DIVERSE EXPLORATION VIA INFOMAX OPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we study the problem of autonomously discovering temporally abstracted actions, or options, for exploration in reinforcement learning. For learning diverse options suitable for exploration, we introduce the *infomax* termination objective defined as the mutual information between options and their corresponding state transitions. We derive a scalable optimization scheme for maximizing this objective via the termination condition of options, yielding the InfoMax Option Critic (IMOC) algorithm. Through illustrative experiments, we empirically show that IMOC learns diverse options and utilizes them for exploration. Moreover, we show that IMOC scales well to continuous control tasks.

1 INTRODUCTION

Abstracting a course of action as a higher-level action, or an *option* (Sutton et al., 1999), is a key ability for reinforcement learning (RL) agents in several aspects, including *exploration*. In RL problems, an agent learns to approximate an optimal policy only from experience, given no prior knowledge. This leads to the necessity of exploration: an agent needs to explore the poorly known states for collecting environmental information, sometimes sacrificing immediate rewards. For statistical efficiency, it is important to explore the state space in a *deep* and directed manner, rather than taking uniformly random actions (Osband et al., 2019). Options can represent such directed behaviors by capturing long state jumps from their starting regions to terminating regions. It has been shown that well-defined options can facilitate exploration by exploiting an environmental structure (Barto et al., 2013) or, more generally, by reducing decision steps (Fruit and Lazaric, 2017).

A key requirement for such explorative options is *diversity*. If all options have the same terminating region, they will never encourage exploration. Instead, options should lead to a variety of regions for encouraging exploration. However, automatically discovering diverse options in a scalable, online manner is challenging due to two difficulties: generalization and data limitation. Generalization with function approximation (Sutton, 1995) is important for scaling up RL methods to large or continuous domains. However, many existing option discovery methods for exploration are graph-based (e.g., Machado et al. (2017)) and incompatible with function approximation, except for that by Jinnai et al. (2020). Discovering options online in parallel with polices requires us to work with limited data sampled from the environment and train the model for evaluating the diversity in a data-efficient manner.

To address these difficulties, we introduce the *infomax* termination objective defined as the mutual information (MI) between options and their corresponding state transitions. This formulation reflects a simple inductive bias: for encouraging exploration, options should terminate in a variety of regions per starting regions. Thanks to the information-theoretical formulation, this objective is compatible with function approximation and scales up to continuous domains. A key technical contribution of this paper is the optimization scheme for maximizing this objective. Specifically, we employ a simple classification model over options as a *critic* for termination conditions, which makes our method data-efficient and tractable in many domains.

The paper is organized as follows. After introducing background and notations, we present the infomax termination objective and derive a practical optimization scheme using the termination gradient theorem (Harutyunyan et al., 2019). We then implement the infomax objective on the option-critic architecture (OC) (Bacon et al., 2017) with algorithmic modifications, yielding the InfoMax Option Critic (IMOC) algorithm. Empirically, we show that (i) IMOC improves exploration in structured environments, (ii) IMOC improves exporation in lifelong learning, (iii) IMOC is scalable to MuJoCo continuous control tasks, and (iv) the options learned by IMOC are diverse and meaningful. We then relate our method to other option-learning methods and the *empowerment* concept (Klyubin et al., 2005), and finally give concluding remarks.

2 Background and Notation

We assume the standard RL setting in the Markov decision process (MDP), following Sutton and Barto (2018). An MDP $\mathcal{M}$ consists of a tuple $(\mathcal{X}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{X}$ is the set of states, $\mathcal{A}$ is the set of actions, $p : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \rightarrow [0, 1]$ is the state transition function, $r : \mathcal{X} \times \mathcal{A} \rightarrow [r_{\min}, r_{\max}]$ is the reward function, and $0 \leq \gamma \leq 1$ is the discount factor. A policy is a probability distribution over actions conditioned on a state x, $\pi : \mathcal{X} \times \mathcal{A} \rightarrow [0, 1]$. For simplicity, we consider the episodic setting where each episode ends when a terminal state x_T is reached. In this setting, the goal of an RL agent is to approximate a policy that maximizes the expected discounted cumulative reward per episode:

$$J^{\mathrm{RL}}(\pi) = \mathbb{E}_{\pi, x_0}\left[\sum_{t=0}^{T-1} \gamma^t R_t\right], \tag{1}$$

where $R_t = r(x_t, a_t)$ is the reward received at time t, and x_0 is the initial state of the episode. Relatedly, we define the action-value function $Q^\pi(x_t, a_t) \overset{\text{def}}{=} \mathbb{E}_{x_t, a_t, \pi}\left[\sum_{t'=t}^{T-1} \gamma^{t'-t} R_{t'}\right]$ and the state-value function $V^\pi(x_t) \overset{\text{def}}{=} \mathbb{E}_{x_t, \pi} \sum_a \pi(a|x_t) Q^\pi(x_t, a)$.

Assuming that π is differentiable by the policy parameters θ_π, a simple way to maximize the objective (1) is the policy gradient method (Williams, 1992) that estimates the gradient by:

$$\nabla_{\theta_\pi} J^{\mathrm{RL}}(\pi) = \mathbb{E}_{\pi, x_t}\left[\nabla_{\theta_\pi} \log \pi(a_t|x_t) \hat{A}(x_t, a_t)\right], \tag{2}$$

where $\hat{A}(x_t, a_t)$ is the estimation of the advantage function $A^\pi(x_t, a_t) \overset{\text{def}}{=} Q^\pi(x_t, a_t) - V^\pi(x_t)$. A common choice of $\hat{A}(x_t, a_t)$ is N-step TD error $\sum_{i=0}^{N} \gamma^i R_{t+i} + \gamma^N \hat{V}(x_{t+N}) - \hat{V}(x_t)$, where N is a fixed rollout length (Mnih et al., 2016).

2.1 Options Framework

Options (Sutton et al., 1999) provide a framework for representating temporally abstracted actions in RL. An option $o \in \mathcal{O}$ consists of a tuple $(\mathcal{I}^o, \beta^o, \pi^o)$, where $\mathcal{I}^o \subseteq \mathcal{X}$ is the initiation set, $\beta^o : \mathcal{X} \rightarrow [0, 1]$ is a termination function with $\beta^o(x)$ denoting the probability that option o terminates in state x, and π^o is *intra-option* policy. Following related studies (Bacon et al., 2017; Harutyunyan et al., 2019), we assume that $\mathcal{I}^o = \mathcal{X}$ and learn only β^o and π^o. Letting x_s denote an option-starting state and x_f denote an option-terminating state, we can write the option transition function as:

$$P^o(x_f|x_s) = \beta^o(x_f)\mathbb{I}_{x_f = x_s} + (1 - \beta^o(x_s)) \sum_x p^{\pi^o}(x|x_s) P^o(x_f|x), \tag{3}$$

where $\mathbb{I}$ is the indicator function and p^{π^o} is the *policy-induced* transition function $p^{\pi^o}(x'|x) \overset{\text{def}}{=} \sum_{a \in \mathcal{A}} \pi^o(a|x) p(x'|x, a)$. We assume that all options eventually terminate so that P^o is a valid probability distribution over x_f, following Harutyunyan et al. (2019).

To present option-learning methods, we define two option-value functions: $Q_\mathcal{O}$ and $U_\mathcal{O}$, where $Q_\mathcal{O}$ is the option-value function denoting the value of selecting an option o at a state x_t defined by $Q_\mathcal{O}(x_t, o) \overset{\text{def}}{=} \mathbb{E}_{\pi, \beta, \mu}\left[\sum_{t'=t}^{T-1} \gamma^{t'-t} R_t\right]$. Analogously to Q^π and V^π, we let $V_\mathcal{O}$ denote the marginalized option-value function $V_\mathcal{O}(x) \overset{\text{def}}{=} \sum_o \mu(o|x) Q_\mathcal{O}(x, o)$, where $\mu(o|x_s) : \mathcal{X} \times \mathcal{O} \rightarrow [0, 1]$ is the policy over options. Function $U_\mathcal{O}(x, o) \overset{\text{def}}{=} (1 - \beta^o(x)) Q_\mathcal{O}(x, o) + \beta^o(x) V_\mathcal{O}(x)$ is called the option-value function *upon arrival* (Sutton et al., 1999) and denotes the value of reaching a state x_t with o and not having selected the new option.

2.2 Option Critic Architecture

OC (Bacon et al., 2017) provides an end-to-end algorithm for learning π^o and β^o in parallel. To optimize π^o, OC uses the intra-option policy gradient method that is the option-conditional version of the gradient estimator (2), $\nabla_{\theta_{\pi^o}} J^{\mathrm{RL}}(\pi^o) = \mathbb{E}\big[\nabla_{\theta_{\pi^o}} \log \pi^o(a_t|x_t)\hat{A}^o(x_t, a_t)\big]$, where $\hat{A}^o$ is an estimation of the option-conditional advantage A^{π^o}.

For optimizing β^o, OC directly maximizes $Q_{\mathcal{O}}$ using the estimated gradient:

$$\nabla_{\theta_{\beta^o}} Q_{\mathcal{O}}(x, o) = \gamma \mathbb{E}\left[-\nabla_{\theta_{\beta^o}} \beta^o(x)\Big(Q_{\mathcal{O}}(x, o) - V_{\mathcal{O}}(x)\Big)\right]. \tag{4}$$

Intuitively, this decreases the termination probability $\beta^o(x)$ when holding an o is advantageous, i.e., $Q_{\mathcal{O}}(x) - V_{\mathcal{O}}(x)$ is positive, and vice versa. Our method basically follows OC but has a different objective for learning β^o.

2.3 Termination Critic

Recently proposed termination critic (TC) (Harutyunyan et al., 2019) optimizes β^o by maximizing the information-theoretic objective called *predictability*:

$$J^{\mathrm{TC}}(P^o) = -H(X_f|o), \tag{5}$$

where H denotes entropy and X_f is the random variable denoting the option-terminating states. Maximizing $-H(X_f|o)$ makes the terminating region of an option smaller and more predictable. In other words, we can *compress* terminating regions by optimizing the objective (5). To derivate this objective by the beta parameters θ_{β^o}, Harutyunyan et al. (2019) introduced the termination gradient theorem:

Theorem 1. *Let β^o be parameterized with a sigmoid function and ℓ_{β^o} denote the logit of β^o. We have*

$$\nabla_{\theta_\beta} P^o(x_f|x_s) = \sum_x P^o(x|x_s)\nabla_{\theta_\beta}\ell_{\beta^o}(x)(\mathbb{I}_{x_f=x} - P^o(x_f|x)), \tag{6}$$

Leveraging the theorem 1, TC performs gradient ascent using the estimated gradient:

$$\nabla_{\theta_{\beta^o}} J^{\mathrm{TC}}(P^o) = -\mathbb{E}_{x_s,x,x_f}\left[\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\left(\Big(\log P^o_\mu(x) - \log P^o_\mu(x_f)\Big) + \Big(1 - \frac{P^o(x_f|x_s)P^o_\mu(x)}{P^o_\mu(x_f)P^o(x|x_s)}\Big)\right)\right].$$

where $P^o_\mu(x)$ is the marginalized distribution of option-terminating states.

Contrary to the termination objective of OC (4), this objective does not depend on state values, making learned options robust against the reward structure of the environment. Our method is inspired by TC and optimizes a similar information-theoretic objective, not for predictability but for *diversity*. Also, our infomax objective requires an estimation of $\hat{p}(o|x_s, x_f)$ instead of the option transition model $P^o(x_f|x_s)$, which makes our method tractable in more environments.

3 InfoMax Option Critic

We now present the key idea behind the InfoMax Option Critic (IMOC) algorithm. We first formulate the infomax termination objective based on the MI maximization, then derive a practical gradient estimation for maximizing this objective on β^o, utilizing the termination gradient theorem 1.

To evaluate the diversity of options, we use the MI between options and option-terminating states conditioned by option-starting states:

$$J^{\mathrm{IMOC}} = I(X_f; O|X_s) = H(X_f|X_s) - H(X_f|X_s, O), \tag{7}$$

where I denotes conditional MI $I(A; B|Z) = H(A|Z) - H(A|B, Z)$, X_s is the random variable denoting an option-starting state, and O is the random variable denoting an option. We call this objective the *infomax* termination objective. Let us interpret $X_f|X_s$ as the random variable denoting a state transition induced by an option. Then maximizing the MI (7) (i) diversifies a state transition $X_f|X_s$ and (ii) makes an option-conditional state transition $X_f|X_s, o$ more deterministic. Note that the marginalized MI $I(X_f; O)$ also makes sense in that it prevents the terminating region of each option from being too broad, as predictability (5) does. However, in this study, we focus on the conditional objective since it is easier to optimize.

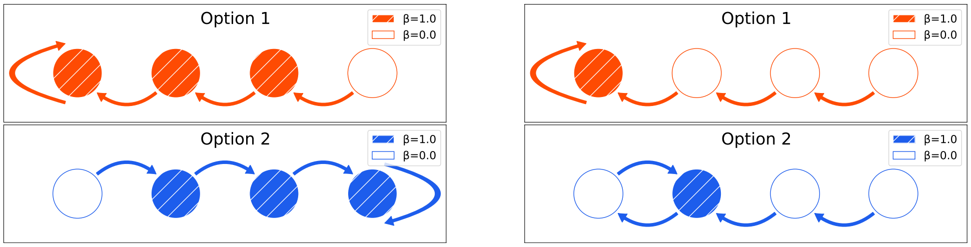


Figure 1: Two insances of InfoMax options in the four state deterministic chain. **Left:** Options are diverse but all state transitions per option are one-step. **Right:** Options enable relatively long state transitions but option-policies are the same at some states.

To illustrate the limitation of infomax options, we conducted an analysis in a toy four-state *deterministic chain* environment, which has four states and two deterministic actions (go left and go right) per each state. Since deriving the exact solution is computationally difficult, we searched options that maximize $H(X_f|X_s)$ from deterministic options that has deterministic option-policies and termination functions (thus has the minimum $H(X_f|X_s, O)$). Among multiple solutions, Figure 1 shows two interesting instances of deterministic infomax options when $|\mathcal{O}| = 2$. The left options enable diverse behaviors per state, although they fail to capture long-term behaviors generally favorable in the literature (e.g., Mann et al. (2015)). On the other hand, the right options enable relatively long, two step state transitions, but they are the same and the rightmost state and the next one. Furthermore, an agent can be caught in a small loop that consists of the leftmost state and the next one. This example shows that (i) we can obtain short and diverse options with only a few options, (ii) to obtain long and diverse options, we need sufficiently many options, and (iii) an agent can be caught in a small loop with only a few options, failing to visit diverse states. As we show in Appendix A, this 'small loop' problem cannot happen with four options. Thus, the number of options is important when we are to maximize the MI (7) and a limitation of this method. However, in experiments, we show that we can practically learn diverse options with relatively small number of options.

For maximizing the MI by gradient ascent, we now derive the gradient of the infomax termination objective (7). First, we estimate the gradient of the objective using the option transition model P^o and marginalized option-transition model $P(x_f|x_s) = \sum_o \mu(o|x_s)P^o(x_f|x_s)$.

Proposition 1. *Let β^o be parameterized with a sigmoid function. Given a trajectory $\tau = x_s, \ldots, x, \ldots, x_f$ sampled by π^o and β^o, we can obtain unbiased estimations of $\nabla_{\theta_\beta} H(X_f|X_s)$ and $\nabla_{\theta_\beta} H(X_f|X_s, O)$ by*

$$\nabla_{\theta_\beta} H(X_f|X_s) = \mathbb{E}_{x_s,x,x_f,o}\left[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log P(x|x_s) - \log P(x_f|x_s)\Big)\right] \tag{8}$$

$$\nabla_{\theta_\beta} H(X_f|X_s, O) = \mathbb{E}_{x_s,x,x_f,o}\left[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log P^o(x|x_s) - \log P^o(x_f|x_s)\Big)\right] \tag{9}$$

where $\ell_{\beta^o}(x)$ denotes the logit of $\beta^o(x)$.

Note that the additional term β^o is necessary because x is not actually a terminating state. The proof follows section 4 in Harutyunyan et al. (2019) and is given in Appendix B.1.

The estimated gradient of the infomax termination objective (7) can now be written as:

$$\begin{aligned}&\nabla_{\theta_\beta} I(X_f; O|X_s) = \nabla_{\theta_\beta} H(X_f|X_s) - \nabla_{\theta_\beta} H(X_f|X_s, O)\\ &= \mathbb{E}_{x_s,x,x_f,o}\left[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\left(\log P(x|x_s) - \log P(x_f|x_s) - (\log P^o(x|x_s) - \log P^o(x_f|x_s))\right)\right],\end{aligned} \tag{10}$$

which means that we can optimize this objective by estimating P^o and P. However, estimating the probability over the state space can be difficult, especially when the state space is large, as common in the deep RL setting. Hence, we reformulate the gradient using Bayes' rule in a similar way as Gregor et al. (2017). The resulting term consists of the reverse option transition $p(o|x_s, x_f)$ that denotes the probability of having an o given a state transition x_s, x_f.

Proposition 2. *We now have*

$$\begin{aligned}\nabla_{\theta_\beta} I(X_f; O|X_s) =& \nabla_{\theta_\beta} H(X_f|X_s) - \nabla_{\theta_\beta} H(X_f|X_s, O) \\ =& \mathbb{E}_{x_s,x,x_f,o}\left[\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log p(o|x_s,x) - \log p(o|x_s,x_f)\Big)\right] \end{aligned} \tag{11}$$

The proof is given in Appendix B.2. In the following sections, we estimate the gradient (11) by learning a classification model over options $\hat{p}(o|x_s, x_f)$ from sampled option transitions.

4 Algorithm

In this section, we introduce modifications for adjusting the OC (Bacon et al., 2017) to our infomax termination objective. Specifically, we implement IMOC on top of Advantage-Option Critic (AOC), a synchronous variant of A2OC (Harb et al., 2018), yielding the Advantage-Actor InfoMax Option Critic (A2IMOC) algorithm. To stably estimates $\hat{p}(o|x_s, x_f)$ for updating θ_β, we sample recent option state transitions o, x_s, x_f from We follow AOC for optimizing option-policies except the following modifications and give a full description of A2IMOC in Appendix C.1. In continuous control experiments, we also used Proximal Policy InfoMax Option Critic (PPIMOC) that is an implementation of IMOC based on PPO (Schulman et al., 2017). We give details of PPIMOC in Appendix C.2.

Upgoing Option-Advantage Estimation Previous studies (e.g., Harb et al. (2018)) estimated the advantage $\hat{A}^{o_t}(x_t)$ ignoring the future rewards after the current option o_t terminates. Since longer rollout length often helps speed up learning (Sutton and Barto, 2018), it is preferable to extend this estimation to use all available future rewards. However, future rewards after option termination heavily depends on the selected option, often leading to underestimation of $\hat{A}^o$. Thus, to effectively use future rewards, we introduce an *upgoing* option-advantage estimation (UOAE). Let $t + k$ denote the time step where the current option o_t terminates in a sampled trajectory. Then, UOAE estimates the advantage by:

$$\hat{A}^o_{\text{UOAE}} = -Q_{\mathcal{O}}(x_t, o_t) + \begin{cases} \sum_{i=0}^{k} \gamma^i R_{t+i} + \underbrace{\max\left(\sum_{j=k}^{N} \gamma^j R_{t+j},\ \gamma^k V_{\mathcal{O}}(x_{t+k})\right)}_{\text{upgoing estimation}} & (k < N) \\ \sum_{i=0}^{N} \gamma^i R_{t+i} + \gamma^N U_{\mathcal{O}}(x_{t+N}, o_t) & (otherwise) \end{cases}. \tag{12}$$

Similar to upgoing policy update (Vinyals et al., 2019), the idea is to be optimistic about the future rewards after option termination by taking the maximum with $V_{\mathcal{O}}$.

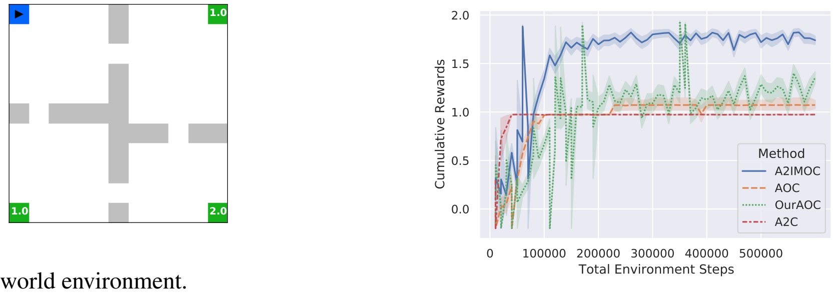
Policy Regularization Based on Mutual Information To perform MI maximization not only on termination functions but also on option-policies, we introduce a policy regularization based on the maximization of the conditional MI, $I(A; O|X_s)$, where A is the random variable denoting an action. This MI can be interpreted as a local approximation of the infomax objective (7), assuming that each action leads to different terminating regions. Although optimizing the infomax termination objective diversifies option-policies implicitly, we found that this regularization helps learn diverse option-policies reliably. Letting π_μ denote the marginalized policy $\pi_\mu(a|x) \overset{\text{def}}{=} \sum_o \mu(o|x)\pi^o(a|x)$, we write $I(A; O|X_s)$ as:

$$I(A; O|X_s) = H(A|X_s) - H(A|O, X_s) = \mathbb{E}_{x_s}\left[H(\pi_\mu(x_s))\right] - \mathbb{E}_{x_s,o}\left[H(\pi^o(x_s))\right].$$

We use this regularization with the entropy bonus (maximization of $H(\pi^o)$) common in policy gradient methods (Williams and Peng, 1991; Mnih et al., 2016) and write the overall regularization term as

$$c_{H_\mu} H(\pi_\mu(x)) + c_H H(\pi^o(x)), \tag{13}$$

where c_{H_μ} and c_H are weights of each regularization term. Note that we add this regularization term on not only option-starting states but all sampled states. This introduces some bias, which we did



(a) Gridworld environment.
Blue grid is the start and green grids are goals.

(b) Performance progression.

Figure 2: Single Task learning in Gridworld Four Rooms.

not find to be harmful when c_{H_μ} is reasonably small. To approximate $H(\pi_\mu)$, we employ $\hat{\mu}$ that is the empirical estimation of μ. Using $\hat{\mu}$, $H(\pi_\mu)$ is computed by $\pi_\mu(a|x) \approx \sum_o \hat{\mu}(o|x)\pi^o(a|x)$ for descrete action spaces and approximated by Monte Carlo method for continuous action spaces. We show the details in Appendix C.1.

5 Experiments

We conducted a series of experiments to show two use cases of IMOC: exploration in structured environments and exploration for lifelong learning (Brunskill and Li, 2014). In this section, we used four options for all option-learning methods and compared the number of options in Appendix D.7.

5.1 Single Task Learning in Structured Environments

We consider two 'Four Rooms' domains, where diverse options are beneficial for utilizing environmental structures.

Gridworld Four Rooms with Suboptimal Goals First, we tested IMOC in a variant of the classical Four Rooms Gridworld (Sutton et al., 1999) with suboptimal goals. An agent is initially placed at the upper left room and receives a positive reward only at goal states: two closer goals with $+1$ reward and the farthest goal with $+2$ reward, as shown in Figure 2a. The episode ends when an agent reaches one of the goals. The optimal policy is aiming the farthest goal in the lower right room without converging to suboptimal goals. Thus, an agent is required to learn multimodal behaviors leading to multiple goals, which options can help. In this environment, we compared A2IMOC with A2C (Mnih et al., 2016; Wu et al., 2017), AOC, and our tuned version of AOC (our AOC) with all enhancements presented in section 4 to highlight the effectiveness of the termination objective among all of our improvements.[1] We show the progress of average cumulative rewards over ten trials in Figure 2b. A2IMOC performed the best and found the optimal goal in most trials. AOC and our AOC also occasionally found the optimal goal, while A2C overfitted to either of the suboptimal goals through all trials.

Figure 3 illustrates learned option-polices and termination functions of each compared method.[2] Terminating regions learned with A2IMOC is diverse. For example, option 0 mainly terminates in the right rooms while option 3 terminates in the left rooms. We see that termination regions learned by A2IMOC are diverse and clearly separated per each option. Although all option policies converged to the same near optimal one, we show that A2IMOC diversifies option-policies at the beginning of learning in Appendix D.4. On the other hand, terminating regions learned with AOC overlap each other, and notably, option 3 has no terminating region. We assume this is because the loss function (4) decreases the terminating probability when the advantage is positive. We can see the same tendency in our AOC, although we cannot see the vanishment of the terminating regions.

[1] Note that we did not include ACTC (Harutyunyan et al., 2019) for comparison since we failed to reliably reproduce the reported results with our implementation of ACTC.

[2] Note that we choose the best model from multiple trials for visualization throughout the paper.

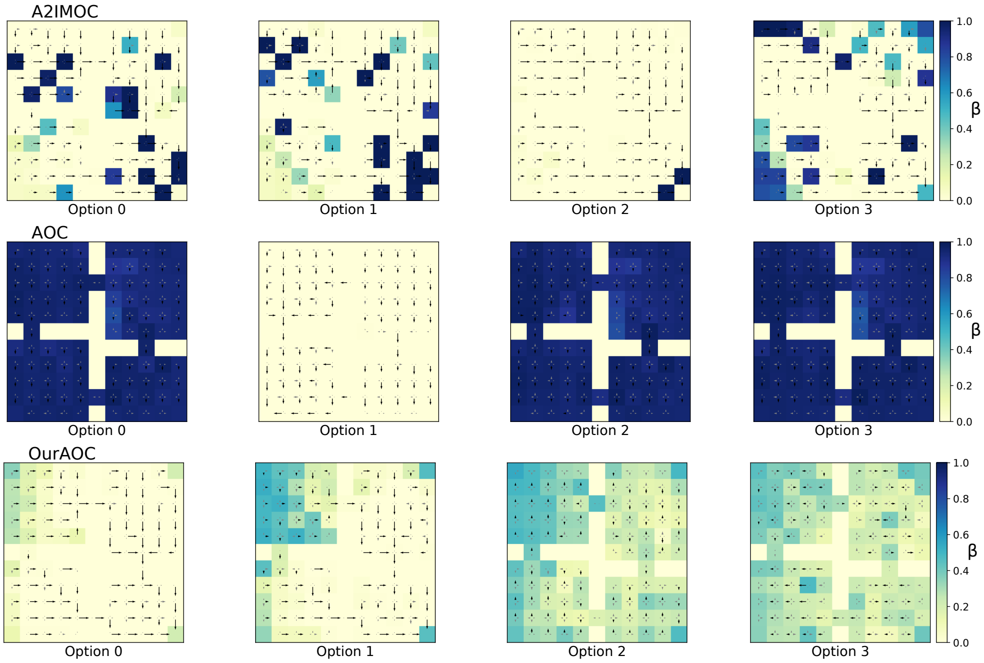
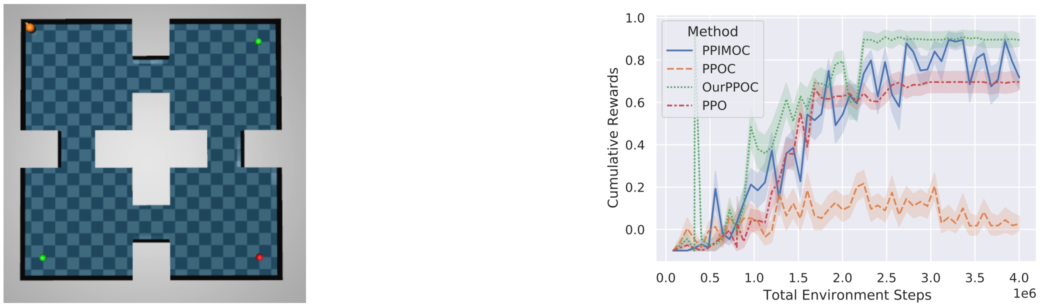


Figure 3: Learned option-policies (π^o) and termination probabilities (β^o) for each option in Gridworld Four Rooms after 6×10^5 steps. Arrows show the probablities of each action and heatmaps show probabilities of each β^o. **First row:** A2IMOC. Terminating regions are clearly different each other and there are a few overlapped regions. **Second row:** AOC. Almost all states has high termination probablity with option $0, 1, 3$ and option 2 has no terminating region. **Third row:** Our AOC. Termination regions are not clearly separated per each option.

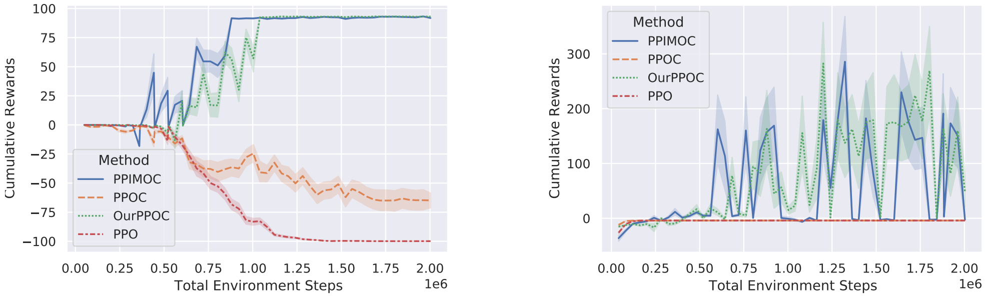


(a) MuJoCo Point Four Rooms. The agent is an orange ball and there are three goals: green goals are suboptimal (+0.5) and the red one in optimal (+1.0).

(b) Performance progression.

Figure 4: Single Task learning in MuJoCo Point Four Rooms.

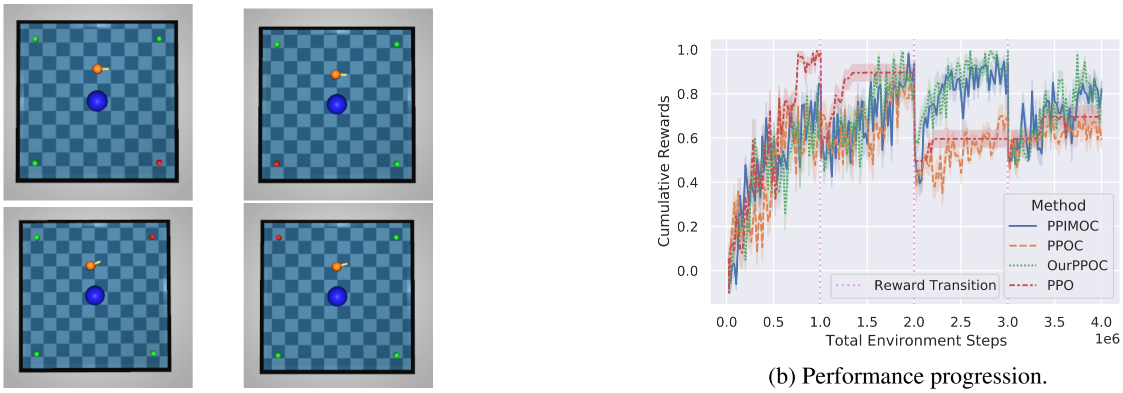
MuJoCo Point Four Rooms To show the scalability of IMOC in continuous domains, we conducted experiments in a similar four rooms environment, based on the MuJoCo (Todorov et al., 2012) physics simulator and "PointMaze" environment in rllab (Duan et al., 2016). This environment follows the Gridworld Four Rooms and has three goals as shown in Figure 4a: two green goals with $+0.5$ reward, and a red one with $+1.0$. An agent controls the rotation and velocity of the orange ball and receives a positive reward only at the goals. In this environment, we compared PPIMOC with PPOC (Klissarov et al., 2017), PPO, and our tuned version of PPOC (our PPOC) that is the same as PPIMOC except for the termination objective. An important difference is that PPOC uses a parameterized μ trained by policy gradient, while PPIMOC and our PPOC use ϵ-greedy for option selection. Figure 4b show the progress of average cumulative rewards over five trials. PPIMOC found the optimal goal four times in five trials and performed slightly bettern than PPO. In a qualitative



(a) Performance progression in Mountain Car.

(b) Performance progression in Cartpole swing up.

Figure 5: Single Task learning in MuJoCo Point Four Rooms.



(b) Performance progression.

(a) MuJoCo Point Billiard. Four goals periodically move clockwise.

Figure 6: Lifelong learning in MuJoCo Point Billiard.

analysis, we observed the same tendency in learned options as the Gridworld experiment, where the details are given in Appendix D.5.

5.2 Single Task Learning in Classical Continuous Control

Additionally, we test PPIMOC on two classical, hard-exploration control problems: Mountain Car (Barto et al., 1983) and Cartpole swingup (Moore, 1990). In Mountain Car, PPIMOC and our PPOC successfully learns to reache the goal, while PPO and PPOC converged to run around the start posision. In Cartpole swing up, PPIMOC and our PPOC performed better than PPO and PPOC, but still failed to learn stable behaviors.

5.3 Exploration for Lifelong Learning

As another interesting application of IMOC, we consider the lifelong learning setting. Specifically, we tested IMOC in ’Point Billiard’ environment. In this environment, an agent receives a positive reward only when the blue *objective* ball reaches the goal, pushed by the agent (orange ball). Figure 6a shows all four configurations of Point Billiard that we used. There are four goals: green goals with $+0.5$ reward and a red one with $+1.0$ reward. The positions of four goals move clockwise after 1M environmental steps and agents need to adapt to the new positions of goals. We compared PPIMOC with PPO, PPOC, and our PPOC in this environment. Figure 6b shows the progress of average cumulative rewards over five trials. Both PPIMOC and our PPOC performed the best and adapted to all reward transitions. On the other hand, PPO and PPOC struggle to adapt to the second transition, where the optimal goal moves behind the agent. The ablation study given in Appendix D.6 shows that UOAE (12) works effectively in this task. However, without UOAE, PPIMOC still outperformed PPO. Thus, we argue that having diverse terminating regions itself is beneficial for adapting to new reward functions in environments with subgoals.

6 Related Work

Options for Exploration Options (Sutton et al., 1999) in RL are widely studied for many applications, including speeding up planning (Mann and Mannor, 2014) and transferring skills (Konidaris and Barto, 2007; Castro and Precup, 2010). However, as discussed by Barto et al. (2013), their benefits for exploration are less well recognized. Many existing methods focused on discovering subgoals that effectively decompose the problem then use such subgoals for encouraging exploration. Subgoals are discovered based on various properties, including graphical features of state transitions (Simsek and Barto, 2008; Machado et al., 2017; Jinnai et al., 2019) and causality (Jonsson and Barto, 2006; Vigorito and Barto, 2010). In contrast, our method directly optimizes termination functions instead of discovering subgoals, capturing environmental structures implicitly. From a theoretical perspective, Fruit and Lazaric (2017) analyzed that good options can improve the exploration combined with state-action visitation bonuses. Using infomax options with visitation bonuses would be an interesting future direction.

End-to-end learning of Options While many studies attempted to learn options and option-policies separately, Bacon et al. (2017) proposed OC to train option-policies and termination functions in parallel. OC has been extended with various types of inductive biases, including deliberation cost (Harb et al., 2018), interest (Khetarpal et al., 2020), and safety (Jain et al., 2018). Our study is directly inspired by an information-theoretic approach presented by Harutyunyan et al. (2019), as we noted in section 2.

Mutual Information and Skill Learning MI often appears in the literature of *intrinsically motivated* (Singh et al., 2004) reinforcement learning, as a driver of goal-directed behaviors. A well-known example is the *empowerment* (Klyubin et al., 2005; Salge et al., 2013), which is obtained by maximizing the MI between sequential k actions and the resulting state $I(a_t, ..., a_{t+k}; x_{t+k}|x_t)$. Some works (Mohamed and Rezende, 2015; Zhao et al., 2020) implemented lower bound maximization of empowerment as intrinsic rewards for RL agents, encouraging goal-directed behaviors in the absence of extrinsic rewards We can interpret our objective $I(X_f; O|X_s)$ as empowerment between limited action sequences and states corresponding to options. Gregor et al. (2017) employed this interpretation and introduced a method for maximizing the variational lower bound of this MI via option-policies, using the same model as our $\hat{p}$, while we aim to maximize the MI via termination functions. MI is also used for intrinsically motivated discovery of skills, assuming that diversity is important to acquire useful skills. Eysenbach et al. (2019) proposed to maximize MI between skills and states $I(O; X)$, extended to the conditional one $I(O; X'|X)$ by Sharma et al. (2020). Although our study shares the same motivation for using MI as these methods, i.e. diversifying sub-policies, the process of MI maximization is significantly different: our method optimizes termination functions, while their methods optimize conditional policies by using MI as intrinsic rewards.

7 Conclusion

We presented a novel end-to-end option learning algorithm InfoMax Option Critic (IMOC) that uses the infomax termination objective to diversify options. Empirically, we showed that IMOC improves exploration in structured environments and for lifelong learning, even in continuous control tasks. We also quantitatively showed the diversity of learned options. An interesting future direction would be combining our method for learning termination conditions with other methods for learning option-policies, e.g., by using MI as intrinsic rewards. A limitation of the infomax objective presented in this study is that it requires *on-policy* data for training. Hence, another interesting line of future work is extending IMOC to use for *off-policy* option discovery.

References

P. Bacon, J. Harb, and D. Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1726–1734, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14858.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.*, 13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077. URL `https://doi.org/10.1109/TSMC.1983.6313077`.

A. G. Barto, G. D. Konidaris, and C. M. Vigorito. Behavioral hierarchy: Exploration and representation. In G. Baldassarre and M. Mirolli, editors, *Computational and Robotic Models of the Hierarchical Organization of Behavior*, pages 13–46. Springer, 2013. doi: 10.1007/978-3-642-39875-9_2. URL `https://doi.org/10.1007/978-3-642-39875-9_2`.

E. Brunskill and L. Li. Pac-inspired option discovery in lifelong reinforcement learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 316–324. JMLR.org, 2014. URL `http://proceedings.mlr.press/v32/brunskill14.html`.

P. S. Castro and D. Precup. Using bisimulation for policy transfer in mdps. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1907`.

Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1329–1338. JMLR.org, 2016. URL `http://proceedings.mlr.press/v48/duan16.html`.

B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL `https://openreview.net/forum?id=SJx63jRqFm`.

R. Fruit and A. Lazaric. Exploration-exploitation in mdps with options. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 576–584, 2017. URL `http://proceedings.mlr.press/v54/fruit17a.html`.

K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Skc-Fo4Yg`.

J. Harb, P. Bacon, M. Klissarov, and D. Precup. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3165–3172, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17421`.

A. Harutyunyan, W. Dabney, D. Borsa, N. Heess, R. Munos, and D. Precup. The termination critic. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2231–2240, 2019. URL `http://proceedings.mlr.press/v89/harutyunyan19a.html`.

M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=SJ6yPD5xg`.

A. Jain, K. Khetarpal, and D. Precup. Safe option-critic: Learning safety in the option-critic architecture. *CoRR*, abs/1807.08060, 2018. URL `http://arxiv.org/abs/1807.08060`.

Y. Jinnai, J. W. Park, D. Abel, and G. D. Konidaris. Discovering options for exploration by minimizing cover time. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International*

Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 3130–3139. PMLR, 2019. URL http://proceedings.mlr.press/v97/jinnai19b.html.

Y. Jinnai, J. W. Park, M. C. Machado, and G. D. Konidaris. Exploration in reinforcement learning with deep covering options. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeIyaVtwB.

A. Jonsson and A. G. Barto. Causal graph based decomposition of factored mdps. *J. Mach. Learn. Res.*, 7:2259–2301, 2006. URL http://jmlr.org/papers/v7/jonsson06a.html.

K. Khetarpal, M. Klissarov, M. Chevalier-Boisvert, P. Bacon, and D. Precup. Options of interest: Temporal abstraction with interest functions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 4444–4451. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5871.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

M. Klissarov, P. Bacon, J. Harb, and D. Precup. Learnings options end-to-end for continuous action tasks. *CoRR*, abs/1712.00004, 2017. URL http://arxiv.org/abs/1712.00004.

A. S. Klyubin, D. Polani, and C. L. Nehaniv. All else being equal be empowered. In M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, editors, *Advances in Artificial Life, 8th European Conference, ECAL 2005, Canterbury, UK, September 5-9, 2005, Proceedings*, volume 3630 of *Lecture Notes in Computer Science*, pages 744–753. Springer, 2005. doi: 10.1007/11553090_75. URL https://doi.org/10.1007/11553090_75.

G. D. Konidaris and A. G. Barto. Building portable options: Skill transfer in reinforcement learning. In M. M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 895–900, 2007. URL http://ijcai.org/Proceedings/07/Papers/144.pdf.

M. C. Machado, M. G. Bellemare, and M. H. Bowling. A laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2295–2304, 2017. URL http://proceedings.mlr.press/v70/machado17a.html.

T. A. Mann and S. Mannor. Scaling up approximate value iteration with options: Better policies with fewer iterations. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 127–135. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/mann14.html.

T. A. Mann, S. Mannor, and D. Precup. Approximate value iteration with temporally extended actions. *J. Artif. Intell. Res.*, 53:375–438, 2015. doi: 10.1613/jair.4676. URL https://doi.org/10.1613/jair.4676.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1928–1937, 2016. URL http://jmlr.org/proceedings/papers/v48/mniha16.html.

S. Mohamed and D. J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information*

Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2125–2133, 2015. URL http://papers.nips.cc/paper/5668-variational-information-maximisation-for-intrinsically-motivated-reinforcement-learning.

A. W. Moore. Efficient memory-based learning for robot control. Technical Report UCAM-CL-TR-209, University of Cambridge, Computer Laboratory, Nov. 1990. URL https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-209.pdf.

I. Osband, B. V. Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20:124:1–124:62, 2019. URL http://jmlr.org/papers/v20/18-339.html.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.

C. Salge, C. Glackin, and D. Polani. Approximation of empowerment in the continuous domain. *Advances in Complex Systems*, 16(2-3), 2013. doi: 10.1142/S0219525912500798. URL https://doi.org/10.1142/S0219525912500798.

A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6120.

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015a. URL http://proceedings.mlr.press/v37/schulman15.html.

J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015b. URL http://arxiv.org/abs/1506.02438.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

Ö. Simsek and A. G. Barto. Skill characterization based on betweenness. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1497–1504. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3411-skill-characterization-based-on-betweenness.

S. P. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1281–1288, 2004. URL http://papers.nips.cc/paper/2552-intrinsically-motivated-reinforcement-learning.

R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

R. S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pages 1038–1044. MIT Press, 1995. URL http://papers.nips.cc/paper/1109-generalization-in-reinforcement-learning-successful-examples-using-sparse-coarse-coding.

R. S. Sutton, D. Precup, and S. P. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109.

C. M. Vigorito and A. G. Barto. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Trans. Auton. Ment. Dev.*, 2(2):132–143, 2010. doi: 10.1109/TAMD.2010.2050205. URL https://doi.org/10.1109/TAMD.2010.2050205.

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z. URL https://doi.org/10.1038/s41586-019-1724-z.

R. Williams and J. Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3:241–, 09 1991. doi: 10.1080/09540099108946587.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5279–5288, 2017. URL http://papers.nips.cc/paper/7112-scalable-trust-region-method-for-deep-reinforcement-learning-using-kronecker-factored-approximation.

R. Zhao, P. Abbeel, and S. Tiomkin. Efficient online estimation of empowerment for reinforcement learning. *CoRR*, abs/2007.07356, 2020. URL https://arxiv.org/abs/2007.07356.

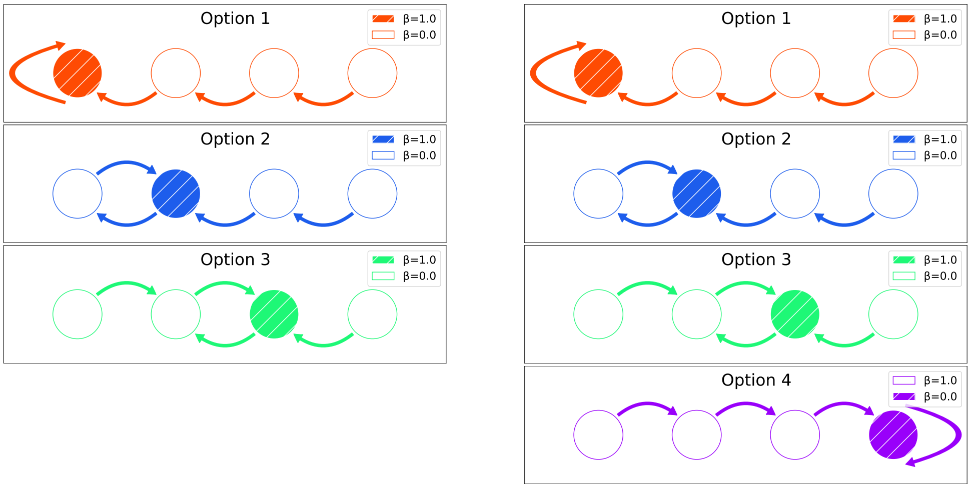


Figure 7: InfoMax options in the four state deterministic chain. **Left:** With three options. **Right:** With four options.

A MORE ANALYSIS ON DETERMINISTIC CHAIN EXAMPLE

Figure 7 shows infomax options with three options and four options in the four state deterministic chain example. Among multiple solutions, we selected options with an absorbing state per option (i.e., $\beta^o(x) = 1.0$ for only one x), which are partially the same as the right options in Figure 1. With four options, $\Pr(x_f|x_s) = 0.25$ for all x_f and x_s, thus $H(X_f|X_s)$ is the maximum. This example shows that we need sufficiently many options for maximizing the MI, otherwise an agent can be caught in a small loop as we described in Section 3.

B OMITTED PROOFS

B.1 PROOF OF PROPOSITION 1

First, we repeat the assumption 2 in Harutyunyan et al. (2019).

Assumption 1. *The distribution $d^\mu(\cdot|o)$ over the starting states of an option o under policy μ is independent of its termination condition β^o.*

Note that this assumption does not strictly hold since ϵ-Greedy option selection depends on β^o via $Q_{\mathcal{O}}$. However, since this dependency is not so strong, we found that β^o reliably converged in our experiments.

Lemma 1. *Assume that the distribution $d^\mu(\cdot|o)$ over the starting state of an option o under policy μ is independent with β^o. Then the following equations hold.*

$$\nabla_{\theta_\beta} H(X_f|X_s) = -\sum_{x_s,o} d^\mu(x_s,o) \sum_x P^o(x|x_s)\nabla_{\theta_\beta}\ell_{\beta^o}(x)\Big[\log P(x|x_s) + 1 - \sum_{x_f} P^o(x_f|x)\Big(\log P(x_f|x_s) + 1\Big)\Big] \tag{14}$$

$$\nabla_{\theta_\beta} H(X_f|X_s,O) = -\sum_{x_s,o} d^\mu(x_s,o) \sum_x P^o(x|x_s)\nabla_{\theta_\beta}\ell_{\beta^o}(x)\Big[\log P^o(x|x_s) + 1 - \sum_{x_f} P^o(x_f|x)\Big(\log P^o(x_f|x_s) + 1\Big)\Big], \tag{15}$$

Sampling x_s, x, x_f, o from d^μ and P^o,

$$\nabla_{\theta_\beta} H(X_f|X_s) = \mathbb{E}_{x_s,x,x_f,o}\Big[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log P(x|x_s) - \log P(x_f|x_s)\Big)\Big] \tag{16}$$

$$\nabla_{\theta_\beta} H(X_f|X_s,O) = \mathbb{E}_{x_s,x,x_f,o}\Big[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log P^o(x|x_s) - \log P^o(x_f|x_s)\Big)\Big]. \tag{17}$$

Proof of Lemma 1

Proof. First, we prove Equation (14). Let $d^\mu(x_s)$ denote the probability distribution over x_s under the policy μ, or the marginal distribution of $d^\mu(x_s|o)$, and $d^\mu(x_s,o)$ denote the joint distribution of x_s and o. Then, we have:

$$\begin{aligned}
&\nabla_{\theta_\beta} H(X_f|X_s) = -\nabla_{\theta_\beta} \sum_{x_s} d^\mu(x_s) \sum_{x_f} P(x_f|x_s) \log P(x_f|x_s) \\
&= -\sum_{x_s} d^\mu(x_s) \sum_{x_f} \Big(\nabla_{\theta_\beta} P(x_f|x_s) \log P(x_f|x_s) + P(x_f|x_s) \frac{\nabla_{\theta_\beta} P(x_f|x_s)}{P(x_f|x_s)} \Big) \\
&= -\sum_{x_s} d^\mu(x_s) \sum_{x_f} \nabla_{\theta_\beta} P(x_f|x_s) \Big(\log P(x_f|x_s) + 1 \Big) \\
&= -\sum_{x_s} d^\mu(x_s) \sum_{x_f} \sum_{o} \mu(o|x_s) \underbrace{\nabla_{\theta_\beta} P^o(x_f|x_s)}_{\text{Apply theorem (6)}} \Big(\log P(x_f|x_s) + 1 \Big) \\
&= -\sum_{x_s} d^\mu(x_s) \sum_{x_f} \sum_{o} \mu(o|x_s) \sum_{x} P^o(x|x_s) \nabla_{\theta_\beta} \ell_{\beta^o}(x) (\mathbb{I}_{x_f=x} - P^o(x_f|x)) \Big(\log P(x_f|x_s) + 1 \Big) \\
&= -\sum_{x_s} d^\mu(x_s) \sum_{o} \mu(o|x_s) \sum_{x} P^o(x|x_s) \nabla_{\theta_\beta} \ell_{\beta^o}(x) \sum_{x_f} (\mathbb{I}_{x_f=x} - P^o(x_f|x)) \Big(\log P(x_f|x_s) + 1 \Big) \\
&= -\underbrace{\sum_{x_s,o} d^\mu(x_s,o)}_{\text{sample}} \underbrace{\sum_{x} P^o(x|x_s)}_{\text{sample}} \nabla_{\theta_\beta} \ell_{\beta^o}(x) \times \Big[\log P(x|x_s) + 1 - \underbrace{\sum_{x_f} P^o(x_f|x)}_{\text{sample}} \Big(\log P(x_f|x_s) + 1 \Big) \Big].
\end{aligned}$$

Sampling x_s, x, x_f, o, we get (16).

Then we prove Equation (15).

$$\begin{aligned}
&\nabla_{\theta_\beta} H(X_f|X_s, O) = -\nabla_{\theta_\beta} \sum_{x_s,o} d^\mu(x_s,o) \sum_{x_f} P^o(x_f|x_s) \log P^o(x_f|x_s) \\
&= -\sum_{x_s,o} d^\mu(x_s,o) \sum_{x_f} \Big(\nabla_{\theta_\beta} P^o(x_f|x_s) \log P^o(x_f|x_s) + P^o(x_f|x_s) \frac{\nabla_{\theta_\beta} P^o(x_f|x_s)}{P^o(x_f|x_s)} \Big) \\
&= -\sum_{x_s,o} d^\mu(x_s,o) \sum_{x_f} \underbrace{\nabla_{\theta_\beta} P^o(x_f|x_s)}_{\text{Apply theorem (6)}} \Big(\log P^o(x_f|x_s) + 1 \Big) \\
&= -\sum_{x_s,o} d^\mu(x_s,o) \sum_{x_f} \sum_{x} P^o(x|x_s) \nabla_{\theta_\beta} \ell_{\beta^o}(x) (\mathbb{I}_{x_f=x} - P^o(x_f|x)) \Big(\log P^o(x_f|x_s) + 1 \Big) \\
&= -\sum_{x_s,o} d^\mu(x_s,o) \sum_{x} P^o(x|x_s) \nabla_{\theta_\beta} \ell_{\beta^o}(x) \sum_{x_f} (\mathbb{I}_{x_f=x} - P^o(x_f|x)) \Big(\log P^o(x_f|x_s) + 1 \Big) \\
&= -\underbrace{\sum_{x_s,o} d^\mu(x_s,o)}_{\text{sample}} \underbrace{\sum_{x} P^o(x|x_s)}_{\text{sample}} \nabla_{\theta_\beta} \ell_{\beta^o}(x) \times \Big[\log P^o(x|x_s) + 1 - \underbrace{\sum_{x_f} P^o(x_f|x)}_{\text{sample}} \Big(\log P^o(x_f|x_s) + 1 \Big) \Big]
\end{aligned}$$

□

Sampling x_s, x, x_f, o, we get (17).

B.2 Proof of Proposition 2

Proof. First, we have that:

$$
\begin{aligned}
\log P^o(x_f|x_s) - \log P(x_f|xs) &= \log \frac{P^o(x_f|x_s)}{P(x_f|xs)} \\
&= \log \frac{\Pr(x_f|x_s, o)}{P(x_f|xs)} \\
&= \log \frac{\Pr(x_s, x_f, o)\Pr(x_s)}{\Pr(x_s, x_f)\Pr(x_s, o)} \\
&= \log \frac{\Pr(x_s, x_f, o)}{\Pr(x_s, x_f)\Pr(o|x_s)} \\
&= \log \frac{\Pr(o|x_s, x_f)\Pr(x_s, x_f)}{\Pr(x_s, x_f)\Pr(o|x_s)} \\
&= \log \frac{p(o|x_s, x_f)}{\mu(o|x_s)}
\end{aligned}
$$

Using this equation, we can rewrite the equation (10) as:

$$
\begin{aligned}
\nabla_{\theta_\beta} I(X_f; O|X_s) &= \nabla_{\theta_\beta} H(X_f|X_s) - \nabla_{\theta_\beta} H(X_f|X_s, O) \\
&= \mathbb{E}_{x_s, x, x_f, o}\Big[-\nabla_{\theta_\beta}\ell_{\beta^o}(x)\beta^o(x)\Big(\log P(x|x_s) - \log P(x_f|x_s) - \log P^o(x|x_s) + \log P^o(x_f|x_s)\Big)\Big] \\
&= \mathbb{E}_{x_s, x, x_f, o}\Big[\nabla_{\theta_\beta}\ell_{\beta^o}(x)\Big(\Big(\log P^o(x|x_s) - \log P(x|x_s)\Big) - \Big(\log P^o(x_f|x_s) - \log P(x_f|x_s)\Big)\Big)\Big] \\
&= \mathbb{E}_{x_s, x, x_f, o}\Big[\nabla_{\theta_\beta}\ell_{\beta^o}(x)\Big(\log \frac{p(o|x_s, x)}{\mu(o|x_s)} - \log \frac{p(o|x_s, x_f)}{\mu(o|x_s)}\Big)\Big] \\
&= \mathbb{E}_{x_s, x, x_f, o}\Big[\nabla_{\theta_\beta}\ell_{\beta^o}(x)\Big(\log p(o|x_s, x) - \log p(o|x_s, x_f)\Big)\Big]
\end{aligned}
$$

□

C Implementation Details

C.1 The whole algorithm of A2IMOC

Algorithm 1 shows a full description of A2IMOC. It follows the architecture of A2C (Mnih et al., 2016; Wu et al., 2017) and has multiple synchronous actors and a single learner. At each optimization step, we update π^o, $Q_{\mathcal{O}}$, and β^o from online trajectories collected by actors. We update $\hat{p}(o|x_s, x_f)$ for estimating the gradient (11) and $\hat{\mu}(o|x_s)$ for entropy regularization (13). To learn $\hat{p}$ and $\hat{\mu}$ stably, we maintain a replay buffer $B_{\mathcal{O}}$ that stores option-transitions, implemented by a LIFO queue. Note that using older o, x_s, x_f sampled from the buffer can introduce some bias in the learned $\hat{p}$ and $\hat{\mu}$, since they depend on the current π^o and β^o. However, we found that this is not harmful when the capacity of the replay buffer is reasonably small.

We also add maximization of the entropy of β^o to the loss function for preventing the termination probability saturating on zero or one. Then the full objective of β^o is written as:

$$
\log \hat{p}(o|x_s, x) - \log \hat{p}(o|x_s, x_f) + c_{H_\beta} H(\beta^o(x)),
$$

where c_{H_β} is a weight of entropy bonus.

C.2 Implementation of PPIMOC

For continuous control tasks, we introduce PPIMOC on top of PPO, with the following modifications to A2IMOC.

Algorithm 1 Advantage-Actor InfoMax Option Critic (A2IMOC)

```
Given: Initial option-value $Q_\mathcal{O}$, option-policy $\pi^o$, and termination function $\beta^o$.
Let $B_\mathcal{O}$ be a replay buffer for storing option-transitions.
for $k = 1, ...$ do
    for $i = 1, 2, ..., N$ do                                  ▷ Collect experiences from environment
        Sample termination variable $b_i$ from $\beta^{o_i}(x_i)$
        if $b_i = 1$ then
            Store option transition $x_s, x_f, o_i$ to $B_\mathcal{O}$
        end if
        Choose next option $o_{i+1}$ by $\epsilon$-Greedy
        Receive reward $R_i$ and state $x_{i+1}$, taking $a_i \sim \pi_{o_{i+1}}(x_i)$
    end for
    for all $x_i$ in the trajectory do ▷ Train option-policy, option-values, and termination function
        Update $\pi^o(a_i|x_i)$ with PG via the UOAE advantage (12) and policy regularization (13)
        Update $Q_\mathcal{O}(x_i, o)$ via the optimistic TD error (12)
        if $o_i$ has already terminated then
            Update $\beta^o(x_i)$ via (11) and the maximization of $c_{H_\beta} H(\beta^o(x_i))$
        end if
    end for
    Train $\hat{p}$ and $\hat{\mu}$ by option-transitions sampled from $B_\mathcal{O}$
end for
```

Upgoing Option-Advantage Estimation for GAE To use an upgoing option-advantage estimation (12) with Generalized Advantage Estimator (GAE) (Schulman et al., 2015b) common with PPO, we introduce an upgoing *general* option advantage estimation (UGOAE). Letting δ denote the TD error corresponding to the marginalized option-state-values, $\delta_t = R_t + \gamma V_\mathcal{O}(x_{t+1}) - V_\mathcal{O}(x_t)$, we write the GAE for marginalized policy π_μ as $\hat{A}^o_\mu = \sum_{i=0}^{N} (\gamma\lambda)^i \delta_{t+i}$, where λ is a coefficient. Supposing that o_t terminates at the $t + k$ step and letting δ^o denote the TD error corresponding to an option-state value $\delta^o_t = R_t + \gamma Q_\mathcal{O}(x_{t+1}, o) - Q_\mathcal{O}(x_t, o)$, we formulate UGOAE by:

$$\hat{A}^o_{\text{UGOAE}} = \begin{cases} \sum_{i=0}^{k} (\gamma\lambda)^i \delta^o_{t+i} + \underbrace{\max(\sum_{i=k+1}^{N} (\gamma\lambda)^i \delta_{t+i}, 0)}_{\text{upgoing estimation}} & (k < N) \\ \sum_{i=0}^{N-1} (\gamma\lambda)^i \delta^o_{t+i} + (\gamma\lambda)^N \left(R_{t+N} + \gamma U_\mathcal{O}(x_{t+N+1}) - Q_\mathcal{O}(x_{t+N}, o)\right) & (otherwise). \end{cases} \tag{18}$$

The idea is the same as UOAE (12) and is optimistic about the advantage after option termination.

Clipped β^o Loss In our preliminary experiments, we found that performing multiple steps of optimization on the gradient (11) led to destructively large updates and resulted in the saturation of β^o to zero or one. Hence, to perform PPO-style multiple updates on β^o, we introduce a clipped loss for β^o:

$$\nabla_{\theta_\beta} \text{clip}(\ell_{\beta^o}(x) - \ell_{{\beta^o}_{\text{old}}}(x), -\epsilon_\beta, \epsilon_\beta) {\beta^o}_{\text{old}}(x) \Big(\log p(o|x_s, x) - \log p(o|x_s, x_f) \Big), \tag{19}$$

where ϵ_β is a small coefficient, ${\beta^o}_{\text{old}}$ is a β^o before the update, and $\text{clip}(x, -\epsilon, \epsilon) = \max(-\epsilon, \min(\epsilon, x))$. Clipping makes the gradient zero when β^o is sufficiently different than ${\beta^o}_{\text{old}}$ and inhibits too large updates.

D Experimental Details

D.1 Network Architecture

Figure 8 illustrates the neural network architecture used in our experiments. In Gridworld experiments, we used the same state encoder for all networks and we found that it is effective for diversifying π^o as an auxiliary loss (Jaderberg et al., 2017). However, in MuJoCo experiments, we found that sharing

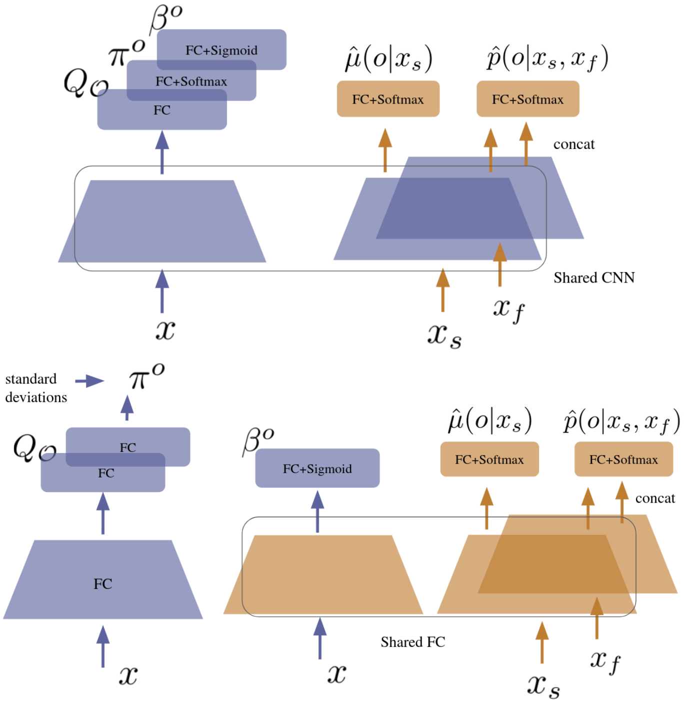


Figure 8: Neural Network architecture used for the Gridworld experiments (top) and the MuJoCo tasks (bottom).

the state encoder can hurt the policy update because the magnitude of β^o loss is larger even if clipped loss (19) is used. As a remedy for this, we used two encoders in MuJoCo experiments: one is for π^o and $Q_{\mathcal{O}}$, and the other is for β^o, $\hat{p}$, and $\hat{\mu}$.

In Gridworld experiments, we represented a state as an image and encoded it by a convolutional layer with 16 filters of size 4×4 with stride 1, followed by a convolutional layer with 16 filters of size 2×2 with stride 1, followed by a fully connected layer with 128 units. In MuJoCo experiments, we encode the state by two fully connected layers with 64 units. π^o is parameterized as a Gaussian distribution with separated networks for standard derivations per option, similar to Schulman et al. (2015a). We used ReLU as an activator for all hidden layers and initialized networks by the orthogonal (Saxe et al., 2014) initialization in all experiments. Unless otherwise noted, we used the default parameters in PyTorch (Paszke et al., 2019) 1.5.0.

D.2 Hyperparameters

When evaluating agents, we used ϵ-Greedy for selecting options with ϵ_{opt} and did not use deterministic evaluation (i.e., an agent samples actions from π^o) in all experiments. We show the algorithm-specific hyperparameters of A2IMOC in Table 1. In Gridworld experiments, we used $\epsilon = 0.1$ for AOC. Our AOC implementation is based on the released code[3] and uses truncated N-step advantage. Other parameters of AOC and A2C are the same as A2IMOC. We also show the hyperparameters of PPIMOC in Table 2. For PPOC, we used $c_{\mu\text{ent}} = 0.001$ for the weight of the entropy $H(\mu(x))$. Our PPOC implementation is based on the released code[4] and uses N-step (not truncated) GAE for computing advantage. PPOC and PPO shares all other parameters with PPIMOC.

[3] https://github.com/jeanharb/a2oc_delib

[4] https://github.com/mklissa/PPOC

Description	Value
γ	0.99
Optimizer	RmsProp (Tieleman and Hinton, 2012)
RmsProp Learning Rate	2×10^{-3}
ϵ_{opt}	$0.8 \rightarrow 0.2$
Number of timesteps per rollout	20
Number of actors	12
c_{H_μ}	0.04
c_H	0.01
$c_{H_\beta} H(\beta^o(x))$	0.01
Gradient clipping	1.0
Capacity of $B_{\mathcal{O}}$	480
Batch size for training $\hat{p}$ and $\hat{\mu}$	240

Table 1: Hyperparameters of A2IMOC in Gridworld experiments.

Description	Value
γ	0.99
Optimizer	Adam (Kingma and Ba, 2015)
Adam Learning Rate	3×10^{-4}
Adam ϵ	1×10^{-4}
ϵ_{opt}	$0.4 \rightarrow 0.1$ (Cartpole swingup), $0.2 \rightarrow 0.1$ (Otherwise)
ϵ_β	0.05 (Lifelong Billiard),0.1 (otherwise)
Number of timesteps per rollout	256
Number of actors	16
GAE parameter (λ)	0.95
Number of epochs for PPO	10
Minibatch size for PPO	1024
c_{H_μ}	0.004
c_H	0.001
$c_{H_\beta} H(\beta^o(x))$	0.01
Gradient clipping	0.5
Number of monte carlo rollout to estimate $H(\pi_\mu)$	20
Capacity of $B_{\mathcal{O}}$	4096
Batch size for training $\hat{p}$ and $\hat{\mu}$	2048

Table 2: Hyperparameters of PPIMOC in MuJoCo and Classical Control experiments.

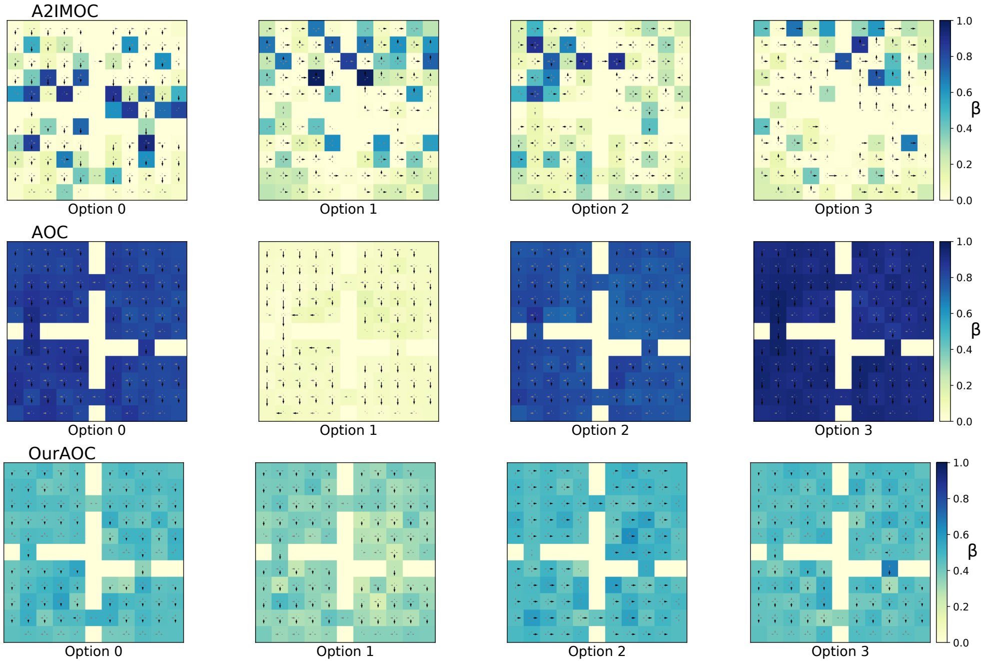


Figure 9: Learned option-policies (π^o) and termination functions (β^o) for each option in Gridworld Four Rooms after 5×10^4 steps. Arrows show the probablities of each action and heatmaps show probabilities of each β^o. **First row:** A2IMOC. Option 0 tends to go down, option 1 and 2 tend to go right, and option 3 tends to go up. **Second row:** AOC. All options tend to go down. **Third row:** Our AOC. Option 0, 1, and 3 tend to go down and option 2 tends to do right.

D.3 Environmental Details

In the Gridworld experiment, an agent can select four actions: go up, go down, go left and go right. With the probability 0.1, the agent takes a uniformly random action. If the agent reaches one of goals, it receives $+1.0$ or $+2.0$ reward. Otherwise, an action penalty -0.002 is given. The maximum episode length is 100.

MuJoCo Point environments are implemented based on "PointMaze" in rllab (Duan et al., 2016) with some modifications, mainly around collision detection. The maximum episode length is 1000. In Four Rooms task, an agent receives $+0.5$ or $+1.0$ reward when it reaches a goal. Otherwise, an action penalty -0.0001 is given. This reward structure is the same in Billiard Task: an agent receives a goal reward when the object ball reaches a goal, otherwise it receives penalty.

D.4 Early Option-Policies learned in Gridworld Four Rooms

Figure 9 shows early option-polices and termination probabilities in Gridworld Four Rooms experiment. We can see that A2IMOC learned the most diverse option-policies.

D.5 Qualitative analysis of Point Four Rooms Experiment

Figure 10 shows the visualizations of learned option-polices and termination functions in MuJoCo Point Four Rooms, averaged over 100 uniformly sampled states per each position. Arrows show the expected moving directions, computed from rotations and option-policies. Terminating regions and option-policies learned with PPIMOC are diverse. For example, option 1 tends to go down while option 2 tends to go right. In the sampled trajectory of PPIMOC, we can see that it mainly used option 1 but occasionally switched to option 0 and option 2 for reaching the goal, and switched to option 3 around the goal. Contrary, for reaching the goal PPOC only used option 3 that does not terminate in any region. Options learned by Our PPOC is almost the same: termination probability is high around the upper left corner and option-policies direct downward.

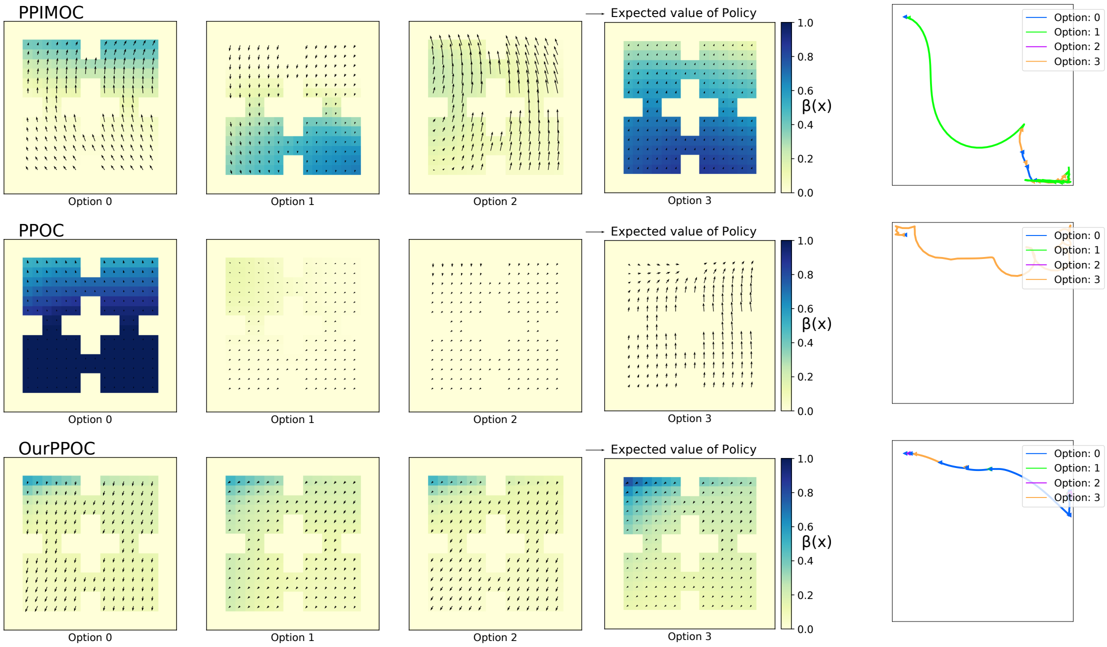


Figure 10: **Left:** Learned option-policies (π^o) and termination functions (β^o) in MuJoCo Point Four Rooms experiment. Arrows show the expected moving direction of the agent and heatmaps show probabilities of each β^o. **Right:** Sampled trajectories of each method. **First row:** PPIMOC. Terminattion regions are clearly separated and option-polices are diverse. **Second row:** PPOC. Option 0 terminates at almost everywhere, while option2 and 3 does not terminate anywhere. **Third row:** Our PPOC. All options are almost the same.

D.6 Ablation Studies

We conducted ablation studies with three variants of A2IMOC/PPIMOC:

- $c_{H_\mu} = 0$: Do not use the policy regularization based on MI (13).
- N-step Advantange: Use N-step advantage or N-step GAE instead of UOAE (12) UGOAE (18).
- Truncated N-step Advantange: Compute advantage ignoring future rewards instead of using UOAE or UGOAE.

Figure 11 shows all results in three tasks. We can see that UOAE is effective in all tasks, since both N-step advantage and truncated N-step advantage performed worse than UOAE. The policy regularization based on MI (13) is effective only in the Point Billiard lifelong learning task.

D.7 Number of Options

Figure 12 shows the performance of IMOC with varying the number of options. Two options perfomed worse in all experiments and we need four or more options to make use of IMOC. However, when we increase the number of options to six and eight, we don't see any peroformance improvement from four options, despite of our analysis that we need sufficiently many options to cover the state space Appendix A. This would be an interesting problem for future works.

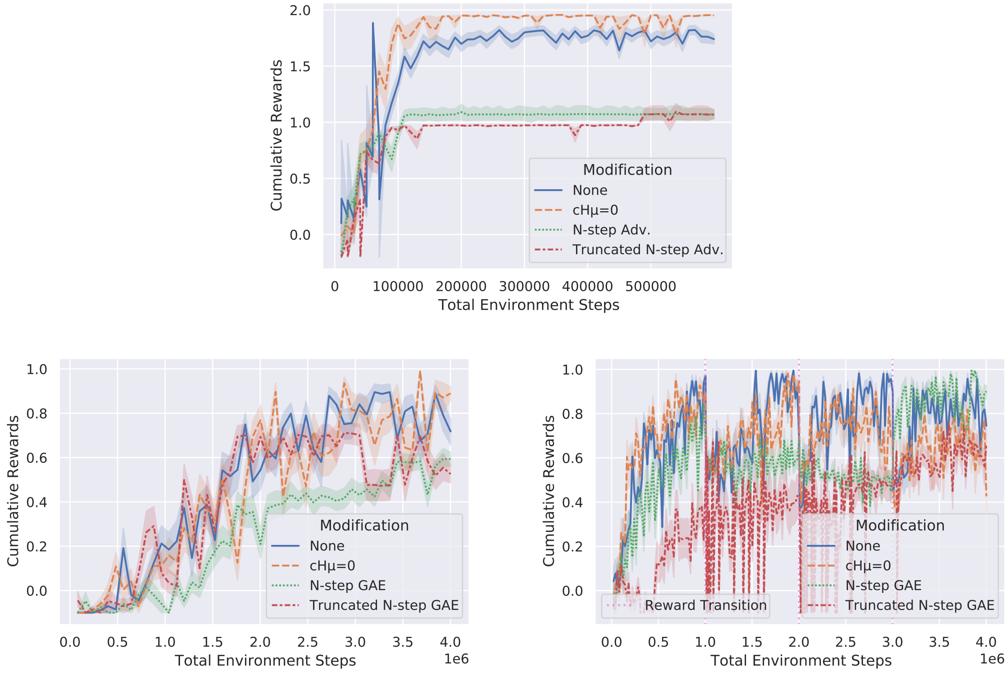


Figure 11: Ablation studies. **Top:** Peformance progression of A2IMOC in Gridworld Four Rooms. **Bottom:** Peformance progression of PPIMOC in MuJoCo Point Four Rooms (left) and MuJoCo Point Billiard (right).

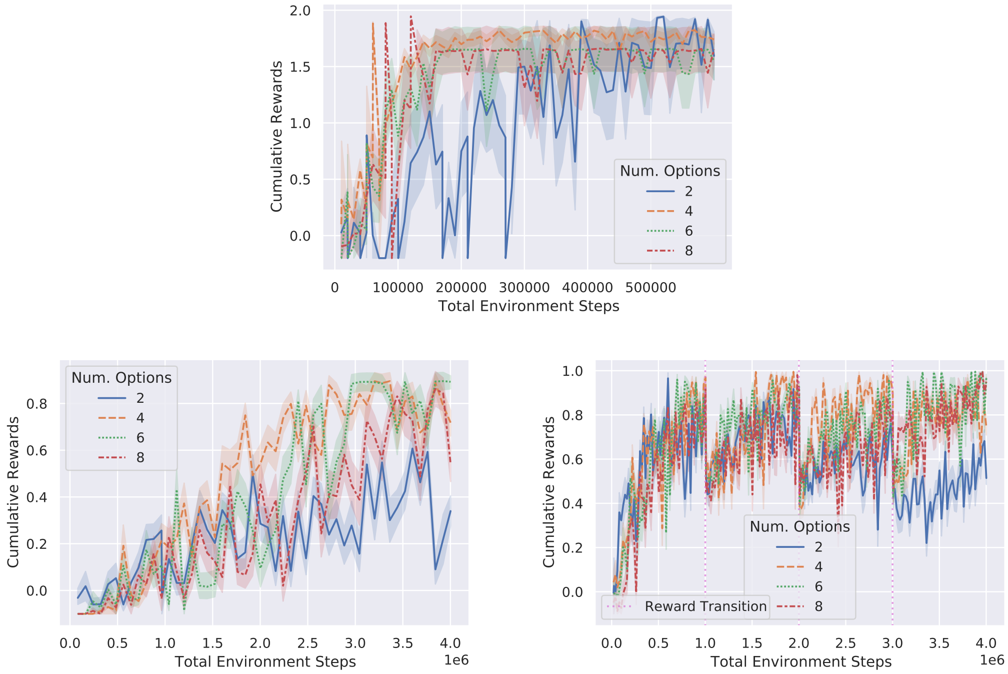


Figure 12: Ablation studies. **Top:** Peformance progression of A2IMOC in Gridworld Four Rooms. **Bottom:** Peformance progression of PPIMOC in MuJoCo Point Four Rooms (left) and MuJoCo Point Billiard (right).