbol{\theta}_b}(b_t|s_t, o_{t-1})\tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o_t|o_{t-1}, s_t, b_t)\pi_{lo}^{\boldsymbol{\theta}_{lo}}(a_t|s_t, o_t)\right]\left[\prod_{t=1}^{T-1} P(s_{t+1}|s_t, a_t)\right]. \end{aligned} \tag{2}$$

Note that $T \in \mathbb{N} \cup \{\infty\}$. With respect to flat RL, in the options framework we additionally define: the state-option-action value function $Q_{\boldsymbol{\theta}}^h(s, o, a) = \mathbb{E}_{\tau_h}[\sum_{t=0}^{\infty} \gamma^t r(s_t, o_t, a_t)|S_0 = s, O_0 = o, A_0 = a]$, the state-option value function $V_{\boldsymbol{\theta}}^h(s, o) = \mathbb{E}_{a\sim\pi_{lo}^{\boldsymbol{\theta}_{lo}}(\cdot|s,o)}[Q_{\boldsymbol{\theta}}^h(s, o, a)]$ and the normalized discounted state-option visitation frequency induced by the triplet $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_b^{\boldsymbol{\theta}_b}\}$ and by $S_0 = s_0$ and $O_0 = o_0$ $\rho_{\boldsymbol{\theta},s_0,o_0}^h(s, o) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, o_t = s; D, \boldsymbol{\theta}, P)$.

**Imitation learning:** Given a task and an expert, such as our foraging experiment and a person playing the game, Imitation Learning (IL) infers the underlying expert distribution via a set of demonstrations (state-action samples) generated while performing the task. When we assume the expert behavior follows a hierarchical policy with true parameters $(\boldsymbol{\theta}_{hi}^*, \boldsymbol{\theta}_{lo}^*, \boldsymbol{\theta}_b^*)$, we refer to the process of estimating $(\boldsymbol{\theta}_{hi}^*, \boldsymbol{\theta}_{lo}^*, \boldsymbol{\theta}_b^*)$ through a finite sequence of expert demonstrations $\tau =$

$(s_{0:T}, a_{0:T})$ with $2 \leq T < \infty$ as HIL. One way to formulate this problem is through:

$$\max_{(\boldsymbol{\theta}_{hi}, \boldsymbol{\theta}_{lo}, \boldsymbol{\theta}_b) \in \Theta} \mathcal{L}(\boldsymbol{\theta}_{hi}, \boldsymbol{\theta}_{lo}, \boldsymbol{\theta}_b), \tag{3}$$

where $\mathcal{L}(\boldsymbol{\theta}_{hi}, \boldsymbol{\theta}_{lo}, \boldsymbol{\theta}_b)$ denotes the marginal log-likelihood and is equivalent to the logarithm of the joint probability of generating the expert demonstrations $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_T, a_T\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}_{hi}, \boldsymbol{\theta}_{lo}, \boldsymbol{\theta}_b) = \log \mathbb{P}_D^{\boldsymbol{\theta}}(\tau) = \log \sum_{o_{0:T}, b_{1:T}} \mathbb{P}_D^{\boldsymbol{\theta}}(\tau_h), \tag{4}$$

where, $\mathbb{P}_D^{\boldsymbol{\theta}}(\tau_h)$ in (4) is defined in (2).

## 3 Method

In the following, we describe in detail the DOC algorithm, which is a deep learning-based reformulation of the hierarchical actor-critic algorithm presented in [18, 25] following the earlier actor-critic [26]. The goal of these algorithms is to optimize a performance objective, the option-state value function, over a family of parameterized policies $\pi_{\boldsymbol{\theta}}$ by performing gradient descent. Note that, in the options framework the option-state value function is induced by the triplet $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_b^{\boldsymbol{\theta}_b}\}$ which follows the OPGM in Fig. 1. To streamline the notation, we denote the expectation using $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ and the value function of interest becomes

$$\begin{aligned} V_{\boldsymbol{\theta}}^h(s, o) &= \mathbb{E}_{\boldsymbol{\theta}}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \gamma^t \eta c(s_t, o_t, s_t, b_{t+1}, s_{t+1}, o_{t+1}) | S_0 = s, O_0 = o], \\ &= \mathbb{E}_{\boldsymbol{\theta}}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \gamma^{t+1} \eta \pi_b^{\boldsymbol{\theta}_b}(b_{t+1} = 1 | s_{t+1}, o_t) | S_0 = s, O_0 = o]. \end{aligned} \tag{5}$$

As in [25], in Eq. (5): $c(s_t, o_t, s_t, b_{t+1}, s_{t+1}, o_{t+1}) := \gamma \eta \pi_b^{\boldsymbol{\theta}_b}(b_{t+1} = 1 | s_{t+1}, o_t)$, with $\eta$ a hyperparameter, is an additional cost which penalizes frequent switching between options with the goal of enhancing their interpretability. Lemma 1 formulates the Bellman equation for the value function in (5).

**Lemma 1.** *Considering the OPGM in Fig. 1, the expected discounted state-option value function in (5) satisfies the following Bellman equation:*

$$\begin{aligned} V_{\boldsymbol{\theta}}^h(s, o) &= \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s, o) \Bigg[ r(s, a) + \gamma \sum_{s'} P(s'|a, s) \bigg( \pi_{b_t}^{\boldsymbol{\theta}_b}(b = 0 | s', o) V_{\boldsymbol{\theta}}^h(s', o) \\ &\quad + \pi_{b_t}^{\boldsymbol{\theta}_b}(b = 1 | s', o) \big( \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s') V_{\boldsymbol{\theta}}^h(s', o') - \eta \big) \bigg) \Bigg] \\ &= \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s, o) Q_{\boldsymbol{\theta}}^h(s, o, a). \end{aligned}$$

*Proof.* Provided in the supplementary material. □

Based on Lemma 1 and assuming differentiability of $\pi_{lo}^{\boldsymbol{\theta}_{lo}}$ and $\pi_b^{\boldsymbol{\theta}_b}$ with respect to $\boldsymbol{\theta}_{lo}$ and $\boldsymbol{\theta}_b$, we formulate the policy gradient theorem for both the low-level and the termination policy.

**Theorem 1.** *Considering the OPGM in Fig. 1 and the Bellman equation in Lemma 1 the gradient of $V_{\boldsymbol{\theta}}^h(s_0, o_0)$ with respect to $\boldsymbol{\theta}_{lo}$ is*

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_{lo}} V_{\boldsymbol{\theta}}^h(s_0, o_0) &\propto \sum_s \sum_o \rho_{\boldsymbol{\theta}, s_0, o_0}^h(s, o) \sum_a \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s, o) Q_{\boldsymbol{\theta}}^h(s, o, a) \\ &\approx \mathbb{E}_{\boldsymbol{\theta}, s, o, a} \bigg[ Q_{\boldsymbol{\theta}}^h(s, o, a) \nabla_{\boldsymbol{\theta}_{lo}} \log \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s, o) \bigg| S_0 = s_0, O_0 = o_0 \bigg], \end{aligned} \tag{6}$$

*and with respect to $\boldsymbol{\theta}_b$ is*

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s_0, o_0) &\propto -\sum_s \sum_o \rho_{\boldsymbol{\theta}, s_0, o_0}^h(s, o) \nabla_{\boldsymbol{\theta}_b} \pi_{b_t}^{\boldsymbol{\theta}_b}(b = 1 | s', o) \Gamma_{\boldsymbol{\theta}}^h(s', o) \\ &\approx \mathbb{E}_{\boldsymbol{\theta}, s, o} \bigg[ - \nabla_{\boldsymbol{\theta}_b} \pi_b^{\boldsymbol{\theta}_b}(b = 1 | s', o) \Gamma_{\boldsymbol{\theta}}^h(s', o) \bigg| S_0 = s_0, O_0 = o_0 \bigg], \end{aligned} \tag{7}$$


(a) (b) (c)

Figure 2: Fig. 2a and 2b are two snapshots of the foraging game in which the forager is about to collect a few coins. Fig. 2c instead shows a top view of the environment where the clusters of coins are indicated by circles.

*where*

$$\Gamma^h_{\boldsymbol{\theta}}(s', o) = V^h_{\boldsymbol{\theta}}(s', o) - \sum_{o'} \pi^{\boldsymbol{\theta}_{hi}}_{hi}(o'|s') V^h_{\boldsymbol{\theta}}(s', o') + \eta.$$

*Proof.* The proof for both (6) and (7) follows the same structure of the policy gradient theorem for the flat RL case in [27] and has been originally given in [18]. We report in the supplementary material our version of the proof, consistent with our notation and formulation of the OPGM. □

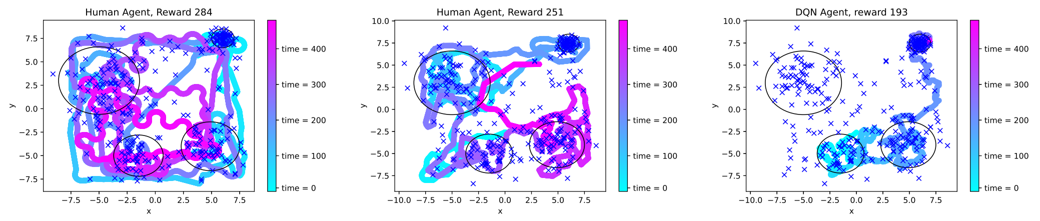
Lemma 1, with Theorem 1, are the main components of the hierarchical actor and critic updates in the DOC algorithm which is summarized in Algorithm 1 in Supplementary Material. Note that, the DOC uses as many replay buffers as the number of options with the same working principle described in [1]; moreover, it requires an initialization for $\{\pi^{\boldsymbol{\theta}_{hi}}_{hi}, \pi^{\boldsymbol{\theta}_{lo}}_{lo}, \pi^{\boldsymbol{\theta}_{b}}_{b}\}$ and $Q^h_{\boldsymbol{\theta}}(s, o, a)$. The triplet $\{\pi^{\boldsymbol{\theta}_{hi}}_{hi}, \pi^{\boldsymbol{\theta}_{lo}}_{lo}, \pi^{\boldsymbol{\theta}_{b}}_{b}\}$ is initialized using HIL and, specifically, we use the Baum-Welch algorithm for HIL in [22, 23] which maximizes a surrogate of (3) and returns a pre-trained hierarchical policy. Note also that, the DOC does not update $\pi^{\boldsymbol{\theta}_{hi}}_{hi}$ since we assume it has been already learnt during the HIL procedure. Finally, we define as many critic-networks $Q^h_{\boldsymbol{\theta}_Q}(s, o, a)$ as options, and initialize them using the Hierarchical Neural TD(0) in Algorithm 2 in the Supplementary Material.

## 4 Experimental results

In the following section, we describe the results obtained using HIL with the DOC algorithm on the foraging task and compare them with the human and the deep Q-learning (DQN) agents. As mentioned, humans collect on average $243.98$ coins and each episode lasts $8$ minutes. Fig. 3a illustrates a human trajectory which consists of 28k steps. For the sake of imitation learning efficiency, the trajectories are further processed and the amount of steps is reduced to $4.3$k. After performing the training, we evaluate the algorithms using episodes of different lengths as summarized by Table 1. We show that, despite requiring more than $4.3$k steps of the processed human trajectories, our algorithm learns how to solve the task and achieves human-like performance without requiring an excessive number of training steps. Additional information on neural networks design and hyperparameters choices are reported in the Supplementary Material[2].

**Deep Q-learning:** We train from scratch a deep RL agent using deep Q-learning [1]. We postulate that humans exploit both egocentric and allocentric strategies when navigating [28]; therefore, we define the state vector as $\boldsymbol{s} = \{x, y, \psi, \chi\}$ where $x, y$ are the coordinates of the agent in the original frame, $\psi$ is a categorical variable describing whether the agent can see a coin or not in its vicinity, $\psi \in$ {see coin, no coins}, and $\chi$ is another categorical variable describing the direction of the closest coin the agent has in its view, $\chi \in$ {east, northeast, north, northwest, west, southwest, south, southeast, no coins}. Then, the agent action is defined as $a \in$ {east, northeast, north, northwest, west, southwest, south, southeast}. The transition occurs always deterministically, and the rewards are simply represented by the coins in the environment; $r(\boldsymbol{s}, a) = 1$ for each coin collected when applying action $a$ in state $\boldsymbol{s}$. Each coin pops

[2]Code at https://figshare.com/s/1fd76320638959eb1b77




(a) (b) (c)

Figure 3: Fig. 3b and Fig. 3a show two trajectories from the human players while Fig. 3c shows a trajectory of the DQN agent. The evolution of the trajectories in time is illustrated using the color bar on the right side of the figures.

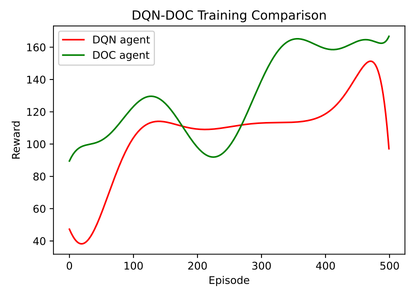



Figure 4: Fig. 4 shows the smoothed training trend for deep Q-learning and the DOC algorithm. Note that for the deep Q-learning we perform an update of the Q-network at each episode for a total of 500 improvement steps. Whereas, on the DOC we perform only 100 improvements since, for each 5 episodes, 4 are simply stored in the buffer.

up when the agent is at a distance $d = 0.8$, and it is collected at $d = 0.3$. The agent is trained for $500$ episodes and the training progress is illustrated in Fig. 4. The algorithm returns a Q-network that we denote with $Q^*$ and the agent action in state $\boldsymbol{s}$ is sampled according to $a \sim \mathcal{N}(\mu^*, 1.5^2)$ where $\mu^* = \arg\max_a Q^*(\boldsymbol{s}, a)$. We then evaluate the agent for $100$ episodes per seed for $40$ different random seeds. We run multiple evaluations, varying the length of the episodes in order to judge how many steps are needed to achieve human-level performance. As Table 1 shows, the DQN agent does not improve by increasing the episodes length and it remains far from achieving human-like performance.

**Behavioral cloning and hierarchical Imitation Learning:** We select the human trajectory in Fig. 3b as expert data. We process the data as described earlier to make it suitable for the designed state and action space, and then we train a simple policy network $\pi_{\boldsymbol{\theta}}$ using Behavioral Cloning (BC) [29] and a triplet $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_{b}^{\boldsymbol{\theta}_{b}}\}$ using the Baum-Welch algorithm for HIL in [22, 23] with two options ($|\mathcal{O}| = 2$). As previously, we evaluate the algorithms on $100$ episodes over $40$ seeds and run multiple evaluations varying the episodes length: the BC agent collects $14.4$ on average and, like the DQN, it does not improve with increased episode length. In this case, this happens because the BC agent suffers from compounding errors, which lead to a series of unexplored states and poor performance [30] (see Fig. 5a). On the other hand, the Baum-Welch algorithm for HIL achieves better performance (see Table 1), but it is prone to convergence to local optima. This translates to an option activation policy in Fig. 5c which is difficult to interpret and does not reflect any specific task-oriented division between the options, nor does it improve with increased episode length. To tackle this issue, based on foraging behavioral studies in both humans and animals [31, 32, 33], we pre-train the high level policy $\pi_{hi}^{\boldsymbol{\theta}_{hi}}$ before applying HIL in order to incorporate human priors not only at the low level of the hierarchy but also at the high level. During the foraging process, the resources in the same cluster are depleted over time, and the forager is forced to resolve the trade off between exploration and exploitation (see Preliminaries: Foraging Task). As a result, we can represent one option (option 1 in blue in Fig. 5c) as an "exploration" option where the agent navigates through the environment looking for potentially rewarding areas and of the other option (option 2 in red in Fig. 5c)

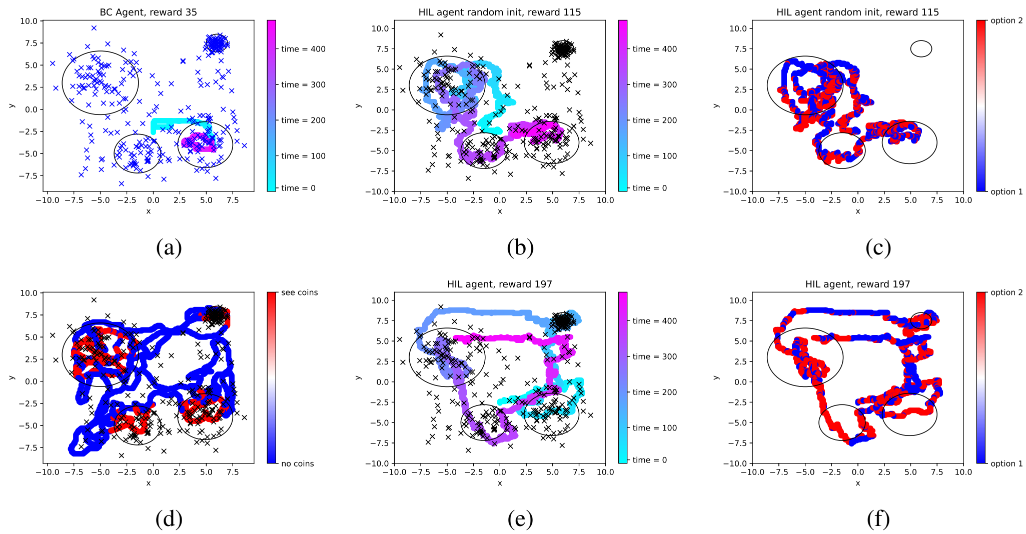


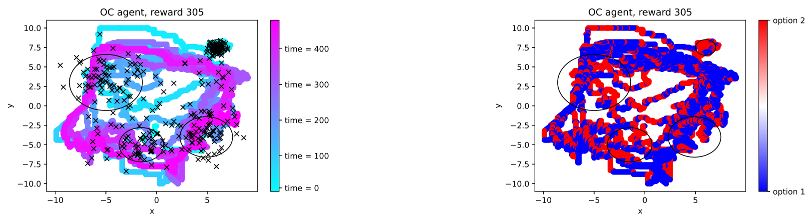

Figure 5: Fig. 5a shows a trajectory for the behavioral cloning agent. Fig. 5b and Fig. 5c show an episode for a randomly initialized hierarchical agent trained with HIL. In particular, Fig. 5c depicts the agent's options usage where the color blue stands for Option 1 and red for Option 2. Similarly, Fig. 5e and Fig. 5f show a trajectory for the pre-initialized HIL agent where the initialization of the high-level policy is based on the data in Fig. 5d, where the different labels are denoted by the different colors.

as an "exploitation" option where the agent tries to collect as many coins as possible within a given cluster. Following this reasoning, we pre-train $\pi_{hi}^{\boldsymbol{\theta}_{hi}}$ with supervised learning using the trajectory in Fig. 5d with states as inputs and options as labels, and we subsequently use it as initialization for the HIL algorithm. This last agent outperforms the random initialized HIL agent, demonstrates greater ability to generalize, and improves when we increase the episode length. Moreover, it shows a much more interpretable set of options, Fig. 5e and 5f, with respect to the random initialized HIL agent in Fig. 5b and 5c. More details on the performance are available in Table 1.

**Deep Option-critic:** As a final step, we take the pre-initialized HIL triplet $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_{b}^{\boldsymbol{\theta}_{b}}\}$ and use it as initial hierarchical policy in the DOC algorithm (Algorithm 1 in the Supplementary Material). Furthermore, we initialize a critic network for each option using the Hierarchical Neural TD(0) algorithm in Algorithm 2 in the Supplementary Material. The evolution of the training progress is shown in Fig. 4. It is performed for 500 episodes where, for every five episodes, four trajectories are simply stored in the replay buffer, and only during the fifth episode does the off-policy update in Algorithm 1 occur. The final agent is then evaluated as before (details are reported in Table 1.) For episodes with 10k steps, this agent can achieve a level of performance comparable to humans, and it outperforms humans for episodes with 12k steps. The best trajectory obtained by the OC agent during the evaluation step is depicted in Fig. 6.

## 5 Discussion and Conclusions

As Table 1 shows, the combination of human priors, HIL, and the DOC algorithm yields an average performance that, for longer episodes, is comparable to human agents. This result was achieved by first including human priors in the hierarchical policy to improve HIL. As mentioned, the imitation learning methods based on maximizing the log-likelihood (cf. Section 2) converge to local optima, and this can only be dampened by non-trivial network initialization. Then, the policy was trained from expert demonstrations using HIL and was combined with a deep HRL. Though the combination of human priors, HIL, and the DOC algorithm performs at a human-like level, there is room for further improvement. Foraging tasks facilitate the discretization of distinct options, but this is not possible with all tasks used to study human cognition. Therefore, we plan to work on methods which can extract human and animals priors from processed brain imaging data (using functional magnetic




(a) (b)

Figure 6: Best trajectory for the final agent trained with DOC and human priors. Fig. 6b depicts the agent's options usage where the blue color indicates Option 1 and red Option 2. Option 1 can be seen as the exploration option while Option 2 as the exploitation option.

Table 1: Summary of the performance.

| | Average | Std | Max | Min |
|---|---|---|---|---|
| Human (8 minutes per episode) | 243.98 | **21.24** | 284 | **195** |
| DQN (6000 steps per episode) | 125.32 | 26.46 | 193 | 15 |
| DQN (10000 steps per episode) | 132.38 | 25.01 | 203 | 23 |
| DQN (12000 steps per episode) | 134.06 | 24.59 | 206 | 17 |
| HIL (6000 steps per episode) | 46.10 | 26.06 | 143 | 5 |
| HIL (10000 steps per episode) | 47.83 | 27.36 | 146 | 5 |
| HIL (12000 steps per episode) | 48.88 | 27.36 | 156 | 5 |
| Human Priors HIL (6000 steps per episode) | 141.95 | 25.38 | 237 | 72 |
| Human Priors HIL (10000 steps per episode) | 177.75 | 26.35 | 272 | 94 |
| Human Priors HIL (12000 steps per episode) | 189.59 | 27.69 | 279 | 104 |
| DOC+Human Priors HIL (6000 steps per episode) | 213.30 | 36.11 | 288 | 98 |
| DOC+Human Priors HIL (10000 steps per episode) | 244.41 | 33.56 | 302 | 151 |
| DOC+Human Priors HIL (12000 steps per episode) | **253.06** | 32.12 | **305** | 170 |

resonance imaging (fMRI) and other methods) in an attempt to test these neuro-inspired algorithms in other cognitive domains. Moreover, Table 1 shows that the DOC agent can perform sufficiently (Max column in Table 1); but it presents higher variance in the results, with lower minima (Std and Min columns in Table 1), indicating that it is less robust than humans. Improving robustness, reliability, and safety while maintaining optimal performance and efficiency represents a main challenge for the future. Note that in this paper, we focused on off-policy methods whose learning is generally oscillatory (Fig. 4) due to the bias introduced in estimating the critic. Future work will focus on hierarchical on-policy methods, which are more stable but also more inefficient and could therefore further benefit by the combination with HIL. Finally, for all RL problems, we specify the simplest reward possible, which is sparse in the state space and does not account for other features of the experiment, such as time, that humans consider. The implications are thus: despite the simplicity of the reward and the absence of intermediate signals, our method accomplishes interesting results; but at the same time, it raises additional questions. Would better reward specification further improve results? Can we make inferences about "hierarchical" rewards directly from human behavior? These questions remain open and will be the topic of future discussions.

In conclusion, this work contributes to the rapidly growing body of literature at the intersection of RL and neuroscience. Our method achieves human-like performance on a foraging task in a reasonably efficient manner. This represents an important step towards developing methods that can be applied to real-world decision-making scenarios where samples are expensive and efficiency is crucial.

## References


[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[2] Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. "Reinforcement learning, fast and slow". In: *Trends in cognitive sciences* 23.5 (2019), pp. 408–422.

[3] Matthew M Botvinick, Yael Niv, and Andew G Barto. "Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective". In: *Cognition* 113.3 (2009), pp. 262–280.

[4] Matthew Michael Botvinick. "Hierarchical reinforcement learning and decision making". In: *Current opinion in neurobiology* 22.6 (2012), pp. 956–962.

[5] Johannes Bill, Hrag Pailian, Samuel J Gershman, and Jan Drugowitsch. "Hierarchical structure is employed by humans during visual motion perception". In: *Proceedings of the National Academy of Sciences* 117.39 (2020), pp. 24581–24589.

[6] Dan Schwartz, Mariya Toneva, and Leila Wehbe. "Inducing brain-relevant bias in natural language processing models". In: *arXiv preprint arXiv:1911.03268* (2019).

[7] Mariya Toneva and Leila Wehbe. "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)". In: *arXiv preprint arXiv:1905.11833* (2019).

[8] Nils Kolling, Timothy EJ Behrens, Rogier B Mars, and Matthew FS Rushworth. "Neural mechanisms of foraging". In: *Science* 336.6077 (2012), pp. 95–98.

[9] Benjamin Y Hayden, John M Pearson, and Michael L Platt. "Neuronal basis of sequential foraging decisions in a patchy environment". In: *Nature neuroscience* 14.7 (2011), p. 933.

[10] Sara M Constantino and Nathaniel D Daw. "Learning the opportunity cost of time in a patch-foraging task". In: *Cognitive, Affective, & Behavioral Neuroscience* 15.4 (2015), pp. 837–853.

[11] Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L Griffiths, and Alexei A Efros. "Investigating human priors for playing video games". In: *arXiv preprint arXiv:1802.10217* (2018).

[12] Eliza Kosoy, Jasmine Collins, David M Chan, Sandy Huang, Deepak Pathak, Pulkit Agrawal, John Canny, Alison Gopnik, and Jessica B Hamrick. "Exploring exploration: Comparing children with RL agents in unified environments". In: *arXiv preprint arXiv:2005.02880* (2020).

[13] Léonard Hussenot, Robert Dadashi, Matthieu Geist, and Olivier Pietquin. "Show me the Way: Intrinsic Motivation from Demonstrations". In: *arXiv preprint arXiv:2006.12917* (2020).

[14] Yilun Du and Karthic Narasimhan. "Task-Agnostic Dynamics Priors for Deep Reinforcement Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 1696–1705. URL: http://proceedings.mlr.press/v97/du19e.html.

[15] David D Bourgin, Joshua C Peterson, Daniel Reichman, Stuart J Russell, and Thomas L Griffiths. "Cognitive model priors for predicting human decisions". In: *International conference on machine learning*. PMLR. 2019, pp. 5133–5141.

[16] Tejas D Kulkarni, Karthik R Narasimhan, Ardavan Saeedi, and Joshua B Tenenbaum. "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation". In: *arXiv preprint arXiv:1604.06057* (2016).

[17] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. "Data-efficient hierarchical reinforcement learning". In: *arXiv preprint arXiv:1805.08296* (2018).

[18] Pierre-Luc Bacon, Jean Harb, and Doina Precup. "The option-critic architecture". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[19] Gabriel V Cruz Jr, Yunshu Du, and Matthew E Taylor. "Pre-training neural networks with human demonstrations for deep reinforcement learning". In: *arXiv preprint arXiv:1709.04083* (2017).

[20] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. "Reinforcement learning from imperfect demonstrations". In: *arXiv preprint arXiv:1802.05313* (2018).

[21] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. "Deep Q-learning from demonstrations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[22] Zhiyu Zhang and Ioannis Paschalidis. "Provable Hierarchical Imitation Learning via EM". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 883–891.

[23] Vittorio Giammarino and Ioannis Ch Paschalidis. "Online Baum-Welch algorithm for Hierarchical Imitation Learning". In: *arXiv preprint arXiv:2103.12197* (2021).

[24] Richard S Sutton, Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial intelligence* 112.1-2 (1999), pp. 181–211.

[25] 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 the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[26] Vijay R Konda and John N Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems*. Citeseer. 2000, pp. 1008–1014.

[27] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. "Policy gradient methods for reinforcement learning with function approximation." In: *Advances in Neural Information Processing Systems*. Vol. 99. Citeseer. 1999, pp. 1057–1063.

[28] Janet D Feigenbaum and Robin G Morris. "Allocentric versus egocentric spatial memory after unilateral temporal lobectomy in humans." In: *Neuropsychology* 18.3 (2004), p. 462.

[29] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. "An algorithmic perspective on imitation learning". In: *arXiv preprint arXiv:1811.06711* (2018).

[30] Stéphane Ross and Drew Bagnell. "Efficient reductions for imitation learning". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 661–668.

[31] Christopher Kalff, Thomas Hills, and Jana Malte Wiener. "Human foraging behavior: A virtual reality investigation on area restricted search in humans". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 32. 32. 2010.

[32] Thomas T Hills, Michael N Jones, and Peter M Todd. "Optimal foraging in semantic memory." In: *Psychological review* 119.2 (2012), p. 431.

[33] Marcos Bella-Fernández, Manuel Suero Suñé, and Beatriz Gil-Gómez de Liaño. "Foraging behavior in visual search: A review of theoretical and mathematical models in humans and animals". In: *Psychological Research* (2021), pp. 1–19.

# A Supplementary material

## A.1 Ethics Statement

Reinforcement learning has the potential to improve decision making across many important application domains. These techniques will become practical only if they can yield excellent, human-level performance using reasonable computational effort. Our work makes progress towards accomplishing this goal by addressing sample complexity using human priors. We do not believe our contributions introduce ethical issues that could negatively impact society. Moreover, all the human data we used throughout the work and the human experiments have been approved by the pertinent institutional Review Board.

## A.2 Human Experiment

**Subjects:** Subjects consisted of male and female, neurologically healthy, English speaking volunteers between the ages of 18-35 with normal or corrected to normal vision. Subjects were recruited from the university and the surrounding community. Individuals with a history of drug abuse, use of psychoactive medication, neurological or psychiatric disorders, or learning disabilities were excluded. Additionally, participants with a history of motion sickness when watching or playing video games were also excluded. We recruited seventeen subjects for this study and we obtained five full datasets that are used here. Two subjects were found to be ineligible due to exclusionary criteria, three withdrew due to motion sickness, two were lost to follow up after the first day, five were excluded for repeatedly falling asleep during scanning.

**Experimental procedure:** This experiment was conducted over two days. On the first day, naive subjects were presented with the task on a computer in a behavioral testing room. Subjects were instructed to freely explore the environment and collect as many coins as possible. Subjects were not told anything about the distribution or total number of coins. Subjects were able to see a running count of the coins they had collected for each run. Subjects performed 10 runs, each lasting 8 minutes. For the second day, participants performed the same task in the MRI scanner for 10 eight-minute runs.

**Task:** The task consisted of a $160m \times 160m$ virtual "open-field" paradigm surrounded by four differently colored and textured walls. 325 coins were distributed throughout the environment, of which 100 were randomly distributed and 225 were distributed according to four different Gaussian distributions of varying sizes: 75 according to $\mathcal{N}((6, 7.5), 0.5^2)$, 40 according to $\mathcal{N}((-1.5, -5.0), 1.1^2)$, 60 according to $\mathcal{N}((-5, 3), 1.8^2)$ and 50 according to $\mathcal{N}((4.9, -4), 1.3^2)$. The Subject's starting location was random at the beginning of each run. Subjects could move forward and turn left or right either with the arrow keys on the first day or the button box on the second day. They could not move backwards.

## A.3 Proof of Lemma 1

Considering the OPGM in Fig. 1, the expected discounted state-option value function in (5) satisfies the following Bellman equation

$$\begin{aligned}
V^h_{\boldsymbol{\theta}}(s,o) = & \sum_a \pi^{\boldsymbol{\theta}_{lo}}_{lo}(a|s,o)\Bigg[r(s,a) + \gamma \sum_{s'} P(s'|a,s)\Bigg(\pi^{\boldsymbol{\theta}_b}_{b_t}(b=0|s',o)V^h_{\boldsymbol{\theta}}(s',o) \\
& + \pi^{\boldsymbol{\theta}_b}_{b_t}(b=1|s',o)\big(\sum_{o'} \pi^{\boldsymbol{\theta}_{hi}}_{hi}(o'|s')V^h_{\boldsymbol{\theta}}(s',o') - \eta\big)\Bigg)\Bigg] \\
= & \sum_a \pi^{\boldsymbol{\theta}_{lo}}_{lo}(a|s,o)Q^h_{\boldsymbol{\theta}}(s,o,a).
\end{aligned}$$

*Proof.* Note that, for the sake of brevity in the following proof we use the notation $S = s$ only the first time to define the value of the random variable $S$, and then, we retain only $s$ to simplify the notation. The same is done for options and actions.

$$
\begin{aligned}
V_{\boldsymbol{\theta}}^h(s,o) =& \mathbb{E}_{\boldsymbol{\theta}}[\sum_{t=0}^{\infty}\gamma^t r(S_t,A_t) - \gamma^{t+1}\eta\pi_b^{\boldsymbol{\theta}_b}(B_{t+1}=1|S_{t+1},O_t)|S_0=s,O_0=o] \\
=&\mathbb{E}_{\boldsymbol{\theta}}[r(S_0=s,A_0=a) - \gamma\eta\pi_b^{\boldsymbol{\theta}_b}(B_1=b=1|S_1=s',O_0=o)+ \\
&\gamma\sum_{t=0}^{\infty}\big(\gamma^t r(S_{t+1},A_{t+1}) - \gamma^{t+1}\eta\pi_b^{\boldsymbol{\theta}_b}(B_{t+2}=1|S_{t+2},O_{t+1})\big)|S_0=s,O_0=o] \\
=&\sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\sum_{s'}P(s'|a,s)\sum_b \pi_b^{\boldsymbol{\theta}_b}(b|s',o)\sum_{o'}\tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(O_{t+1}=o'|o,s',b)\Big[r(s,a) \\
&-\gamma\eta\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o) \\
&+\gamma\mathbb{E}_{\boldsymbol{\theta}}[\sum_{t=0}^{\infty}\big(\gamma^t r(S_{t+1},A_{t+1}) - \gamma^{t+1}\eta\pi_{b_t}^{\boldsymbol{\theta}_b}(B_{t+2}=1|S_{t+2},O_{t+1})\big)|S_1=s',O_1=o']\Big] \\
=&\sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\sum_{s'}P(s'|a,s)\sum_b \pi_{b_t}^{\boldsymbol{\theta}_b}(b|s',o)\sum_{o'}\tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o'|o,s',b)\Big[r(s,a) \\
&-\eta\gamma\pi_{b_t}^{\boldsymbol{\theta}_b}(b=1|s',o)+\gamma V_{\boldsymbol{\theta}}^h(s',o')\Big] \\
=&\sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\Big[r(s,a)+\gamma\sum_{s'}P(s'|a,s)\Big(\pi_b^{\boldsymbol{\theta}_b}(b=0|s',o)V_{\boldsymbol{\theta}}^h(s',o) \\
&+\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(\sum_{o'}\pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o')-\eta\big)\Big)\Big] \\
=&\sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)Q_{\boldsymbol{\theta}}^h(s,o,a).
\end{aligned}
$$

□

### A.4 Proof of Theorem 1

Considering the OPGM in Fig. 1 and the Bellman equation in Lemma 1 the gradient of $V_{\boldsymbol{\theta}}^h(s,o)$ with respect to $\boldsymbol{\theta}_{lo}$ is

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}_{lo}}V_{\boldsymbol{\theta}}^h(s_0,o_0) &\propto \sum_s\sum_o \rho_{\boldsymbol{\theta},s_0,o_0}^h(s,o)\sum_a \nabla_{\boldsymbol{\theta}_{lo}}\pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)Q_{\boldsymbol{\theta}}^h(s,o,a) \\
&\approx \mathbb{E}_{\boldsymbol{\theta},s,o,a}\Big[Q_{\boldsymbol{\theta}}^h(s,o,a)\nabla_{\boldsymbol{\theta}_{lo}}\log\pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\Big|S_0=s_0,O_0=o_0\Big],
\end{aligned}
\tag{8}
$$

and with respect to $\boldsymbol{\theta}_b$ is

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}_b}V_{\boldsymbol{\theta}}^h(s_0,o_0) &\propto -\sum_s\sum_o \rho_{\boldsymbol{\theta},s_0,o_0}^h(s,o)\nabla_{\boldsymbol{\theta}_b}\pi_{b_t}^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o) \\
&\approx \mathbb{E}_{\boldsymbol{\theta},o,s}\Big[-\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o)\Big|S_0=s_0,O_0=o_0\Big],
\end{aligned}
\tag{9}
$$

where

$$
\Gamma_{\boldsymbol{\theta}}^h(s',o) = V_{\boldsymbol{\theta}}^h(s',o) - \sum_{o'}\pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o')+\eta.
$$

*Proof.* We start by proving (8):

$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}_{lo}} V^h_{\boldsymbol{\theta}}(s,o) = \nabla_{\boldsymbol{\theta}_{lo}} \sum_a \Big\{ \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) \Big\} \\
&= \sum_a \Big\{ \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) + \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \nabla_{\boldsymbol{\theta}_{lo}} Q^h_{\boldsymbol{\theta}}(s,o,a) \Big\} \\
&= \sum_a \Big\{ \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) + \gamma \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \sum_{s'} P(s'|a,s) \times \\
&\Big( \pi_b^{\boldsymbol{\theta}_b}(b=0|s',o) \nabla_{\boldsymbol{\theta}_{lo}} Q^h_{\boldsymbol{\theta}}(s',o) + \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o) \big( \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s') \nabla_{\boldsymbol{\theta}_{lo}} Q^h_{\boldsymbol{\theta}}(s',o') \big) \Big) \Big\} \\
&= \sum_a \Big\{ \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) + \gamma \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \sum_{s'} P(s'|a,s) \times \\
&\sum_b \pi_b^{\boldsymbol{\theta}_b}(b|s',o) \sum_{o'} \tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o'|o,s,b) \Big( \nabla_{\boldsymbol{\theta}_{lo}} V^h_{\boldsymbol{\theta}}(s',o') \Big) \Big\} \\
&= \sum_a \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) + \gamma \sum_{s'} \sum_{o'} \mathbb{P}(s',o'|s,o) \Big( \nabla_{\boldsymbol{\theta}_{lo}} V^h_{\boldsymbol{\theta}}(s',o') \Big).
\end{aligned}
$$

We define

$$
\beta_k(s,o) = \begin{cases} \gamma^k \sum_{s_k} \sum_{o_k} \mathbb{P}(s_k,o_k|s,o) & \text{if } k > 0 \\ 1 & \text{if } k=0, s_k = s, o_k = o \\ 0 & \text{if } k = 0, s_k \neq s, o_k \neq o \end{cases}
$$

and proceeding recursively we obtain

$$
\nabla_{\boldsymbol{\theta}_{lo}} V^h_{\boldsymbol{\theta}}(s,o) = \sum_{s_k} \sum_{o_k} \Big( \sum_{k=0}^{\infty} \beta_k(s,o) \Big) \sum_{a_k} \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a_k|s_k,o_k) Q^h_{\boldsymbol{\theta}}(s_k,o_k,a_k). \tag{10}
$$

For the sake of simplicity, we change the notation and we rewrite (10) as

$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}_{lo}} V^h_{\boldsymbol{\theta}}(s_0,o_0) \propto \sum_s \sum_o \rho^h_{\boldsymbol{\theta},s_0,o_0}(s,o) \sum_a \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) \\
&\approx \mathbb{E}_{\boldsymbol{\theta},s,o} \Big[ \sum_a \nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}}(s,o,a) \Big| S_0 = s_0, O_0 = o_0 \Big] \\
&= \mathbb{E}_{\boldsymbol{\theta},s,o} \Big[ \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \frac{\nabla_{\boldsymbol{\theta}_{lo}} \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)}{\pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)} Q^h_{\boldsymbol{\theta}}(s,o,a) \Big| S_0 = s_0, O_0 = o_0 \Big] \\
&= \mathbb{E}_{\boldsymbol{\theta},s,o} \Big[ \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \big( \nabla_{\boldsymbol{\theta}_{lo}} \log \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \big) Q^h_{\boldsymbol{\theta}}(s,o,a) \Big| S_0 = s_0, O_0 = o_0 \Big] \\
&= \mathbb{E}_{\boldsymbol{\theta},s,o,a} \Big[ Q^h_{\boldsymbol{\theta}}(s,o,a) \nabla_{\boldsymbol{\theta}_{lo}} \log \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) \Big| S_0 = s_0, O_0 = o_0 \Big],
\end{aligned}
$$

which concludes the first part of the proof. Similarly, we prove (9). We rewrite

$$\begin{aligned}
Q_{\boldsymbol{\theta}}^h(s,o,a) =& \Big[r(s,a) + \gamma \sum_{s'} P(s'|a,s)\Big(\pi_b^{\boldsymbol{\theta}_b}(b=0|s',o)V_{\boldsymbol{\theta}}^h(s',o) \\
&+ \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(\sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') - \eta\big)\Big)\Big] \\
=& \Big[r(s,a) + \gamma \sum_{s'} P(s'|a,s)\Big(\big(1-\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big)V_{\boldsymbol{\theta}}^h(s',o) \\
&+ \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(\sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') - \eta\big)\Big)\Big] \\
=& \Big[r(s,a) + \gamma \sum_{s'} P(s'|a,s)\Big(V_{\boldsymbol{\theta}}^h(s',o) - \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(V_{\boldsymbol{\theta}}^h(s',o) \\
&- \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') + \eta\big)\Big)\Big],
\end{aligned}$$

and

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}_b} Q_{\boldsymbol{\theta}}^h(s,o,a) \\
=&\gamma \sum_{s'} P(s'|a,s)\Big(\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o) - \nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(V_{\boldsymbol{\theta}}^h(s',o) \\
&- \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') + \eta\big) - \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o) \\
&- \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o')\big)\Big) \\
=&\gamma \sum_{s'} P(s'|a,s)\Big( - \nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big(V_{\boldsymbol{\theta}}^h(s',o) \\
&- \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') + \eta\big) + \big(1-\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\big)\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o) \\
&+ \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o')\Big) \\
=&-\gamma \sum_{s'} P(s'|a,s)\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o) \\
&+ \gamma \sum_{s'} P(s'|a,s)\sum_b \pi_b^{\boldsymbol{\theta}_b}(b|s',o)\sum_{o'} \tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o'|o,s',b)\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o')
\end{aligned}$$

where

$$\Gamma_{\boldsymbol{\theta}}^h(s',o) = V_{\boldsymbol{\theta}}^h(s',o) - \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s')V_{\boldsymbol{\theta}}^h(s',o') + \eta.$$

Then,

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s,o) = \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\nabla_{\boldsymbol{\theta}_b} Q_{\boldsymbol{\theta}}^h(s,o,a) \\
=&-\gamma \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\sum_{s'} P(s'|a,s)\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o) \\
&+ \gamma \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\sum_{s'} P(s'|a,s)\sum_b \pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\sum_{o'} \tilde{\pi}_{hi}^{\boldsymbol{\theta}_{hi}}(o'|o,s',b)\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o') \\
=&-\gamma \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o)\sum_{s'} P(s'|a,s)\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o) \\
&+ \gamma \sum_{o'}\sum_{s'} P(s',o'|s,o)\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s',o').
\end{aligned}$$

Proceeding recursively as for the first part of the proof we obtain

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s,o) &= -\sum_{s_k}\sum_{o_k}\left(\sum_{k=0}^{\infty}\beta_k(s,o)\right)\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s_k,o_k)\Gamma_{\boldsymbol{\theta}}^h(s_k,o_k)\\ &\approx \mathbb{E}_{\boldsymbol{\theta},s_k,o_k}\left[-\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s_k,o_k)\Gamma_{\boldsymbol{\theta}}^h(s_k,o_k)\middle|S_0=s,O_0=o\right],\end{aligned}$$

and by changing notation we obtain

$$\nabla_{\boldsymbol{\theta}_b} V_{\boldsymbol{\theta}}^h(s_0,o_0) \approx \mathbb{E}_{\boldsymbol{\theta},s',o}\left[-\nabla_{\boldsymbol{\theta}_b}\pi_b^{\boldsymbol{\theta}_b}(b=1|s',o)\Gamma_{\boldsymbol{\theta}}^h(s',o)\middle|S_0=s_0,O_0=o_0\right],$$

which concludes the proof. $\square$

### A.5 Algorithms

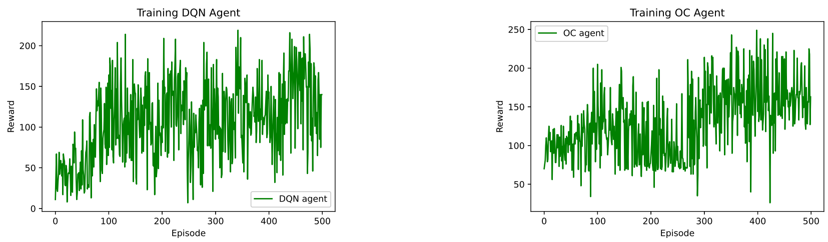
**Algorithm 1** Deep Option-Critic

**Init:** $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_{b}^{\boldsymbol{\theta}_{b}}\}$, $Q^h_{\boldsymbol{\theta}_Q}(s,o,a)$ and $\eta$
**Require:** $s_0 \sim D$ and $o_0 \sim \pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_0)$. A replay buffer for each $o \in \mathcal{O}$ to store $(s_t, a_t, r_t, c_t, s_{t+1})$ for all $t$.
**for** $t = 0, \ldots, T$ **do**
  $a_t \sim \pi_{lo}^{\boldsymbol{\theta}_{lo}}(\cdot|s_t, o_t)$.
  Observe $r_t$ and $s_{t+1}$.
  $b_{t+1} \sim \pi_{b}^{\boldsymbol{\theta}_{b}}(\cdot|s_{t+1}, o_t)$
  **if** $b_{t+1} = 1$ **then**
    $c_t \leftarrow \eta$
    $o_{t+1} \sim \pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_{t+1})$
  **else**
    $c_t \leftarrow 0$
    $o_{t+1} = o_t$
  **end if**
  Store $(s_t, a_t, r_t, c_t, s_{t+1})$ in the replay buffer corresponding to $o_t$.
  **if** $t >$ batch size **then**
    Sample a batch of $(s, a, r, c, s')$ from the replay buffer corresponding to $o$.
    $\tilde{r} = r - c$
    $y = \tilde{r} + \gamma \pi_{b}^{\boldsymbol{\theta}_{b}}(0|s', o) \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s', o) Q^h_{\boldsymbol{\theta}_Q}(s', o, a) + \gamma \pi_{b}^{\boldsymbol{\theta}_{b}}(1|s', o) \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s') \sum_a \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s', o') Q^h_{\boldsymbol{\theta}_Q}(s', o', a)$ for all the elements of the mini-batch.
    **Update Critic Q:** Perform gradient descent on $(y - Q^h_{\boldsymbol{\theta}_Q}(s,o,a))^2$ over the mini-batch.
    **Update** $\pi_{lo}^{\boldsymbol{\theta}_{lo}}$**:** $\boldsymbol{\theta}_{lo} \leftarrow \boldsymbol{\theta}_{lo} + \alpha_{lo} \nabla_{\boldsymbol{\theta}_{lo}} \log \pi_{lo}^{\boldsymbol{\theta}_{lo}}(a|s,o) Q^h_{\boldsymbol{\theta}_Q}(s,o,a)$
    **Update** $\pi_{b}^{\boldsymbol{\theta}_{b}}$**:** $\boldsymbol{\theta}_{b} \leftarrow \boldsymbol{\theta}_{b} - \alpha_b \nabla_{\boldsymbol{\theta}_b} \pi_{b}^{\boldsymbol{\theta}_{b}}(1|s', o) \big(V^h_{\boldsymbol{\theta}_Q}(s', o) - \sum_{o'} \pi_{hi}^{\boldsymbol{\theta}_{hi}}(o'|s') V^h_{\boldsymbol{\theta}_Q}(s', o') + \eta\big)$
  **end if**
  $o_t \leftarrow o_{t+1}$, $s_t \leftarrow s_{t+1}$
**end for**

**Algorithm 2** Hierarchical Neural TD(0)

**Init:** $\{\pi_{hi}^{\boldsymbol{\theta}_{hi}}, \pi_{lo}^{\boldsymbol{\theta}_{lo}}, \pi_{b}^{\boldsymbol{\theta}_{b}}\}$ and $Q^h_{\boldsymbol{\theta}_Q}(s,o,a)$.
**Require:** Replay Buffer for each $o \in \mathcal{O}$ to store $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ for all $t$, $s_0 \sim D$ and $a_0 \in \mathcal{A}$. $o_0 \sim \pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_0)$
**for** $t = 0, \ldots, T$ **do**
  Apply action $a_t$ in state $s_t$ given $o_t$ and observe $r_t$ and $s_{t+1}$.
  Then, sample $b_{t+1} \sim \pi_{b}^{\boldsymbol{\theta}_{b}}(\cdot|s_{t+1}, o_t)$, $o_{t+1} \sim \pi_{hi}^{\boldsymbol{\theta}_{hi}}(\cdot|s_{t+1}, b_t, o_t)$ and $a_{t+1} \sim \pi_{lo}^{\boldsymbol{\theta}_{lo}}(\cdot|s_{t+1}, o_{t+1})$.
  Store $(s_t, a_t, r_t, s_{t+1}, o_{t+1}, a_{t+1})$ in the replay buffer corresponding to $o_t$.
  **if** $t >$ batch size **then**
    Sample a batch of $(s, a, r, s', o', a')$ from the replay buffer corresponding to $o$.
    $y = r + \gamma Q^h_{\boldsymbol{\theta}_Q}(s', a', o')$ for all $j$ in the mini-batch.
    Perform gradient descent on $(y - Q^h_{\boldsymbol{\theta}_Q}(s,a,o))^2$ over the mini-batch.
  **end if**
  $a_t \leftarrow a_{t+1}$, $s_t \leftarrow s_{t+1}$, $o_t \leftarrow o_{t+1}$
**end for**

### A.6 Experimental results




(a) (b)

Figure 7: Fig. 7a and Fig 7b show the unsmoothed training trend for deep Q-learning and the DOC algorithm respectively.

In the following we provide more details on the experimental results presented in Section 4. Specifically, the tables below show the neural network architectures and the main hyperparameters we have used throughout the paper. For more details the code is available at `https://figshare.com/s/1fd76320638959eb1b77`.

Table 2: DQN architecture.

| Layer type | Dimension | Activation Function |
|---|---|---|
| Input | observation size | linear |
| Dense | 256 | Relu |
| Output | action space | linear |

Table 3: Other hyperparameters.

| Hyperparameter | value |
|---|---|
| $\gamma$ | 0.99 |
| exploration type | $\epsilon$-greedy |
| $\epsilon$ | $\max(\frac{0.5}{\bmod (episode, 10)}, 0.1)$ |
| Buffer size | 30000 |
| Batch size | 256 |

Table 4: $\pi_{\boldsymbol{\theta}}$ for behavioral cloning (BC) architecture.

| Layer type | Dimension | Activation Function |
|---|---|---|
| Input | observation size | linear |
| Dense | 512 | Relu |
| Output | action space | softmax |

Table 5: Other hyperparameters BC.

| Hyperparameter | value |
|---|---|
| Epochs | 200 |
| Loss type | Cross Entropy |

Table 6: $\pi_{lo}^{\boldsymbol{\theta}_{lo}}$, $\pi_{b}^{\boldsymbol{\theta}_{b}}$ and $\pi_{hi}^{\boldsymbol{\theta}_{hi}}$ and critic architecture.

| $\pi_{hi}^{\boldsymbol{\theta}_{hi}}$ | | |
|---|---|---|
| Layer type | Dimension | Activation Function |
| Input | observation size | linear |
| Dense | 5 | Relu |
| Output | $\lvert\mathcal{O}\rvert$ | softmax |
| $\pi_{lo}^{\boldsymbol{\theta}_{lo}}$ for each option | | |
| Layer type | Dimension | Activation Function |
| Input | observation size | linear |
| Dense | 128 | Relu |
| Output | $\lvert\mathcal{A}\rvert$ | softmax |
| $\pi_{b}^{\boldsymbol{\theta}_{b}}$ for each option | | |
| Layer type | Dimension | Activation Function |
| Input | observation size | linear |
| Dense | 10 | Relu |
| Output | $\lvert\mathcal{B}\rvert$ | softmax |
| Critic for each option | | |
| Layer type | Dimension | Activation Function |
| Input | observation size | linear |
| Dense | 256 | Relu |
| Dense | 256 | Relu |
| Output | $\lvert\mathcal{A}\rvert$ | linear |

Table 7: Other hyperparameters for HIL.

| Hyperparameter | value |
|---|---|
| Epochs | 10 |
| Optimizer | Adamax |
| Optimization Algorithm | Mini-Batch Gradient |
| Mini-Batch size | 32 |
| Full Gradient steps per iterations | 10 |
| Learning rate | $10^{-1}$ |

Table 8: Other hyperparameters DOC.

| Hyperparameter | value |
|---|---|
| $\gamma$ | 0.99 |
| $\eta$ | 0.00001 |
| Buffer size | 30000 |
| Batch size | 512 |
| Optimizer | Adamax |
| Optimization Algorithm | Gradient Descent |
| Learning rate $\pi_{lo}$ | $10^{-3}$ |
| Learning rate $\pi_b$ | $10^{-5}$ |