Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills

Vctor Campos [1] **Alex Trott** [2] **Caiming Xiong** [2] **Richard Socher** [2] **Xavier Giro-i-Nieto** [3] **Jordi Torres** [1]

Abstract

Acquiring abilities in the absence of a task-oriented reward function is at the frontier of reinforcement learning research. This problem has been studied through the lens of *empowerment*, which draws a connection between option discovery and information theory. Information-theoretic skill discovery methods have garnered much interest from the community, but little research has been conducted in understanding their limitations. Through theoretical analysis and empirical evidence, we show that existing algorithms suffer from a common limitation – they discover options that provide a poor coverage of the state space. In light of this, we propose *Explore, Discover and Learn* (EDL), an alternative approach to information-theoretic skill discovery. Crucially, EDL optimizes the same information-theoretic objective derived from the empowerment literature, but addresses the optimization problem using different machinery. We perform an extensive evaluation of skill discovery methods on controlled environments and show that EDL offers significant advantages, such as overcoming the coverage problem, reducing the dependence of learned skills on the initial state, and allowing the user to define a prior over which behaviors should be learned.

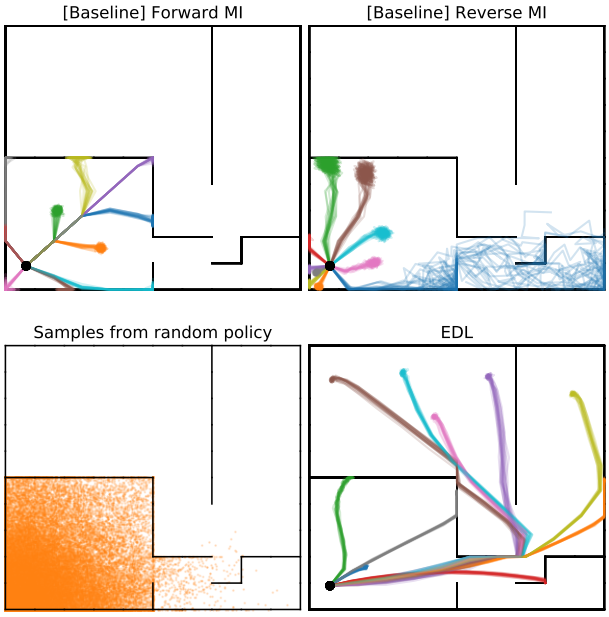

Figure 1. Skills learned on a maze with bottleneck states. Each colored line represents a trajectory initiated at the black dot by a different skill. Multiple rollouts per skill are reported in order to account for the stochasticity of the policy. The bottom left plot depicts states visited by a policy with random weights, showing which states are reachable by the agent at the beginning of training. Existing methods fail at expanding this set of states, and end up committing to behaviors discovered by the random policy. On the other hand, EDL discovers skills that provide a better coverage of the state space.

1. Introduction

Reinforcement learning (RL) algorithms have recently achieved outstanding goals thanks to advances in simulation (Todorov et al., 2012; Bellemare et al., 2013), efficient and scalable learning algorithms (Mnih et al., 2015; Lillicrap et al., 2016; Schulman et al., 2015; 2017; Espeholt et al., 2018), function approximation (LeCun et al., 2015; Goodfellow et al., 2016), and hardware accelerators (NVIDIA, 2017; Jouppi et al., 2017). These landmarks include outperforming humans in board (Silver et al., 2017) and computer games (Mnih et al., 2015; Vinyals et al., 2019), and solving complex robotic control tasks (Andrychowicz et al., 2017; Akkaya et al., 2019). Typically, training aims to solve a particular task, relying on task-specific reward functions to measure progress and drive learning. This contrasts with how intelligent creatures learn in the absence of external supervisory signals, acquiring abilities in a task-agnostic manner by exploring the environment. Methods for training models without expert supervision have already obtained promising results in fields like natural language process-

[1]Barcelona Supercomputing Center [2]Salesforce Research [3]Universitat Politcnica de Catalunya. Correspondence to: Vctor Campos <victor.campos@bsc.es>.

Preprint. Under review.

ing (Radford et al., 2019; Devlin et al., 2019) and computer vision (Hénaff et al., 2019; He et al., 2019). In RL, analogous "unsupervised" methods are often aimed at learning generically useful behaviors for interacting within some environment, behaviors that may naturally accelerate learning once one or more downstream tasks become available.

The idea of unsupervised RL is often formulated through the lens of *empowerment* (Salge et al., 2014), which formalizes the notion of an agent discovering *what* can be done in an environment while learning *how* to do it. Central to this formulation is the concept of mutual information, a tool from information theory. Mohamed & Rezende (2015) derived a variational lower bound on the mutual information which can be used to learn options (Sutton et al., 1999) in a task-agnostic fashion. Following classical empowerment (Salge et al., 2014), options are discovered by maximizing the mutual information between sequences of actions and final states. This results in *open loop* options, where the agent commits to a sequence of actions *a priori* and follows them regardless of the observations received from the environment. Gregor et al. (2016) developed an algorithm to learn *closed loop* options, whose actions are conditioned on the state, by maximizing the mutual information between states and some latent variables instead of action sequences. This approach has been extended by several works, which are surveyed in Section 2. Despite the interest in developing information-theoretic skill discovery methods, very little research has been conducted in understanding the limitations of these algorithms. We distinguish two categories of such limitations. The first type has to do with the nature of the objective itself, e.g. the difficulty of purely information-theoretic methods for capturing human priors (Achiam et al., 2018). In this work, we focus on the second group – those issues introduced when adapting the objective to current optimization methods.

In order to maximize empowerment, an agent needs to learn to control the environment while discovering available options. It should not aim for states where it has the most control according to its current abilities, but for states where it expects it will achieve the most control *after* learning (Gregor et al., 2016). We empirically observe that this is not achieved by existing methods, which prematurely commit to already discovered options instead of exploring the environment to unveil novel ones. Figure 1 (top) showcases this failure mode when deploying existing algorithms on a 2D maze. We provide theoretical analysis showing that these methods tend to reinforce already discovered behaviors at the expense of exploring in order to discover new ones, resulting in behaviors that exhibit poor coverage of the available state space. Figure 1 (bottom right) depicts the skills discovered by our proposed *Explore, Discover and Learn* (EDL) paradigm, a three-stage methodology that is able to discover skills with much better coverage.

Our contributions can be summarized as follows. (1) We provide theoretical analysis and empirical evidence showing that existing skill discovery algorithms fail at learning state-covering skills. (2) We propose an alternative approach to information-theoretic option discovery, *Explore, Discover and Learn* (EDL), that overcomes the limitations of existing methods. Crucially, EDL achieves this while optimizing the same information-theoretic objective as previous methods. (3) We validate the presented paradigm by implementing a solution that follows the three-stage methodology. Through extensive evaluation in controlled environments, we demonstrate the effectiveness of EDL, showcase its advantages over existing methods, and analyze its current limitations and directions for future research.

2. Information-theoretic skill discovery

This section presents a generic mathematical framework that can be used to formulate, analyze and compare information-theoretic skill discovery methods in the literature. Let us consider a Markov Decision Process (MDP) $\mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, p)$ with state space $\mathcal{S}$, action space $\mathcal{A}$ and transition dynamics p. We learn latent-conditioned policies $\pi(a|s, z)$, and define *skills* or *options* as the policies obtained when conditioning π on a fixed value of $z \in \mathcal{Z}$ (Eysenbach et al., 2019; Achiam et al., 2018). Let $S \sim p(s)$ be a random variable denoting states such that $S \in \mathcal{S}$, and $Z \sim p(z)$ be a random variable for latent variables. Using notation from information theory, we use $I(\cdot;\cdot)$ and $H(\cdot)$ to refer to the mutual information and Shannon entropy, respectively. Information-theoretic skill discovery methods seek to find a policy that maximizes the mutual information between S and Z. Due to symmetry, this measure can be expressed in the following two forms:

$$I(S;Z) = H(Z) - H(Z|S) \qquad \textit{// reverse} \qquad (1)$$
$$= H(S) - H(S|Z) \qquad \textit{// forward} \qquad (2)$$

For presentation clarity, we follow Gregor et al. (2016) and refer to Equations 1 and 2 as the *reverse* and *forward* forms of the mutual information, respectively.

Our goal is to analyze which fundamental design choices are responsible for the properties and limitations of such algorithms. We classify existing skill discovery methods depending on the form of the mutual information they optimize, and implement a canonical algorithm for each form that allows for fair comparison. The following subsections describe existing skill discovery methods as well as the specific implementations considered in this work.

2.1. Reverse form of the mutual information

The objective can be derived by expanding the definition of the mutual information in Equation 1, and then leveraging the non-negativity property of the KL divergence to compute

a variational lower bound (Barber & Agakov, 2003):

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{3}$$

$$\geq \mathbb{E}_{s,z\sim p(s,z)}[\log q_\phi(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{4}$$

where $q_\phi(z|s)$ is fitted by maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. This implicitly approximates the unknown posterior as $p(z|s) \approx \rho_\pi(z|s)$, where $\rho_\pi(z|s)$ is the empirical posterior induced by the policy.

Note that computing this measure will require sampling from two distributions, $p(z)$ and $p(s,z)$. The distribution over latent variables $p(z)$ can be learned as part of the optimization process or fixed beforehand, with the latter often yielding superior results (Eysenbach et al., 2019). However, sampling from the joint distribution over states and latents $p(s,z)$ is more problematic. A common workaround consists in assuming a generative model of the form $p(s,z) = p(z)p(s|z) \approx p(z)\rho_\pi(s|z)$, where $\rho_\pi(s|z)$ is the stationary state-distribution induced by $\pi(a|s,z)$ (Gregor et al., 2016).

This category includes a variety of methods with slight differences. VIC (Gregor et al., 2016) considers only the final state of each trajectory and a learnable prior $p(z)$. SNN4HRL (Florensa et al., 2017) introduces a task-specific proxy reward, which encourages exploration and can be understood as a bonus to increase the entropy of the stationary state-distribution. DIAYN (Eysenbach et al., 2019) additionally minimizes the mutual information between actions and skills given the state, resulting in a formulation that resembles MaxEnt RL (Haarnoja et al., 2017). VALOR (Achiam et al., 2018) considers the posterior over sequences of states instead of individual states in order to encourage learning dynamical modes rather than goal-attaining modes. DISCERN (Warde-Farley et al., 2019) and Skew-Fit (Pong et al., 2019) aim at learning a goal-conditioned policy in an unsupervised fashion, which can be understood as skill discovery methods where Z takes the form of states sampled from a buffer of previous experience.

We consider a variant of VIC (Gregor et al., 2016) with a fixed prior $p(z)$ and where all states in a trajectory are considered in the objective[1]. This method can be seen as a version of DIAYN (Eysenbach et al., 2019) where the scale of the entropy regularizer $H(a|s,z)$ is set to 0. The variational lower bound in Equation 4 is optimized by training the policy $\pi(a|s,z)$ using the reward function

$$r(s,z') = \log q_\phi(z'|s) - \log p(z'),\ z' \sim p(z) \tag{5}$$

[1]The original implementation by Gregor et al. (2016) considered final states only, thus providing a sparser reward signal to the policy.

2.2. Forward form of the mutual information

A similar lower bound to that in Equation 4 can be derived by expanding the forward form of the mutual information in Equation 2:

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(s|z)] - \mathbb{E}_{s\sim p(s)}[\log p(s)] \tag{6}$$

$$\geq \mathbb{E}_{s,z\sim p(s,z)}[\log q_\phi(s|z)] - \mathbb{E}_{s\sim p(s)}[\log p(s)] \tag{7}$$

where $q_\phi(s|z)$ is fitted by maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. This amounts to approximating $p(s|z)$ with the stationary state-distribution of the policy, $p(s|z) \approx \rho_\pi(s|z)$.

To the best of our knowledge, DADS (Sharma et al., 2019) is the only method within this category. DADS follows a model-based setup where $I(S_{t+1};Z|S_t)$ is maximized. This is achieved by modelling changes in the state, $\Delta s = s_{t+1} - s_t$. When evaluated on locomotion environments that encode the position of the agent in the state vector, this setup favors the discovery of gaits that move in different directions. Similarly to methods in Section 2.1, $p(s,z)$ is approximated by relying on the stationary state-distribution induced by the policy and $p(s) \approx \rho_\pi(s) = \mathbb{E}_z\left[\rho_\pi(s|z)\right]$. We will consider a model-free variant of DADS where the variational lower bound in Equation 7 is optimized by training the policy $\pi(a|s,z)$ with a reward function

$$r(s,z') = \log q_\phi(s|z') - \log \frac{1}{L}\sum_{i=1}^{L} q_\phi(s|z_i),\ z', z_i \sim p(z) \tag{8}$$

where $p(s)$ is approximated using q_ϕ and L random samples from the prior $p(z)$ as done by Sharma et al. (2019). When using a discrete prior, we marginalize over all skills.

2.3. Limitations of existing methods

Recall that maximizing empowerment implies fulfilling two tasks, namely discovering what is possible in the environment and learning how to achieve it. In preliminary experiments, we observed that existing methods discovered skills that provide a poor coverage of the state space. This suggests a limited capability for discovering what options are available.

Maximizing the mutual information between states and latents requires knowledge of some distributions. Methods based on the forward form of the mutual information make use of $p(s|z)$ and $p(s)$, whereas those using the reverse form employ $p(z|s)$. Note that none of these are known *a priori*, so the common practice is to approximate them using the distributions induced by the policy. Distributions over states are approximated with the stationary state-distribution of the

policy, $p(s|z) \approx \rho_\pi(s|z)$ and $p(s) \approx \rho_\pi(s) = \mathbb{E}_z\left[\rho_\pi(s|z)\right]$. The posterior $p(z|s)$ is approximated with the empirical distribution induced by running the policy, $p(z|s) \approx \rho_\pi(z|s)$. In practice, these distributions are estimated via maximum likelihood using rollouts from the policy.

We analyze the asymptotic behavior of the reward function for existing methods under the aforementioned approximations through a theoretical lens. The analysis considers an agent aiming to discover N discrete skills, and perfect estimations of all distributions. Our main result shows that the agent receives larger rewards for visiting known states than discovering new ones. Known states can receive a reward of up to $r_{\text{max}} = \log N$. On the other hand, previously unseen states will receive a smaller reward, $r_{\text{new}} = 0$. These observations hold for the forward and reverse forms of the mutual information, and provide theoretical insight for why existing methods do not discover state-covering skills. We refer the reader to the Supplementary Material (SM) for a detailed derivation of the results.

In order to provide preliminary evidence for this result, we deploy the described algorithms on a 2D maze with bottleneck states (see Section 4 for details on the experimental setup). As shown in Figure 1 (top), existing methods fail at exploring the maze and most options just visit different regions of the initial room. Figure 1 (bottom left) depicts states visited by a policy with the same architecture, but random weights. Note that existing algorithms do not expand the set of states visited by this random policy, but simply identify and reinforce different modes of behavior among them. This observation confirms that existing formulations fail at discovering available options, and motivates our study of alternative methods for option discovery.

3. Proposed method

Maximizing the mutual information between states and latent variables requires access to unknown distributions, which existing methods approximate using the distributions induced by the policy. Instead of encouraging the agent to discover available options, this approximation reduces the problem to that of reinforcing already discovered behaviors. Since the policy is initialized randomly at the beginning of training, the discovered options seldom explore further than a random policy.

We propose an alternative approach, *Explore, Discover and Learn* (EDL), for modelling these unknown distributions and performing option discovery. Existing methods make use of the state distribution $p(s) \approx \rho_\pi(s) = \mathbb{E}_z\left[\rho_\pi(s|z)\right]$, which focuses $p(s)$ around states where the policy receives a high reward. This dependency contributes to the pathological learning dynamics described above. To break this dependency, EDL makes use of a *fixed* distribution over states $p(s)$ and is agnostic to the method by which this distribution is discovered/obtained. For a given distribution over states, EDL makes use of variational inference techniques to model $p(s|z)$ and $p(z|s)$. As its name suggests, EDL is composed of three stages: (*i*) exploration, (*ii*) skill discovery, and (*iii*) skill learning. These can be studied and improved upon independently, and the actual implementation of each stage will depend on the problem being addressed. The compartmentalization of these facets of the objective, together with the inclusion of a fixed distribution over states, are the key features of EDL. Table 1 positions this new approach with respect to existing ones.

Exploration. In the absence of any prior knowledge, a reasonable choice for the distribution over states $p(s)$ is a uniform distribution over all $\mathcal{S}$, which will encourage the discovery of state-covering skills. This stage comes with the challenge of being able to generate or sample from the distribution of states that the learned skills should ultimately cover. This is generally a difficult problem, for which we consider possible solutions. When an oracle is available, it can be queried for samples belonging to the set of valid states. If such an oracle is not available, one can train an exploration policy that induces a uniform distribution over states. Finding these policies is known as the problem of maximum entropy exploration, for which provably efficient algorithms exist under certain conditions (Hazan et al., 2019). When interested in some particular modes of behavior, one can leverage a more specific state distribution or adopt a non-parametric solution by sampling states from a dataset of extrinsically generated experience (Guss et al., 2019). Note that unlike approaches in imitation learning (Ho & Ermon, 2016), learning from demonstrations (Hester et al., 2017; Večerík et al., 2017), and learning from play (Lynch et al., 2019), EDL does not require access to trajectories or actions emitted by an expert policy.

Skill discovery. Whereas existing methods sample skill $z \sim p(z)$ directly as an input to the skill-conditioned policy, EDL requires an indirect approach wherein latent skills are inferred from $p(s)$. More concretely, given a distribution over states, or samples from it, we treat skill discovery as learning to model $p(z|s)$ and $p(s|z)$. We turn to variational inference techniques for this purpose, and Variational Autoencoders (VAE) (Kingma & Welling, 2014) in particular. Fortunately, we can approximate both distributions by training a VAE on samples from $p(s)$ – the encoder q_ψ models $p(z|s)$, whereas the decoder q_ϕ models $p(s|z)$. Intuitively, this process determines which latent codes are assigned to each region of the state space, and which states should be visited by each skill. The fact that exploration and skill discovery are disentangled enables learning variational posteriors for different $p(z)$ priors without needing to re-learn a new skill-condition policy every time. This is an interesting property, as the task of defining the prior over skills is

Assumptions	MI form	Methods
$p(z)$: fixed $p(z\|s) \approx \rho_\pi(z\|s)$ $p(s\|z) \approx \rho_\pi(s\|z)$ $p(s) \approx \rho_\pi(s) = \mathbb{E}_z\left[\rho_\pi(s\|z)\right]$	Forward	DADS (Sharma et al., 2019)
	Reverse	VIC (Gregor et al., 2016), SNN4HRL (Florensa et al., 2017), DIAYN (Eysenbach et al., 2019), VALOR (Achiam et al., 2018), DISCERN (Warde-Farley et al., 2019), Skew-Fit (Pong et al., 2019)
$p(z), p(s)$: fixed $p(s\|z)$, $p(z\|s)$: modelled with VI	Forward	EDL (ours)

Table 1. Types of methods depending on the considered generative model and the version of the mutual information (MI) being maximized. Distributions denoted by ρ are induced by running the policy in the environment, whereas p is used for the true and potentially unknown ones. The dependency of existing methods on $\rho_\pi(s|z)$ causes pathological training dynamics by letting the agent influence over the states considered in the optimization process. EDL relies on a fixed distribution over states $p(s)$ to break this dependency and makes use of variational inference (VI) techniques to model $p(s|z)$ and $p(z|s)$.

not straightforward. In contrast, previous methods perform exploration and skill discovery at the same time, so that modifying $p(z)$ inevitably involves exploring the environment from scratch.

Skill learning. The final stage consists in training a policy $\pi_\theta(s, z)$ that maximizes the mutual information between states and latent variables. EDL adopts the forward form of the mutual information. The reader is referred to the SM for a detailed explanation of this choice. Since $p(s)$ is fixed, Equation 7 can be maximized in a reinforcement learning-styled setup with the reward function

$$r(s, z') = \log q_\phi(s|z'),\ z' \sim p(z) \tag{9}$$

where $q_\phi(s|z)$ is given by the decoder of the VAE trained on the skill discovery stage. This final stage can be seen as training a policy that mimics the decoder *within* the MDP, i.e. a policy that will visit the state that the decoder would generate for each latent code z. Note that the reward function is fixed, unlike that in previous methods which continuously changes depending on the behavior of the policy.

4. Experiments

Some previous works have evaluated skill discovery methods on complex environments, such as robotic locomotion (Todorov et al., 2012) or 3D navigation (Beattie et al., 2016), whose complexity renders policy learning difficult. This burden falls on the underlying RL algorithm, which needs to learn a more complicated policy in order to achieve the desired behavior. Note that this does not necessarily make the task of discovering options more difficult. As an example, consider the process of discovering useful locomotion skills. These options will likely require the agent to move in different directions, no matter if it is controlling a simple point mass or a complex humanoid.

In this work, we take a different approach and consider controlled synthetic environments. These are fully-continuous 2D mazes where the agent observes its current position and outputs actions that control its location, which is affected by collisions with walls. Varying the maze topology allows for an analysis of skill discovery methods in the face of specific challenges, providing insight on the properties and limitations of these algorithms.

All experiments consider discrete priors over skills. This choice allows for a fair comparison between methods, as those based on the reverse form of the mutual information are not straightforward to combine with continuous priors. We consider the two methods described in Section 2 as baselines. The skill discovery stage in EDL is performed with a VQ-VAE (van den Oord et al., 2017) to handle the discrete prior, and the real-valued codes it discovers are used to condition the policy. We adopt the common distributional assumption for continuous data where $p(s|z)$ is Gaussian (Kingma & Welling, 2014), which does not consider the actual connectivity within the MDP and results in reward functions that can become fraught with local optima. For this reason, we used Sibling Rivalry (Trott et al., 2019) to escape local optima during the skill learning stage in some environments. Note that the described implementation is just a possible solution that follows the proposed paradigm, which is not limited to the specific choices made in our experimental setup.

Figures 1–5 visualize multiple rollouts per skill to account for the stochasticity of the policy. The initial state is denoted by a black dot and the color of the rollout denotes the skill upon which it was conditioned, thus figures are best viewed in color. We refer the reader to the SM for a detailed description of the experimental setup and the hyperparameters.

Exploration with SMM. In the absence of any prior knowledge, we would like to discover skills across the whole state space by defining a uniform distribution over states, $p(s)$. In the controlled environments considered in this work, this can be achieved by sampling states from an oracle. In order to understand the impact of not having access to an oracle,

we employ State Marginal Matching (SMM) (Lee et al., 2019) with a uniform target distribution to perform the exploration stage in EDL. Evaluation is performed on a simple maze where the forward and reverse baselines already fail to learn state-covering skills, as depicted in Figure 2 (top). In contrast, Figure 2 (bottom) shows how EDL can learn state-covering skills even in the realistic scenario where an oracle is not available[2].

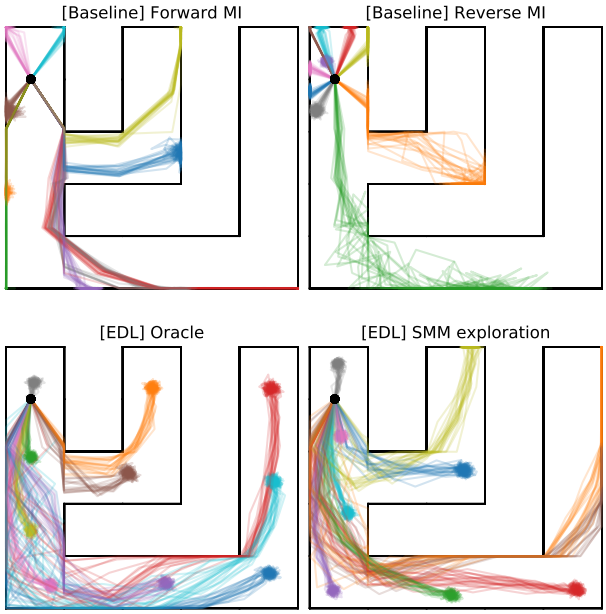

Figure 2. Impact of replacing the oracle with State Marginal Matching (SMM) in the exploration stage. *Top:* baselines fail at discovering skills that reach the right side of the maze. *Bottom:* EDL discovers skills that are spread across the whole maze, even when replacing the oracle with SMM. We observed that SMM tended to collect more samples near the walls, which explains the slight difference in the discovered options.

Impact of the initial state. Baseline methods rely on $\rho_\pi(s|z)$ to perform skill discovery, which is initially induced by a random policy. This introduces a strong dependence on the distribution over initial states, $p(s_0)$. Changes to $p(s_0)$ might make some behaviors harder to learn, e.g. reaching a certain position becomes more difficult the further an agent spawns from it. As long as all options are still achievable, a change in $p(s_0)$ should have little impact on what options are deemed important. We evaluate this phenomenon on two corridor-shaped mazes, which have the same topology but differ in the position of the initial state. We will refer to these environments as E_{center} and E_{left}, in which the agent spawns in the center and the left section of the corridor, respectively. Figure 3 (top) shows how the baselines discover completely different skills depending on $p(s_0)$. When replicating this experiment using EDL with SMM exploration, we get two different setups. Figure 3 (bottom left) shows the result of performing exploration and skill discovery in E_{center} and then learning skills in both E_{center} and E_{left}. Figure 3 (bottom right) depicts the impact of performing exploration and skill discovery in E_{left} instead. Skills learned in both setups are very similar, with differences coming from the slightly different distribution over states collected by SMM.

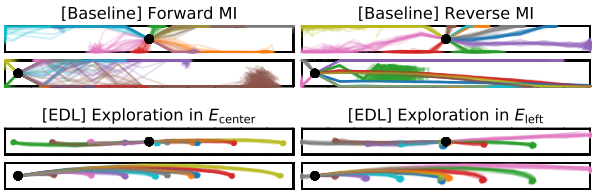

Figure 3. Impact of the distribution over initial states, $p(s_0)$. *Top:* baselines are very sensitive to $p(s_0)$ and discover very different skills depending on this distribution. *Bottom:* we report two different experiments with EDL. The setup on the left performs exploration and skill discovery in E_{center} and then learns skills in both E_{center} and E_{left}. The one on the right performs exploration and skill discovery in E_{left} instead. Options discovered by EDL are very similar in both setups.

Encouraging specific behaviors. In many settings, the user has some knowledge about which areas of state space will be most relevant for downstream tasks. Existing methods can leverage prior knowledge by maximizing $I(f(S); Z)$ instead of $I(S; Z)$, where $f(S)$ is a function of the states. For instance, this function can compute the center of mass of a robot in order to encourage the discovery of locomotion skills (Eysenbach et al., 2019). However, this method fails at incorporating more complex priors, such as encouraging the agent to only learn locomotion skills that move in specific directions. EDL offers more flexibility for leveraging priors through the definition of $p(s)$, e.g. by drawing samples from a dataset of human play (Guss et al., 2019). We simulate this scenario by performing skill discovery with an oracle that samples states uniformly from a subset of the state space. Figure 4 reports results in a tree-shaped maze, where we introduce the prior that skills should visit the right side of the maze only. EDL effectively incorporates this prior, and learns state-covering skills in its absence.

Impact of bottleneck states. The maze with bottleneck states from Figure 1, where baseline approaches fail to explore a large extent of the state space, is a challenging environment where the limitations of EDL can be evaluated. We were unable to explore this type of maze effectively with SMM. Given that SMM relies on a curiosity-like bonus (Lee et al., 2019), we attribute this failure to well-known issues of

[2]We succeeded at training skills discovered by EDL without Sibling Rivalry. However, it greatly reduced the number of runs in the grid search that got trapped in local optima. The presented results used Sibling Rivalry to take advantage of this fact and reduce variance in the results.

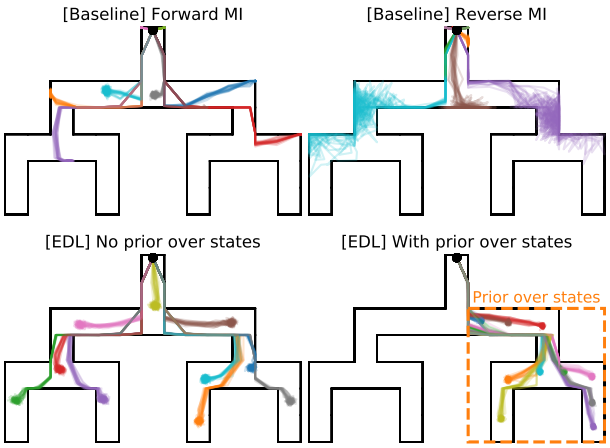

Figure 4. Incorporating priors over skills, where we are interested in learning skills on the right side of the maze. *Top:* this type of prior cannot be incorporated into baseline methods, whose discovered options are agnostic to it. *Bottom:* in the absence of a prior, EDL learns options across the whole state space. When incorporating the prior, the agent devotes all its capacity to learning skills within the region of interest.

these methods such as derailment and detachment (Ecoffet et al., 2019). Note that these problems are related to the sub-optimality of the density estimation method and RL solver, as shown by the bounds derived by Hazan et al. (2019). In light of this, we rely on an oracle to simulate perfect exploration and evaluate the skill discovery/learning stages of EDL. On this maze, the reward functions that EDL introduces create deceptive local optima in which policies tend to get stuck (visualizations are reported in the SM). Sibling Rivalry proved crucial to avoid these failures, and allowed the policy to learn the skills depicted in Figure 1 (bottom right). These observations suggest that the main bottlenecks for the proposed approach to skill discovery are maximum entropy exploration and avoiding local optima when learning to maximize the EDL reward (Equation 9). Given that EDL decouples the process in three stages, advances in these fields are straightforward to incorporate and will boost the performance of this type of option discovery method.

Interpolating between skills. The skill discovery stage in EDL with a categorical prior $p(z)$ can be seen as the process of learning a discrete number of goals, together with an embedded representation for each of them. In the experimental setup presented in this work, each embedded representation corresponds to one of the continuous vectors in the VQ-VAE's codebook, z_i, whereas each goal state is given by $g_i = \text{argmax}_s q_\phi(s|z_i)$. The idea of goal embeddings was introduced as part the Universal Value Function Approximators (UVFA) framework (Schaul et al., 2015). UVFAs can generalize to unseen goals, and here we explore how the policies learned by EDL generalize to unseen latents z – where we construct new latents by interpolating the ones discovered by EDL. The results in Figure 5 suggest that the policy learns to generalize, with interpolated skills reaching states that come from the interpolation in Euclidean space of the goals of the original skills. Additional visualizations are included in the SM.

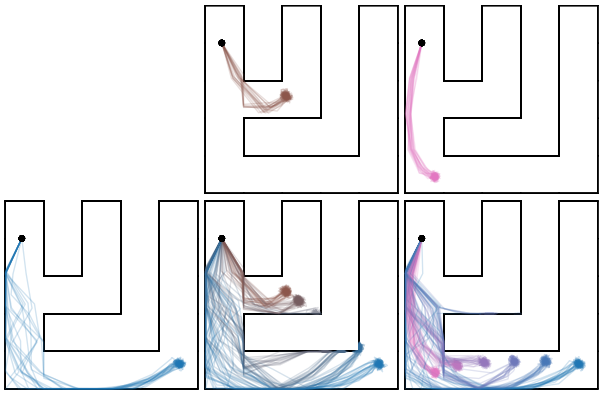

Figure 5. Interpolating skills learned by EDL. Interpolation is performed at the latent variable level by blending the z vector of two skills. The first row and column show the original skills being interpolated, which were selected randomly from the set of learned options. When plotting interpolated skills, we blend the colors used for the original skills.

5. Related work

Option discovery. Temporally-extended high-level primitives, also known as options, are an important resource in the RL toolbox (Parr & Russell, 1998; Sutton et al., 1999; Precup, 2001). The process of defining options involves task-specific knowledge, which might be difficult to acquire and has motivated research towards methods that automatically discover such options. These include learning options while solving the desired task (Bacon et al., 2017), leveraging demonstrations (Fox et al., 2017), training goal-oriented low-level policies (Nachum et al., 2018), and meta-learning primitives from a distribution of related tasks (Frans et al., 2018). Skills discovered by information-theoretic methods such as the ones considered in this work have also been used as primitives for Hierarchical RL (Florensa et al., 2017; Eysenbach et al., 2019; Sharma et al., 2019).

Intrinsic rewards. Agents need to encounter a reward before they can start learning, but this process might become highly inefficient in sparse reward setups when relying on standard exploration techniques (Osband et al., 2016). This issue can be alleviated by introducing intrinsic rewards, i.e. denser reward signals that can be automatically computed. These rewards are generally task-agnostic and might come from state visitation pseudo-counts (Bellemare et al., 2016; Tang et al., 2017), unsupervised control

tasks (Jaderberg et al., 2017), learning to predict environment dynamics (Houthooft et al., 2016; Pathak et al., 2017; Burda et al., 2018), self-imitation (Oh et al., 2018), and self-play (Sukhbaatar et al., 2017; Liu et al., 2019).

Novelty Search. Discovering a set of diverse and task-agnostic behaviors in the absence of a fitness function has been explored in the evolutionary computation community (Lehman & Stanley, 2011a;b). Quality Diversity algorithms aim at combining the best of both worlds by optimizing task-specific fitness functions while encouraging diverse behaviors in a population of agents (Cully et al., 2015; Mouret & Clune, 2015; Pugh et al., 2016). These methods rely on a behavior characterization function, which is tasked with summarizing the behavior of an agent into a vector representation. There have been efforts towards learning such functions (Meyerson et al., 2016), but it is still a common practice for practitioners to design a different function for each task (Conti et al., 2017).

Goal-oriented RL. The standard RL framework can be extended to consider policies and reward functions that are conditioned on some goal $g \in \mathcal{G}$ (Schaul et al., 2015). Given a known distribution over goals $p(g)$ that the agent should achieve, this setup allows for efficient training techniques involving experience relabeling (Andrychowicz et al., 2017) and reward shaping (Trott et al., 2019). Defining such distribution requires expert domain knowledge, an assumption that is not always fulfilled. As a result, methods that can learn to reach *any* given state have garnered research interest (Warde-Farley et al., 2019; Pong et al., 2019). These approaches can be seen as skill discovery algorithms where $\mathcal{Z} = \mathcal{S}$, i.e. where each goal state defines a different skill. This raises the question of whether methods that can reach any state are superior to those learning a handful of skills. We argue that the latter offer important benefits in terms of exploration when used by a meta-controller to solve downstream tasks. When $p(z)$ is a simple distribution, the meta-controller benefits from a reduced search space, which is one of the motivations behind building hierarchies and options (Precup, 2001). On the other hand, exploring with state-reaching policies involves a search space of size $|\mathcal{S}|$. This figure will quickly increase as the complexity of the environment grows, making exploration inefficient. Moreover, this setup assumes that the meta-controller is to be able to sample from $\mathcal{S}$ in order to emit goals for the goal-conditioned policy.

6. Discussion

We provide theoretical and empirical evidence that poor state space coverage is a predominant failure mode of existing skill discovery methods. The information-theoretic objective requires access to unknown distributions, which these methods approximate with those induced by the policy. These approximations lead to pathological training dynamics where the agent obtains larger rewards by visiting already discovered states rather than exploring the environment. We propose *Explore, Discover and Learn* (EDL), a novel option discovery approach that leverages a fixed distribution over states and variational inference techniques to break the dependency on the distributions induced by the policy. Importantly, this alternative approach optimizes the same objective derived from information theory used in previous methods. EDL succeeds at discovering state-covering skills in environments where previous methods failed. It offers additional advantages, such as being more robust to changes in the distribution of the initial state and enabling the user to incorporate priors over which behaviors are considered useful. Our experiments suggest that EDL discovers a meaningful latent space for skills even when tasked with learning a discrete set of options, whose latent codes can be combined in order to produce a richer set of behaviors.

The proposed EDL paradigm is not limited to the implementation considered in this work. Each of the three stages of the method poses its own challenges, but can benefit from advances in their respective research directions as well. This modular design allows us to incorporate recent advances to our implementation such as exploration with State Marginal Matching (Lee et al., 2019), vector quantization techniques to impose discrete priors in VAEs (van den Oord et al., 2017), and relabeling techniques to optimize deceptive reward functions (Trott et al., 2019). Future breakthroughs in these directions could contribute towards scaling up skill discovery methods to richer environments, potentially leading to the emergence of complex behaviors (Jaderberg et al., 2019).

There are several research directions to be explored in future work. Improvements in pure exploration methods would make EDL applicable to a broader range of environments. Despite the existence of strong theoretical results (Hazan et al., 2019), these approaches involve the optimization of reward functions that are challenging for current algorithms (Ecoffet et al., 2019). In our experiments, we adopted the common distributional assumption for continuous data where $p(s|z)$ is Gaussian (Kingma & Welling, 2014). This assumption was responsible for the deceptive reward functions discovered in our experiments, and might be detrimental in some other environments. This motivates research towards discovering embedding spaces for states where distances are related to the closeness of states within the MDP (Florensa et al., 2019), and learning reward functions that reflect similarity in controllable aspects of the environment (Warde-Farley et al., 2019).

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation and the European Regional Development Fund under contracts TEC2016-75976-R and TIN2015-65316-P, by the BSC-CNS Severo Ochoa program SEV-2015-0493, and grant 2017-SGR-1414 by Generalitat de Catalunya. Víctor Campos was supported by Obra Social "la Caixa" through La Caixa-Severo Ochoa International Doctoral Fellowship program.

References

Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *NIPS*, 2017.

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *AAAI*, 2017.

Barber, D. and Agakov, F. V. The IM algorithm: a variational approach to information maximization. In *NIPS*, 2003.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2013.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Capdevila, J., Cerquides, J., and Torres, J. Mining urban events from the tweet stream through a probabilistic mixture model. *Data mining and knowledge discovery*, 2018.

Chou, P.-W., Maturana, D., and Scherer, S. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In *ICML*, 2017.

Conti, E., Madhavan, V., Such, F. P., Lehman, J., Stanley, K. O., and Clune, J. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint arXiv:1712.06560*, 2017.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. Robots that can adapt like animals. *Nature*, 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *ICML*, 2018.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *ICLR*, 2019.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. In *ICLR*, 2017.

Florensa, C., Degrave, J., Heess, N., Springenberg, J. T., and Riedmiller, M. Self-supervised learning of image embedding for continuous control. *arXiv preprint arXiv:1901.00943*, 2019.

Fox, R., Krishnan, S., Stoica, I., and Goldberg, K. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. Meta learning shared hierarchies. In *ICLR*, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Guss, W. H., Codel, C., Hofmann, K., Houghton, B., Kuno, N., Milani, S., Mohanty, S., Liebana, D. P., Salakhutdinov, R., Topin, N., et al. The minerl competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:1904.10079*, 2019.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *ICML*, 2017.

Hazan, E., Kakade, S. M., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *ICML*, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., et al. Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *NIPS*, 2016.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In *NIPS*, 2016.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*, 2017.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 2019.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. In-datacenter performance analysis of a tensor processing unit. In *ISCA*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 2015.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Lehman, J. and Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 2011a.

Lehman, J. and Stanley, K. O. Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*. Springer, 2011b.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

Liu, H., Trott, A., Socher, R., and Xiong, C. Competitive experience replay. In *ICLR*, 2019.

Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. Learning latent plans from play. *arXiv preprint arXiv:1903.01973*, 2019.

Meyerson, E., Lehman, J., and Miikkulainen, R. Learning behavior characterizations for novelty search. In *GECCO*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *NIPS*, 2015.

Mouret, J.-B. and Clune, J. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *NeurIPS*, 2018.

NVIDIA. Nvidia tesla v100 gpu architecture. 2017.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. In *ICML*, 2018.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. In *ICML*, 2016.

Parr, R. and Russell, S. J. Reinforcement learning with hierarchies of machines. In *NIPS*, 1998.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

Pong, V. H., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Precup, D. Temporal abstraction in reinforcement learning. 2001.

Pugh, J. K., Soros, L. B., and Stanley, K. O. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 2016.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Salge, C., Glackin, C., and Polani, D. Empowerment – an introduction. In *Guided Self-Organization: Inception*. Springer, 2014.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *ICML*, 2015.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *ICML*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 2017.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 1999.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *NIPS*, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

Trott, A., Zhen, S., Xiong, C., and Socher, R. Sibling rivalry. In *NeurIPS*, 2019.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *NeurIPS*, 2017.

Večerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 2019.

Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *ICLR*, 2019.

A. Theoretical analysis of existing methods

This section provides insight for why existing methods do not encourage the discovery state-covering skills from a theoretical lens. This is achieved by analyzing the reward function of these methods, and studying its asymptotic behavior for known and novel states. Our main result shows that the agent receives larger rewards for visiting known states than discovering new ones. The following subsections contain the derivation of this result, and Figure 6 provides a numerical example on a gridworld environment.

A.1. Reverse form of the mutual information

The objective for these methods is

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{10}$$

$$\approx \mathbb{E}_{s,z\sim p(s,z)}[\log \rho_\pi(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{11}$$

where the unknown posterior $p(z|s)$ is approximated by the distribution induced by the policy, $\rho_\pi(z|s)$. This distribution is estimated with a model $q_\theta(z|s)$ trained via maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. For this analysis, however, we will assume access to a perfect estimate of $\rho_\pi(z|s)$. When considering the discovery of N discrete skills under a uniform prior, the reward in Equation 5 becomes

$$r(s,z') = \log \rho_\pi(z'|s) - \log p(z') \tag{12}$$

$$= \log \rho_\pi(z'|s) + \log N \tag{13}$$

where $z' \sim p(z)$. We will assume that $\sum_{i=1}^{N} \rho_\pi(z_i|s) = 1$ in our analysis.

Maximum reward for know states. The reward function encourages policies to discover skills that visit disjoint regions of the state space where $\rho_\pi(z'|s) \to 1$:

$$r_{\text{max}} = \log 1 + \log N = \log N \tag{14}$$

Reward for previously unseen states. Note that $\rho_\pi(z|s)$ is not defined for unseen states, and we will assume a uniform prior over skills in this undefined scenario, $\rho_\pi(z|s) = 1/N, \forall z$:

$$r_{\text{new}} = \log \frac{1}{N} + \log N = 0 \tag{15}$$

Alternatively, one could add a *background* class to the model in order to assign null probability to unseen states (Capdevila et al., 2018). This differs from the setup in previous works, reason why it was considered in the analysis. However, note that the agent gets a larger penalization for visiting new states in this scenario:

$$r'_{\text{new}} = \lim_{\rho_\pi(z'|s)\to 0} \log \rho_\pi(z'|s) + \log N = -\infty \tag{16}$$

These observations explain why the learned skills provide a poor coverage of the state space.

A.2. Forward form of the mutual information

The objective for these methods is

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(s|z)] - \mathbb{E}_{s\sim p(s)}[\log p(s)] \tag{17}$$

$$= \mathbb{E}_{s,z\sim p(s,z)}[\log \rho_\pi(s|z)] - \mathbb{E}_{s\sim \rho_\pi(s)}[\log \rho_\pi(s)] \tag{18}$$

where the unknown distributions $p(s|z)$ and $p(s)$ are approximated using the stationary state-distribution, $p(s|z) \approx \rho_\pi(s|z)$ and $p(s) \approx \rho_\pi(s) = \mathbb{E}_z\left[\rho_\pi(s|z)\right]$. The stationary state-distribution is estimated with a model $q_\theta(s|z)$ trained via maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. For this analysis, however, we will assume access to a perfect estimate of $\rho_\pi(s|z)$. When considering the discovery of N discrete skills, the reward in Equation 8 can be expanded as follows:

$$r(s,z') = \log \rho_\pi(s|z') - \log \frac{1}{N}\sum_{\forall z_i} \rho_\pi(s|z_i) \tag{19}$$

$$= \log \frac{\rho_\pi(s|z')}{\sum_{\forall z_i} \rho_\pi(s|z_i)} + \log N \tag{20}$$

$$= \lim_{\epsilon\to 0} \log \frac{1}{1 + \sum_{\forall z_i \neq z'} \frac{\rho_\pi(s|z_i)+\epsilon}{\rho_\pi(s|z')+\epsilon}} + \log N \tag{21}$$

where $z', z_i \sim p(z)$ and we added $\epsilon \to 0$ in the last step to prevent division by 0.

Maximum reward for know states. As observed by Sharma et al. (2019), this reward function encourages skills to be predictable (i.e. $\rho_\pi(s|z') \to 1$) and diverse (i.e. $\rho_\pi(s|z_i) \to 0, \forall z_i \neq z'$):

$$r_{\text{max}} = \log 1 + \log N = \log N \tag{22}$$

Reward for previously unseen states. In novel states, $\rho_\pi(s|z_i) \to 0, \forall z_i$:

$$r_{\text{max}} = \lim_{\epsilon\to 0} \log \frac{1}{1 + \sum_{\forall z_i \neq z'} \frac{\epsilon}{\epsilon}} + \log N \tag{23}$$

$$= \log \frac{1}{1 + (N-1)} + \log N \tag{24}$$

$$= \log \frac{1}{N} + \log N \tag{25}$$

$$= 0 \tag{26}$$

This result shows that visiting known states instead of exploring unseen ones provides larger rewards to the agent, producing options that provide a poor coverage of the state space.

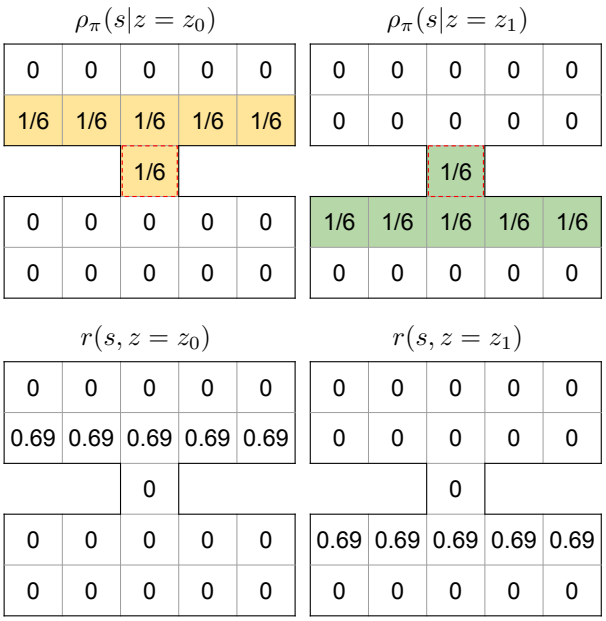

Figure 6. Analysis of the reward landscape on a toy gridworld with two skills, assuming perfect density estimation. Under this assumption, both forms of the mutual information generate the same reward landscape. Each column depicts a different skill, and all rollouts always start from the central tile which is highlighted in red. Skills are rewarded for visiting known states where they are maximally distinguishable, but receive no reward for visiting novel states.

B. Choice of mutual information's form

The main novelty of EDL is an alternative for modelling the unknown distributions, which in principle could work with either form of the mutual information. For the sake of comparison with previous works, all experiments consider discrete skills. This was achieved through a categorical posterior $p(z|s)$ that was approximated with a VQ-VAE (van den Oord et al., 2017). The encoder of the VQ-VAE takes an input x, produces output $z_e(x)$, and maps it to the closest element in the codebook, $e \in \mathbb{R}^{K\times D}$. The posterior categorical distribution $q(z|x)$ probabilities are defined as one-hot as follows:

$$q(z=k|x) = \begin{cases} 1 & \text{for } k = \text{argmin}_j ||z_e(x) - e_j||_2 \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

One could consider the reverse form of the mutual information and train the policy with a reward function as follows:

$$r(s,z) = q(z|s) \quad (28)$$

where we assumed a uniform prior over z and removed the constant $\log p(z)$ term from the reward.

We can foresee two issues with this reward function. It is sparse, i.e. many states provide no reward at all, which might hinder training unless proper exploration strategies are used (Ecoffet et al., 2019; Trott et al., 2019). A similar behavior was observed in existing methods using the reverse form of the mutual information (c.f. Figure 9). Moreover, the fact that many states produce a maximum reward of 1 might lead to unpredictable skills when paired with an entropy bonus. Such unpredictability might not be desirable when training a metra-controller to solve a downstream task by combining the learned skills (Sharma et al., 2019).

C. Implementation Details

Environments. The maze environments are adapted from the open-source implementation[3] by Trott et al. (2019). The agent does not observe the walls, whose location needs to be inferred from experience and makes exploration difficult. The initial state for each episode is sampled from a 1×1 tile. See Table 2 for details about the environments and the topology of each maze.

Parameter	Value
State space	$\mathcal{S} \in \mathbb{R}^2$
Action space	$\mathcal{A} \in [-0.95, 0.95]^2$
Episode length	50
Size: Bottleneck maze (Figure 1)	10×10
Size: Square maze (Figure 2)	5×5
Size: Corridor maze (Figure 3)	1×12
Size: Tree maze (Figure 4)	7×7

Table 2. Environment details.

RL Agents. Policy networks emit the parameters of a Beta distribution (Chou et al., 2017), which are then shifted and scaled to match the task action range. Entropy regularization is employed to prevent convergence to deterministic behaviors early in training. We use a categorical distribution with uniform probabilities for the skill prior $p(z)$. Agents are trained with PPO (Schulman et al., 2017) and the Adam optimizer (Kingma & Ba, 2014). Hyperparameters are tuned for each method independently using a grid search. See Table 3 for details.

Exploration. When relying on State Marginal Matching (SMM) (Lee et al., 2019) for exploration, we implement the version that considers a mixture of policies with a uniform target distribution $p^*(s)$. The density model $q(s)$ is approximated with a VAE. We use states in the replay buffer

[3] https://github.com/salesforce/sibling-rivalry

Hyperparameter	Value
Discount factor	0.99
λ_{GAE}	0.98
λ_{entropy}	$\{0.001, 0.005, 0.01, 0.025\}$
$\epsilon_{\text{SiblingRivalry}}$	$\{2.5, 5.0, 7.5\}$
Optimizer	Adam
Learning rate	$\{0.0003, 0.001\}$
Learning rate schedule	Constant
Advantage normalization	Yes
Input normalization	{Yes, No}
Hidden layers	2
Units per layer	128
Non-linearity	ReLU
Horizon	2500
Batch size	250
Number of epochs	4

Table 3. Hyperparameters used in the experiments. Values between brackets were used in the grid search, and tuned independently for each method.

Hyperparameter	Value
Discount factor	0.99
Target smoothing coefficient	0.005
Target update interval	1
α_{entropy}	$\{0.1, 1, 10\}$
β_{VAE}	$\{0.01, 0.1, 1\}$
Optimizer	Adam
Policy: Learning rate	0.001
SMM discriminator: Learning rate	0.001
VAE: Learning rate	0.01
Learning rate schedule	Constant
Policies in the mixture	4
Input normalization	No
Policy: Hidden layers	2
SMM discriminator: Hidden layers	2
VAE encoder: Hidden layers	2
VAE decoder: Hidden layers	2
Units per layer	128
Non-linearity	ReLU
Gradient steps	1
Batch size	128
Replay buffer size	50k

Table 4. Hyperparameters used for exploration using SMM. Values between brackets were used in the grid search, and tuned independently for each environment. Training ends once the buffer is full.

as a non-parametric approach to sampling from the desired $p(s)$ (Warde-Farley et al., 2019). Sampling states from the replay buffer is similar to a uniform Historical Averaging strategy. This worked well in our experiments, but exponential sampling strategies might be needed in other environments to avoid oversampling states collected by the initially random policies (Hazan et al., 2019). Our implementation follows the open-source code released by the authors[4], which relies on SAC for policy optimization. Hyperparameters are tuned for each environment independently using a grid search. See Table 4 for details.

Skill discovery. The skill discovery stage in the proposed method is done with a VQ-VAE (van den Oord et al., 2017), which allows learning discrete latents. We implement the version that relies on a commitment loss to learn the dictionary. The size of the codebook is set to the number of desired skills. Hyperparameters are tuned for each environment and exploration method independently using a grid search. See Table 5 for details.

D. Figure details

All experiments in the paper consider agents that learn 10 skills. This value was selected to provide a good balance between learning a variety of behaviors and ease of visualization. Given the stochastic nature of the learned policies, we report 20 rollouts per skill. When visualizing states visited by a random policy, we collect 100 rollouts with each (untrained) skill. Trajectories from these skills highly overlap with each other, so we use a single color for all of them to reduce clutter.

E. Additional visualizations

We include visualizations that provide further insight about the results presented in the paper, and that could not be included there due to space constraints. These include the goal states discovered by methods using the forward for of the mutual information (Figure 7), visualization of the reward landscape of each method (Figures 8, 9 and 10), and additional skill interpolations (Figure 11).

[4] https://github.com/RLAgent/state-marginal-matching

Hyperparameter	Value
Code size	16
$\beta_{\text{commitment}}$	$\{0.25, 0.5, 0.75, 1.0, 1.25\}$
Optimizer	Adam
Learning rate	0.0002
Learning rate schedule	Constant
Batch size	256
Number of samples	4096
Input normalization	Yes
Encoder: Hidden layers	2
Decoder: Hidden layers	2
Units per layer	128
Non-linearity	ReLU

Table 5. Hyperparameters used for training the VQ-VAE in the skill discovery stage. Values between brackets were used in the grid search, and tuned independently for each environment and exploration method.

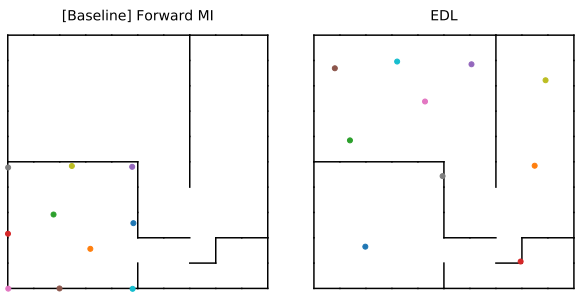

Figure 7. Goal states discovered by methods using the forward form of the mutual information in Figure 1. We define a goal state as the most likely state under $q_\phi(s|z)$ for each skill, i.e. $g_i = \text{argmax}_s q_\phi(s|z_i)$. The baseline method relies on the stationary state-distribution induced by the policy to discover goals. This policy seldom leaves the initial room, limiting the goals that can be discovered. In contrast, the uniform distribution over states in EDL enables the discovery of goals across the whole maze.

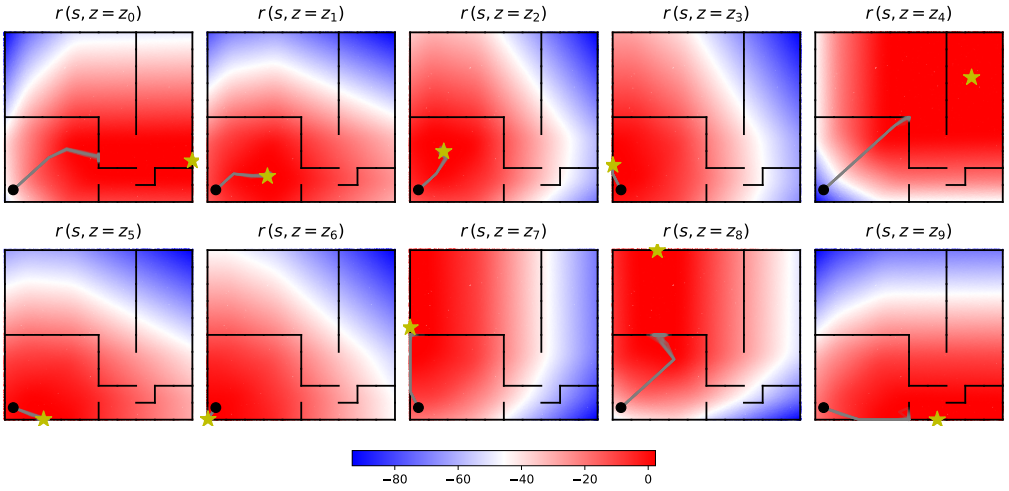

Figure 8. Reward landscape per skill at convergence for the agent in Figure 1 (left). Trajectories from each skill starting from the black dot are plotted in gray. The yellow star indicates the point of maximum reward for each skill. For some skills, this point belongs to an unexplored region of the state space, contrary to the intuition in Section A. Note that this is due to the Gaussian assumption over $p(s|z)$ in the density model.

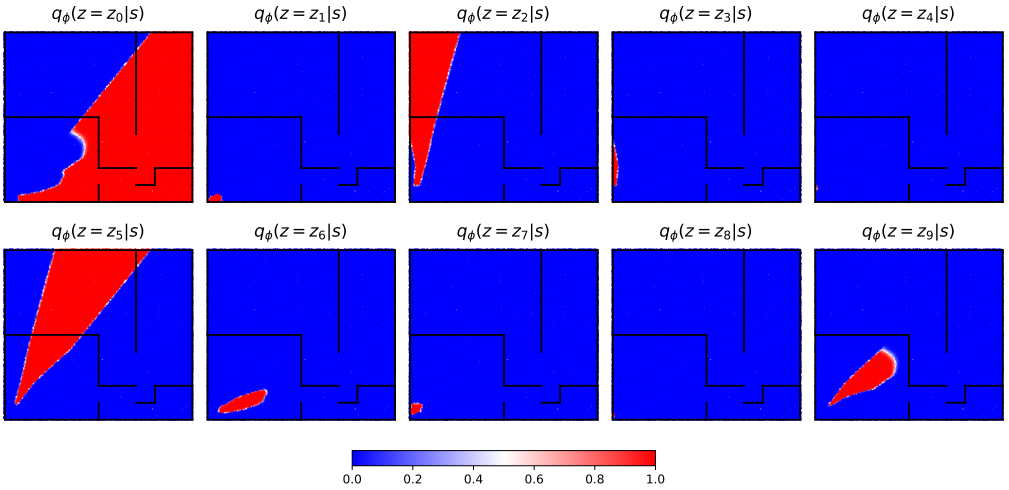

Figure 9. Approximate posterior $q_\phi(z|s)$ at convergence for the agent in Figure 1 (middle). Recall that the reward function for this agent is $r(s, z) = \log q_\phi(z|s) - \log p(z)$, and $\log p(z)$ is constant in our experiments due to the choice of prior over latent variables. The state space is partitioned in disjoint regions, so that skills only need to enter their corresponding region in order to maximize reward. Note how $q_\phi(z|s)$ extrapolates this partition to states that have never been visited by the policy. When combined with an entropy bonus, this reward landscape results in skills that produce highly entropic trajectories within each region.

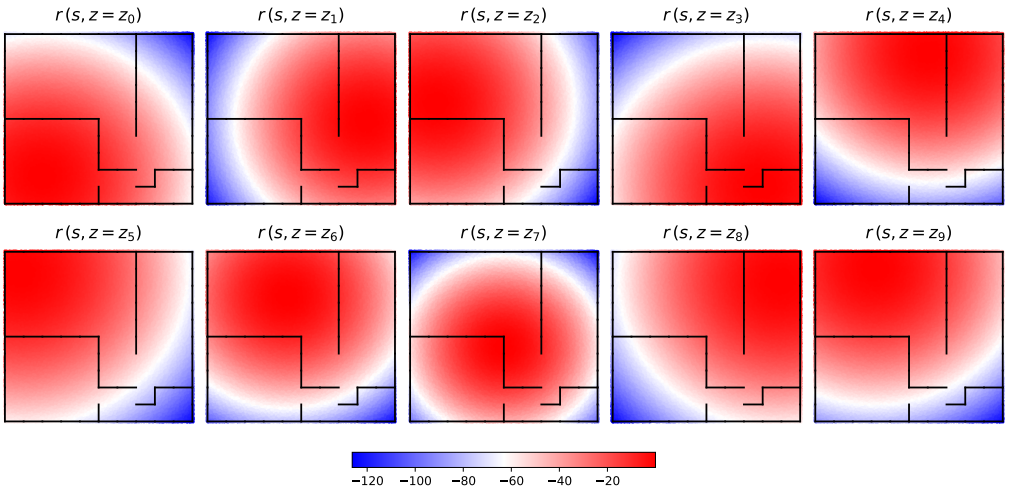

Figure 10. Reward landscape per skill at convergence for the agent in Figure 1 (right). The reward functions follow a bell shape centered at each of the centroids in Figure 7 (right). These are dense signals that ease optimization, but training is prone to falling in local optima due to their deceptive nature.

Figure 11. Interpolating skills learned by EDL. Interpolation is performed at the latent variable level by blending the z vector of two skills. The first row and column show the original skills being interpolated, which were selected randomly from the set of learned options. When plotting interpolated skills, we blend the colors used for the original skills.