Off-Policy Empowerment Bandits: Learning Large Abstract Skillsets without a Simulator

Anonymous Author(s)
Affiliation
Address
email

Abstract

General purpose agents will need large skillsets that can be executed in stochastic environments. Empowerment is a compelling objective for learning diverse skillsets in stochastic settings because empowerment enables agents to learn abstract skills that target groupings of states. A key problem with using empowerment for building skillsets is that it can require executing a prohibitively large and highly redundant set of skills in the environment, which is why prior empowerment algorithms have needed a simulator of the environment in order to work. To make empowerment a more practical objective for skill learning, we introduce Off-Policy Empowerment Bandits, an algorithm that can optimize empowerment using a replay buffer of past agent interactions with the environment. We test our approach in a variety of highly stochastic domains, including ones with high-dimensional observations, and our approach is able to learn skillsets with levels of empowerment equivalent to that achieved by the leading empowerment skill-learning algorithm which requires a simulator of the environment.

1 Introduction

General purpose agents that operate in the real world will need to be able to execute a diverse set of skills in a highly stochastic world. For instance, a household robot of the future will need to be able to perform the large number of skills involved in cooking and cleaning while the nearby human members of the household are continually moving in unexpected ways and having random conversations with each other and the robot. Even when humans are not nearby, the simple act of the robot looking in different directions will be highly stochastic as objects that are relevant to the robot will be observed in unexpected places. For example, a robot trying to clean dishes may look down into the sink and see a sponge, when the sponge is often located on the kitchen counter.

A major problem in AI research is that it unclear whether the dominant paradigms in unsupervised skill learning are capable of learning large skillsets in environments with realistic levels of randomness. The dominant approach to unsupervised skill learning, unsupervised Goal-Conditioned RL (GCRL), which trains agents to learn skills that target particular regions of the spate space, has repeatedly demonstrated that it can learn diverse skillsets in deterministic settings where specific regions of the state space can be consistently achieved (Ecoffet *et al.*, 2019; Mendonca *et al.*, 2021; Nair *et al.*, 2018; Pong *et al.*, 2019; Campos *et al.*, 2020; Pitis *et al.*, 2020; Held *et al.*, 2017; Kim *et al.*, 2023). However, recent work has shown that when applied to stochastic settings where specific regions of the state space cannot be consistently achieved, GCRL struggles to learn larger skillsets (Anonymous *et al.*, 2024).

Empowerment (Klyubin *et al.*, 2005; Salge *et al.*, 2013; Jung *et al.*, 2012; Mohamed & Rezende, 2015; Gregor *et al.*, 2016; Eysenbach *et al.*, 2018), another approach to learning large skillsets, does offer some mathematical advantages when it comes to skill learning in stochastic domains but is still

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

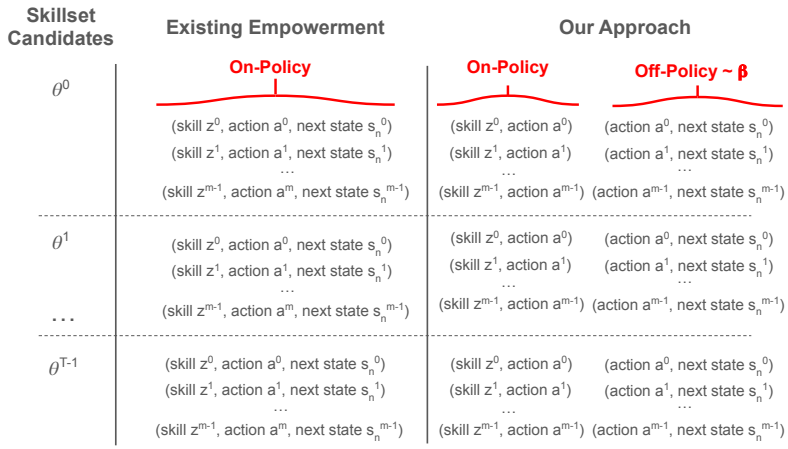

Figure 1: **Approach Overview.** Existing empowerment approaches can require a prohibitive amount of environment exploration because computing the mutual information of a single skillset θ can require executing many skills in the environment in order to obtain the needed (skill z, action a, next state s_n) tuples, and empowerment methods need to evaluate many skillsets. Our mutual information-like objective limits the required environment interaction because it can be optimized with (s_0, a, s_n) transitions from a replay buffer β.

an impractical approach to building skills. As the maximum mutual information between skills and states, empowerment encourages agents to learn a large set of skills, in which each skill targets a distinct grouping of states. Thus, instead of forcing agents to learn skills that target specific states like GCRL, empowerment enables agents to learn more abstract skills that can target sets of states that are not necessarily close to one another in the state space. The flexibility to learn abstract skills can be helpful in stochastic domains where skills cannot always achieve tight regions of states. For instance, continuing the household robot example mentioned earlier, empowerment could help the robot learn a skill to pick up the sponge from the sink. The skill could target a grouping of states, in which each state in the group describes a scenario where the robot holds the sponge but may differ in the background noise that occurs or the different lighting and shadows that the robot sees when looking at the sink.

Yet despite its potential to help agents learn large abstract skillsets, empowerment is not yet a practical approach for skill learning because it typically requires an infeasible amount of interaction with the environment. Computing the mutual information between skills and states for a single skillset (e.g., a single skill-conditioned policy) requires significant interaction in the environment. In order to obtain an estimate of the mutual information of a skillset (i.e., measure how diverse the skillset is), many skills need to be executed to understand the relationship between skills and their skill-terminating states. Empowerment, however, is the maximum mutual information between skills and states so empowerment can require computing the mutual information for a large number of candidate skillsets. For instance, in one of the leading empowerment approaches, candidate skillsets include those formed by small changes to each of the thousands of parameters that make up a skill-conditioned policy (Anonymous *et al.*, 2024). This is an infeasible amount of interaction with the environment, which is why this approach and other empowerment approaches require that the agent has access to a simulator of the environment, enabling agents to simulate executing thousands of actions in parallel (Anonymous *et al.*, 2024; Levy *et al.*, 2023; Gu *et al.*, 2021).

It is important to note that although the quantity of interaction typically required by empowerment is infeasible, the interaction that is needed is often highly redundant as the agent will frequently execute the same action. For instance, when executing skills from candidate skillsets formed by a small change to one of the parameters of the skill-conditioned policy, this interaction will be highly repetitive because a small change to a single parameter should not noticeably change the marginal distribution over actions produced by the candidate skillsets. Further, over time as the agents makes small changes to its skillset in order to improve mutual information, these changes should often be additive, in which the agent adds skills to reach new states. The agent will generally not add skills that execute actions that reach states that can already be achieved because that would lower mutual

information of the skillset. Thus over time, the agent should be frequently repeating the distinct actions it has discovered. Given the infeasibly large but also highly redundant environment interaction that empowerment requires, an important question is whether there is a way to reuse past experience to compute empowerment in a more sample efficient manner.

We introduce a new algorithm, *Off-Policy Empowerment Bandits* (OPEB), that can optimize empowerment by reusing past experience. The key ingredient is our objective which can form a tight bound with the mutual information between skills and states, but also be optimized with transition data. Figure 1 provides an overview of our approach. We validate our approach in a series of stochastic domains, including ones with high-dimensional image observations. Our key result is that the empowerment levels of the skillsets learned by our approach match the empowerment achieved by the leading empowerment algorithm (Anonymous *et al.*, 2024), which assumed the agent had access to a simulator as the algorithm required several orders of magnitude more environment interaction than our approach.

2 Background

2.1 Problem Setting

We consider the typical unsupervised skill-learning problem setting in which an agent interacts in a controlled Markov process, which is a Markov Decision Process (MDP) without a reward function. The controlled Markov process is defined by the tuple $(\mathcal{S}, p(s_0), \mathcal{A}, p(s_{t+1}|s_t, a_t))$, in which $\mathcal{S}$ is the space of states; $p(s_0) : \Delta(\mathcal{S})$ is the initial state distribution; $\mathcal{A}$ is the space of actions; $p(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represents the transition dynamics distribution.

The goal of the unsupervised skill-learning setting is for agents to learn a diverse skillset that can be used for downstream tasks. We will define a skillset by the tuple $(\mathcal{Z}, l, \theta, p(s_n|s_0, l, \theta, z))$. $\mathcal{Z}$ is the space of skills. l represents the parameter(s) that define the distribution over skills $p(z|l)$. For instance, in our approach the distribution over skills takes the form of a uniform distribution over a d-dimensional cube, and l is the scalar value that reflects the side length of each of the d-dimensions of the cube. In our 2-dimensional tasks, for instance, skills are sampled from a square with side length l. θ represents the set of parameters that define the skill-conditioned policy $\pi(a|s, z, \theta)$. In our case, θ is the set of weights and biases that make up the skill-conditioned policy neural network, which takes as input the state and skill and outputs the mean of the distribution over actions. $p(s_n|s_0, l, \theta, z)$ is the transition function that outputs the skill-terminating state s_n given the start state s_0, skill distribution parameter l, skill-conditioned policy parameters θ, and the specific skill z.

2.2 Empowerment

Empowerment is an objective function for learning diverse skillsets. The empowerment of a state s_0 is typically defined

$$\mathcal{E}(s_0) = \max_{l,\theta} I(Z; S_n|s_0) = \max_{l,\theta} H(Z|s_0) - H(Z|s_0, S_n) \tag{1}$$

$$= \max_{l,\theta} \mathbb{E}_{z\sim p(z|s_0), s_n \sim p(s_n|s_0,z)}[\log p(z|s_0, s_n) - \log p(z|s_0)]. \tag{2}$$

$I(Z; S_n|s_0)$ is the mutual information between skills and skill-terminating states for the state s_0 under consideration. In line 2, l and θ have been marginalized in the distributions $p(z|s_0) = p(z|s_0, l)$ and $p(s_n|s_0, z) = p(s_n|s_0, z, \theta)$. Thus, the goal in empowerment is to learn a skillset defined by (l, θ) with high mutual information between skills and states. Per line 1, a skillset has high mutual information if it is diverse because diverse skillsets have (i) relatively high $H(Z|s_0)$ (i.e., the number of skills in the skillset is large) and (ii) relatively low $H(Z|s_0, S_n)$ (i.e., skills target distinct states).

2.3 Empowerment Optimization Challenges

Optimizing the empowerment objective poses several challenges. One key difficulty with empowerment is that the mutual information term is a function of a posterior probability $p(z|s_0, s_n)$ that is intractable to compute in domains with continuous state, action, and skill spaces because the probability requires integrating over the state, action, and skill random variables in the skill trajectory. Mohamed & Rezende (2015) provided a solution to this problem — replace the problematic

posterior $p(z|s_0, s_n)$ with a variational posterior $q_\phi(z|s_0, s_n)$ parameterized by ϕ, and the resulting objective shown in equation 3 is a lower bound on empowerment, sometimes referred to as variational empowerment $\mathcal{E}^V$.

$$\mathcal{E}^V(s_0) = \max_{l,\theta} I^V(Z; S_n|s_0) = \max_{l,\theta} H(Z|s_0) + \mathbb{E}_{z\sim p(z|s_0), s_n\sim p(s_n|s_0,z)}[\log q_\phi(z|s_0, s_n)] \quad (3)$$

The tightness of the bound between $I(Z; S_n)$ and $I^V(Z; S_n)$ depends on the KL divergence between the true and variational posteriors (Barber & Agakov, 2003). For instance, if the variational posterior takes its typical form as a diagonal gaussian (i.e., $q_\phi(z|s_0, s_n) = \mathcal{N}(z; [\mu, \sigma] = f_\phi(s_0, s_n))$), and the true posterior also takes a similar form to a diagonal gaussian (e.g., a diverse skillset in which skills target distinct grouping of states), then the bound has the potential to be tight. On the other hand, the bound can be loose for less diverse skillsets in which distant skills z target the similar states s_n.

The next major challenge is how to learn a skillset defined by (l, θ) that produces large variational mutual information I^V. A popular approach has been to optimize equation 3 with Reinforcement Learning (Gregor *et al.*, 2016; Eysenbach *et al.*, 2018; Achiam *et al.*, 2018; Baumli *et al.*, 2020; Choi *et al.*, 2021). When the distribution over skills is fixed as most of these algorithms do, the objective looks like a goal-conditioned RL problem in which the reward is 0 for the first $n - 1$ actions and then the final step reward $R(s_{n-1}, a_{n-1}, s_{n-1}, z, \theta) = \log q_{\phi^*}(z|s_0, s_n)$. ϕ^* is used in place of ϕ because prior to applying RL, an inner optimization loop is executed to update the parameters ϕ so that the variational distribution $q_\phi(z|s_0, s_n)$ is closer to the true posterior $p(z|s_0, s_n)$ and thus the mutual information bound is tighter. ϕ is updated using a maximum likelihood objective $\mathbb{E}_{z\sim p(z), s_n\sim p(s_n|s_0,s_n)}[\log q_\phi(z|s_0, s_n)]$. Importantly, because the output of this maximum likelihood problem, ϕ^*, is a function of the skill-conditioned policy parameters θ, which are needed to produce the s_n samples in the maximum likelihood problem, that means the reward in the RL problem is also a function of θ, which is why θ was also included as an input into the reward function.

The major issue with applying RL to this objective is the objective does not directly encourage diverse skillset learning, as noted by others (Levy *et al.*, 2023; Campos *et al.*, 2020; Anonymous *et al.*, 2024). Instead, the objective encourages updates to θ that cause the agent to go states where the agent already visits. This is because the variational posterior $q_{\phi^*}(z|s_0, s_n)$ reflects the relationship between skills and states for the current skill-conditioned policy parameters θ. When $q_{\phi^*}(z|s_0, s_n)$ is then used as a reward that means an agent that pursues some skill z is encouraged to go states that z typically visits as these will have high $q_{\phi^*}(z|s_0, s_n)$. But this can be achieved by making little changes to θ. Indeed, the reward for a potential new θ that caused many skills to reach new states (i.e., produced higher mutual information), would be $\log 0$, which is undefined.

2.4 Empowerment Bandits

Recent work by Anonymous *et al.* (2024) (paper also submitted to RLBRew) introduced an approach, Empowerment Bandits (EB), that makes a few key changes to the empowerment objective in order to better align the objective towards diverse skillset learning. The first change is to move the skillset parameters l and θ so that they are conditioning variables for the mutual information term as opposed to marginalized variables in the $p(z|s_0)$ and $p(s_n|s_0, z)$ distributions. That is, the updated empowerment objective seeks to maximize the mutual information term $I^V(Z; S_n|s_0, l, \theta)$ instead of $I^V(Z; S_n|s_0)$. If l and θ are not among the conditioning variables, then the objective will continue to encourage stagnant policies because the agent will only seek to go to states where the variational posterior $q_\phi(z|s_0, s_n)$ is high, which are the states the agent already goes to. However, if l and θ are added to the set conditioning variables, then the agent can learn the variational posterior and mutual information for different skillsets defined by l and θ and then choose skillsets that are more diverse. The next modification is to change the learned action space from the primitive action space to the skillset parameters θ and l. Specifically, agents will now learn two policies. One policy f_λ will take as input the skill start state s_0 and the skill distribution parameter l and output θ. The other skillset parameter policy f_ψ will take as input the skill start state s_0 and output l, which is side length of the d-dimensional cube that represents the uniform distribution over skills. Anonymous *et al.* (2024) thus redefine the variational lower bound on empowerment to be

$$\mathcal{E}^V(s_0) = \max_{l,\theta} I^V(Z; S_n|s_0, l, \theta) \quad (4)$$

$$= \max_{l,\theta} H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|s_0,l), s_n\sim p(s_n|s_0,l,\theta,z)}[\log q_{\phi^*}(z|s_0, l, \theta, s_n)], \quad (5)$$

$$\theta = f_\lambda(s_0, l); l = f_\psi(s_0).$$

ϕ^* is again the output of the inner loop maximum likelihood problem that seeks to reach a tighter bound between I^V and the true mutual information.

Anonymous *et al.* (2024) optimize this empowerment objective as two nested bandit problems. In the inner bandit problem, the trained policy is f_λ and the reward $R(s_0, l, \theta)$ is the variational lower bound on mutual information $I^V(Z; S_n|s_0, l, \theta)$. That is, the agent is encouraged to find θ that produce more diverse skillsets. The outer bandit problem seeks to train the policy f_ψ that learns to output the skill distribution parameter l. The reward $R(s_0, l)$ is $I^V(Z; S_n|s_0, l, \theta)$, in which θ is the greedy output of $f_\lambda(s_0, l)$. f_λ will thus be encouraged to output a larger l if the agent is able to execute a more diverse skillset from a larger d-dimensional cube of goals, but the agent can also lower or maintain the current size of the skill distribution if that produces more mutual information. In summary, the proposed objective from Anonymous *et al.* (2024) directly encourages learning diverse skillsets because actions are now skillset and the reward for a certain skillset (l, θ) is the mutual information of that skillset. In their experiments, Empowerment Bandits became the first unsupervised skill learning algorithm to learn skillsets in stochastic domains showing that empowerment does have the potential to learn large abstract skillsets.

The major problem with Empowerment Bandits is that the approach requires a prohibitive amount of environment interaction that can only be completed with a simulator of the environment. The skillset candidates that EB evaluates are formed by making small changes to a single parameter of the θ and this process is performed for all $|\theta|$ parameters. Then for each of the tens of thousands of skillset candidates, in order to determine the variational posterior q^* for that skillset, many skills need to be executed to obtain the needed (z, s_n) tuples. In total, this can often require hundreds of thousands of skills to be executed in the environment. This is not feasible, which is why Anonymous *et al.* (2024) required that the agent had access to a simulator of the environment, which enables thousands of skills to be executed in parallel.

3 Off-Policy Empowerment Bandits

Building off of the work of Anonymous *et al.* (2024), we introduce a new algorithm, Off-Policy Empowerment Bandits (OPEB), that can compute empowerment without a simulator. Key to our approach is our reward objective $R(s_0, l, \theta)$, which like the $I^V(Z; S_n|s_0, l, \theta)$ reward in Empowerment Bandits, can form a tight bound to the mutual information $I(Z; S_n|s_0, l, \theta)$. However, unlike $I^V(Z; S_n|s_0, l, \theta)$, which required a prohibitive amount of on-policy tuples (z, a, s_n) to optimize its parameters ϕ^*, the parameters that make up our reward objective can be optimized in part with (state s_0, action a, state s_n) transition data from a replay buffer. A limitation of our approach is that the actions that skills execute are open-loop sequences of primitive actions. That is, an action a consists of a concatenation of n primitive actions $[a_0, a_1, \ldots, a_{n-1}]$. This is a limitation as because it restricts the agent to only learning short-horizon skills. However, computing the diversity of a skillset employing short open-loop actions is still a difficult problem because (i) there can be a lot of redundancy in the action sequences (i.e., numerous action sequences can reach the same skill-terminating state s_n) and (ii) the amount of environment interaction is still prohibitively large with an approach like Empowerment Bandits and in practice would require a simulator.

3.1 Deriving the Reward Objective

Next, we will partly walk through a derivation of our reward objective $R(s_0, l, \theta)$ that can both form a tight bound to $I(Z; S_n|s_0, l, \theta)$ and be optimized with transition data from a replay buffer instead of extensive environment interaction. The purpose of the initial steps of this derivation are to obtain an objective that both forms a tight bound on mutual information but also takes a form such that a latent variable model can be integrated so that the joint distribution of random variables can be changed so that on-policy (z, a, s_n) tuples are not needed. Beginning with the mutual information objective that we want to maximize $I(Z; S_n|s_0, l, \theta)$, a lower bound of this term is $I(Z; Z_n|s_0, l, \theta)$, in which $Z \to S_n \to Z_n$ forms a Markov chain given the tuple (s_0, l, θ) (e.g., z_n is some function of s_0, l, θ, s_n):

$$I(Z; S_n|s_0, l, \theta) \geq I(Z; Z_n|s_0, l, \theta) = H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|l), z_n\sim p(z_n|s_0,l,\theta,z)}[\log p(z|s_0, l, \theta, z_n)]$$

We will sometimes refer to the distribution $p_\alpha(z_n|s_0, l, \theta, s_n)$ as the abstraction distribution as it seeks to map state s_n to a latent variable z_n. The bound is due to the Data-Processing Inequality

(Cover & Thomas, 2006). Next, the mutual information term $I(Z; Z_n|s_0, l, \theta)$ is upper bounded by the objective $I^J(Z; Z_n|s_0, l, \theta) = H(Z|s_0, l) + \log(\mathbb{E}_{z\sim p(z|l), z_n\sim p(z_n|s_0,l,\theta,z)}[p(z|s_0, l, \theta, z_n)])$. We use the notation I^J because the upper bound results from Jensen's Inequality, which states that $\log \mathbb{E}(X) \geq \mathbb{E}[\log(X)]$, in which X is some random variable. The upper bound of I^J can be a tight bound when I^J is large because then the posterior probabilities $p(z|s_0, l, \theta, z_n)$ are likely similarly large. If this is the case, then the bound between I^J and I^V will be tight due to the concavity of the log function. Figure 3 provides a illustration of this idea. As a consequence of the potential tightness of the upper bound, if the agent is able to find a θ that produces large $I^J(Z; Z_n|s_0, l, \theta)$, this can also produce a large $I(Z; Z_n|s_0, l, \theta)$, which would then mean the mutual information $I(Z; S_n|s_0, l, \theta)$ is large, which is our goal.

From the objective $I^J(Z; Z_n|s_0, l, \theta)$, now a latent variable model $p_\eta(z_n|s_0, l, \theta, a)$ can be integrated, forming a lower bound objective of I^J that can be optimized in part with off-policy transition data $(s_0, a, s_n) \sim \beta$ instead of on-policy $(z, a, s_n|s_0, l, \theta)$ tuples. The latent variable model will take as input s_0, l, θ, and the action a and output a distribution over the latent variable z_n. By integrating the latent variable model using a three-step process consisting of (i) the importance sampling change of distribution trick, (ii) another application of Jensen's Inequality, and (iii) replacing the intractable posterior $p(z|s_0, l, \theta, z_n)$ with the variational posterior $q_\phi(z|s_0, l, \theta, z_n)$, the following lower bound objective is obtained (see section B of the Appendix for the full derivation):

$$I^J(Z; Z_n|s_0, l, \theta) \geq H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|l), a\sim\pi(a|s,z,\theta), z_n\sim p_\eta(z_n|s_0,l,\theta,a)}[\log q_\phi(z|s_0, l, \theta, z_n)] \\ - \mathbb{E}_{(a,s_n)\sim p(a,s_n|s_0,l,\theta)}[D_{KL}(p_\eta(z_n|s_0, l, \theta, a)||p_\alpha(z_n|s_0, l, \theta, s_n))] \quad (6)$$

The reward we will use for our nested bandit problems will be the same as 6, except it will use the optimized η^*, α^*, and ϕ^* for its parameters.

This reward function in line 6 provides an intuitive way to measure the diversity of a skillset, which should be expected from an objective that can form a tight bound with the mutual information $I(Z; S_n|s_0, l, \theta)$. The sum of the first two terms $H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|l), a\sim\pi(a|s,z,\theta), z_n\sim p_{\eta^*}(z_n|s_0,l,\theta,a)}[\log q_{\phi^*}(z|s_0, l, \theta, z_n)]$ equals the mutual information $I^V(Z; Z_n|s_0, l, \theta)$, which lower bounds $I(Z; Z_n|s_0, l, \theta)$, which itself is a lower bound of the mutual information $I(Z; A|s_0, l, \theta)$ due to the Data-Processing Inequality because $Z \to A \to Z_n$ form a Markov chain. $I(Z; A|s_0, l, \theta)$ measures the diversity of actions produced by the skillset, ignoring the states to which those actions lead. Skillsets in which different skills execute distinct actions will produce higher $I(Z; A)$, while skillsets in which many skills execute similar actions will produce lower $I(Z; A)$. Note that high $I^V(Z; Z_n|s_0, l, \theta)$ means the skillset (l, θ) given some skill z outputs an action a that can be encoded to a z_n by the latent variable model $p_{\eta^*}(z_n|s_0, l, \theta, a)$ and then decoded back to a skill z close to the original skill z by the variational posterior $q_{\phi^*}(z|s_0, l, \theta, z_n)$. While the first two terms thus measure the diversity of actions by the skillset, the last term checks whether those distinct actions target overlapping states. Consider the situation in which two skills z_0 and z_1 that are far apart in the skill space execute different actions a_0 and a_1 that result in the same state s_n. In order to make the second term in line 6 high, the latent variable model $p_{\eta^*}(z_n|s_0, l, \theta, a)$ may have mapped actions a_0 and a_1 to distinct z_n^0 and z_n^1 variables, respectively, that the variational posterior $q_{\phi^*}(z|s_0, l, \theta, z_n)$ could then decode near the original skills z_0 and z_1. However, if this were the case, due to the KL divergence in the third term of 6, the distribution $p_{\alpha^*}(z_n|s_0, l, \theta, s_n)$ would be forced to learn a wide distribution to cover both z_n^0 and z_n^1, which would lead to a high KL divergence and penalize the skillset.

Further, in addition to providing an intuitive way to measure the diversity of a skillset, optimizing the reward objective in line 6 with respect to the parameters η, α, and ϕ does not require on-policy (z, a, s_n) tuples that require extensive interaction. The first two terms that combine to form $I^V(Z; Z_n|s_0, l, \theta)$ only require samples of $(z, a, z_n|s_0, l, \theta)$ in which z_n is sampled using the latent variable model p_η, so these two terms do not require any environment interaction. The last term which measures the KL divergence between the latent variable model p_η and the abstraction distribution p_α requires samples of $(a, s_n|s_0, l, \theta)$ (i.e., (state s_0, action a, next state s_n) transition data from the skillset under consideration). However, because (a) the skillsets that will be evaluated will be slight changes to the current skillset and (b) changes to the skillset should largely be additive with new skills executing new actions that need to new states because adding redundant skills is discouraged, a large portion of the needed transition data can often be supplied by sampling (s_0, a, s_n) from the replay buffer β.

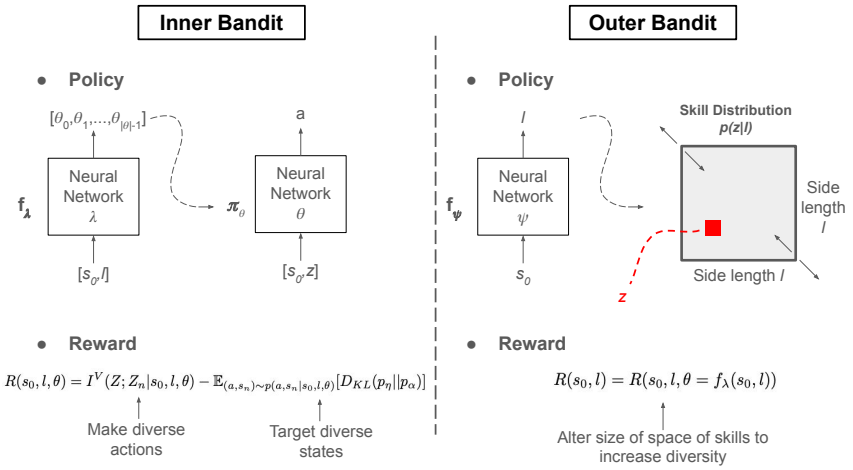

Figure 2: **Overview of bandit problems.** (Left) Agent learns policy f_λ to output θ, which are the parameters of the skill-conditioned policy neural network. Agent is rewarded for θ that produce more diverse skillsets. (Right) Agents learns policy f_ψ which outputs a scalar value l, representing the side length of the d-dimensional cube uniform skill distribution. Policy receives higher reward for skill distributions that produce a more diverse skillset.

There will be times, particularly early in training, in which a candidate skillset executes actions that are not in the replay buffer. In that case, there will not be (s_0, a, s_n) transitions from which the exact KL divergence can be estimated and instead the agent will need to rely on the KL divergence estimates from similar actions. If the KL divergence term for nearby actions is relatively low, the reward for the candidate skillset that includes the new actions should be higher than the current skillset because $I^V(Z; Z_n|s_0, l, \theta)$ should be larger with more distinct actions. If the skillset was updated to include these new actions, then when interacting with the environment in the future, the agent can execute the new actions and the (s_0, a, s_n) transitions will be replaced in the replay buffer. Then in a subsequent update phase, the agent could remove this action if it was redundant or keep the action it it was a new action. In addition, sampling (state s_0, action a, next state s_n) transitions that are not used in the current skillset should not affect the η^*, α^*,and ϕ^* parameters that are learned. For instance, if the action a from the transition is not executed by the current skillset and nor is s_n reached by the current skillset reached from some other action, then the only term in the objective that is affected by this transition is the KL divergence term and so p_η and p_α could map the action and next state to the same z_n. If the action is not executed by the current skillset but the state s_n is reached by some other action, then p_η could map the action from the transition to the same z_n as the other action that yields the state s_n.

With the reward function derived, Figure 2 provides visualization of the inner and outer bandit problems.

3.2 Algorithm

Next we detail how the Off-Policy Empowerment Bandits algorithm works. OPEB learns an increasingly diverse skillset by repeating a two phase process. In the first phase, the agent interacts with the environment by greedily executing a certain number of skills from its skillset. A skill $z \sim p(z|l)$ is sampled, and then an action $a \sim \pi(a|s, z, \theta)$ is executed returning a skill-terminating state $s_n \sim p(s_n|s_0, a)$. The (s_0, a, s_n) tuples are stored in a replay buffer β. In the second phase, the agent uses its dataset of transitions to learn a skillset (l, θ) with larger mutual information $I(Z; S_n|s_0, l, \theta)$.

The second phase consists of a sequence of three steps. The first step is to optimize the reward parameters η, α, and ϕ for a variety of $(\tilde{l}, \tilde{\theta})$ skillsets, in which $(\tilde{l}, \tilde{\theta})$ are noisy versions of the current (l, θ) skillsets. The purpose of optimizing the reward parameters is because the higher the reward the tighter the bound can be between the reward and the true mutual information $I(Z; S_n|s_0, \tilde{l}, \tilde{\theta})$ of the

candidate skillset. The objective for optimizing the reward parameters is

$$J(\eta, \alpha, \phi) = \mathbb{E}_{\tilde{l},\tilde{\theta},z\sim p(z|\tilde{l}),a\sim p(a|s,z,\tilde{\theta}),z_n\sim p(z_n|s_0,\tilde{l},\tilde{\theta})}[\log q(z|s_0,\tilde{l},\tilde{\theta},z_n)] \\ - \mathbb{E}_{\tilde{l},\tilde{\theta},(a,s_n)\sim\beta,z_n\sim p_\eta(z_n|s_0,\tilde{l},\tilde{\theta},a)}[\log p_\eta(z_n|s_0,\tilde{l},\tilde{\theta},a) - \log p_\alpha(z_n|s_0,\tilde{l},\tilde{\theta},s_n)]. \tag{7}$$

Note that the objective for updating the parameters of the reward function samples the (s_0, a, s_n) tuples from a replay buffer β, which is the benefit of our approach relative to the work of Anonymous *et al.* (2024). In the case of a diverse skillset in with different skills z target distinct states s_n, the outcome of this optimization should be that the latent variable model p_η encode actions a and the abstraction distribution p_α encodes targeted states s_n to similar latent variables z_n (i.e., low D_{KL}), which can then be decoded by the variational posterior q_ϕ to the skills z that execute the action a. Given that η^*, α^*, and $\phi*$ have been computed, the rewards $R(s_0, l, \theta)$ should be closer to the true mutual information.

In the second step of the update process, the inner bandit actor-critic is updated so that f_λ outputs θ that produce more diverse skillsets. In the third step, the outer bandit actor-critic is updated so that f_ψ outputs skill space size parameter l associated with more diverse skillsets. For detailed objective functions for the bandit actor-critics, please see section C of the appendix.

Similar to Anonymous *et al.* (2024), learning the functions p_η, p_α, q_ϕ, Q_γ^{KL} (used to estimate the KL divergence), and Q_κ (critic for f_λ), all of which take as input θ is intractable due to the potentially huge size of θ. We overcome this obstacle by applying the same solution used by Anonymous *et al.* (2024), in which parameter-specific versions of these functions are implemented and trained in parallel using multiple accelerators and the parallelization capabilities of modern deep learning frameworks (e.g., JAX) (Bradbury *et al.*, 2018). For more detail on how the parameter-specific functions are implemented see section D of the appendix.

4 Experiments

The main purpose of our experiments is to evaluate whether our approach can learn skillsets with empowerment levels that match those achieved by existing empowerment methods that require a simulator of the environment. We evaluate our algorithm in the same four stochastic environments that Empowerment Bandits was implemented in. A description of each of these environments is provided below. An image of random action sequences taken in these domains is provided in section E of the appendix.

1. **Stochastic Four Rooms Navigation:** In this domain, a two-dimensional point agent navigates in an environment with four walled rooms by executing two-dimensional $(\Delta x, \Delta y)$ actions. The domain is highly stochastic because after each action is completed, the agent is moved to the same (x offset, y offset) location in a randomly sampled room. The abstract states that can be targeted are the (x offset, y offset) from the center of a room.
2. **Stochastic Four Room Pick-and-Place:** This is the same environment as the navigation task except there is now an object (the red triangle in the second row of Figure 4) that can be moved if the agent is within a certain distance of the object.
3. **RGB QR Code Navigation:** In this domain an agent learns to navigate amid a continually changing RGB-colored QR code background. Observations are 507-dimensional RGB images and are highly stochastic as the colored-QR code image fully changes after each action.
4. **RGB QR Code Pick-and-Place:** This environment is the same as the navigation task except there is a now an object that can be moved if the object is in reach.

In all domains, there is a single skill start state s_0 and skills consist of 5 primitive actions.

The main baseline we compare to is Empowerment Bandits, which is the only other unsupervised skill-learning algorithm that has been able to learn skills in stochastic domains. However, Empowerment Bandits requires a simulator of the environment as it requires multiple orders of magnitude more interaction with the environment than ours. For additional perspective, we also show the results of GCRL and Variational Intrinsic Control (VIC) (Gregor *et al.*, 2016) provided in the Empowerment Bandits paper in these exact same domains. In the stochastic four rooms domains, the GCRL approach

Table 1: Variational empowerment (nats) of learned skillsets for all baselines over five random seeds and one standard deviation of error.

TASK	OURS	EB	VIC	GCRL
FOUR ROOMS NAV.	**5.8± 0.3**	5.1± 0.3	0.2± 0.4	0.3± 0.4
FOUR ROOM P.-AND-P.	7.6± 0.4	**8.7± 0.3**	-0.1± 0.3	3.9± 0.6
RGB QR NAV.	**4.0± 0.3**	3.5± 0.1	-0.4± 0.0	-0.4± 0.3
RGB QR P.-AND-P.	**6.1± 0.5**	6.0± 0.2	-0.6± 0.1	-2.6± 5.8

used is the variant of GCRL that forms a lower bound to mutual information (Choi *et al.*, 2021). In the image domains, the GCRL approach is RIG (Nair *et al.*, 2018), which performs GCRL in a latent space learned separately by a VAE. VIC was used to represent the category of previous empowerment methods such as DIAYN and VALOR that often struggled to learn diverse skillsets because the reward function encouraged stagnant skillsets. Both the GCRL and VIC had access to a simulator of the environment. We plan to add versions that use off-policy data, but the performance of these methods with biased (z, s_n) tuples should be expected to perform worse than their counterparts with access to the simulator.

For a more detailed discussion of related work see section F in the appendix. We will move this section to the main body of the paper in a future version of the paper.

4.1 Results

The main result is that Off-Policy Empowerment Bandits was able to overall match the performance of Empowerment Bandits per table 1 while requiring orders of magnitude fewer interactions with the environment. Figure 5 shows the mean performance and one standard deviation over time for OPEB in all tasks. In three out of the four tasks, OPEB was able to learn skillsets with larger empowerment and in the remaining task was still able to learn a large skillset. Note that negative values shown by other baselines in the table can occur when the skill-learning algorithm performs poorly and $I^V(Z; S_n)$ forms a loose lower bound on $I(Z; S)$.

Additional evidence that our approach is working as expected can be seen in the visualizations of the empowerment-related entropies $H(S_n)$, $H(S_n|Z)$, $H(Z)$, and $H(Z|S_n)$, which are provided for each task in section H of the appendix. The $H(S_n)$ visual shows the skill-terminating states from 1000 skills randomly sampled from the skill space. In all tasks, the skill-terminating states s_n nearly uniformly cover the reachable state space. The $H(S_n|Z)$ visuals show the particular states targeted by individual skills. As these visuals show, each skill targets the appropriate abstract state in each task: (x, y) offset locations in the stochastic four rooms tasks and (x, y) positions in the RGB QR Code tasks. The $H(Z)$ and $H(Z|Z_n)$ visual for each task show the learned skill space $p(z|l)$ and samples from the variational posterior $q_\phi(z|z_n)$. As would be expected from a skillset with high mutual information, the posterior forms a narrow distribution around the original skill. Further, the $H(Z|Z_n)$ visuals shows samples from the variational posterior in which the input z_n are taken both from the abstraction distribution $p_\alpha(z_n|s_n)$ and the latent variable model $p_\eta(z_n|a)$. Given that outputs of the variational posterior from the two sources of the input z_n, both target the original skill z means the algorithm is working as expected and the KL divergence between the latent variable model p_η and the abstraction distribution p_α is low.

5 Conclusion

Empowerment has the potential to help agents learn the large abstract skillsets needed to operate in realistic environments. However, in order to be a practical objective for building skillsets, the objective needs to require a tractable amount of interaction with the environment. We take a step in this direction by introducing a version of empowerment that can learn skillsets using prior transition data. Our approach was able to learn large skillsets in stochastic settings while requiring orders of magnitude less environment interaction than the leading empowerment approach.

References

Achiam, Joshua, Edwards, Harrison, Amodei, Dario, & Abbeel, Pieter. 2018. Variational Option Discovery Algorithms. *CoRR*, **abs/1807.10299**.

Anonymous, A., Anonymous, B., & Anonymous, C. 2024. Learning Abstract Skillsets with Empowerment Bandits.

Barber, David, & Agakov, Felix. 2003. The IM Algorithm: A Variational Approach to Information Maximization. *Page 201–208 of: Proceedings of the 16th International Conference on Neural Information Processing Systems*. NIPS'03. Cambridge, MA, USA: MIT Press.

Baumli, Kate, Warde-Farley, David, Hansen, Steven, & Mnih, Volodymyr. 2020. Relative Variational Intrinsic Control. *CoRR*, **abs/2012.07827**.

Bradbury, James, Frostig, Roy, Hawkins, Peter, Johnson, Matthew James, Leary, Chris, Maclaurin, Dougal, Necula, George, Paszke, Adam, VanderPlas, Jake, Wanderman-Milne, Skye, & Zhang, Qiao. 2018. *JAX: composable transformations of Python+NumPy programs*.

Burda, Yuri, Edwards, Harrison, Storkey, Amos J., & Klimov, Oleg. 2018. Exploration by Random Network Distillation. *CoRR*, **abs/1810.12894**.

Campos, Víctor, Trott, Alexander, Xiong, Caiming, Socher, Richard, Giró-i-Nieto, Xavier, & Torres, Jordi. 2020. Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills. *CoRR*, **abs/2002.03647**.

Choi, Jongwook, Sharma, Archit, Lee, Honglak, Levine, Sergey, & Gu, Shixiang Shane. 2021. Variational Empowerment as Representation Learning for Goal-Based Reinforcement Learning. *CoRR*, **abs/2106.01404**.

Cover, Thomas M., & Thomas, Joy A. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience.

Ecoffet, Adrien, Huizinga, Joost, Lehman, Joel, Stanley, Kenneth O., & Clune, Jeff. 2019. Go-Explore: a New Approach for Hard-Exploration Problems. *CoRR*, **abs/1901.10995**.

Eysenbach, Benjamin, Gupta, Abhishek, Ibarz, Julian, & Levine, Sergey. 2018. Diversity is All You Need: Learning Skills without a Reward Function. *CoRR*, **abs/1802.06070**.

Gregor, Karol, Rezende, Danilo Jimenez, & Wierstra, Daan. 2016. Variational Intrinsic Control. *CoRR*, **abs/1611.07507**.

Gu, Shixiang Shane, Diaz, Manfred, Freeman, Daniel C., Furuta, Hiroki, Ghasemipour, Seyed Kamyar Seyed, Raichuk, Anton, David, Byron, Frey, Erik, Coumans, Erwin, & Bachem, Olivier. 2021. *Braxlines: Fast and Interactive Toolkit for RL-driven Behavior Engineering beyond Reward Maximization*.

Hafner, Danijar, Lillicrap, Timothy P., Fischer, Ian, Villegas, Ruben, Ha, David, Lee, Honglak, & Davidson, James. 2018. Learning Latent Dynamics for Planning from Pixels. *CoRR*, **abs/1811.04551**.

Held, David, Geng, Xinyang, Florensa, Carlos, & Abbeel, Pieter. 2017. Automatic Goal Generation for Reinforcement Learning Agents. *CoRR*, **abs/1705.06366**.

Jung, Tobias, Polani, Daniel, & Stone, Peter. 2012. Empowerment for Continuous Agent-Environment Systems. *CoRR*, **abs/1201.6583**.

Kim, Seongun, Lee, Kyowoon, & Choi, Jaesik. 2023. *Variational Curriculum Reinforcement Learning for Unsupervised Discovery of Skills*.

Klyubin, A.S., Polani, D., & Nehaniv, C.L. 2005. Empowerment: a universal agent-centric measure of control. *Pages 128–135 Vol.1 of: 2005 IEEE Congress on Evolutionary Computation*, vol. 1.

Lange, Sascha, & Riedmiller, Martin. 2010. Deep auto-encoder neural networks in reinforcement learning. *Pages 1–8 of: The 2010 International Joint Conference on Neural Networks (IJCNN)*.

Lee, Lisa, Eysenbach, Benjamin, Parisotto, Emilio, Xing, Eric P., Levine, Sergey, & Salakhutdinov, Ruslan. 2019. Efficient Exploration via State Marginal Matching. *CoRR*, **abs/1906.05274**.

Levy, Andrew, Rammohan, Sreehari, Allievi, Alessandro, Niekum, Scott, & Konidaris, George. 2023. *Hierarchical Empowerment: Towards Tractable Empowerment-Based Skill Learning*.

Liu, Hao, & Abbeel, Pieter. 2021. Behavior From the Void: Unsupervised Active Pre-Training. *CoRR*, **abs/2103.04551**.

Mazzaglia, Pietro, Çatal, Ozan, Verbelen, Tim, & Dhoedt, Bart. 2021. Self-Supervised Exploration via Latent Bayesian Surprise. *CoRR*, **abs/2104.07495**.

Mendonca, Russell, Rybkin, Oleh, Daniilidis, Kostas, Hafner, Danijar, & Pathak, Deepak. 2021. Discovering and Achieving Goals via World Models. *CoRR*, **abs/2110.09514**.

Mohamed, Shakir, & Rezende, Danilo Jimenez. 2015. *Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning*.

Nair, Ashvin, Pong, Vitchyr, Dalal, Murtaza, Bahl, Shikhar, Lin, Steven, & Levine, Sergey. 2018. Visual Reinforcement Learning with Imagined Goals. *CoRR*, **abs/1807.04742**.

Park, Seohong, & Levine, Sergey. 2023. *Predictable MDP Abstraction for Unsupervised Model-Based RL*.

Park, Seohong, Choi, Jongwook, Kim, Jaekyeom, Lee, Honglak, & Kim, Gunhee. 2022. Lipschitz-constrained Unsupervised Skill Discovery. *CoRR*, **abs/2202.00914**.

Park, Seohong, Lee, Kimin, Lee, Youngwoon, & Abbeel, Pieter. 2023a. *Controllability-Aware Unsupervised Skill Discovery*.

Park, Seohong, Rybkin, Oleh, & Levine, Sergey. 2023b. *METRA: Scalable Unsupervised RL with Metric-Aware Abstraction*.

Pathak, Deepak, Agrawal, Pulkit, Efros, Alexei A., & Darrell, Trevor. 2017. Curiosity-driven Exploration by Self-supervised Prediction. *CoRR*, **abs/1705.05363**.

Pathak, Deepak, Gandhi, Dhiraj, & Gupta, Abhinav. 2019. Self-Supervised Exploration via Disagreement. *CoRR*, **abs/1906.04161**.

Pitis, Silviu, Chan, Harris, Zhao, Stephen, Stadie, Bradly C., & Ba, Jimmy. 2020. Maximum Entropy Gain Exploration for Long Horizon Multi-goal Reinforcement Learning. *CoRR*, **abs/2007.02832**.

Pong, Vitchyr H., Dalal, Murtaza, Lin, Steven, Nair, Ashvin, Bahl, Shikhar, & Levine, Sergey. 2019. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. *CoRR*, **abs/1903.03698**.

Rajeswar, Sai, Mazzaglia, Pietro, Verbelen, Tim, Piché, Alexandre, Dhoedt, Bart, Courville, Aaron, & Lacoste, Alexandre. 2023. *Mastering the Unsupervised Reinforcement Learning Benchmark from Pixels*.

Rudolph, Max, Chuck, Caleb, Black, Kevin, Lvovsky, Misha, Niekum, Scott, & Zhang, Amy. 2024. *Learning Action-based Representations Using Invariance*.

Salge, Christoph, Glackin, Cornelius, & Polani, Daniel. 2013. Empowerment - an Introduction. *CoRR*, **abs/1310.1863**.

Sekar, Ramanan, Rybkin, Oleh, Daniilidis, Kostas, Abbeel, Pieter, Hafner, Danijar, & Pathak, Deepak. 2020. Planning to Explore via Self-Supervised World Models. *CoRR*, **abs/2005.05960**.

Shyam, Pranav, Jaskowski, Wojciech, & Gomez, Faustino. 2018. Model-Based Active Exploration. *CoRR*, **abs/1810.12162**.

Srinivas, Aravind, Laskin, Michael, & Abbeel, Pieter. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. *CoRR*, **abs/2004.04136**.

Strouse, DJ, Baumli, Kate, Warde-Farley, David, Mnih, Vlad, & Hansen, Steven. 2021. Learning more skills through optimistic exploration. *CoRR*, **abs/2107.14226**.

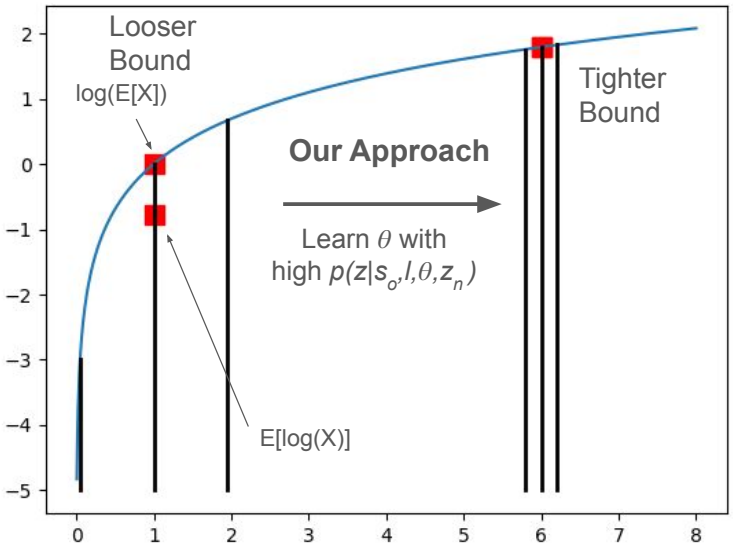

Figure 3: The intuition behind our use of I^J as a reward function even though it upper bounds I^V is that I^J will frequently be maximized when all the posterior probabilities $p(z|s_0, l, \theta, z_n)$ are high. If that is the case, then the difference between I^J and I^V can be small and the bound will be tight.

van den Oord, Aäron, Li, Yazhe, & Vinyals, Oriol. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, **abs/1807.03748**.

Yarats, Denis, Fergus, Rob, Lazaric, Alessandro, & Pinto, Lerrel. 2021. Reinforcement Learning with Prototypical Representations. *CoRR*, **abs/2102.11271**.

Zhang, Amy, McAllister, Rowan, Calandra, Roberto, Gal, Yarin, & Levine, Sergey. 2020. Learning Invariant Representations for Reinforcement Learning without Reconstruction. *CoRR*, **abs/2006.10742**.

Zou, Qiming, & Suzuki, Einoshin. 2024. Compact Goal Representation Learning via Information Bottleneck in Goal-Conditioned Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

A Illustration of Potential Tight Bound

Figure 3 provides some intuition for why the upper bound of I^J relative to I^V can be a tight bound when I^J is high.

B Derivation of Reward Function $R(s_0, l, \theta)$

Below is the derivation of the reward function $R(s_0, l, \theta)$ which is a lower bound on $I^J(Z; Z_n|s_0, l, \theta)$.

$$I^J(Z; Z_n|s_0, l, \theta) = H(Z|s_0, l) + \log\left(\mathbb{E}_{z\sim p(z|l), z_n\sim p(z_n|s_0,l,\theta,z)}[p(z|s_0, l, \theta, z_n)]\right) \tag{8}$$

$$= H(Z|s_0, l) + \log\left(\mathbb{E}_{(z,a,s_n,z_n)\sim p_1(z,a,s_n,z_n)}\left[\frac{p_0(z, a, s_n, z_n)}{p_1(z, a, s_n, z_n)} p(z|s_0, l, \theta, z_n)\right]\right) \tag{9}$$

$$\geq H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|l), a\sim\pi(a|s_0,z,\theta), z_n\sim p_\eta(z_n|s_0,l,\theta,a)}[\log p(z|s_0, l, \theta, z_n)] \tag{10}$$

$$- \mathbb{E}_{(a,s_n)\sim p(a,s_n|s_0,l,\theta), z_n\sim p_\eta(z_n|s_0,l,\theta,a)}[\log p_\eta(z_n|s_0, l, \theta, a) - \log p_\alpha(z_n|s_0, l, \theta, s_n)]$$

$$\geq H(Z|s_0, l) + \mathbb{E}_{z\sim p(z|l), a\sim\pi(a|s_0,z,\theta), z_n\sim p_\eta(z_n|s_0,l,\theta,a)}[\log q_\phi(z|s_0, l, \theta, z_n)] \tag{11}$$

$$- \mathbb{E}_{(a,s_n)\sim p(a,s_n|s_0,l,\theta)}[D_{KL}(p_\eta(z_n|s_0, l, \theta, a)||p_\alpha(z_n|s_0, l, \theta, s_n))]. \tag{12}$$

Line 8 provides the definition of $I^J(Z; Z_n|s_0, l, \theta)$. Line 9 performs the change of distribution trick used in importance sampling. That is, given two distributions $p_0(x)$ and $p_1(x)$, an expectation with respect to the distribution $p_0(x)$ can be switched to an expectation with respect to the distribution $p_1(x)$:

$$\mathbb{E}_{x\sim p_0(x)}[f(x)] = \mathbb{E}_{x\sim p_1(x)}\left[\frac{p_0(x)}{p_1(x)} f(x)\right]. \tag{13}$$

In our case, $p_0(z, a, s_n, z_n) = p(z|l)\pi(a|s_0, z, \theta)p(s_n|s_0, a)p_\alpha(z_n|s_0, l, \theta, s_n)$ and $p_1(z, a, s_n, z_n) = p(z|l)\pi(a|s_0, z, \theta)p(s_n|s_0, a)p_\eta(z_n|s_0, l, \theta, a)$. Line 10 applies Jensen's Inequality. Line 11 replaces the intractable posterior $p(z|s_0, l, \theta, z_n)$ with the variational posterior $q_\phi(z|s_0, l, \theta, z_n)$ forming a lower bound (Barber & Agakov, 2003).

C Bandit Actor-Critic Objective Functions

The inner bandit actor-critic is updated so that f_λ outputs more diverse skillsets θ given a particular l. To do this, we use supervised learning to update the critic for the inner bandit actor, $Q_\kappa(s_0, l, \theta)$, so that it outputs a Q-value close to the target reward $R(s_0, l, \theta)$. For the target reward, we use values sampled from the expectation

$$\text{Target}_\kappa \sim \mathbb{E}_{z\sim p(z|l), a\sim\pi}[\mathbb{E}_{z_n\sim p_{\eta^*}}[\log q_{\phi^*}(z|s_0, l, \theta, z_n)] - Q^{KL}_\gamma(s_0, l, \theta, a)], \tag{14}$$

in which $Q^{KL}_\gamma(s_0, l, \theta, a)$ is a function trained to approximate the KL divergence for an action a taken by the skillset (l, θ). Q^{KL}_γ is trained using samples of (s_0, a, s_n) from the replay buffer. The inner bandit actor is then updated using the objective $J(\gamma) = Q_\kappa(s_0, l, \theta = f_\lambda(s_0, l))$.

The outer bandit actor critic is updated so that f_ψ outputs a skill distribution size l that produces a more diverse skillset. The outer bandit critic, $Q_\nu(s_0, l)$ is updated using supervised learning to output values closer to those output by the inner bandit critic $Q_\kappa(s_0, l, \theta = f_\psi(s_0, l))$. The outer bandit actor is then updated using the objective $J(\psi) = Q(s_0, l = f_\psi(s_0))$.

D Parameter-Specific Functions and Distributions

We apply the same approach as Anonymous *et al.* (2024) to handle the large action space of θ. In order to find the gradient of f_λ that outputs θ, the agent needs to know how the reward $R(s_0, l, \theta)$ reacts to slightly changes in each of the $|\theta|$ parameters in θ. Accordingly, we learn parameter-specific critic functions $Q_{\kappa_0}, Q_{\kappa_1}, \ldots, Q_{\kappa_{|\theta|-1}}$. Each Q_{κ_i} estimates the reward $R(s_0, l, \tilde{\theta}_i)$, in which $\tilde{\theta}_i$ consists of a noisy value of the i-th parameter of θ and the rest are greedy values from $f_\gamma(s_0, l)$. Note that the $\tilde{\theta}_i$

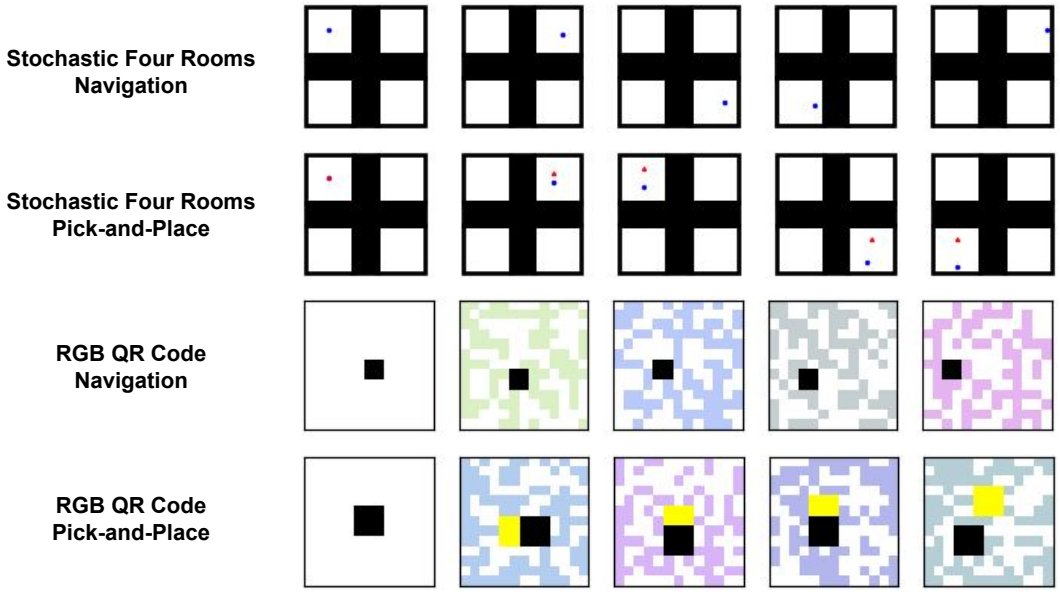

Figure 4: Figure shows images of the four environment we applied our algorithm to. The image shows a random action sequence being executed.

input into the critic Q_{κ_i} is just a scalar value of the noisy i-th parameter. Q_{κ_i} requires the parameters η, ϕ, α and γ for the KL critic so we learn parameter-specific versions of all of those functions.

The parameter-specific functions are all trained in parallel using multiple accelerators and the vectorization functions in JAX (Bradbury *et al.*, 2018). For instance, given 4 GPUs and $|\theta| = 1000$, each GPU would compute the updates for 250 of the parameter-specific functions in parallel. The optimization objectives for the parameter-specific functions are the same as those provided in section 3.2.

The only objective that changes is the one for the inner bandit actor f_λ. Instead of using a single critic to update its policy, f_λ will now use all $|\theta|$ critics to update its policy. This is done with the objective

$$J(\lambda) = \sum_{i=0}^{|\theta|-1} Q_{\kappa_i}(s_0, l, \theta_i), \tag{15}$$
$$\theta_i = f_\lambda(s_0, l)[i],$$

in which $f_\lambda(s_0, l)[i]$ samples the i-th component of the θ vector.

E Environment screenshots

Figure 4 shows images from the four environments we used.

F Related Work

Unsupervised RL Our work falls under a large class of algorithms known as Unsupervised RL. Unsupervised RL algorithms seek to gather information about the world (e.g., exploratory data, skills, world models) either without or with limited human supervision in the form of reward functions or manually crafted goal spaces. One large subclass within unsupervised RL are pure exploration methods which seek to cover the state space by maximizing uncertainty in a learned model Pathak *et al.* (2017); Mazzaglia *et al.* (2021); Shyam *et al.* (2018); Sekar *et al.* (2020); Rajeswar *et al.* (2023); Pathak *et al.* (2019); Burda *et al.* (2018) or maximizing state entropy Lee *et al.* (2019); Liu & Abbeel (2021); Yarats *et al.* (2021); Liu & Abbeel (2021). One general issue with these methods is how exploration can be distilled into reusable skills. The other major subclass within unsupervised RL

is unsupervised skill-learning which includes algorithms that try to learn diverse skill sets without supervision. In addition to the GCRL and empowerment-based skill-learning methods that have been mentioned previously, there are algorithms that combine mutual information objective with exploration bonuses Strouse *et al.* (2021); Park & Levine (2023). There is also another class of algorithms Park *et al.* (2022, 2023a,b) that draw an interesting contrast with empowerment. Instead of taking the empowerment approach of trying to maximize the number of distinct skills learned, these approaches learn small skillsets, but encourage each skill to learn a distinct long-horizon policy. A key limitation of prior unsupervised RL methods is that they have not demonstrated that they can scale to domains with significant stochasticity, in which it is more difficult to learn a model of the transition dynamics or count states or learn skills to achieve particular states.

Representation Learning Also related to our work is the large class of representation learning algorithms, which seek to learn abstract representations. Many representation learning algorithms optimize a reconstruction loss (Lange & Riedmiller, 2010; Hafner *et al.*, 2018) or a contrastive loss (van den Oord *et al.*, 2018; Srinivas *et al.*, 2020). However, both of these losses encourage agents to learn very detailed representations and so may not be helpful for learning abstract states in stochastic domains. There are representation learning techniques that have been able to successfully learn abstract representations (Zhang *et al.*, 2020; Zou & Suzuki, 2024; Rudolph *et al.*, 2024). However, these algorithms have required supervision in the form of reward functions or hand-crafted goal spaces.

G Learning Curves

Figure 5 shows the mean and a standard deviation over time for OPEB in all experiments.

H Entropy Visualizations

Figures 6,7,8, and 9 provide entropy visualizations for each task.

I Compute Requirements

Our experiments were done with a university-level budget. All experiments were conducted with 8 RTX 4090 GPUs but only required a short 1-2 hour period of time.

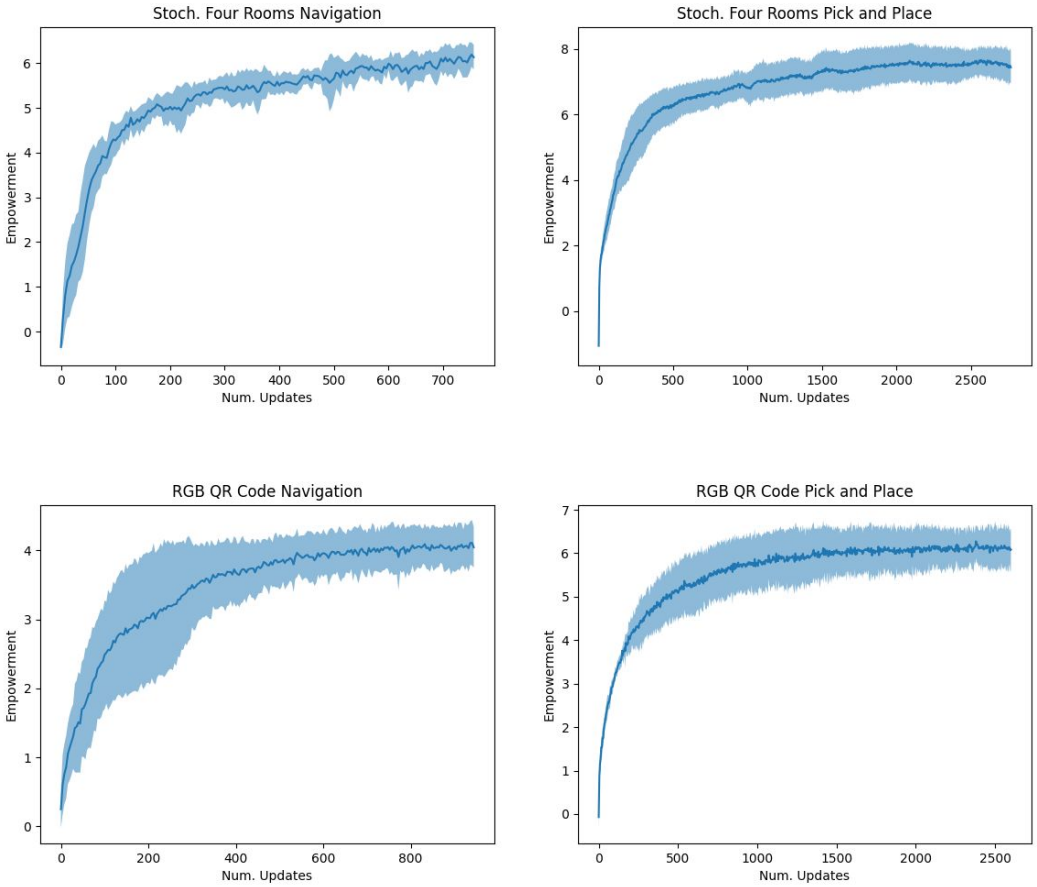

Figure 5: Off-Policy Empowerment Bandits learning curves for all four tasks. Mean and one standard deviation are shown over 5 random seeds. Empowerment is measured in nats. Num. updates refers to the number of updates to the skill parameter policies f_λ and f_ψ

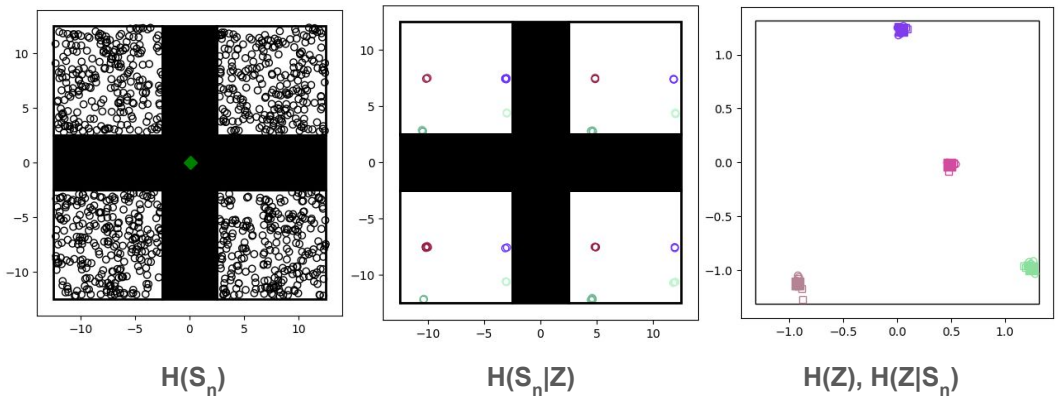

Figure 6: Entropy visualizations for the stochastic four rooms navigation task. Left figure visualizes $H(S_n)$ by showing the skill-terminating states s_n from 1000 randomly sampled skills from the skill space. The skill-terminating states s_n uniformly cover the reachable state space as would be expected from a well-performing empowerment-based skill learning method. Middle figure visualizes $H(S_n|Z)$ by showing the skill-terminating states associated with 4 random selected goals. The s_n belonging to a specific goals are in the same color. Per figure, each skill targets an abstract (x, y) offset location. The last figure visualizes both $H(Z)$ and $H(Z|S_n)$. The skill space in this task is a square and is shown by the black outlined square within the axes. The colored squares are randomly sampled skills z. The empty circles are samples from the variational posterior $q_\phi(z|z_n)$, in which the z_n were sampled from the abstraction dist $p_\alpha(z_n|s_n)$. As would be expected from a skillset with high mutual information, the samples from the posterior are near original skill. This figure also shows empty colored squares which are again samples from the variational posterior, but this time the z_n are sampled from the latent variable model $p_\eta(z_n|a)$. This shows our algorithm is working as expected as the KL divergence between p_η and p_α is low.

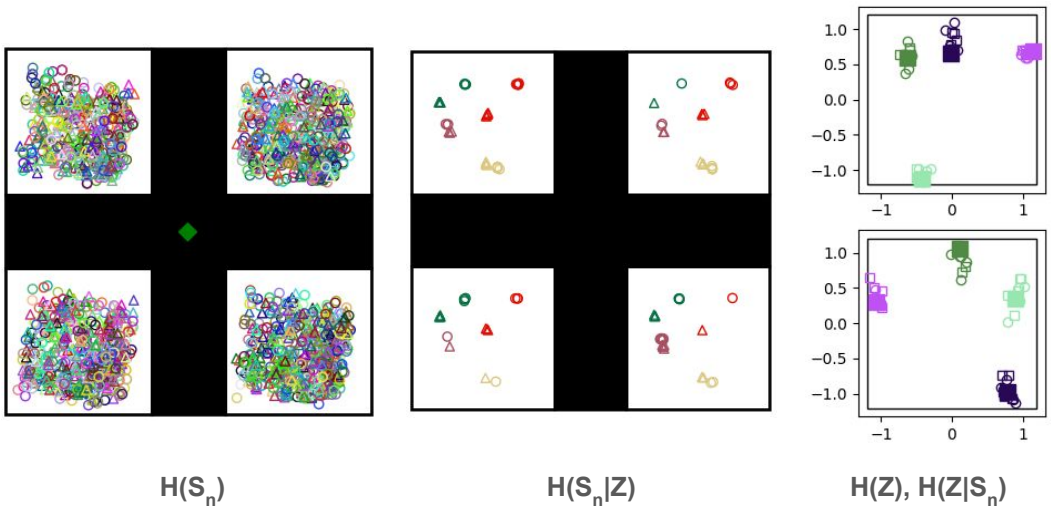

Figure 7: Entropy visualizations for the stochastic four rooms pick and place task. Left figure visualizes $H(S_n)$ by showing the skill-terminating states s_n from 1000 randomly sampled skills from the skill space. The skill-terminating states s_n uniformly cover much of the reachable state space as would be expected from a well-performing empowerment-based skill learning method. Middle figure visualizes $H(S_n|Z)$ by showing the skill-terminating states associated with 4 random selected goals. The s_n belonging to a specific goals are in the same color. Per figure, each skill targets an abstract (x, y) offset location for both the agent and object. The last figure visualizes both $H(Z)$ and $H(Z|S_n)$. The skill space in this task is 4-dimensional and is shown by two squares, which are the black outlined squares within the axes. The colored squares are randomly sampled skills z. The first two dimensions of the skill are shown by a square in the top part of the figure and the last two dimensions are shown by a square in the bottom part of the figure. The empty circles are samples from the variational posterior $q_\phi(z|z_n)$, in which the z_n are sampled from the abstraction dist $p_\alpha(z_n|s_n)$. As would be expected from a skillset with high mutual information, the samples from the posterior are near original skill. This figure also shows empty colored squares which are again samples from the variational posterior, but this time the z_n are sampled from the latent variable model $p_\eta(z_n|a)$. This shows our algorithm is working as expected as the KL divergence between p_η and p_α is low.

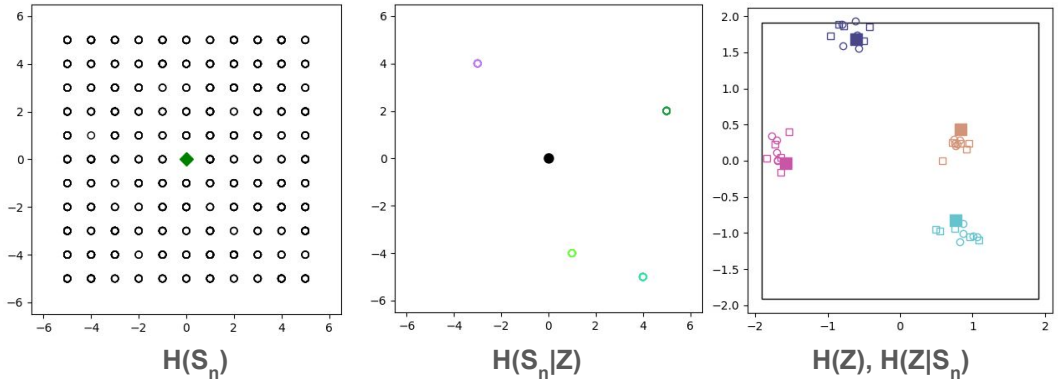

Figure 8: Entropy visualizations for the RGB QR code navigation task. Left figure visualizes $H(S_n)$ by showing the skill-terminating states s_n from 1000 randomly sampled skills from the skill space. Note that the state the agent actually receives is an 507-dimensional image, but we plot here the low-dimensional (x, y) location. The skill-terminating states s_n uniformly cover the reachable state space as would be expected from a well-performing empowerment-based skill learning method. Middle figure visualizes $H(S_n|Z)$ by showing the skill-terminating states associated with 4 random selected goals. The s_n belonging to a specific goals are in the same color. Per figure, each skill targets an abstract (x, y) location. The last figure visualizes both $H(Z)$ and $H(Z|S_n)$. The skill space in this task is a square and is shown by the black outlined square within the axes. The colored squares are randomly sampled skills z. The empty circles are samples from the variational posterior $q_\phi(z|z_n)$, in which the z_n were sampled from the abstraction dist $p_\alpha(z_n|s_n)$. As would be expected from a skillset with high mutual information, the samples from the posterior are near original skill. This figure also shows empty colored squares which are again samples from the variational posterior, but this time the z_n are sampled from the latent variable model $p_\eta(z_n|a)$. This shows our algorithm is working as expected as the KL divergence between p_η and p_α is low.

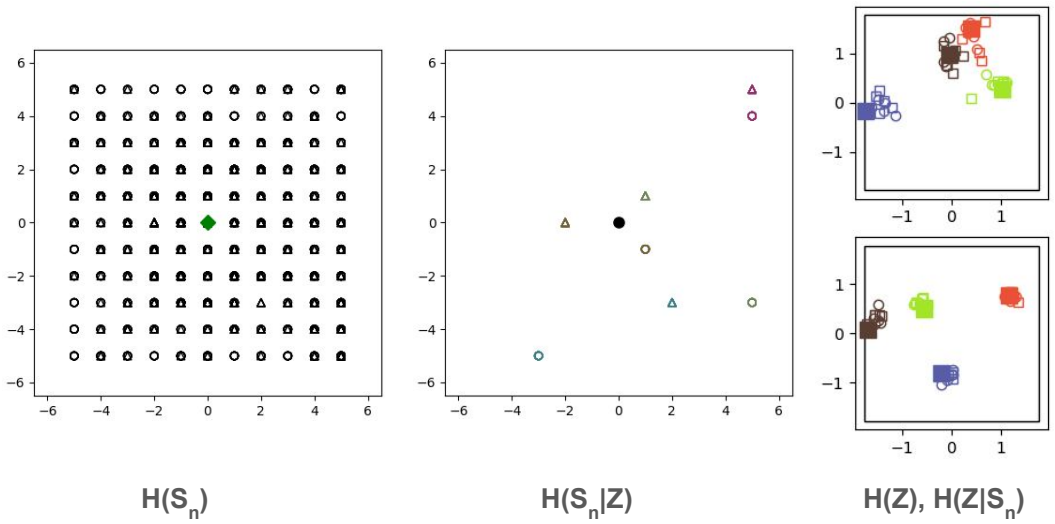

Figure 9: Entropy visualizations for the RGB QR code pick and place task. Left figure visualizes $H(S_n)$ by showing the skill-terminating states s_n from 1000 randomly sampled skills from the skill space. Note that the state the agent actually receives is an 507-dimensional image, but we plot here the low-dimensional (x, y) locations of the agent (circle) and object (triangle).The skill-terminating states s_n uniformly cover much of the reachable state space as would be expected from a well-performing empowerment-based skill learning method. Middle figure visualizes $H(S_n|Z)$ by showing the skill-terminating states associated with 4 random selected goals. The s_n belonging to a specific goals are in the same color. Per figure, each skill targets an abstract (x, y) offset location for both the agent and object. The last figure visualizes both $H(Z)$ and $H(Z|S_n)$. The skill space in this task is 4-dimensional and is shown by two squares, which are the black outlined squares within the axes. The colored squares are randomly sampled skills z. The first two dimensions of the skill are shown by a square in the top part of the figure and the last two dimensions are shown by a square in the bottom part of the figure. The empty circles are samples from the variational posterior $q_\phi(z|z_n)$, in which the z_n are sampled from the abstraction dist $p_\alpha(z_n|s_n)$. As would be expected from a skillset with high mutual information, the samples from the posterior are near original skill. This figure also shows empty colored squares which are again samples from the variational posterior, but this time the z_n are sampled from the latent variable model $p_\eta(z_n|a)$. This shows our algorithm is working as expected as the KL divergence between p_η and p_α is low.