Unsupervised Skill Discovery with Bottleneck Option Learning

Jaekyeom Kim [*1] **Seohong Park** [*1] **Gunhee Kim** [1]

Abstract

Having the ability to acquire inherent skills from environments without any external rewards or supervision like humans is an important problem. We propose a novel unsupervised skill discovery method named *Information Bottleneck Option Learning (IBOL)*. On top of the linearization of environments that promotes more various and distant state transitions, IBOL enables the discovery of diverse skills. It provides the abstraction of the skills learned with the information bottleneck framework for the options with improved stability and encouraged disentanglement. We empirically demonstrate that IBOL outperforms multiple state-of-the-art unsupervised skill discovery methods on the information-theoretic evaluations and downstream tasks in MuJoCo environments, including Ant, HalfCheetah, Hopper and D'Kitty. Our code is available at `https://vision.snu.ac.kr/projects/ibol`.

1. Introduction

Deep reinforcement learning (RL) has recently shown great advancement in solving various tasks, from playing video games (Mnih et al., 2013; 2015; Berner et al., 2019) to controlling robot navigation (Kahn et al., 2018). While the standard RL is to maximize rewards from environments as a form of supervision, there has been a surge of interest in unsupervised learning without the assumption of extrinsic rewards (Sukhbaatar et al., 2018; Shyam et al., 2019). Discovering inherent skills in environments without supervision is important and desirable for multiple reasons. First, since it is still challenging to define an effective reward function for practical tasks (Hadfield-Menell et al., 2017; Dulac-Arnold et al., 2019), unsupervised skill discovery helps reduce the burden of it by identifying effective skills for environments. Second, in sparse-reward environments, learned skills can encourage the exploration for encountering rewards, not only by providing useful primitives for the exploration but also by reducing the effective horizon. Third, those skills can be directly used to solve downstream tasks, for example, by employing a meta-controller on top of the discovered skills in a hierarchical manner (Achiam et al., 2018; Eysenbach et al., 2019; Sharma et al., 2020b). Finally, discovered skills could help better understand environments by providing interpretable sets of behaviors.

Unsupervised skill discovery can be formalized with the *options* framework (Sutton et al., 1999), which generalizes primitive actions with the notion of *options*. For ease of learning, options, or synonymously *skills*, are often formulated by introducing a skill latent parameter z to an ordinary policy, resulting in a skill policy with a form of $\pi(a|s, z)$ keeping the same z for multiple steps or the full episode horizon (Gregor et al., 2016; Achiam et al., 2018; Eysenbach et al., 2019; Sharma et al., 2020b). In recent research on the unsupervised skill discovery problem, information-theoretic approaches have been prevalent (Gregor et al., 2016; Achiam et al., 2018; Eysenbach et al., 2019; Sharma et al., 2020b).

In this work, we propose a novel unsupervised skill discovery method named *Information Bottleneck Option Learning (IBOL)*, whose two major novelties over existing approaches are (i) the *linearizer* and (ii) the *information bottleneck*-based skill learning. First, the linearizer is a kind of low-level policy to be suitable for skill discovery by converting a given environment into one with simplified dynamics. It reduces the skill discovery algorithm's burden to learn how to make transitions to diverse states in a given environment without any external rewards, which is not a straightforward job with fairly complex dynamics such as Ant and Humanoid from MuJoCo (Todorov et al., 2012). Once the linearizer is trained, it can be reused for multiple training sessions with different skill discovery approaches. Figure 1 compares the qualitative visualization of the skills learned by different methods in the locomotion (*i.e.* x-y) plane, in Ant. As shown, DIAYN (Eysenbach et al., 2019), VALOR (Achiam et al., 2018) and DADS (Sharma et al., 2020b) with the linearizer (with suffix '-L') learn far more diverse skills than the same methods without the linearizer.

Leveraging the environment simplified with the linearizer,

[*]Equal contribution [1]Department of Computer Science and Engineering, Seoul National University, South Korea. Correspondence to: Gunhee Kim <gunhee@snu.ac.kr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

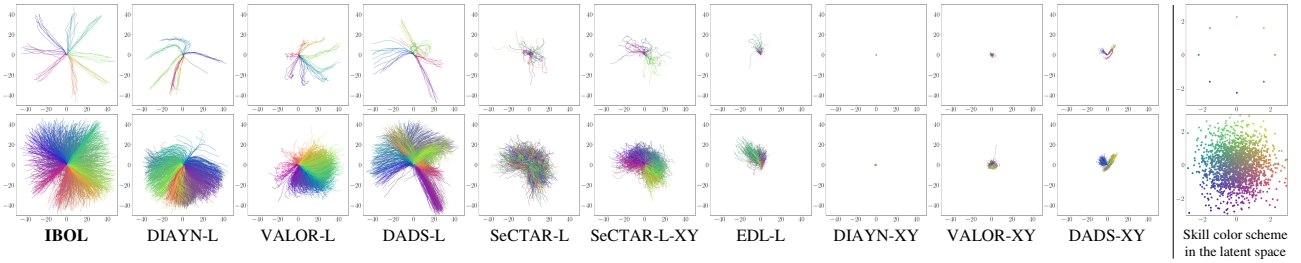


Figure 1. Visualization of the x-y traces of skills discovered by each algorithm in Ant, where the colors represent the two-dimensional skill latents used for the sampling of the skills (see the color scheme on the right). (Top) Trajectories of the six roll-outs from each of the eight different skill latents. (Bottom) Trajectories of 2000 skill latents sampled from the standard normal distribution.

IBOL discovers and learns skills based on the information bottleneck (IB) framework (Tishby et al., 2000; Alemi et al., 2017). Compared to prior approaches, IBOL can introduce some desirable properties to the learned skills. It discovers and learns skills with the skill latent variable Z in a more disentangled way, which makes the learned skills better interpretable with respect to Z. Interpretable models help understand their behaviors and provide intuition about their further uses (Adel et al., 2018). Figure 1 demonstrates that the skill trajectories instantiated by IBOL have a visually simpler and more predictable mapping with the skill latents, which is one of the main requirements for increasing interpretability (Adel et al., 2018). Moreover, the skills learned by IBOL cover the locomotion plane more uniformly and widely. Finally, with the IB-style objective, the skill latent variable Z is learned to be not only informative about the discovered skills but also parsimonious to keep unrelated information about the skills.

Our key contributions can be summarized as follows.

- To the best of our knowledge, our method is the first to separate the problem of making transitions in the state space from skill discovery, simplifying the environment dynamics with independent pre-training, whose learning cost is amortized across multiple skill discovery trainings. It aids skill discovery methods to learn diverse skills by making the environment dynamics as linear as possible.
- We propose a novel skill discovery method with information bottleneck, which provides multiple benefits including learning skills in a more disentangled and interpretable way with respect to skill latents and being robust to nuisance information.
- Our method shows superior performance to various state-of-the-art unsupervised skill discovery methods including DADS (Sharma et al., 2020b), DIAYN (Eysenbach et al., 2019) and VALOR (Achiam et al., 2018) in multiple MuJoCo (Todorov et al., 2012) environments. To verify this, we measure the information-theoretic metrics and the performance on four downstream tasks.

2. Preliminaries and Related Work

We review previous information-theoretic approaches to unsupervised skill discovery and discuss their limitations.

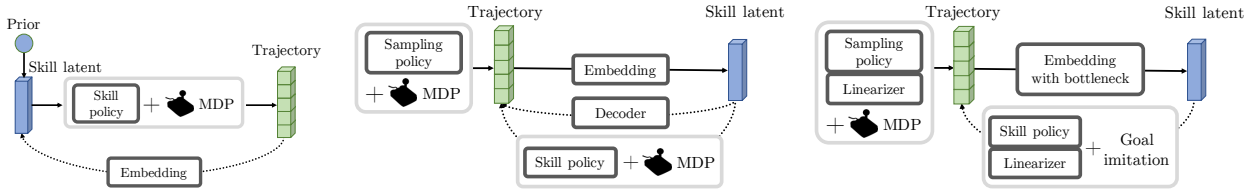
Preliminaries. We consider a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p)$ *without external rewards*. $\mathcal{S}$ and $\mathcal{A}$ respectively denote the state and action spaces, and $p(s_{t+1}|s_t, a_t)$ is the transition function where $s_t, s_{t+1} \in \mathcal{S}$ and $a_t \in \mathcal{A}$. Given a policy $\pi(a_t|s_t)$, a trajectory $\tau = (s_0, a_0, \dots, s_T)$ follows the distribution $\tau \sim p(\tau) = p(s_0) \prod_{t=0}^{T-1} \pi(a_t|s_t) p(s_{t+1}|s_t, a_t)$. Within the options framework (Sutton et al., 1999), we formulate the unsupervised skill discovery problem as learning a latent-conditioned skill policy $\pi(a_t|s_t, z)$ where $z \in \mathcal{Z}$ represents the *skill latent*. We consider continuous skill latents $z \in \mathbb{R}^d$. $h(\cdot)$ and $I(\cdot;\cdot)$ denote differential entropy and mutual information, respectively.

We introduce existing skill discovery methods in two groups: *latent-first* and *trajectory-first* methods.

Latent-first methods. Skill discovery methods in this category, such as VIC (Gregor et al., 2016), DIAYN (Eysenbach et al., 2019), VALOR (Achiam et al., 2018), DADS (Sharma et al., 2020b) and HIDIO (Zhang et al., 2021), first sample a skill latent z and then trajectories conditioned on z, as illustrated in Figure 2a. They aim to increase $I(Z;S)$, the mutual information between the skill latent and state variables. VALOR (Achiam et al., 2018), which incorporates VIC and DIAYN as its special forms (Achiam et al., 2018), optimizes a lower bound of the identity $I(Z;S) = h(Z) - h(Z|S)$. Its objective is to maximize

$$\mathbb{E}_{z\sim p(z)}\left[\mathbb{E}_{\tau\sim p(\tau|z)}[\log p_D(z|s_{0:T})] + \beta \cdot \sum_{t=0}^{T-1} h(A_t)\right],$$

where A_t is the action variable that follows $\pi(a_t|s_t, z)$, β is



(a) Latent-first methods. (b) Trajectory-first methods. (c) IBOL.

Figure 2. Architecture overview of (a) latent-first methods, (b) trajectory-first methods and (c) IBOL.

the entropy coefficient, $p(z)$ is the prior distribution over z, and $p_D(z|s_{0:T})$ is a trainable decoder that reconstructs the original z given $s_{0:T}$. Achiam et al. (2018) show that this objective has an equivalency to β-VAE (Higgins et al., 2017) with the structure of z (input) $\rightarrow \tau$ (latent) $\rightarrow z$ (reconstruction). However, this objective does not take advantage of the benefits that the VAE formulations can provide, such as the theoretical connection to more disentangled and interpretable z (Achille & Soatto, 2018b;a; Chen et al., 2018). DADS (Sharma et al., 2020b) optimizes the other identity $I(Z;S) = h(S) - h(S|Z)$, using a skill dynamics model $q(s_{t+1}|s_t, z)$ that predicts the next state conditioned on z. While the learned dynamics model enables model-based planning, it lacks an explicit mapping from states to skill latents, and thus hardly obtains disentangled skill latents z.

Trajectory-first methods. Another group of methods first samples trajectories and then encodes them into skill latents using the variational autoencoder (VAE) (Kingma & Welling, 2014), as visualized in Figure 2b. This category includes SeCTAR (Co-Reyes et al., 2018), EDL (Campos Camúñez et al., 2020) and OPAL (Ajay et al., 2020). SeCTAR and EDL have separate objectives for their *exploratory policies* to sample diverse trajectories by maximizing $h(p(\tau))$ or $h(S)$. OPAL assumes an offline RL setting where a fixed set of trajectories is priorly given. While these methods employ the VAE with the usual direction of $\tau \rightarrow z \rightarrow \tau$ (EDL has $s \rightarrow z \rightarrow s$) that encourages disentangled representations, they have a limitation that the exploratory policy *only* maximizes the diversity of trajectories. On the contrary, our IBOL method, which also falls into this category, *jointly* maximizes both the diversity and discriminability of trajectories (Section 3.3), which leads to a significant improvement in performance (Section 4).

Finally, all of the prior works learn the skill policies on top of raw environment dynamics. Although dealing with raw dynamics is not highly demanding in simple environments, it could hinder the skill learning in environments with complex dynamics such as Ant and Humanoid from MuJoCo (Todorov et al., 2012). IBOL solves the issue by *linearizing* the environment dynamics ahead of skill discovery so that it can acquire diverse skills by reaching different states more easily in the simplified environment dynamics. Furthermore, we find that the linearization benefits other existing skill discovery methods too (Section 4).

3. Information Bottleneck Option Learning (IBOL)

We decompose the skill discovery problem into two separate phases. Firstly, IBOL trains the *linearizer* that lifts the burden from the skill discovery algorithm to generate diverse states and trajectories under complex environment dynamics (Section 3.1). Secondly, on top of the pre-trained linearizer, IBOL learns to map trajectories into the continuous skill latent space, with the information bottleneck principle (Tishby et al., 2000; Alemi et al., 2017) (Section 3.2). Figure 2c provides the conceptual illustration of IBOL. Algorithm 1 overviews the training of the linearizer in the first phase and Algorithm 2 describes the skill discovery process in the second.

3.1. Linearization of Environments

The *linearizer* π^{lin} is a pre-trained low-level policy that aims to "linearize" the environment dynamics. It takes as input *goals* produced by IBOL's policies for skill discovery (will be discussed in Section 3.2), and translates them into raw actions in the direction of a given goal while interacting with the environment. We define the linearizer $\pi^{\text{lin}}(a_t|s_t, g_t)$ as a goal-conditioned policy (Schaul et al., 2015), which takes both a state $s_t \in \mathcal{S}$ and a goal $g_t \in \mathcal{G}$ as input and outputs a probability distribution over actions $a_t \in \mathcal{A}$. The goal space $\mathcal{G}$ is defined as $\mathcal{G} = [-1, 1]^{dim(\mathcal{S})}$, which has the same dimensionality as the state space (up to 47 in our experiments). Each goal dimension provides a signal for the direction in the corresponding state dimension. We assume that a goal $g_t \in \mathcal{G}$ is given at every ℓ-th time step such that $t \equiv 0 \pmod{\ell}$ (called a *macro* time step), and otherwise kept fixed, *i.e.* $g_t = g_{t-1}$ for $t \not\equiv 0 \pmod{\ell}$.

We sample goals $(g_0, g_\ell, g_{2\ell}, \ldots)$ at the beginning of each roll-out and train the linearizer with a reward function of

$$R^{\text{lin}}(s_t, g_t, a_t, s_{t+1}) = \frac{1}{\ell}(s_{(c+1)\cdot\ell} - s_{c\cdot\ell})^\top g_t, \quad (1)$$

where $c = \left\lfloor \frac{t}{\ell} \right\rfloor$. It corresponds to the inner product of the goal g_t and the state difference between macro time steps:

Algorithm 1 (Phase 1) Training Linearizer

Initialize linearizer π^{lin}.
while *not converged* **do**
  for $i = 1$ **to** n **do**
    Sample goals $(g_0^{(i)}, g_\ell^{(i)}, g_{2\ell}^{(i)}, \dots)$.
    Sample trajectory using π^{lin} and goals.
    Compute linearizer reward R^{lin} using Equation (1).
    Add trajectory to replay buffer.
  end for
  Update π^{lin} using collected samples from replay buffer with SAC (Haarnoja et al., 2018).
end while

Algorithm 2 (Phase 2) Skill Discovery

Load pre-trained linearizer π^{lin}.
Initialize sampling policy π_{θ_s}, trajectory encoder p_ϕ, skill policy π_{θ_z}.
while *not converged* **do**
  for $i = 1$ **to** n **do**
    Sample trajectory using π_{θ_s} on top of π^{lin}.
  end for
  Compute objective from Equation (5).
  Compute its gradient w.r.t. ϕ, θ_z.
  Compute its policy gradient w.r.t. θ_s.
  Jointly update π_{θ_s}, p_ϕ, π_{θ_z} with gradients.
end while

$(s_{(c+1)\cdot\ell} - s_{c\cdot\ell})$. Intuitively, each goal dimension value (ranging from -1 to $+1$) indicates the desired direction and the degree of change in the corresponding state dimension.

The inner product in the reward function has several advantages for skill discovery compared to the Euclidean distance in prior approaches (Nachum et al., 2018; 2019). First, unlike the Euclidean distance that needs to specify the valid range of each state dimension, the inner product only takes care of directions in the state space. Thus, training of the linearizer requires no additional supervision on specifying valid goal spaces or state ranges. Second, by setting some dimensions of a goal to be (near-)zero values, we can ignore changes in the corresponding state dimensions, which is not achievable with the Euclidean distance. This enables IBOL's policies for skill discovery to ignore nuisance dimensions without manually specifying them (Section 3.2).

We find that the linearizer benefits not only IBOL but also other skill discovery methods since it can promote reaching diverse and distant states easier, as shown in Figure 1.

3.2. Skill Discovery with Bottleneck Learning

On top of the pre-trained and fixed linearizer π^{lin}, we learn policies that produce *goals* and acquire a continuous set of skills that is not only distinguishable and diverse but also disentangled and interpretable. The linearizer alone is highly limited to discovery abstractive and informative skills, since it is trained with the inner product reward function and thus optimized for transitioning to distant states rather than the mapping with the latent space. Additionally, IBOL can fix possibly imperfect linearization with the linearizer by combining appropriate high-level goals. In Section 4, we will demonstrate that how such limitations of the linearizer can be resolved by the following skill discovery process.

In contrast to previous skill discovery methods that maximize $I(S; Z)$ (Gregor et al., 2016; Eysenbach et al., 2019; Achiam et al., 2018; Sharma et al., 2020b), IBOL consists of the following three learnable components based on the information bottleneck:

1. The sampling policy $\pi_{\theta_s}(g_t|s_t)$ produces diverse and easily mappable trajectories.
2. The trajectory encoder $p_\phi(z|s_{0:T})$ encodes the state trajectories into the skill latent space.
3. The skill policy $\pi_{\theta_z}(g_t|s_t, z)$ learns to imitate the skills given their latents.

Note that the sampling and skill policies produce goals g_t instead of raw actions a, as they operate on top of the linearizer. We will first start with the sampling policy π_{θ_s} and introduce our IB objective for the trajectory encoder p_ϕ. We then show that it naturally leads to the emergence of the skill policy π_{θ_z} as a variational approximation to the sampling policy π_{θ_s}.

IBOL's objective. Assuming trajectories generated by the sampling policy, $\{\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(n)}\}$, our objective is to embed the state trajectories $\{s_{0:T}^{(1)}, \dots, s_{0:T}^{(n)}\}$ into the skill latent space $\mathcal{Z}$. We encode the *state* trajectory $s_{0:T}$, not the *whole* trajectory τ, because an outside observer can only see the agent's state, not its underlying actions or goals. However, the encoded skill latent z should contain sufficient information about the underlying goals so that the whole trajectory is reproducible from z. Furthermore, since raw states often contain nuisance information not pertaining to skill discovery, z is encouraged to minimally contain unnecessary or noisy information in the states irrelevant to the goals. This leads to the Information Bottleneck objective (Tishby et al., 2000; Alemi et al., 2017) over the structure of $S_{0:T}$ (input) $\rightarrow Z$ (latent) $\rightarrow G_{0:T-1}$ (target).

Formally, let us first define the *sampling policy* parameterized by θ_s as $\pi_{\theta_s}(g_t|s_t)\colon \mathcal{S} \rightarrow \mathcal{P}(\mathcal{G})$, which maps a

state to a probability distribution over goals. A trajectory $\tau = (s_0, g_0, s_1, \ldots, g_{T-1}, s_T)$ obtained by the sampling policy is acquired from the distribution $\tau \sim p_{\theta_s}(\tau) = p(s_0)\prod_{t=0}^{T-1}\pi_{\theta_s}(g_t|s_t)p(s_{t+1}|s_t, g_t)$. Under the distribution $p_{\theta_s}(\tau)$, let S_t be a random variable corresponding to s_t and G_t be a random variable for g_t. We define the *trajectory encoder* parameterized by ϕ as $p_\phi(z|s_{0:T})\colon \mathcal{S}^{T+1} \to \mathcal{P}(\mathcal{Z})$ that maps a state trajectory to a probability distribution over skill latents z in the skill space $\mathcal{Z}$. Let Z be a random variable for z.

We formulate our IB objective as follows. First, given S_t, the skill latent Z should be informative about the goal G_t that the sampling policy has produced, which leads to the *prediction term* $I(Z; G_t|S_t)$. Second, Z should be penalized for preserving information about the state trajectory but unrelated to the goals, which corresponds to the *compression term* $I(Z; S_{0:T})$. Summing these up, we obtain the following objective:

$$\text{maximize } \mathbb{E}_t[I(Z; G_t|S_t) - \beta \cdot I(Z; S_{0:T})], \quad (2)$$

where $\mathbb{E}_t$ is the expectation over $\{0, 1, \ldots, T-1\}$, and β is a constant that controls the weight of the compression term.

Lower bound optimization. Since the objective is practically intractable, we derive its lower bound (Alemi et al., 2017) as follows (see Appendix B for the full derivation):

$$\begin{aligned}
&\mathbb{E}_t[I(Z; G_t|S_t) - \beta \cdot I(Z; S_{0:T})] \quad (3)\\
&= \mathbb{E}_{\substack{\tau\sim p_{\theta_s}(\tau),t,\\ z\sim p_\phi(z|s_{0:T})}}\left[\log \frac{p_{\theta_s}(g_t|s_t, z)}{\pi_{\theta_s}(g_t|s_t)} - \beta \log \frac{p_\phi(z|s_{0:T})}{p_\phi(z)}\right]\\
&\geq \mathbb{E}_{\tau\sim p_{\theta_s}(\tau)}\Big[\mathbb{E}_{z\sim p_\phi(z|s_{0:T}),t}\Big[\log \pi_{\theta_z}(g_t|s_t, z) \quad (4)\\
&\qquad - \log \pi_{\theta_s}(g_t|s_t)\Big] - \beta \cdot D_{\mathrm{KL}}(p_\phi(Z|s_{0:T})\|r(Z))\Big],
\end{aligned}$$

where D_{KL} denotes the Kullback-Leibler (KL) divergence. Here we use two variational approximations: the *skill policy*'s output distribution $\pi_{\theta_z}(g_t|s_t, z)$ is a variational approximation of $p_{\theta_s}(g_t|s_t, z)$ and $r(z)$ is that of the marginal distribution $p_\phi(z)$. In Equation (4), the first term $\log \pi_{\theta_z}(g_t|s_t, z)$ makes the skill policy $\pi_{\theta_z}(g_t|s_t, z)$ imitate the sampling policy's output given the skill latent z; thus we call this the *imitation term*. The third term $-\beta D_{\mathrm{KL}}(p_\phi(Z|s_{0:T})\|r(Z))$ is the *compression term* that forces the output distributions of the trajectory encoder to be close to $r(z)$. We will revisit the second term $-\log \pi_{\theta_s}(g_t|s_t)$ later.

We fix $r(z)$ to $\mathcal{N}(0, I)$ as in Alemi et al. (2017) for the following reasons. First, it enables us to analytically compute the KL divergence. Second, more importantly, it induces disentanglement between the dimensions of z (Achille & Soatto, 2018b;a; Chen et al., 2018). Disentangled representations lead to more interpretable skills with respect to their skill latents z. In Appendix C, we provide further details on how the compression term encourages the disentanglement of skill latent dimensions.

It is worth noting that the first and third terms in Equation (4) are related to the β-VAE objective (Kingma & Welling, 2014; Higgins et al., 2017; Alemi et al., 2017) and previous skill discovery methods that use trajectory VAEs (Co-Reyes et al., 2018; Ajay et al., 2020). The first and the third term correspond to the reconstruction loss and the KL divergence loss in β-VAEs, respectively. One important difference is that we reconstruct not the original state trajectories but their underlying goals. It eliminates the need for state decoders or sampling with the skill policy during training.

3.3. Training

The trajectory encoder and the skill policy can be trained using the reparameterization trick as in VAEs (Kingma & Welling, 2014); we optimize those two terms in Equation (4) with respect to their parameters, θ_z and ϕ. Note that the skill policy does not interact with the environment during training and the second term $-\log \pi_{\theta_s}(g_t|s_t)$ is independent of these parameters.

The sampling policy $\pi_{\theta_s}(g_t|s_t)$ can be trained with the same objective of Equation (4). This is the key difference with prior trajectory-first methods that employ similar VAE architectures (Campos Camúñez et al., 2020; Co-Reyes et al., 2018; Ajay et al., 2020) (Section 2). They either have a separate objective for training their sampling policies (Campos Camúñez et al., 2020; Co-Reyes et al., 2018) or assume the offline RL setting (Ajay et al., 2020). In contrast, we jointly train all the components with the same objective.

There are several merits of using the same objective. First, the second term $-\log \pi_{\theta_s}(g_t|s_t)$, referred to as the *entropy term*, encourages the sampling policy to produce diverse trajectories. In deterministic environments, maximizing this term is equivalent to maximizing the entropy of whole trajectories, as $h(p_{\theta_s}(\tau)) = T \cdot \mathbb{E}_{\tau\sim p_{\theta_s}(\tau),t}[-\log \pi_{\theta_s}(g_t|s_t)] + (\text{const})$. Note that this entropy term often remains constant in IB literature (Alemi et al., 2017), assuming that the training data (*e.g.* images) are given, whereas we can diversify the "training data" too. Second, optimizing the whole Equation (4) makes the sampling policy generate trajectories that are not only diverse but also *easily encoded into the skill space for the trajectory encoder and skill policy* thanks to the first and third terms, which helps the learning of the two components as well. This is not achievable when the sampling policy is trained with a diversity maximizing objective only. In Section 4, we will demonstrate how taking into account both diversity and encodability leads to a huge difference in performance, comparing with baselines without such consideration.

Practical training. Since the expectation in Equation (4) involves the sampling policy's roll-outs in the environment, we optimize the sampling policy via the policy gradient method. However, there exists one practical difficulty when training IBOL. Since the sampling policy $\pi_{\theta_s}(g_t|s_t)$ lacks a variable about the context (*e.g.* z) compared to the skill policy $\pi_{\theta_z}(g_t|s_t, z)$, π_{θ_s} is less expressive than π_{θ_z}, which could end up with a suboptimal convergence. To solve this issue, we introduce a new context parameter $u \in \mathcal{U}$ with its prior $p(u)$ to the sampling policy, redefining it as $\pi_{\theta_s}(g_t|s_t, u) : \mathcal{S} \times \mathcal{U} \to \mathcal{P}(\mathcal{G})$. The new parameter u for π_{θ_s} plays a similar role to the skill latent z for π_{θ_z}. We also fix $p(u) = \mathcal{N}(0, I)$ as in $r(z)$. To obtain roll-outs from the sampling policy, we first sample u from its prior, and then keep sampling goals with the fixed u.

Given that $r(z)$ and $p(u)$ are identical, we additionally include an auxiliary term $\mathbb{E}_{u\sim p(u),\tau\sim p_{\theta_s}(\tau|u)}[\lambda \cdot p_\phi(u|s_{0:T})]$ to further stabilize the training. This term guides the output of the trajectory encoder p_ϕ to u, which is from $p(u) = r(z)$, operating compatibly with the compression term.

Finally, with the revised sampling policy, we approximate the second term in Equation (4) as done in DADS (Sharma et al., 2020b): $\pi_{\theta_s}(g_t|s_t) = \int_u \pi_{\theta_s}(g_t|s_t, u)p(u|s_t)du \approx \int_u \pi_{\theta_s}(g_t|s_t, u)p(u)du \approx \frac{1}{L}\sum_{i=1}^{L} \pi_{\theta_s}(g_t|s_t, u_i)$ for $u_i \overset{\text{i.i.d.}}{\sim} p(u)$, where L is the number of samples from the prior. Therefore, the final objective of our method is

$$\mathbb{E}_{\substack{u\sim p(u),\\ \tau\sim p_{\theta_s}(\tau|u)}} \left[\mathbb{E}_{\substack{z\sim p_\phi(z|s_{0:T}),t \\ u_i \overset{\text{i.i.d.}}{\sim} p(u)}} \left[J^{\mathrm{P}} \right] - \beta \cdot J^{\mathrm{C}} + \lambda \cdot p_\phi(u|s_{0:T}) \right]$$

$$\text{where } J^{\mathrm{P}} = \log \pi_{\theta_z}(g_t|s_t, z) - \log \left(\frac{1}{L} \sum_{i=1}^{L} \pi_{\theta_s}(g_t|s_t, u_i) \right)$$

$$J^{\mathrm{C}} = D_{\mathrm{KL}}(p_\phi(Z|s_{0:T}) \| r(Z)). \tag{5}$$

4. Experiments

We compare our IBOL with other state-of-the-art methods in multiple aspects. First, we visualize the learned skills with the trajectory plots and the rendered scenes from environments (Section 4.1). Second, we quantitatively evaluate the skill discovery methods in terms of multiple information-theoretic metrics (Section 4.2). Third, we evaluate the trained policies on the downstream tasks with different configurations (Section 4.3). Finally, we present additional behaviors of IBOL in the absence of the locomotion signals and with the distorted goal space (Section 4.4). Please refer to Appendix for additional results.

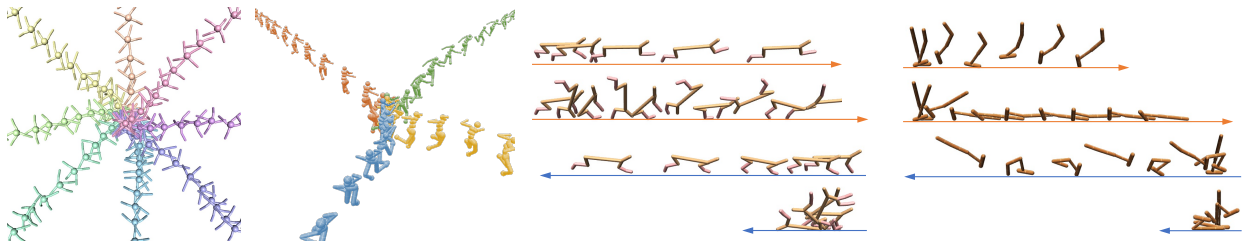
Experiment setup and baselines. We experiment with MuJoCo environments (Todorov et al., 2012) for multiple tasks: Ant, HalfCheetah, Hopper and Humanoid from OpenAI Gym (Brockman et al., 2016) with the setups by Sharma et al. (2020b) and D'Kitty from ROBEL (Ahn et al., 2020) adopting the configurations by Sharma et al. (2020a). We use D'Kitty with the random dynamics setting; in each episode, multiple properties of the environment, such as its joint dynamics, friction and height field, are randomized, which provides an additional challenge to agents. We mainly compare our method with recent information-theoretic unsupervised skill discovery methods, VALOR (Achiam et al., 2018), DIAYN (Eysenbach et al., 2019) and DADS (Sharma et al., 2020b). Since IBOL operates on top of the linearized environments, we also consider a variant of each algorithm that uses the same linearizer, denoted with the suffix '-L' (*e.g.* VALOR-L). In Ant experiments, we use the suffix '-XY' to refer to the methods with the x-y *prior* (Sharma et al., 2020b), which forces them to focus exclusively on the locomotion skills by restricting the observation space of the trajectory encoder (or the skill dynamics model in DADS) to the x-y coordinates.

Implementation. For experiments, we use pre-trained linearizers with two different random seeds on each environment. When training the linearizers, we sample a goal g at the beginning of each roll-out and fix it within that episode to learn consistent behaviors, as in SNN4HRL (Florensa et al., 2016). We consider continuous priors for skill discovery methods. Especially, we use the standard normal distribution for $p(u)$ and $r(z)$ in IBOL and for $p(z)$ in other methods. Further details are described in Appendix I.

4.1. Visualization of Learned Skills

Figure 3 shows that IBOL, with no extrinsic rewards, discovers diverse locomotion skills for Ant and Humanoid and multiple skills with various speeds and poses in both directions for HalfCheetah and Hopper. We present the discovery of orientation primitives for Ant in Section 4.4 and additional results including the videos of the discovered skills at `https://vision.snu.ac.kr/projects/ibol`.

Figure 1 demonstrates that while all the algorithms mainly discover locomotion skills, IBOL discovers visually less entangled primitives with the most diverse directions compared to the latent-first and trajectory-first baselines. We train IBOL, DIAYN-L, VALOR-L, DADS-L, SeCTAR-L, SeCTAR-L-XY and EDL-L on Ant with the skill latent variables of $d = 2$, where SeCTAR-L-XY is equipped with the x-y *prior* (Sharma et al., 2020b). We qualitatively examine their trajectories in the x-y plane; since the x-y dimensions are interpretable and have a large range of values, they can illustrate the characteristic differences between skill discovery algorithms well. We also train DIAYN-XY, VALOR-XY and DADS-XY to enforce them to discover skills on the x-y plane without the linearizer. We observe that the linearizer significantly improves not only the diversity of trajectories but also the correspondence between skill latents and trajec-



(a) Ant (b) Humanoid (c) HalfCheetah (d) Hopper

Figure 3. Examples of rendered scenes illustrating the skills that IBOL discovers with no rewards in MuJoCo environments. (a) Ant moving in various directions. (b) Humanoid running in different directions. (c) (Top to Bottom) HalfCheetah running forward, rolling forward, running backward and flipping backward. (d) (Top to Bottom) Hopper hopping forward, crawling forward, jumping backward and flipping backward.

tories by reducing the burden of making transitions in the x-y dimensions.

4.2. Information-Theoretic Evaluations

We present the metrics that evaluate the unsupervised skill discovery methods without the need for external tasks. While the quantities between skill latents Z and state sequences $S_{0:T}$ generated with π_{θ_z} are attractive, the high dimensionality of $S_{0:T}$ makes it a less viable choice. One workaround is to examine only the last states S_T instead of the whole sequences, as S_T still characterizes skills in environments to some degree. That is, we can simply estimate $I(Z;S_T)$ instead of $I(Z;S_{0:T})$ to measure how informative Z is. This can also be viewed as follows: in $I(Z;S_{0:T}) = I(Z;S_T) + \sum_{i=0}^{T-1} I(Z;S_i|S_{i+1:T})$, only the first term $I(Z;S_T)$ is taken into account, as $I(Z;S_i|S_{i+1:T}) = h(Z|S_{i+1:T}) - h(Z|S_{i:T})$ and adding S_i to $S_{i+1:T}$ to the condition would change only little entropy of Z.

We also consider metrics for measuring the disentanglement of Z. We find Do & Tran (2020) provide a helpful viewpoint to our evaluation. They suggest that the concept of disentanglement has three considerations: *informativeness*, *separability* and *interpretability*. Informativeness denotes how much information each latent dimension contains about the data, and separability is a concept about *no* information sharing between two latent dimensions on the data. Interpretability considers the alignment between the ground-truth and learned factors. Among them, we do not employ the interpretability measure because the lack of supervision in unsupervised skill discovery prevents achieving a high value (Locatello et al., 2019). For example, if data points are uniformly distributed in a two-dimensional circle, there can be infinite equally good ways to disentangle the data into two axes. To measure informativeness and separability, we use the SEPIN@k and WSEPIN metrics (Do & Tran, 2020) evaluated for skill latents and the last states (detailed in Appendix E).

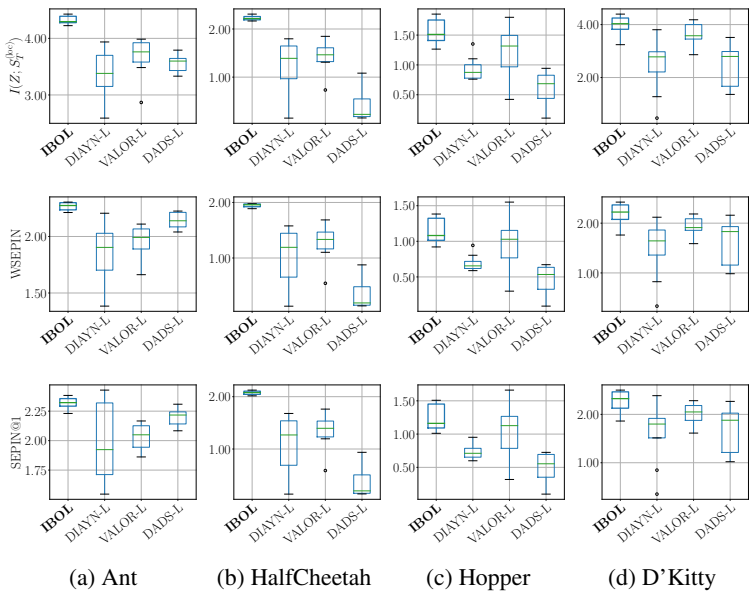
We compare the skill policies trained by IBOL, DIAYN-L, VALOR-L and DADS-L with $d = 2$. We use the three evaluation metrics, $I(Z;S_T^{(\text{loc})})$, SEPIN@1 and WSEPIN on Ant, HalfCheetah, Hopper and D'Kitty, keeping only the state dimensions for the agent's locomotion (*i.e.* x-y coordinates for Ant and D'Kitty and x for the rest) denoted as (loc). One rationale behind it is that the algorithms on the linearized environments successfully discover the locomotion skills (*e.g.* Figure 1). The locomotion coordinates are also suitable for assessing skill *discovery*, since these values can vary in large ranges.

Figure 4 shows the box plots of the results. With the same linearizers, IBOL outperforms the three baselines, DIAYN-L, VALOR-L and DADS-L, in all three information-theoretic evaluation metrics on Ant, HalfCheetah, Hopper and D'Kitty. The plots for $I(Z;S_T^{(\text{loc})})$ show that IBOL can stably discover diverse skills from the environments conditioned on the skill latent parameter Z. Also, the results with WSEPIN and SEPIN@1 suggest that IBOL outperforms the baselines, with regard to both informativeness and separability of Z's individual dimensions. Overall, IBOL shows the lower average deviation compared to the other methods, which demonstrates its stability in learning. For additional analysis and details, please refer to Appendix.

4.3. Evaluation on Downstream Tasks

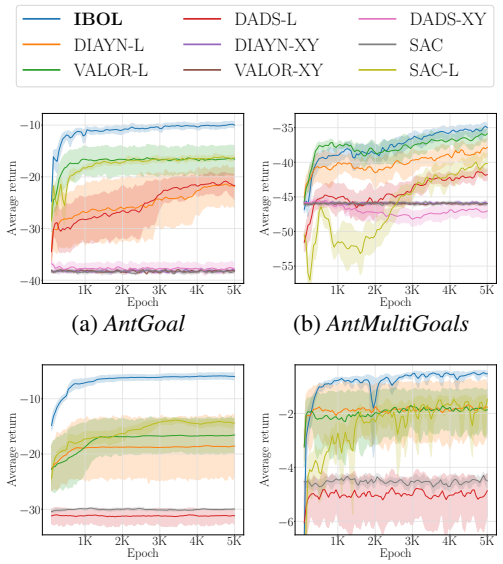
We demonstrate the effectiveness of the abstraction learned by IBOL on downstream tasks. In Ant, we modify the environment to obtain two tasks, *AntGoal* and *AntMultiGoals*, inspired by Eysenbach et al. (2019); Sharma et al. (2020b). In HalfCheetah, we test the methods on two tasks, *CheetahGoal* and *CheetahImitation*.

AntGoal is a task for evaluating how capable the agent is in reaching diverse goals. For every new episode, a goal $w = [w^{(x)}, w^{(y)}]$ is randomly sampled in the x-y plane. The



(a) Ant (b) HalfCheetah (c) Hopper (d) D'Kitty

Figure 4. Comparison of IBOL (ours) with the baseline methods, DIAYN-L, VALOR-L and DADS-L, in the evaluation metrics of $I(Z; S_T^{(\text{loc})})$, WSEPIN and SEPIN@1, on Ant, HalfCheetah, Hopper and D'Kitty. For each method, we use the eight trained skill policies.



(a) *AntGoal* (b) *AntMultiGoals*

(c) *CheetahGoal* (d) *CheetahImitation*

Figure 5. Comparison of IBOL (ours) with the baseline methods on the four downstream tasks. Each line is the mean return over the last 100 epochs at each time step, averaged over eight runs. The shaded areas denote the 95% confidence interval.

agent can observe the goal w at every step, and receives a reward of $\big(-\|w - [s_T^{(x)}, s_T^{(y)}]\|_2\big)$ where $[s_T^{(x)}, s_T^{(y)}]$ is the agent's final position, when each episode ends.

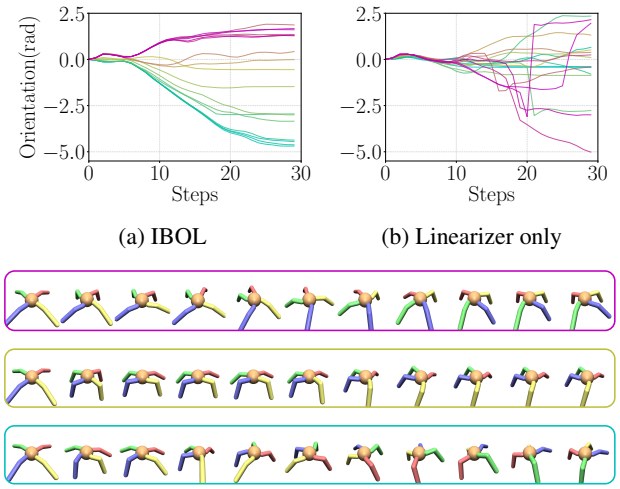
AntMultiGoals is a repeated version of *AntGoal*. At time step $t \equiv 0 \pmod{\eta}$ in each episode, a new goal $w = [w^{(x)}, w^{(y)}]$ is sampled based on the agent's current position, $[s_t^{(x)}, s_t^{(y)}]$, and is held for the next η steps. Similarly to *AntGoal*, at the end of each η-sized chunk (before sampling of a new goal), the agent gains a reward of $\big(-\|w - [s_t^{(x)}, s_t^{(y)}]\|_2\big)$. We set $\eta = 50$.

CheetahGoal is a task similar to *AntGoal* but in HalfCheetah. For each episode, a goal $w^{(x)}$ in the x axis is sampled and observed by the agent at every step. At the end of the episode, the agent receives a reward of $\big(-|w^{(x)} - s_T^{(x)}|\big)$ where $s_T^{(x)}$ is the final position of the agent.

We also experiment with a different type of task, *CheetahImitation*. Each of the skill policies learned by the four skill discovery methods is used to sample 1000 random skill trajectories, all of whose x traces are gathered to form a set of imitation targets. For a new episode of *CheetahImitation*, one imitation target $w = [w_1^{(x)}, \dots, w_T^{(x)}]$, a sequence of T positions in the x axis, is randomly sampled from the set. The goal of this task is to imitate the target w in the x axis; at the t-th step, a reward of $\big(-(w_t^{(x)} - s_t^{(x)})^2\big)$ is given, where the agent perceives the target w as part of its observation. *CheetahImitation* can evaluate the diversity and coverage of skill policies.

For comparison, we employ a meta-controller on top of each skill policy learned by skill discovery methods. The meta-controller iterates observing a state from the environment and picking a *skill* with its own meta-policy, which invokes the pre-trained skill policy with the same skill latent value z for ℓ_m time steps. We employ Soft Actor-Critic (SAC) (Haarnoja et al., 2018) to train the meta-controller, and also compare a pure SAC agent as an additional baseline method.

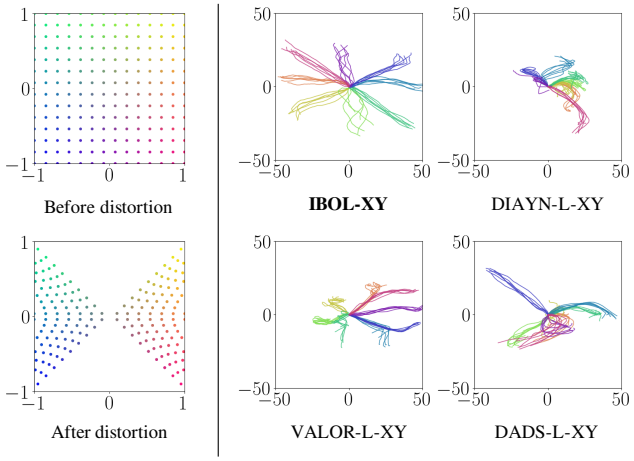
Figure 5 compares the performance of IBOL with the baseline methods on the four tasks: *AntGoal*, *AntMultiGoals*, *CheetahGoal* and *CheetahImitation*. We set $\ell_m = 5$ for *AntMultiGoals* and $\ell_m = 20$ for the others. Figures 5a and 5b suggest that the abstraction by IBOL is more effective for the meta-controller to learn to reach a goal from the initial state, in comparison to the baselines. They confirm that the linearizer greatly helps different skill policies' learning of locomotion in Ant. Figure 5c shows that IBOL provides better abstraction to the meta-controller for reaching goals in HalfCheetah. Also, Figure 5d demonstrates that IBOL's skills can be used to imitate skills not only from itself but also from the other baselines. It supports the improved diversity of skills learned by IBOL. Overall, IBOL presents significantly smaller variances than the other baselines.



(a) IBOL (b) Linearizer only

(c) Rendered scenes

Figure 6. Orientation trajectories from (a) the *skill policy* of IBOL and (b) the *linearizer*. The skill latent value is interpolated from -4 (cyan) to 4 (magenta) for IBOL, while the orientation dimension value of the goal is interpolated from -1 (cyan) to 1 (magenta) for the linearizer (since it is trained with the goal range of $[-1, 1]$). (c) Rendered scenes of IBOL's trajectories from (a).



(a) Distortion scheme (b) Visualization of x-y traces

Figure 7. (a) Distortion scheme of the linearizer. It distorts the x and y dimensions of goals, and produces the corresponding actions for the modified goals. (b) Visualization of the x-y traces of the skills discovered by each algorithm using the distorted linearizer. The same skill latents are used with the top row of Figure 1.

4.4. Additional Observations

We present more experiments on Ant to confirm that IBOL can pick appropriate goals at different states for the linearizer in order to learn skills with high distinguishability.

Learning non-locomotion skills. In the absence of locomotion signals, IBOL can learn orientation primitives, which is not easy unless the skill discovery algorithm produces diverse goals for the linearizer. Figure 6a shows examples of orientation skills by IBOL on Ant with $d = 1$. Figure 6b depicts that using the linearizer alone fails to produce comparable results, while IBOL utilizes various goal dimensions of the linearizer to obtain an interpolable skill set.

Overcoming goal space distortion. We conduct additional experiments to validate IBOL's capability of discovering more discriminable trajectories even under harsh conditions. We distort the linearizer's goal space as Figure 7a, so that reaching vertically distant states becomes more demanding. We train IBOL-XY, DIAYN-L-XY, VALOR-L-XY and DADS-L-XY with $d = 2$ on top of the modified linearizer. Figure 7b suggests that IBOL discovers locomotion skills in various angles including vertical directions in the most visually disentangled manner.

5. Conclusion

We presented Information Bottleneck Option Learning (IBOL) as a novel unsupervised skill discovery method. It first deals with the environment dynamics using the linearizer trained to transition in various directions in the state space. It then discovers skills taking advantage of the information bottleneck framework, which learns the skill latent parameter (or the parameter of the skill policy) as the learned representations of the skills. Our quantitative evaluation showed that the skill latent learned by IBOL provides improved abstraction measured as the disentanglement. We also confirmed that IBOL outperforms other skill discovery methods with notably lower variances and the linearizer benefits both IBOL and other baselines on downstream tasks.

One future challenge may be to deal with environments whose state space is very high dimensional such as vision environments, since goal directions of the linearizer in such domains might not operate well as feasible signals. A possible solution could be adopting state representation learning techniques for RL such as Nachum et al. (2019).

Acknowledgements

We thank the anonymous reviewers for the helpful comments. This work was supported by Samsung Advanced Institute of Technology, the ICT R&D program of MSIT/IITP (No. 2019-0-01309, Development of AI technology for guidance of a mobile robot to its goal with uncertain maps in indoor/outdoor environments) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01082, SW StarLab). Jaekyeom Kim was supported by Hyundai Motor Chung Mong-Koo Foundation. Gunhee Kim is the corresponding author.

References

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

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018a.

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2897–2905, 2018b.

Adel, T., Ghahramani, Z., and Weller, A. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pp. 50–59. PMLR, 2018.

Ahn, M., Zhu, H., Hartikainen, K., Ponte, H., Gupta, A., Levine, S., and Kumar, V. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on Robot Learning*, pp. 1300–1313. PMLR, 2020.

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *ArXiv*, abs/2010.13611, 2020.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *ArXiv*, abs/1606.01540, 2016.

Campos Camúñez, V., Trott, A., Xiong, C., Socher, R., Giró Nieto, X., and Torres Viñals, J. Explore, discover and learn: unsupervised discovery of state-covering skills. In *ICML 2020, Thirty-seventh International Conference on Machine Learning:[posters]*, pp. 1–17, 2020.

Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in neural information processing systems*, pp. 2610–2620, 2018.

Co-Reyes, J. D., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In *ICML*, 2018.

Do, K. and Tran, T. Theory and evaluation metrics for learning disentangled representations. In *International Conference on Learning Representations*, 2020.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

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

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. In *ICLR*, 2016.

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

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S., and Dragan, A. Inverse reward design. In *Advances in Neural Information Processing Systems*, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Kahn, G., Villaflor, A., Ding, B., Abbeel, P., and Levine, S. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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*, 518(7540): 529–533, 2015.

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

Nachum, O., Gu, S., Lee, H., and Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2019.

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

Sharma, A., Ahn, M., Levine, S., Kumar, V., Hausman, K., and Gu, S. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2020a.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020b.

Shyam, P., Jaśkowski, W., and Gomez, F. Model-based active exploration. In *International Conference on Machine Learning*, pp. 5779–5788. PMLR, 2019.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. In *International Conference on Learning Representations*, 2018.

Sutton, R., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112:181–211, 1999.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *ArXiv*, physics/0004057, 2000.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Zhang, J., Yu, H., and Xu, W. Hierarchical reinforcement learning by discovering intrinsic options. *ArXiv*, abs/2101.06521, 2021.