IR-VIC: Unsupervised Discovery of Sub-goals for Transfer in RL

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

[1]Georgia Institute of Technology
[2]Facebook AI Research
{nirbhaym,prithvijit3,mohit.sharma,abhshkdz,parikh,dbatra}@gatech.edu, ramav@fb.com

Abstract

We propose a novel framework to identify sub-goals useful for exploration in sequential decision making tasks under partial observability. We utilize the variational intrinsic control framework (Gregor et.al., 2016) which maximizes empowerment – the ability to reliably reach a diverse set of states and show how to identify sub-goals as states with high *necessary* option information through an information theoretic regularizer. Despite being discovered without explicit goal supervision, our sub-goals provide better exploration and sample complexity on challenging grid-world navigation tasks compared to supervised counterparts in prior work.

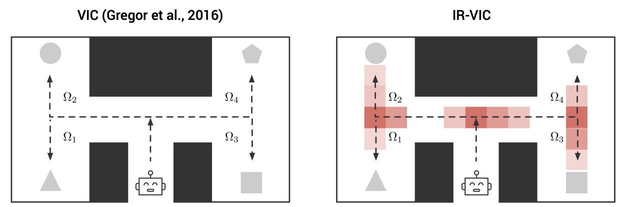

Figure 1: Left: The VIC framework Gregor *et al.* [2016] in a navigation context: an agent learns high-level macro-actions (or options) to reach different states in an environment reliably without any extrinsic reward. Right: IR-VIC identifies sub-goals as states where necessary option information is high (darker shades of red) for an empowered agent. Identification of unsupervised sub-goals leads to improved transfer to novel environments.

1 Introduction

A common approach in reinforcement learning (RL) is to decompose an original decision making problem into a set of simpler decision making problems – each terminating into an identified sub-goal. Beyond such a decomposition or abstraction being evident in humans (*e.g.* adding salt is a sub-goal in the process of cooking a dish) [Hayes-Roth and Hayes-Roth, 1979], sub-goal identification is also useful from a practical perspective of constructing policies that transfer to novel tasks (*e.g.* adding salt is a useful sub-goal across a large number of dishes one might want to cook, corresponding to different 'end' goals).

However, identifying sub-goals that can accelerate learning while also being re-usable across tasks or environments is a challenge in itself. Constructing such sub-goals often requires knowledge of the task structure (supervision) and may fail in cases where 1) dense rewards are absent [Pathak *et al.*, 2017], 2) rewards require extensive hand engineering and domain knowledge (hard to scale), and 3) where the notion of reward may not be obvious [Lillicrap *et al.*, 2015]. In this work, we demonstrate a method for identifying sub-goals in an "unsupervised" manner – without any external rewards or goals. We show that our sub-goals generalise to novel partially observed environments and goal-driven tasks, leading to comparable (or better) performance (via. better exploration) on downstream tasks compared to prior work on goal-driven sub-goals [Goyal *et al.*, 2019].

We study sub-goals in the framework of quantifying the minimum information necessary for taking actions by an agent. van Dijk and Polani [2011] have shown that in the presence of an external goal, the minimum goal information required by an agent for taking an action is a useful measure of sub-goal states. Goyal *et al.* [2019] demonstrate that for action A, state S and a goal G, such sub-goals can be efficiently learnt by imposing a bottleneck on the information $I(A, G|S)$. We show that replacing the goal with an intrinsic objective admits a strategy for discovery of sub-goals in a completely unsupervised manner.

Our choice of intrinsic objective is the Variational Intrinsic Control (VIC) formulation [Gregor *et al.*, 2016] to learn options Ω that maximize the mutual information $I(S_f, \Omega)$, referred to as empowerment, where S_f is the final state in a trajectory [Salge *et al.*, 2013]. To see why this maximizes empowerment, notice that $I(S_f, \Omega) = H(S_f) - H(S_f|\Omega)$, where $H(.)$ denotes entropy. Thus, empowerment maximizes the diversity in final states S_f while learning options highly predictive of S_f. We demonstrate that by limiting the information the agent uses about the selected option Ω while maximizing empowerment, a sparse set of states emerge where the necessary option information $I(\Omega, A|S)$ is high – we interpret these states as our unsupervised sub-goals. We call our approach Information Regularized VIC (IR-VIC). Although IR-VIC is similar in spirit to Goyal *et al.* [2019]; Polani *et al.* [2006], it is important to note that we use latent options Ω instead of external goals – removing any dependence on the task-structure.

To summarize our contributions,

- We propose Information Regularized VIC (IR-VIC), a novel framework to identify sub-goals in a task-agnostic manner, by regularizing relevant option information.
- Theoretically, we show that the proposed objective is a sandwich bound on the empowerment $I(\Omega, S_f)$ – this is the only useful upper bound we are aware of.
- We show that our sub-goals are transferable and lead to improved sample-efficiency on goal-driven tasks in novel, partially-observable environments. On a challenging grid-world navigation task, our method outperforms (a re-implementation of) Goyal *et al.* [2019].

2 Methods

2.1 Notation

We consider a Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{X}, \mathcal{A}, \mathcal{P}, r)$, $s \in \mathcal{S}$ is the state, $x \in \mathcal{X}$ is the partial observation of the state and $a \in \mathcal{A}$ is an action from a discrete action space. $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A}$ denotes an unknown transition function, representing $p\left(s_{t+1} | s_t, a_t\right) : s_t, s_{t+1} \in \mathcal{S}, A_t \in \mathcal{A}$. Both VIC and IR-VIC initially train an option (Ω) conditioned policy $\pi(a_t | \omega, x_t)$, where $\omega \in \{1, \cdots, |\Omega|\}$. During transfer, all approaches (including baselines) train a goal-conditioned policy $\pi(a_t | x_t, g_t)$ where g_t is the goal information at time t.

Following standard practice [Cover and Thomas, 1991], we denote random variables in uppercase (Ω), and items from the sample space of random variables in lowercase (ω).

2.2 Variational Intrinsic Control (VIC)

VIC maximizes the mutual information between options Ω and the final (option termination) state S_f given s_0, *i.e.* $I(S_f, \Omega \mid S_0 = s_0)$, which encourages the agent to learn options that reliably reach a diverse set of states. This objective is estimated by sampling an option Ω from a prior at the start of a trajectory, following it until termination, and inferring the sampled option given the final and initial states. Informally, VIC maximizes the empowerment for an agent, *i.e.* its internal options Ω have a high degree of correspondence to the states of the world S_f that it can reach. VIC formulates a variational lower bound on this mutual information. Specifically, let $p(\omega \mid s_0) = p(\omega)$ be a prior on options (we keep the prior fixed as per Eysenbach *et al.* [2018]), $p^J(s_f \mid \omega, s_0)$ is defined as the (unknown) terminal state distribution achieved when executing the policy $\pi(a_t \mid \omega, s_t)$, and $q_\nu(\omega \mid s_f, s_0)$ denote a (parameterized) variational approximation to the true posterior on options given S_f and S_0. Then:

$$
\begin{aligned}
&I(\Omega, S_f \mid S_0 = s_0) \\
&\geq \mathbb{E}_{\substack{\Omega \sim p(\omega) \\ S_f \sim p^J(s_f | \Omega, S_0 = s_0)}} \left[\log \frac{q_\nu(\Omega \mid S_f, S_0 = s_0)}{p(\Omega)} \right] \\
&= \mathcal{J}_{VIC}(\Omega, S_f; s_0)
\end{aligned}
\tag{1}
$$

2.3 Information Regularized VIC (IR-VIC)

We identify sub-goals as states where the necessary option information required for deciding actions is high. Formally, this means that at every timestep t in the trajectory, we minimize the mutual information $I(\Omega, A_t | S_t, S_0 = s)$, resulting in a sparse set of states where this mutual information remains high despite the minimization. Intuitively, this means that on average (across different options), these states have higher relevant option information that other states (*e.g.* the regions with darker shades of red in Figure 1). Overall, our objective is to maximize:

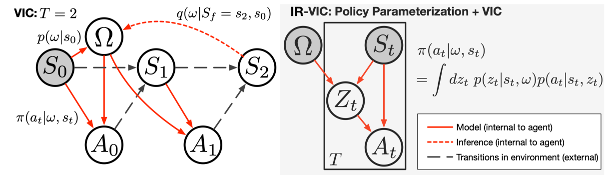

Figure 2: **Illustration of VIC for 2 timesteps. L:** Given a start state S_0, VIC samples option ω and follows policy $\pi(a_t \mid \Omega = \omega, s_t)$ and infers Ω from the terminating state (S_2), optimizing a lower bound on $I(S_2, \Omega \mid S_0)$. **R:** IR-VIC considers a particular parameterization of π and imposes a bottleneck on $I(A_t, \Omega | S_t)$.

$$
\mathcal{J}_{VIC}(\Omega, S_f; s_0) - \beta \sum_t I(\Omega, A_t \mid S_t, S_0 = s_0) \tag{2}
$$

where $\beta > 0$ is a trade-off parameter. Thus, this is saying that one wants options Ω which allow the agent to have a high empowerment, while utilizing the least relevant option information at each step.

Interestingly, Equation 2 has a clear, principled interpretation in terms of the empowerment $I(\Omega, S_f | S_0)$ from the VIC model. We state the following lemma (which follows from recursively applying the chain rule of mutual information and the data-processing inequality [Cover and Thomas, 1991]):

Lemma 2.1. *Let A_t be the action random variable at timestep t and state S_t following an option-conditioned policy $\pi(a_t | s_t, \omega)$. Then, $I(\Omega, A_t | S_t, S_0)$* i.e. *the conditional mutual information between the option Ω and action A_t when summed over all timesteps in the trajectory, upper bounds the conditional mutual information $I(\Omega, S_f | S_0)$ between Ω and the final state S_f – namely the empowerment as defined by Gregor* et al. *[2016]:*

$$
I(\Omega, S_f | S_0) \leq \sum_{t=1}^{f} I(\Omega, A_t | S_t, S_0) = \mathcal{U}_{DS}(\boldsymbol{\tau}, \Omega, S_0) \tag{3}
$$

Implications: With this lens, one can view the optimization problem in Equation 2 as a Lagrangian relaxation of the following constrained optimization problem:

$$
\max \; \mathcal{J}_{VIC} \quad s.t. \; \mathcal{U}_{DS} \leq R \tag{4}
$$

where $R > 0$ is a constant. While upper bounding the empowerment does not directly imply one will find useful sub-goals (meaning it is the structure of the decomposition eq. (3) that is more relevant than the fact that it is an upper bound), this bound might be of interest more generally for representation learning [Achiam *et al.*, 2018; Gregor *et al.*, 2016]. Targeting specific values for the upper bound R can potentially allow us to control how 'abstract' or invariant the latent option representation is relative to the states S_f, leading to solutions that say, neglect unnecessary information in the state representation to allow better generalization. Note that most approaches currently limit the abstraction by constraining the number of discrete options, which (usually) imposes an upper bound on $I(\Omega, S_f) = H(\Omega) - H(\Omega|S_f)$, since $H(\Omega) \geq H(\Omega|S_f)$ and $H \geq 0$ in the discrete case. However, this does not hold for the continuous case, where this result might be more useful. Investigating this is beyond the scope of this current paper, however, as our central aim is to identify useful sub-goals, and not to scale the VIC framework to continuous options.

2.4 Algorithmic Details

Upper Bounds for $I(\Omega, A_t \mid S_t, S_0)$. Inspired by InfoBot [Goyal *et al.*, 2019], we bottleneck the information in a statistic Z_t of the state S_t and option Ω used to parameterize the policy $\pi(A_t \mid \Omega, S_t)$ (fig. 2 right). This is justified by the the data-processing inequality [Cover and Thomas, 1991] for the markov chain $\Omega, S_t \leftrightarrow Z_t \leftrightarrow A_t$, which implies $I(\Omega, A_t \mid S_t, S_0) \leq I(\Omega, Z_t \mid S_t, S_0)$. We can then obtain the following upper bound on $I(\Omega, Z_t \mid S_t, S_0)$: [1]

$$\begin{aligned} &I(\Omega, Z_t \mid S_t, S_0 = s) \\ &\leq \mathbb{E}_{\substack{\Omega \sim p(\omega) \\ S_t \sim p^J(s_t|\Omega, S_0 = s) \\ Z_t \sim p(z_t|S_t,\Omega)}} \left[\log \frac{p(Z_t \mid \Omega, S_t)}{q(Z_t)} \right] \end{aligned} \tag{5}$$

where $q(z_t)$ is a fixed variational approximation (set to $\mathcal{N}(0, \mathtt{I})$ as in InfoBot), and $p_\phi(z_t \mid \omega, s_t)$ is a parameterized encoder. As explained in section 1, the key difference between eq. (5) and InfoBot is that they construct upper bounds on $I(G, A_t \mid S_t, S_0)$ using information about the goal G, while we bottleneck the option-information. One could use the DIAYN objective [Eysenbach *et al.*, 2018] (see more below under related objectives) which also has a $I(A_t, \Omega|S_t)$ term, and directly bottleneck the action-option mutual information instead of eq. (5), but we found that directly imposing this bottleneck often hurt convergence in practice.

We can compute a Monte Carlo estimate of Equation 5 by first sampling an option ω at s_0 and then keeping track of all states visited in trajectory $\boldsymbol{\tau}$. In addition to the VIC term and our bottleneck regularizer, we also include the entropy of the policy over the actions (maximum-entropy RL [Ziebart *et al.*, 2008]) as a bonus to encourage sufficient exploration. We fix the coefficient for maximum-entropy, $\alpha = 10^{-3}$ which consistently works well for our approach as well as baselines. Overall, the IR-VIC objective is:

$$\begin{aligned} \max_{\theta,\phi,\nu} \tilde{J}(\theta, \phi, \nu) = \mathbb{E}_{\substack{\Omega \sim p(\omega) \\ \boldsymbol{\tau} \sim \pi(\cdot|\omega, S_0) \\ Z_t \sim p_\phi(z_t|S_t,\Omega)}} \Bigg[\log \frac{q_\nu(\Omega \mid S_f, S_0)}{p(\Omega)} \\ - \sum_{t=0}^{f-1} \Big(\beta \log \frac{p_\phi(Z_t \mid S_t, \Omega)}{q(Z_t)} + \alpha \log \pi_\theta(A_t \mid S_t, Z_t) \Big) \Bigg] \end{aligned} \tag{6}$$

[1] Similar to VIC, p^J here denotes the (unknown) state distribution at time t from which we can draw samples when we execute a policy. We then assume a variational approximation $q(z_t)$ (fixed to be a unit gaussian) for $p(z_t|S_t)$. Using the fact that $D_{\mathrm{KL}}(p(z_t|s_t)||q(z_t)) \geq 0$ we get the derived upper bound.

where θ, ϕ and ν are the parameters of the policy, latent variable decoder and the option inference network respectively. The first term in the objective promotes high empowerment while learning options; the second term ensures *minimality* in using the options sampled to take actions and the third provides an incentive for *exploration*.

Related Objectives. Eysenbach *et al.* [2018] (DIAYN) attempts to learn skills (similar to options) which can control the states visited by agents while ensuring that all visited states, as opposed to termination states, are used to distinguish skills. Thus, for an option Ω and every state S_t in a trajectory, they maximize $\sum_t I(\Omega, S_t) - I(A_t, \Omega|S_t) + H(A_t|S_t)$, as opposed to $I(\Omega, S_f) - \beta \sum_t I(A_t, \Omega|S_t) + H(A_t|S_t)$ in our objective. With the sum over all timesteps for $I(\Omega, S_t)$, the bound in lemma 2.1 no longer holds true, which also means that there is no principled reason (unlike our model) to scale the second term with β.

The most closely related work to ours is InfoBot [Goyal *et al.*, 2019], which maximizes $\sum_t R(t) - \beta I(Z_t, G|S_t)$ for a goal (G) conditioned policy $\pi(a_t|S_t, G)$. They define states where $I(Z_t, G|S_t)$ is high despite the bottleneck as "decision states". The key difference is that InfoBot requires extrinsic rewards in order to identify goal-conditioned decision states, while our work is strictly more general and scales even in the absence of extrinsic rewards.

Further, in context of both these works, our work provides a principled connection between action-option information regularization $I(A_t, \Omega|S_t)$ and empowerment of an agent. The tools from Lemma 2.1 might be useful for analysing these previous objectives which both employ this technique.

2.5 Transfer to Goal-Driven Tasks

In order to transfer sub-goals to novel environments, Goyal *et al.* [2019] pretrain their model to identify their goal-conditioned decision states, and then study if identifying similar states in new environments can improve exploration when training a new policy $\pi_\gamma(a|s, g)$ from scratch. Given an environment with reward $R_e(t)$, goal G, $\kappa > 0$, and state visitation count $c(S_t)$, their reward is:

$$R_t = R_e(t) + \frac{\kappa}{\sqrt{c(S_t)}} \underbrace{I(G, Z_t|S_t)}_{\text{Pretrained, Frozen}} \tag{7}$$

The count-based reward decays with square root of $c(S_t)$ to encourage the model to explore novel states, and the mutual information between goal G and bottleneck variable Z_t is a smooth measure of whether a state is a sub-goal, and is multiplied with the exploration bonus to encourage visitation of state where this measure is high.

We use an almost identical setup, replacing their decision-state term from supervised pretraining with necessary option information from IR-VIC pretraining:

$$R_t = R_e(t) + \frac{\kappa}{\sqrt{c(S_t)}} \underbrace{I(\Omega, Z_t|S_t, S_0)}_{\text{Pretrained, Frozen}} \tag{8}$$

$I(\cdot)$ is computed with eq. (5) with a frozen parameterized encoder $p(z_t \mid \omega, s_t)$ during transfer. Thus, we incentivize visitation of states where necessary option information is high.

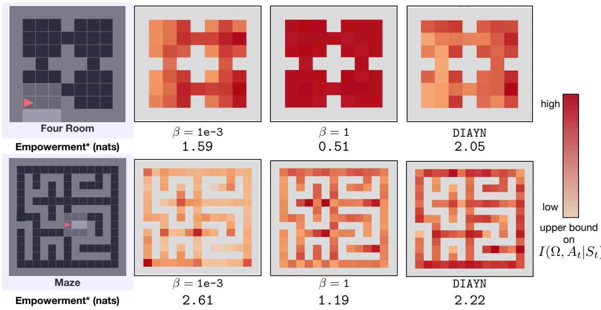

Figure 3: Heatmaps of necessary option information $I(\Omega, Z_t|S_t, S_0)$ (normalized to have maximum value 1) at visited states on simple environments – `4-Room` (top) and `maze` (bottom). First column depicts environment layout, second and third show results for IR-VIC for $\beta = 1e^{-3}$ and $\beta = 1$ respectively, and the fourth column shows DIAYN. * denotes lower bound (Eq. 1) on empowerment.

2.6 IR-VIC for Transfer

Options with partial observability: The methods we have described so far have assumed the true state $s \in \mathcal{S}$ to be known – the VIC framework with explicit options has only been shown to work in fully obervable MDPs [Gregor *et al.*, 2016; Eysenbach *et al.*, 2018]. However, since we are primarily interested improved exploration in downstream partially-observable tasks, we adapt the VIC framework to only use partially-observable information for the parts that we use during transfer. We design our policy (including the encoder $p(Z_t \mid \Omega, S_t)$ used for computing the reward bonus $I(\Omega, Z_t \mid S_t, S_0)$ during transfer) to take as input partial observations $x \in \mathcal{X}$ while allowing the option inference networks (of IR-VIC and DIAYN) to take as input the global (`x, y`) coordinates of the agent (assuming access to the true state $s \in \mathcal{S}$. Note that this privileged information is made available for a single environment in order to discover sub-goals transerable to multiple novel environments (whereas supervised methods such as InfoBot [Goyal *et al.*, 2019] require global (`x, y`) coordinates as goal information across all training environments). Please refer to Algorithm 1 for details about training the policy and inference networks during option-learning and transfer to goal-driven tasks.

Preventing Option Information Leak: We parameterize $p(a_t|z_t, s_t)$ (fig. 2, right) using just the current state s_t, whereas the encoder $p(z_t|\Omega, (s_1, \cdots, s_t))$ uses all previous states since a sequence of state observations $(s_1, \cdots, s_t)$ could potentially be very informative of the Ω being followed, which if provided directly to $p(a_t|\cdot)$ can lead to a leakage of the option information to the actions, rendering the bottleneck on option information imposed via z_t useless. Hence, in our implementation we remove recurrence over partial observations for $p(a_t|z_t, s_t)$ while keeping it in $p(z_t \mid \Omega, s_t)$.

3 Experiments

Environments. We pre-train and test on grid-worlds from the MiniGrid [Chevalier-Boisvert *et al.*, 2018] environments. We first consider a set of simple environments – `4-Room` and `Maze` (see Fig. 3) followed by the `MultiRoomNXSY` also

Algorithm 1 IR-VIC

Require: A parameterized encoder $p_\phi(z_t \mid \omega, x_t)$, policy $\pi(a_t \mid \omega, x_t)$
Require: A parameterized option inference network $q_\nu(\omega \mid s_0, s_f)$
Require: A parameterized goal-conditioned policy $\pi_\gamma(a_t|x_t, g)$
Require: A prior on discrete options $p(\omega) = \frac{1}{|\Omega|}$ and integer H - the length of each option trajectory.
Require: A variational approximation of the option-marginalized encoder $q(z_t)$
Require: A regularization weight β and max-ent coefficient α
Require: A set of training environments $p_{\mathtt{train}}(E)$ and transfer environments $p_{\mathtt{transfer}}(E)$
Unsupervised Discovery
Sample training environment $E_{\mathtt{train}} \sim p_{\mathtt{train}}(E)$
for episodes = 1 to $\mathtt{max}-\mathtt{episodes}$ **do**
  Sample a spawn location $S_0 \sim p(s_0|E_{\mathtt{train}})$ and an option $\Omega \sim p(\omega)$
  Unroll a state-action trajectory τ under $\pi_\theta(a_t|x_t, z_t)$ for H steps with reparametrized $Z_t \sim p_\phi(z_t|x_t, \omega)$
  Infer Ω from $q_\nu(\omega|s_o, s_f)$
  Update the parameters θ, ν and ϕ based on Eqn. 6
end for
Transfer to Goal-Driven Tasks
Sample transfer environment $E_{\mathtt{transfer}} \sim p_{\mathtt{transfer}}(E)$
for episodes = 1 to $\mathtt{max}-\mathtt{episodes}$ **do**
  Sample a goal $G \sim p(g|E_{\mathtt{transfer}})$
  Unroll a state-action trajectory under the goal-conditioned policy $\pi_\gamma(a_t|x_t, g)$
  Update policy parameters γ to maximize the reward given by Eqn. 8
end for

used by Goyal *et al.* [2019]. The `MultiRoomNXSY` environments consist of X rooms of size Y, connected in random orientations. We refer to the ordering of rooms, doors and goal as a 'layout' in the `MultiRoomNXSY` environment – pre-training of options (for IR-VIC and DIAYN) is performed on a single fixed layout while transfer is performed on several different layouts (a layout is randomly selected from a set every time the environment is reset). In all pre-training environments, we fix the option trajectory length H (the number of steps an option takes before termination) to 30 steps.

We use Advantage Actor-Critic (A2C) for all experiments. Since code for InfoBot [Goyal *et al.*, 2019] was not public, we report numbers based on a re-implementation of InfoBot, ensuring consistency with their architectural and hyperparameter choices. We refer the readers to our code[2] for further details.

Baselines. We evaluate the following on quality of exploration and transfer to downstream goal-driven tasks with sparse rewards: 1) InfoBot (our implementation) – which identifies goal-driven decision states by regularizing goal information, 2) DIAYN – whose focus is unsupervised skill acquisition, but has an $I(A_t, \Omega|S_t)$ term which can be used for the bonus in Equation 8, 3) count-based exploration which uses visitation counts as exploration incentive (this corresponds to replacing $I(\Omega, Z_t|S_t, S_0)$ with 1 in Equation 8), 4) a randomly initialized encoder $p(z_t \mid \omega, x_t)$ which is a noisy version of the count-based baseline where the scale of the reward is adjusted to match the count-based baseline 5) how different values of β affect performance and how we choose a β value using a validation set, and 6) a heuristic baseline that uses domain knowledge to identify landmarks such as corners and doorways and provide a higher count-based exploration bonus to these states. This validates the extent to which necessary option information is useful in identifying a sparse set of states that are useful for transfer *vs.* heuristically determined landmarks.

[2]https://github.com/nirbhayjm/irvic

Method	MultiRoomN3S4	MultiRoomN5S4	MultiRoomN6S25
$p_\phi(Z_t\|S_t,\Omega)$ pretrained on	MultiRoomN2S6	MultiRoomN2S6	MultiRoomN2S10
InfoBot [Goyal *et al.*, 2019]	90%	85%	N/A
InfoBot (Our Implementation)	$99.9\%_{\pm 0.1\%}$	$79.1\%_{\pm 11.6\%}$	$90.9\%_{\pm 1.2\%}$
Count-based Baseline	$99.7\%_{\pm 0.1\%}$	$99.7\%_{\pm 0.1\%}$	$86.8.4\%_{\pm 2.2\%}$
DIAYN	$99.7\%_{\pm 0.1\%}$	$95.4\%_{\pm 4.1\%}$	$0.1\%_{\pm 0.1\%}$
Random Network	$99.9\%_{\pm 0.1\%}$	$98.8\%_{\pm 0.7\%}$	$79.5\%_{\pm 5.2\%}$
Heuristic Baseline	N/A	N/A	$85.9\%_{\pm 3.0\%}$
Ours ($\beta = 10^{-2}$)	$99.3\%_{\pm 0.3\%}$	$99.4\%_{\pm 0.2\%}$	$92.9\%_{\pm 1.2\%}$

Table 1: Success rate (mean $\pm$ standard error) of the goal-conditioned policy when trained with different exploration bonuses in addition to the extrinsic reward $R_e(t)$. We report results at 5×10^5 timesteps for MultiRoomN3S4, MultiRoomN5S4 and at $\times 10^7$ timesteps for MultiRoomN6S25. We also report the performance of InfoBot for completeness. Note that for rooms of size 4 (MultiRoomN3S4, MultiRoomN5S4), incentivizing to visit corners and doorways (Heuristic Sub-goals) is equivalent to count-based exploration.

3.1 Qualitative Results

Grid Worlds: Figure 3 shows heatmaps of necessary option information $I(\Omega, A_t|S_t, S_0)$ on `4-Room` and `maze` environments where the initial state is sampled uniformly at random. Stronger regularization ($\beta = 1$) leads to poorer empowerment maximization and in some cases not learning any options (and $I(\Omega, A_t|S_t, S_0)$ collapses to 0 at all states). At lower values of $\beta = \texttt{1e-3}$, we get more discernible states with distinctly high necessary option information. Finally, for `maze` we see that for a similar value of empowerment[3], IR-VIC leads to a more peaky distribution of states with high necessary option information than DIAYN.

3.2 Quantitative Results

Transfer to Goal-Driven Tasks. Next, we evaluate Equation 8, *i.e.* whether providing visitation incentive proportional to necessary option information at a state in addition to sparse extrinsic reward can aid in transfer to goal-driven tasks in different environments. We restrict ourselves to the point-navigation task [Goyal *et al.*, 2019] transfer in the `MultiRoomNXSY` set of partially-observable environments. In this task, the agent learns a policy $\pi(a_t|g_t, s_t)$ where g_t is the vector pointing to the goal from agent's current location at every time step t. The initial state is always the first room, and has to go to a randomly sampled goal location in the last room and is rewarded only when it reaches the goal. Goyal *et al.* [2019] test the efficacy of different exploration objectives [4] and show that this is a hard setting where efficient exploration is necessary. They show that InfoBot outperforms several state-of-the art exploration methods in this environment.

Concretely, we 1) train IR-VIC to identify sub-goals (Equation 2) on `MultiRoomN2S6` and transfer to a goal-driven task on `MultiRoomN3S4` and `MultiRoomN5S4` (similar to Goyal *et al.* [2019]), and 2) train on `MultiRoomN2S10` and transfer to `MultiRoomN6S25`,

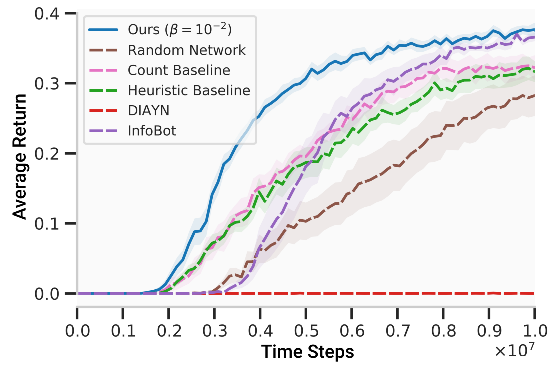

Figure 4: Transfer results on a test set of MultiRoomN6S25 environment layouts after unsupervised pre-training on MultiRoomN2S10. Shaded regions represent standard error of the mean of average return over 10 random seeds.

which is a more challenging transfer task *i.e.* it has a larger upper limit on room size making efficient exploration critical to find doors quickly. For IR-VIC and DIAYN (the two methods that learn options), we pre-train on a single layout of the corresponding `MultiRoom` environment for 10^6 episodes and pick the checkpoints with highest empowerment values across training. For InfoBot (no option learning required), we pre-train as per Goyal *et al.* [2019] on multiple layouts of the `MultiRoom` environment. Transfer performance of all methods is reported on a fixed test set of multiple `MultiRoom` environment layouts and hyperparameters across all methods, *e.g.* β for IR-VIC and InfoBot are selected using a validation set of `MultiRoom` environment layouts.

Overall Trends. Table 1 reports success rate – the % of times the agent reaches the goal and Figure 4 reports the average return when learning to navigate on test environments. The `MultiRoomN6S25` environment provides a sparse decaying reward upon reaching the goal – implying that when comparing methods, higher success rate (Table 1) indicates that the goal is reached more often, and higher return values (Figure 4) indicate that the goal is reached with fewer time steps.

First, our implementation of InfoBot is competitive with Goyal *et al.* [2019][5]. Next, for the MultiRoomN2S6 to N5S4 transfer (middle column), baselines as well as sub-goal identification methods perform well with some models having overlapping confidence intervals despite low success means. In MultiRoomN2S10 to N6S25 transfer, where the latter has a large state space, we find that IR-VIC (at $\beta = 10^{-2}$) achieves the best sample complexity (in terms of average return) and final success, followed closely by InfoBot. Moreover, we find that the heuristic baseline which identifies a sparse set of landmarks (to mimic sub-goals) does not perform well – indicating that it is not easy to hand-specify sub-goals that are useful for the given transfer task. Finally, the randomly initialized encoder as well as DIAYN generalize much worse in this transfer task.

β sensitivity. We sweep over β in log-scale from $\{10^{-1}, \cdots, 10^{-6}\}$, as shown in Figure 5 (except $\beta = 10^{-1}$

[3]Since DIAYN maximizes the mutual information with every state in a trajectory, we report the empowerment for the state with maximum mutual information with the option.

[4]While our focus is on identifying and probing how good sub-goals from intrinsic training are, more broader comparisons to exploration baselines are in InfoBot [Goyal *et al.*, 2019].

[5]We found it important to run all models (inlcuding InfoBot) an order of magnitude more steps compared to Goyal *et al.* [2019], but our models also appear to converge to higher success values.

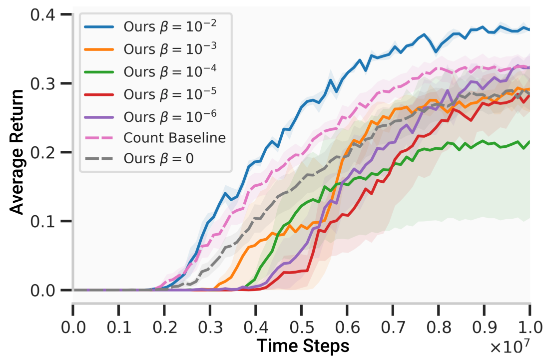

Figure 5: Evaluation of average return on a held-out validation set of `MultiRoomN6S25` environment layouts. For each value of β, pretraning is performed over 3 random seeds with the best seed being picked to measure transfer performance over 3 subsequent random seeds. Shaded regions represent standard error of the mean over the 3 random seeds used for transfer.

which does not converge to > 0 empowerment) and also report $\beta = 0$ which recovers a no information regularization baseline. We find that 10^{-2} works best – with performance tailing off at lesser values. This is intuitive, since for a really large value of β, one does not learn any options (as the empowerment is too low), while for a really small value of β, one might not be able to target necessary option information, getting large "sufficient" (but not necessary) option information for the underlying option-conditioned policy.

We pick the best model for transfer based on performance on the validation environments, and study generalization to novel test environments. Choosing the value of β in this setting is thus akin to model selection. Such design choices are inherent in general in unsupervised representation learning (*e.g.* with K-means and β-VAE Higgins *et al.* [2017]).

4 Related Work

Intrinsic Control and Intrinsic Motivation. Learning how to explore without extrinsic rewards is a foundational problem in Reinforcement Learning [Pathak *et al.*, 2017; Gregor *et al.*, 2016; Schmidhuber, 1990]. Typical curiosity-driven approaches attempt to visit states that maximize the surprise of an agent [Pathak *et al.*, 2017] or improvement in predictions from a dynamics model [Lopes *et al.*, 2012]. While curiosity-driven approaches seek out and explore novel states, they typically do not measure how reliably the agent can reach them. In contrast, approaches for intrinsic control [Eysenbach *et al.*, 2018; Achiam *et al.*, 2018; Gregor *et al.*, 2016] explore novel states while ensuring those states are reliably reachable. Gregor *et al.* [2016] maximize the number of final states that can be reliably reached by the policy, while Eysenbach *et al.* [2018] distinguish an option (which they refer to as a 'skill') at every state along the trajectory, and Achiam *et al.* [2018] learn options for entire trajectories by encoding a sub-sequence of states, sampled at regular intervals. Since we wish to learn to identify useful sub-goals which one can reach reliably acting in an environment rather than just visiting novel states (without an estimate of reachability), we formulate our regularizer in the intrinsic control framework, specifically building on the work of Gregor *et al.* [2016].

Default Behavior and Decision States. Recent work in policy compression has focused on learning a *default policy* when training on a family of tasks, to be able to re-use behavior across tasks. In Teh *et al.* [2017], default behavior is learnt using a set of task-specific policies which then regularizes each policy, while Goyal *et al.* [2019] learn a default policy using an information bottleneck on task information and a latent variable the policy conditions on, identifying sub-goals which they term as "decision states". We devise a similar information regularization objective that learns default behavior shared by all intrinsic options without external rewards so as to reduce learning pressure on option-conditioned policies. Different from these previous approaches, our approach does not need any explicit reward specification when learning options (ofcourse, since we care about transfer we still need to do model selection based on validation environments).

Bottleneck states in MDPs. There is rich literature on identification of bottleneck states in MDPs. The core idea is to either identify all the states that are common to multiple goals in an environment [McGovern and Barto, 2001] or use a diffusion model built using an MDP's transition matrix [Machado *et al.*, 2017]. The key distinction between bottleneck states and necessary-information based sub-goals is that the latter are more closely tied to the information available to the agent and what it can act upon, whereas bottleneck states are more tied to the connectivity structure of an MDP and intrinsic to the environment, representing states which when visited allow access to a novel set of states [Goyal *et al.*, 2019]. However, bottleneck states do not easily apply to partially observed environments and when the transition dynamics of the MDP are not known.

Information Bottleneck in Machine Learning. Since the foundational work of Tishby *et al.* [1999]; Chechik *et al.* [2005], there has been a lot of interest in making use of ideas from information bottleneck (IB) for various tasks such as clustering [Strouse and Schwab, 2017; Still *et al.*, 2004], sparse coding [Chalk *et al.*, 2016], classification using deep learning [Alemi *et al.*, 2016], cognitive science and language [Zaslavsky *et al.*, 2018] and reinforcement learning [Goyal *et al.*, 2019; Strouse *et al.*, 2018]. We apply an information regularizer to an RL agent that results in a set of sparse states where necessary option information is high, which correspond to our sub-goals.

5 Conclusion

We devise a principled approach to identify sub-goals in an environment without any extrinsic reward supervision using a sandwich bound on the empowerment of Gregor *et al.* [2016]. Our approach yields sub-goals that aid efficient exploration on external-reward tasks and subsequently lead to better success rate and sample complexity in novel environments (competitive to supervised sub-goals of Goyal *et al.* [2019]). All our code and environments will be made publicly available.

References

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

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2016.

Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. 2016.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for gaussian variables. *J. of Machine Learning Research*, 6(Jan):165–188, 2005.

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

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

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

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. InfoBot: Transfer and exploration via the information bottleneck. *arXiv [stat.ML]*, January 2019.

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

Barbara Hayes-Roth and Frederick Hayes-Roth. A cognitive model of planning. *Cognitive science*, 3(4):275–310, 1979.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *NIPS*, 2012.

Marios C Machado, Marc G Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In *ICML*, 2017.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML*, 2001.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

Daniel Polani, Chrystopher L Nehaniv, Thomas Martinetz, and Jan T Kim. Relevant information in optimized persistence vs. progeny strategies. In *In: Artificial Life X: Proc. of the Tenth International Conference on the Simulation and Synthesis of Living Systems*. Mit Press, 2006.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment – an introduction. October 2013.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 222–227, Cambridge, MA, USA, 1990. MIT Press.

Susanne Still, William Bialek, and Léon Bottou. Geometric clustering using the information bottleneck method. In S Thrun, L K Saul, and B Schölkopf, editors, *NIPS*. 2004.

D J Strouse and David J Schwab. The information bottleneck and geometric clustering. December 2017.

D J Strouse, Max Kleiman-Weiner, Josh Tenenbaum, Matt Botvinick, and David Schwab. Learning to share and hide intentions using information regularization. August 2018.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *NIPS*, 2017.

N. Tishby, F.C. Pereira, and W. Biale. The information bottleneck method. In *The 37th annual Allerton Conf. on Communication, Control, and Computing*, pages 368–377, 1999.

Sander G van Dijk and Daniel Polani. Grounding subgoals in information transitions. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 105–111, 2011.

Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proc. Natl. Acad. Sci. U. S. A.*, 115(31):7937–7942, July 2018.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.