Unsupervised Discovery of Decision States through Intrinsic Control

Nirbhay Modhe[1] **Mohit Sharma**[1] **Prithvijit Chattopadhyay**[1] **Abhishek Das**[1]
Devi Parikh[1,2] **Dhruv Batra**[1,2] **Ramakrishna Vedantam**[2]

[1]Georgia Tech [2]Facebook AI Research

{nirbhaym,mohit.sharma,prithvijit3,abhshkdz,parikh,dbatra}@gatech.edu

ramav@fb.com

Abstract

Learning diverse and reusable skills in the absence of rewards in an environment is a key challenge in reinforcement learning. One solution to this problem, as has been explored in prior work (Gregor et al., 2016; Eysenbach et al., 2018; Achiam et al., 2018), is to learn a set of *intrinsic* macro-actions or options that reliably correspond to trajectories when executed in an environment. In this options framework, we identify and distinguish between decision-states (e.g. crossroads) where one needs to make a decision, as being distinct from corridors (where one can follow default behavior) in the modeling of options. Our intuition is that identifying decision states would lead to more interpretable behavior from an RL agent, exposing clearly what the underlying options correspond to. We formulate this as an information regularized intrinsic control problem using techniques similar to (Goyal et al., 2019) who applied the information bottleneck to goal-driven tasks. Our qualitative results demonstrate that we learn interpretable decision states in an unsupervised manner by merely interacting with the environment.

1 Introduction

Understanding the right levels of abstractions at which to break down complex tasks (say, "going to the kitchen") is an essential skill for building flexible and scalable reinforcement learning agents (Andreas et al., 2016; Springenberg et al., 2018). Most complex real-world tasks have intrinsic *decision states*, followed by phases where one has a default behavior. For example, when navigating to the kitchen, one might have to head out of the door, take a right, and then take a left.

Thus, the act of "walking to the kitchen" is punctuated by certain *decision states*, where one switches from one mode of behavior to another, but beyond that a lot of the sub-tasks have a default mode of behavior outside of these *decision states* (say when walking straight in corridors). Understanding these *decision states* in an environment (say in context of a navigation task) has the promise to enable better transfer and understanding of the structure in the environment.

While prior work in this domain Goyal et al. (2019) discovers these *decision states* and identifies the default modes of behavior in a task/goal-supervised manner, we aim to do so in an unsupervised manner using agents which explore the environment to understand what they can control.

Our premise is that the decision states can exist in environments independent of the end goals (e.g. crossings in a maze). Thus, we set out to explore if one can identify such decision states in a purely usupervised manner by simply interacting with the environment. We situate ourselves in the hierarchical reinforcement learning framework of options (Kulkarni et al., 2016), learning a breakdown of a higher level option into its constitutent decision states. We next elaborate on some related work in this space spanning learning of options and learning with parameterized *default* policies.

Options Frameworks. Reward-free methods for option discovery (Gregor et al., 2016; Eysenbach et al., 2018; Achiam et al., 2018) have been shown to learn a diverse set of options that are useful for downstream tasks and hierarchical control. These options (which are easiest to think of as macro-actions) are discovered through related but different objectives in prior work – Gregor et al. (2016) maximize the number of final states that can be reliably reached by the policy, Eysenbach et al. (2018) distinguish an option at every state along the trajectory, and Achiam et al. (2018) learns options for entire trajectories by encoding the sequence of states at regular time intervals.

One can understand these works as attempting to maximize intrinsic control (IC), namely understanding the space of macro-actions that meaningfully affect the state of the world. In the most general case, IC frameworks characterize behavior in the form of trajectories $\tau = (s_0, a_0, s_1, \ldots, s_T)$, which are then encoded in a higher level option space Ω. The goal of maximizing the diversity of behavior is achieved by maximizing the entropy $H(\tau)$ of the trajectory distribution while having reliability in producing a specific behavior corresponding to minimizing the entropy $H(\tau|\Omega)$ of the option conditional trajectory distribution, which is essentially viewed as maximizing the mutual information $I(\Omega, \tau)$ between option and trajectory as the objective – $\max_\pi \ \mathbb{E}_{\tau\sim\pi}\left[I(\Omega, \tau)\right]$.

But this is far from perfect, since the set of intrinsic options available to an agent are likely to produce trajectoies with overlapping substructures. For example, when passing through a corridor, the only modes of action are forward or backward irrespective of option, and distinguishing between options in these common states is meaningless. Thus it is crucial to identify *decision states* where options become more distinguished, and gracefully fall back to default operating behavior in common states.

Default Policies. Recent work in policy compression has focused on learning this *default policy* when training on a family of tasks, to be able to re-use behavior common to all tasks. In Galashov et al. (2018); Teh et al. (2017), a default policy is learnt using a set of task-specific policies which in turn acts as a regularizer for each policy, while Goyal et al. (2019) learn a default policy using an information bottleneck on task information and a latent variable the policy is conditioned on. We devise a similar information bottleneck objective but in an unsupervised setting that learns default behavior to be shared by all intrinsic options so as to reduce learning pressure on option-specific policies.

Contributions. Our contributions are as follows: 1) we devise a formulation to compute *decision states* without reward supervision by augmenting intrinsic control objectives with an information theoretic regularizer, 2) we show that the proposed formulation is able to identify *decision states* that are meaningful for discrete 2D environments with countable options.

2 Approach

We first describe the objectives from (Gregor et al., 2016) (VIC) which optimizes for intrinsic control using options. We then describe the information regularizer we construct on top of this approach to identify decision states implicit in the choices of options. The key idea from Gregor et al. (2016) is to maximize the mutual information (Cover & Thomas, 2006) between an option (Ω) and the final state in a trajectory s_f given a current state s_0, i.e. $I(s_f, \Omega|s_0)$. Gregor et al. (2016) formulates the following lower bound on this mutual information:

$$I(\Omega, s_f|s_0) \geq \mathbb{E}_{\Omega\sim p(\Omega|s_0), s_f\sim p^J(s_f|\Omega, s_0)}\left[\log \frac{q_\nu(\Omega|s_f, s_0)}{p(\Omega|s_0)}\right] \tag{1}$$

Where $p(\Omega)$ is a prior on options, $p^J(\cdot|\Omega, s_0)$ is the (unknown) distribution of final states given the option which we can sample from by executing actions in the environment, and $q(\cdot)$ is a variational approximation to the true posterior on options given a final state s_f. We next describe our information bottleneck regularizer which we employ in addition to the intrinsic control objective above.

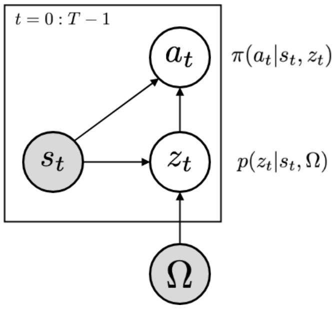


Figure 1: Regularized option conditioned policy $\pi(a_t|s_t, \Omega)$. We impose a bottleneck, minimizing $I(a_t, \Omega|s_t)$ to allow discovery of decision states, where Ω influences actions a_t (despite the bottleneck).

Information Bottleneck Regularizer. The VIC framework utilizes options in decision making by parameterizing a policy $\pi(a_t|s_t, \Omega)$ which modulates the behavior of the policy by conditioning on the option of interest Ω, sampled once for an entire trajectory. Our regularizer focuses on a particular parameterization of this policy, and constructs a bottleneck inspired by (Goyal et al., 2019).

Fig. 1 shows the proposed construction of the option conditioned policy. At the beginning of every episode, an option Ω is sampled which guides the behavior of the agent for the entire episode. At every timestep t in the episode, given the current state (s_t) and the (global) option (Ω), we compute an intermediate representation z_t. This intermediate representation is used along with the state s_t to choose an action a_t to perform at the current timestep t.

Similar to (Goyal et al., 2019), we impose a constraint that says that the actions a_t should have low mutual information with the options, i.e. $\min I(a_t, \Omega | s_t)$. This is a way of providing the inductive bias to the model that there are states with default behavior (where one need not reason about the chosen options). Further, this allows us to identify decision states by tracking states where the mutual information is high (despite the constraint). By data processing inequality (Cover & Thomas, 2006), we know that $I(a_t, \Omega | s_t) \leq I(z_t, \Omega | s_t)$. Thus, we minimize the upper bound instead, yielding the following regularized objective (c.f. Goyal et al. (2019)):

$$\max_{\pi} \; J^{\pi} = \mathbb{E}_{s_1, \cdots, s_f \sim \pi} \Big[I(\Omega, s_f | s_0) - \beta \sum_t I(\Omega, z_t | s_t) \Big] \tag{2}$$

where β controls the strength of the regularization. The first term $I(\Omega, s_f | s_0)$ is optimized using Eqn. 1. We next explain how we do the optimization of the second term. We first write $I(\Omega, z_t | s_t)$ as:

$$\mathbb{E}_{\Omega \sim p(\Omega), z_t \sim p(z_t | s_t, \Omega)} \left[\log \frac{p(z_t | \Omega, s_t)}{p(z_t | s_t)} \right] \tag{3}$$

We then assume a variational approximation $q(z_t)$ for $p(z_t | s_t)$, and using the fact that $D_{\mathrm{KL}}[p(z_t | s_t) || q(z_t)] \geq 0$, we get a the following lower bound (similar to Goyal et al. (2019)):

$$I(\Omega, z_t | s_t) \geq \mathbb{E}_{\Omega \sim p(\Omega), z_t \sim p(z_t | s_t, \Omega)} \left[\log \frac{p(z_t | \Omega, s_t)}{q(z_t)} \right] \tag{4}$$

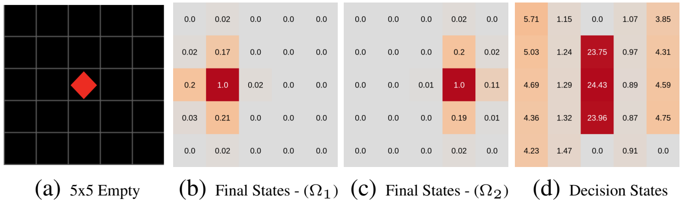
Putting these together, the overall objective that we optimize for is[1]:

$$\max_{\theta, \phi, \nu} \tilde{J}(\theta, \phi, \nu) = \mathbb{E}_{\Omega \sim p(\Omega), z_t \sim p(z_t | s_t, \Omega), s_1, \cdots, s_f \sim \pi_\theta} \left[\log \frac{q_\nu(\Omega | s_f)}{p(\Omega)} - \beta \sum_t \log \frac{p_\phi(z_t | s_t, \Omega)}{q(z_t)} \right] \tag{5}$$

Decision State Identification. We identify *decision states* in the environment as states where Eqn. 4 is relatively high. One could also consider looking at decision-states corresponding to individual options, we chose the expectation over the entire option vocabulary to take into account all options intrinsically available to the agent - we hypothesize, *decision states* agreed upon by options across the entire vocabulary are more likely to correlate to structural regularities in the environment.

3 Experiments

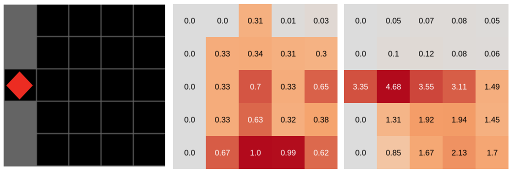
As a proof of concept, we demonstrate the emergence of *decision states* based on the unsupervised objective in two simple two-dimensional grid-world environments with a fully observable state-spaces. Our environments are inherited from MiniGrid set of environments (Chevalier-Boisvert et al., 2018). The agent's action-space includes four cardinal movement directions - `Up`, `Down`, `Left`, `Right` and a `Stay` action. We use a reactive policy and Advantage Actor-Critic (Wu et al., 2017) for our experiments. We train agents in an episodic setting where termination happens after fixed time-steps T (we set $T = 8$). In practice, we found that it was hard to obtain reasonable results by learning both the terms in the objective to be optimized for from scratch and therefore, we optimize the intrinsic objective itself for $\sim 8k$ episodes (i.e., we set $\beta = 0$) after which we turn on the bottleneck term and let β grow linearly for another $\sim 8k$ episodes to get feasible outcomes at convergence.



(a) 5x5 Empty (b) Final States - (Ω_1) (c) Final States - (Ω_2) (d) Decision States

Figure 2: Qualitative results on the 5x5 Empty grid world (Figure. 2(b)) where the agent is always spawed at the center and trained with two discrete intrinsic options. The two learnt options (2(c)) show a final state frequency (normalized) in two distinct halves of the grid. Figure. 2(d) shows the heatmap of KL values (normalized) for $\mathbb{E}_{p(\Omega)}[D_{\mathrm{KL}}(p(z_t | s_t, \Omega) || q(z))]$, indicating decision states.

[1] In our experiments, we further augment the objective with a maximum-entropy bonus to encourage sufficient exploration of the state-space.



(a) Single Corridor

0.0	0.0	0.31	0.01	0.03
0.0	0.33	0.34	0.31	0.3
0.0	0.33	0.7	0.33	0.65
0.0	0.33	0.63	0.32	0.38
0.0	0.67	1.0	0.99	0.62

(b) Final States

0.0	0.05	0.07	0.08	0.05
0.0	0.1	0.12	0.08	0.06
3.35	4.68	3.55	3.11	1.49
0.0	1.31	1.92	1.94	1.45
0.0	0.85	1.67	2.13	1.7

(c) Decision States

Figure 3: Qualitative results on the 5x5 Single-Corridor Room environment (Figure. 3(a)) where the agent is always spawed at the entrance of the room and trained with 32 discrete intrinsic options. The learnt options (3(b)) show a final state frequency (normalized) in the grid. Figure. 3(c) shows the heatmap of KL values (normalized) for $\mathbb{E}_{p(\Omega)}[D_{\mathrm{KL}}(p(z_t|s_t,\Omega)||q(z))]$, indicating decision states.

Empty 5x5 Grid. Fig. 2 shows an empty 5x5 grid where the agent is always spawned at the center of the grid. Our option space is discrete with 2 choices $\Omega = \{\Omega_1, \Omega_2\}$. We observe that the option termination states lie in mutually exclusive halves of the environment. (see Fig. 2(c)). Furthermore, upon identifying *decision states* as per Eq. 4, we find they emerge along the vertical line separating the option termination states (Figure. 2(c)). Presumably, these are the states where an agent needs to undertake option-informed actions in order to decide which cluster of final states to go to. Further, once we get closer to the final states default behavior emerges, i.e. the states in the vicinity of final states are not decision states.

Single-Corridor Room. Fig. 3(a) shows an environment which consists of a single-cell corridor leading up to an empty room. The agent is always spawned at the entrance of the room. Interestingly, in Fig. 3, we found it difficult for reasonable behavior to emerge when the number of options provided to the agent is less and therefore, we conduct our experiments with 32 options. Unlike the empty 5x5 grid, the final states (see Fig. 3(b)) associated with the learnt set of options are more concentrated towards the bottom half of the room. However, note that this is an overcomplete setting – the number of available options is more than the number of cells the agent can occupy. Thus, at the very least, one would expect one option to cover every single cell in the environment. Furthermore, we find that decision states (see Fig. 3(c)) emerge near the mouth of the corridor and along a horizontal line separating the bottom and top halves of the room – where a decision would indicate which half of the room to go to.

Initial State Distribution. From our experiments, we observed that the emergence of interpretable *decision states* while optimizing Eqn. 5 is also sensitive to where the agent is spawned across multiple training episodes. We find that spawning the agent unformly at random at a cell in the environment resulted in a fairly smooth and uniform bottleneck map (similar to Fig. 2(d) and 3(c)), making it hard to reason about the emergent *decision states* in an interpretable manner. Therefore, we stick to spawning the agent at the same location in the environment across multiple episodes of training.[2]

4 Conclusion

We introduce a method to discover *decision states* inherently present in an environment without any task-specific reward supervision by using an intrinsic control objective augemented with an information-theoretic regularizer. As a proof of concept, we show that our proposed formalism is capable of identifying *decision states* for a simple 2D discrete grid-world environment in addition to partitioning the state-space into regions which can be reliably traversed for individual options. In terms of extensions we intend to show the following:

Constrained Environments. We intend to transfer the above objective to more complex (maze-like) environments to demonstrate that *decision states* do indeed emerge at states where a default mode of behavior is unclear. Additionally, we aim to show this with a larger option vocabulary so as to capture more higher-level modes of action corresponding to these *decision states*.

Transfer. Secondly, as a part of future work, we would like to study how this identification of *decision states* and default modes of behavior allow us to adapt to goal-driven settings in the same environments. Given the the assumption that these *decision states* capture structural regularities in the environment, properly utilizing them via a hierarchical planner-controller architecture should lead to improved sample efficiency when transferring to goal-driven tasks.

[2] We understand that this limits the utility of options learned throughout the optimization process as the agent would not have picked an option at multiple cells (except for the fixed spawn location) in the environment.

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018. 1, 6

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. *CoRR*, abs/1611.01796, 2016. URL http://arxiv.org/abs/1611.01796. 1

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. 3

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 2006. 2nd edition. 2, 3

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018. 1

Alexandre Galashov, Siddhant M Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized rl. 2018. 2

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Sergey Levine, and Yoshua Bengio. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019. 1, 2, 3

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016. 1, 2, 6

Tejas D. Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *CoRR*, abs/1604.06057, 2016. URL http://arxiv.org/abs/1604.06057. 1

Jost Tobias Springenberg, Karol Hausman, Martin Riedmiller, Nicolas Heess, and Ziyu Wang. Learning an embedding space for transferable robot skills. 2018. 1

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4496–4506, 2017. 2

Yuhuai Wu, Elman Mansimov, Shun Liao, Roger B. Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *CoRR*, abs/1708.05144, 2017. URL http://arxiv.org/abs/1708.05144. 3

A Appendix

A.1 Training Details

We now go over a few details of the entire training pipeline, starting with representation used for agent's state, the convergence criterion used for the optimization process and how we incorporate the option-curriculum from Achiam et al. (2018) to learn a meaningful and reliable set of options.

State Representation The state of the agent is fully characterized by the grid configuration and the agent's position in the grid. We represent the state as a multi-channel image, where each channel corresponds to a specific object type present in the grid. This multi-channel representation essentially generalizes the notion of an occupancy grid. These grids are then processed using a shallow convolution network. The learnt representation is used by both the option encoder $p(z|s, \Omega)$ and the policy head $\pi(a|s, z)$.

Convergence Criterion We first restate the objective function we optimize in our framework. The overall objective function is as follows:

$$\max_{\theta,\phi,\nu} \tilde{J}(\theta, \phi, \nu) = \mathbb{E}_{\Omega \sim p(\Omega), z_t \sim p(z_t|s_t,\Omega), s_1, \cdots, s_f \sim \pi_\theta} \left[\log \frac{q_\nu(\Omega|s_f)}{p(\Omega)} \; - \beta \sum_t \log \frac{p_\phi(z_t|s_t, \Omega)}{q(z_t)} \right] \tag{6}$$

where the first term $\log \frac{q_\nu(\Omega|s_f)}{p(\Omega)}$ is a variational approximation for $I(\Omega, s_f|s_0)$, mutual information between an option Ω and the final state s_f reached by an agent following option-conditioned policy $\pi(a|s, \Omega)$. It can be interpreted as a measure of control (*empowerment*) agent has in the environment. Empowerment can also be interpreted as the number of states the agent can reliably reach in the environment. Since it is a proxy for coverage of state-space achieved by the agent, we can use it as a convergence criterion for training as we expect the agent to have learnt a reliable and discriminative option space when it reaches high empowerment value. We terminate the optimization process when empowerment staturates after reaching a high value.

Option Curriculum It is standard to have a fixed-sized discrete option space Ω with a uniform pior Gregor et al. (2016). However, learning a meaningful option space with larger option vocabulary size $|\Omega| = K$ has been reported to be difficult Achiam et al. (2018). We adopt a curriculum based approach proposed in Achiam et al. (2018) where vocalubary size is gradually increases as the option decoder $q_\nu(\Omega|s_f)$ becomes more confident in mapping back the final state back to the corresponding option sampled at the beginning of the episode. More concretely, whenever $q_\nu(\Omega|s_f) > 0.75$ (this treshold was chosen through hyperparameter tuning), the option vocabulary size increases according to

$$K \leftarrow \min \left(\text{int}\big(\, 1.5 \times K + 1 \big), K_{max} \right)$$

For our experiments, curriculum starts with $K = 2$ and and the curriculum based learning ends when $K = K_{max}$ ($K_{max} = 32$ for our experiment).