Disentangled Unsupervised Skill Discovery for Efficient Hierarchical Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

A hallmark of intelligent agents is the ability to learn reusable skills purely from unsupervised interaction with the environment. However, existing unsupervised skill discovery methods often learn *entangled* skills where one skill variable simultaneously influences many entities in the environment, making downstream skill chaining extremely challenging. We propose **Disentangled Unsupervised Skill Discovery** (**DUSDi**), a method for learning *disentangled skills* that can be efficiently reused to solve downstream tasks. DUSDi decomposes skills into disentangled components, where each skill component only affects one factor of the state space. Importantly, these skill components can be **concurrently** composed to generate low-level actions, and efficiently chained to tackle downstream tasks through hierarchical Reinforcement Learning. DUSDi defines a novel mutual-information-based objective to enforce disentanglement between the influences of different skill components, and utilizes value factorization to optimize this objective efficiently. Evaluated in a set of challenging environments, DUSDi successfully learns disentangled skills, and significantly outperforms previous skill discovery methods when it comes to applying the learned skills to solve downstream tasks[1].

1 Introduction

Reinforcement learning (RL) algorithms have achieved many successes in complex tasks, from magnetic plasma control [10] to automobile racing [47]. However, applying existing RL algorithms to every new task in a *tabula rasa* manner often results in low sample efficiency that limits RL's broader applicability. Unsupervised skill discovery holds the promise of improving the sample efficiency of Reinforcement Learning, by learning a set of reusable skills through reward-free interaction with the environment that can be later recombined to tackle multiple downstream tasks more efficiently. In practice, prior unsupervised RL skills are represented as a policy that conditions on a skill variable to generate diverse behaviors, and have led to successful and efficient learning of downstream tasks when combined with skill fine-tuning or hierarchical RL skill selection [12, 22].

Despite prior successes, a common limitation of the skills learned by existing unsupervised RL methods is that they are *entangled*: any change in the skill variable causes the agent to induce changes in *multiple dimensions* of the state space simultaneously. Learning to use and recombine these *entangled* skills can be extremely hard for an agent trying to solve downstream tasks, especially in complex domains like multi-agent systems or household humanoid robots, where the agent needs to concurrently change multiple independent dimensions of the state to complete the task. For example, consider an agent learning to drive: if a single skill variable simultaneously changes the speed, steering, and headlights of the car, it will be extremely challenging for the agent to learn how to

[1]Website: https://sites.google.com/view/dusdi

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

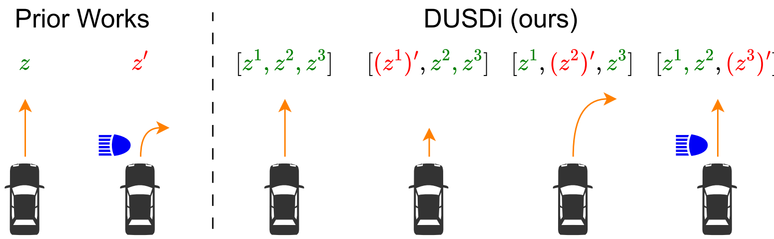


Figure 1: Consider an agent practicing driving skills by learning to control a car's speed (length of orange arrow), steering (curvature of orange arrow), and headlights (blue symbol), **(Left)** previous unsupervised skill discovery methods learn *entangled* skills, where a change in the skill variable can cause all three environment factors to change **(Right)** DUSDi learns *disentangled skills* with concurrent components, where each skill component only affects one factor of the state space, enabling efficient downstream task learning with hierarchical RL.

turn on/off the headlights while keeping the car at the right speed and direction. In contrast, humans naturally have the ability to concurrently and independently adjust the car's acceleration, steering, and headlights based on the car's current speed, surroundings, and lighting conditions. In other words, humans learn *disentangled* skill components where each component only affects one or few state variables, and can be easily recombined into *compositional skills*.

In this work, we aim to create such a mechanism for artificial agents to learn disentangled skills that facilitate solving downstream tasks. We introduce Disentangled Unsupervised Skill Discovery (DUSDi), a novel method for unsupervised discovery of disentangled skills. A key insight of DUSDi is to take advantage of state factorization that is naturally available in unsupervised RL environments [12, 32, 16] (e.g. speed, direction, and lighting conditions of the car in the driving example; the state of different objects in a household environment). These factored state spaces provide a natural inductive bias we leverage for disentanglement: DUSDi decomposes skills into disentangled components, and encourages each skill component to affect only one state factor while discouraging it from affecting any other factors. To that end, DUSDi designs a novel intrinsic reward based on mutual information (MI) between disentangled skills and state factors: the learning agent receives high reward for 1) increasing the MI between a state factor and the skill component assigned to change it, and 2) for decreasing the MI between that skill component and all other state factors.

DUSDi introduces a set of technical innovations to efficiently optimize the proposed mutual information objective. Once the DUSDi skills are learned, they can be used as the low-level policy in a hierarchical reinforcement learning (HRL) setting to tackle downstream tasks. Compared to using entangled skills, a key benefit of using the disentangled DUSDi skills is that they guarantee more efficient exploration during downstream task learning and therefore often lead to significantly better performance. Furthermore, the structured skill space of DUSDi opens up additional possibilities to inject domain knowledge into the learning process to further improve downstream task learning.

DUSDi is easy to implement and can be integrated into any MI-based unsupervised skill discovery approach. In our experiments, we integrate DUSDi with DIAYN [12] and evaluate the performance on four domains: a 2D agent navigation domain, a DMC walker domain, a large-scale multi-agent particle domain, and a 3D realistic simulated robotics domain. Our experiments indicate that DUSDi can indeed learn disentangled skills, and significantly outperforms other Unsupervised Reinforcement Learning methods on solving complex downstream tasks with HRL.

2 Preliminaries

Factored Markov Decision Process (f-MDP) In this work, we consider unsupervised skill discovery in a reward-free Factored Markov Decision Process. Following Osband and Van Roy [30], we define a Factored Markov Decision Process by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$, where $\mathcal{S} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^N$ is a factored state space with N factors such that each state $s \in \mathcal{S}$ consists of N state factors: $s = (s^1, \ldots, s^N), s^i \in \mathcal{S}^i$. $\mathcal{A}$ is the action space, and $\mathcal{P}$ is an unknown Markovian transition model, $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Notice that a factored state space is often naturally available in domains used by prior works [12, 22, 32, 16, 8] as it can naturally represent environments with separate elements (e.g., objects) that can be changed independently. In domains with only image-based (unfactored)

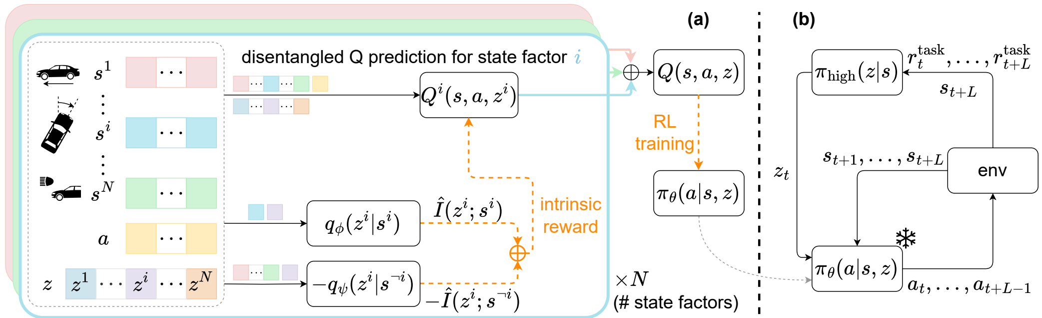


Figure 2: Two learning stages of DUSDi: **(a)** in *disentangled* skill learning stage, DUSDi creates a one-to-one mapping between state factors and skill components — each disentangled skill component z^i only influences state factor s^i. DUSDi designs a novel mutual-information-based intrinsic reward to enforce disentanglement and utilize Q-value decomposition to learn the skill policy π_θ efficiently. **(b)** in the task learning stage, the skill policy is used as a frozen low-level policy and a high-level policy π_{high} is learned to select skill z for every L steps, by maximizing the task reward r^{task}.

observations, a factored state space can be extracted using disentangled representation learning or object-centric representation learning methods [29, 18], which we empirically evaluated in Sec. 4.5.

Mutual-Information-Based Skill Discovery Mutual-information-based skill discovery methods, such as the paradigmatic DIAYN [12], specify the skills with a latent variable $z \in \mathcal{Z}$, and learns a skill-conditioned policy $\pi(a|s, z)$. The optimization objective these methods use to learn the skills is to maximize the mutual information (MI) between the state, s, and the skill latent variable, z: $I(\mathcal{S}; \mathcal{Z})$, which incentivizes the agent to reach diverse and distinguishable states. One popular way to determine the MI, $I(\mathcal{S}; \mathcal{Z})$, is to decompose it as $I(\mathcal{S}; \mathcal{Z}) = H(\mathcal{Z}) - H(\mathcal{Z} \mid \mathcal{S})$, where H denotes entropy. Since the skill variable is typically sampled from a fixed distribution, $H(\mathcal{Z})$ can be assumed constant: maximizing $I(\mathcal{S}; \mathcal{Z})$ is thus equivalent to minimizing $H(\mathcal{Z} \mid \mathcal{S})$. Following the definition of conditional entropy, $-H(\mathcal{Z} \mid \mathcal{S}) = \mathbb{E}_{s,z}[\log p(z|s)]$, DIAYN proposes to approximate $p(z|s)$ with a learned *discriminator* $q(z|s)$ that predicts the skill latent, z, given the state, s.

After discovering the skills, mutual-information-based methods apply them to learn downstream reward-supervised tasks. Many methods (e.g., DIAYN) adopt a hierarchical RL structure for this second phase, where the skill policy is used as a low-level "frozen" element, and a high-level policy $\pi_{\text{high}}(z|s)$ learns to sequentially activate skill z based on observations. The high-level policy is trained to maximize the provided task reward, $\mathcal{R}$, with $\mathcal{Z}$ as the action space.

3 Learning Disentangled Skills with DUSDi

DUSDi acts in two phases: in the first phase, DUSDi learns disentangled skills without external reward (Sec. 3.1). The key to DUSDi's success is to encourage disentanglement between different skill components through a novel learning objective that restricts the effect of each disentangled skill component to independent factors. In the second phase, DUSDi learns to perform downstream tasks with explicit reward supervision using a Hierarchical RL architecture, achieving higher returns than methods with entangled skills (Sec. 3.3). In practice, learning disentangled skills in environments with many factors can be challenging. To address this challenge, we introduce improvements to DUSDi's first phase based on Q-function decomposition (Sec. 3.2). We present the entire DUSDi pipeline in Fig. 2, and the pseudo-code in Appendix A.

3.1 Disentangled Skill Spaces and Learning Objective

DUSDi aims to create disentangled skill components that can be easily recombined to solve downstream tasks. To that end, DUSDi proposes a novel factorization of the latent skill conditioning variable, z, into N independent disentangled components such that the latent space $\mathcal{Z}$ becomes $\mathcal{Z} = \mathcal{Z}^1 \times \cdots \times \mathcal{Z}^N$. We equate N to the number of state factors and consider $z^i \in \mathcal{Z}^i$ the disentangled skill component that affects state factor i. The skill policy $\pi(a|s, z)$ takes in $z \in \mathcal{Z}$, which is a composition of the skill components.

While, in principle, the factored latent space could be discrete or continuous, without loss of generality we assume in this paper that the skill space is discrete, which also leads to more clarity in the presentation. We can then assume that each disentangled component z^i takes the form of an integer, $z^i \in [1, k]$, resulting in a compositional skill, z, with the form of a N-dimensional multi-categorical vector with k^N possible values. During skill training, we independently sample each disentangled component z^i from a fixed uniform distribution $p(z^i)$, similar to Eysenbach et al. [12].

Given this factored skill space, our goal is to learn a skill policy network, $\pi_\theta : \mathcal{S} \times \mathcal{Z} \mapsto \mathcal{A}$, such that each disentangled component $\mathcal{Z}^i$ affects and only affects the value of a state factor, $\mathcal{S}^i$. For each disentangled component and state factor pair $(\mathcal{Z}^i, \mathcal{S}^i)$, we encourage diverse and distinguishable behaviors by maximizing their mutual information $I(\mathcal{S}^i; \mathcal{Z}^i)$. While this objective enables a disentanglement skill component to affect the corresponding factor, it does not restrict the component from affecting other factors. This is undesirable since the resulting skill components would still be entangled in their effects. To prevent that, we propose to ensure that each skill component, $\mathcal{Z}^i$, minimally affects the rest of the state factors, $\mathcal{S}^{\neg i}$, where $\mathcal{S}^{\neg i}$ denotes the subspace formed by all other state factor spaces except $\mathcal{S}^i$: $\mathcal{S}^1 \times \ldots, \mathcal{S}^{i-1} \times \mathcal{S}^{i+1} \times \cdots \times \mathcal{S}^N$. Specifically, we incorporate an entanglement penalty to minimize, $I(\mathcal{S}^{\neg i}; \mathcal{Z}^i)$, which corresponds to the mutual information between a skill component and all other state factors that it should not affect.

Formally, the skill policy aims to maximize the following objective:

$$\mathcal{J}(\theta) = \sum_{i=1}^{N} I(\mathcal{S}^i; \mathcal{Z}^i) - \lambda I(\mathcal{S}^{\neg i}; \mathcal{Z}^i), \tag{1}$$

where $\lambda < 1$ is a hyperparameter that controls the importance of the entanglement penalty relative to the skill-factor association. We restrict λ to be smaller than one for the following reason: in some environments, due to intrinsic dynamical dependencies between state factors themselves, controlling a state factor, $\mathcal{S}^i$, has to introduce some association between $\mathcal{Z}^i$ and other factors in $\mathcal{S}^{\neg i}$, e.g., when controlling an object whose manipulation requires the agent to use other objects as tools. In these cases, as the policy learns to maximize the MI between a skill and a factor, $I(\mathcal{S}^i, \mathcal{Z}^i)$, the MI with other factors, $I(\mathcal{S}^{\neg i}; \mathcal{Z}^i)$, may also increase. For these cases, the use of $\lambda < 1$ will ensure that the entanglement penalty does not overpower the association reward, and the policy is still incentivized to learn disentangled skill components that change S^i distinguishably while introducing minimal changes on other factors. In practice, we simply set $\lambda = 0.1$ in all our experiments.

Optimizing DUSDi's Objective: Directly maximizing the objective in Eq. 1 is intractable. Alternatively, we propose to approximate the objective using a variational lower bound of the mutual information [1]:

$$I(\mathcal{S}^i; \mathcal{Z}^i) = H(\mathcal{Z}^i) - H(\mathcal{Z}^i|\mathcal{S}^i) \geq C + \mathbb{E}_{z,s} \log q^i_\phi(z^i|s^i), \tag{2}$$

where C represents the constant value of $H(\mathcal{Z}^i)$, the entropy of the prior distribution over the skill latent variable, which does not change during training, and q^i_ϕ is a variational distribution.

Similarly, we can approximate the MI in the entanglement penalty by:

$$I(\mathcal{S}^{\neg i}; \mathcal{Z}^i) \geq C + \mathbb{E}_{z,s} \log q^i_\psi(z^i|s^{\neg i}), \tag{3}$$

where q^i_ψ is another variational distribution. Importantly, when these q approximations perfectly recover the posterior distribution of z^i, we obtain equality in Eq. 2 and Eq. 3. We implement the variational distributions, q_ϕ and q_ψ, as neural network discriminators mapping input state factor(s) to the predicted disentangled component values, z^i.

To optimize $\mathcal{J}(\theta)$, we alternate between two steps: 1) performing variational inference to train the discriminators q^i_ϕ and q^i_ψ through gradient ascent, and 2) using q^i_ϕ and q^i_ψ to learn a disentangled skill policy π_θ through RL by maximizing the following intrinsic reward approximating Eq. 1:

$$r_z(s, a) \triangleq \sum_{i=1}^{N} q^i_\phi(z^i|s^i) - \lambda q^i_\psi(z^i|s^{\neg i}) \tag{4}$$

Notice that because of the negative sign in front of the entanglement penalty, $-q^i_\psi(z^i|s^{\neg i})$, we are no longer optimizing a variational lower bound on $\mathcal{J}(\theta)$. Despite that, we found empirically that our optimization procedure works well as an approximation for $\mathcal{J}(\theta)$, possibly because both q^i_ϕ and q^i_ψ quite accurately approximate the underlying simple categorical distribution.

Interestingly, the decomposed nature of our intrinsic reward allows a convenient avenue for shaping skill behaviors based on domain knowledge. In particular, we can restrict a state factor s^i to only take certain values by constraining $q^i_\phi(z^i|s^i)$ accordingly. While not the main focus of this work, we briefly explore this further optimization enabled by DUSDi in Appendix I.

3.2 Accelerating Skill Learning through Q Decomposition

When using reinforcement learning (RL) to optimize the intrinsic reward function defined in Eq. 4, standard RL algorithms treat the reward function as a black box and learn a single value function from the mixture of intrinsic reward terms. While this approach may be sufficient for environments with few state factors, doing so for complex environments with many state factors (large N) often leads to suboptimal solutions. A key reason is that the mixture of $2N$ reward terms leads inevitably to high variance in the reward, making the value of the Q function oscillate. Furthermore, the sum of reward terms obscures information about each term's value, which hinders credit assignment.

DUSDi overcomes this issue by leveraging the fact that the intrinsic reward function in Eq. 4 is a linear sum over terms associated with each disentangled component. Thanks to the linearity of expectation, we can decompose the Q function into N disentangled Q functions as follows:

$$Q_\pi(s, a, z) = \sum_{i=1}^{N} Q^i(s, a, z) \tag{5}$$

where Q^i represents each disentangled Q function, one for each disentangled component. We proof Eq. 5 in Appendix H. The disentangled Q functions can be then updated only with their corresponding intrinsic reward terms, $r^i \triangleq q^i_\phi(z^i|s^i) - \lambda q^i_\psi(z^i|s^{\neg i})$. During policy learning, we sum all disentangled Q functions together to recover the global critic, Q_π, as shown in Fig. 2 (a), top. Compared to learning Q_π directly from all $2N$ reward terms, learning disentangled Q functions significantly reduces reward variance, allowing Q_π to converge faster and more stably.

Notice that the decomposed Q function provides a convenient way to inject domain prior and improve training efficiency. This can be done by constraining each decomposed Q function to only attend to a (often small) subset of the state and skill factors that matter, which we leave for future work.

3.3 Downstream Task Learning

Similar to Eysenbach et al. [12], in DUSDi we utilize hierarchical RL to solve reward-supervised downstream tasks with the discovered skills, as depicted in Fig.2 (b). The skill policy, $\pi_\theta : \mathcal{S} \times \mathcal{Z} \to \mathcal{A}$, acts as the low-level policy and is kept constant while a high-level policy, $\pi_{\text{high}} : \mathcal{S} \to \mathcal{Z}$, learns to select which skill to execute for L steps using the skill latent variable, z. Thus, the skill latent conditioning space, $\mathcal{Z}$, acts as the action space of the high-level policy, π_{high}. As extensively evaluated in our experiments, without any additional "ingredient", performing downstream task learning in the action space formed by DUSDi skills often results in significantly superior performance compared to an action space formed by entangled skills. We show that the superior performance of DUSDi can be explained by **more efficient exploration** when using the DUSDi skills for hierarchical RL, which we elaborate on in Appendix B.

Depending on the nature of the downstream tasks, we can often take further advantage of the disentangled skills learned by DUSDi through leveraging its structure. One such scenario is when the downstream task has a composite reward function consisting of multiple terms. Previous works [15, 42] have shown that when the causal dependencies from action dimensions to reward terms are available (e.g., the reward for speed only depends on actions that affect speed), one can use Causal Policy Gradient (CPG) to decompose the policy update (e.g., only the "speed actions" get updated by the speed reward) and greatly improve sample efficiency, especially when the dependencies are sparse. In downstream task learning, with an action space (of the high-level policy) consisting of the skills learned by DUSDi, we have a convenient way of applying causal policy gradient, where the causal dependencies between the action dimensions (i.e., skill components) and reward terms are often sparse and can be easily obtained by examining the state factor that a skill component is associated with, which we evaluate empirically in Sec. 4.6.

Table 1: Evaluation of skill disentanglement based on the DCI metric, shown as mean and standard deviation across skill policies trained with 3 random seeds.

	2D Gunner		Multi-Particle		iGibson	
	DUSDi (ours)	DIAYN-MC	DUSDi (ours)	DIAYN-MC	DUSDi (ours)	DIAYN-MC
Disentanglement (↑)	**0.864** $\pm$ 0.018	0.016 $\pm$ 0.002	**0.705** $\pm$ 0.037	0.002 $\pm$ 0.000	**0.833** $\pm$ 0.022	0.017 $\pm$ 0.006
Completeness (↑)	**0.864** $\pm$ 0.017	0.024 $\pm$ 0.004	**0.750** $\pm$ 0.041	0.003 $\pm$ 0.000	**0.834** $\pm$ 0.021	0.019 $\pm$ 0.005
Informativeness (↑)	**0.897** $\pm$ 0.012	0.821 $\pm$ 0.010	**0.849** $\pm$ 0.052	0.791 $\pm$ 0.032	**0.854** $\pm$ 0.006	0.752 $\pm$ 0.015

4 Experimental Evaluation

In the evaluation of DUSDi, we aim to answer the following questions: **Q1**: Are skills learned by DUSDi truly disentangled (Sec. 4.2)? **Q2**: Can Q-decomposition improve skill learning efficiency (Sec. 4.3)? **Q3**: Do our disentangled skills perform better when solving downstream tasks compared to other unsupervised reinforcement learning methods (Sec. 4.4)? **Q4**: Can DUSDi be extended to image observation environments (Sec.4.5)? **Q5**: Can we leverage the structured skill space of DUSDi to further improve downstream task learning efficiency (Sec.4.6)?

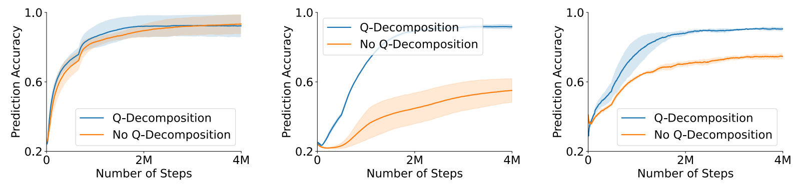
4.1 Evaluation Environments

Previous works [12, 31, 32, 40, 21] extensively rely on standard RL environments such as DMC [43] and OpenAI Fetch [3] to evaluate unsupervised RL methods. However, unlike previous unsupervised skill discovery methods, DUSDi focuses on learning a set of disentangled skill components that can be concurrently executed and re-combined to complete downstream tasks. As such, it only makes sense to examine the performance of DUSDi in challenging tasks that require concurrent control of many environment entities (e.g. multi-agent systems, complex household robots). Previous environments lack this property: in DMC for example, while the state and action space can be very complex, the predominant downstream tasks are just to move the center-of-mass of the agent to different places. In such cases, there is no need for concurrent skill components, and therefore we do not expect large gains from using DUSDi's disentangled skills. Nevertheless, we include an evaluation on the **DMC-Walker** [43] environment to demonstrate that our method is also applicable to those environments, but focus the majority of our evaluation on environments that DUSDi is designed for, including **2D Gunner**, **Multi-Particle** [28], and **iGibson** [24].

The 2D gunner is a relatively simple domain, where a point agent can navigate inside a continuous 2D plane, collecting ammo and shooting at targets. Multi-Particle is a multi-agent domain modified based on [28]. In this domain, a centralized controller simultaneously controls 10 heterogenous point-mass agents to interact with 10 stations, where each agent can only interact with a specific station. We evaluate in this domain to test the scalability of our methods to a large number of state factors. iGibson [24] is a realistic simulated robotics domain, where a mobile manipulator can navigate in a room, inspect the room using its head camera, and interact with electric appliances in the room by pointing a remote control to them and switching them on/off. We evaluate in this domain to examine whether our method can handle home-like environments with complex dynamics. We provide visualizations and additional information about each of the environments in Appendix C.

4.2 Evaluating Skill Disentanglement

First, we examine whether the skills learned by DUSDi are truly disentangled (**Q1**) using the DCI metric proposed by Eastwood and Williams [11]. The DCI metric consists of three terms, namely **disentanglement**, **completeness**, and **informativeness**, explained in detail in Appendix F. We compare against **DIAYN-MC** (Multi-channel DIAYN) that uses the same skill representation as DUSDi but optimizes the DIAYN objective of $I(\mathcal{S}; \mathcal{Z})$, and show results in Table 1. Unsurprisingly, DUSDi significantly outperforms DIAYN-MC, especially on Disentanglement and Completeness, across all three environments. These results indicate that DUSDi learns truly disentangled skills, enabling efficient downstream task learning, as we will show in Sec. 4.4. Qualitatively, we showcase some of the learned skills in `https://sites.google.com/view/dusdi`.



(a) 2D Gunner (b) Multi-Particle (c) iGibson

Figure 3: Evaluation of the effect of Q-decomposition in skill learning. The plots depict the mean and standard deviation of accuracy (↑) when predicting the skill component z^i based on the state factor s^i, computed across 3 training processes. The higher prediction accuracy indicates that the policy learns to control more state factors in more distinguishable ways, leading to more efficient downstream task learning.

4.3 Evaluating Skill Learning Efficiency with Q-decomposition

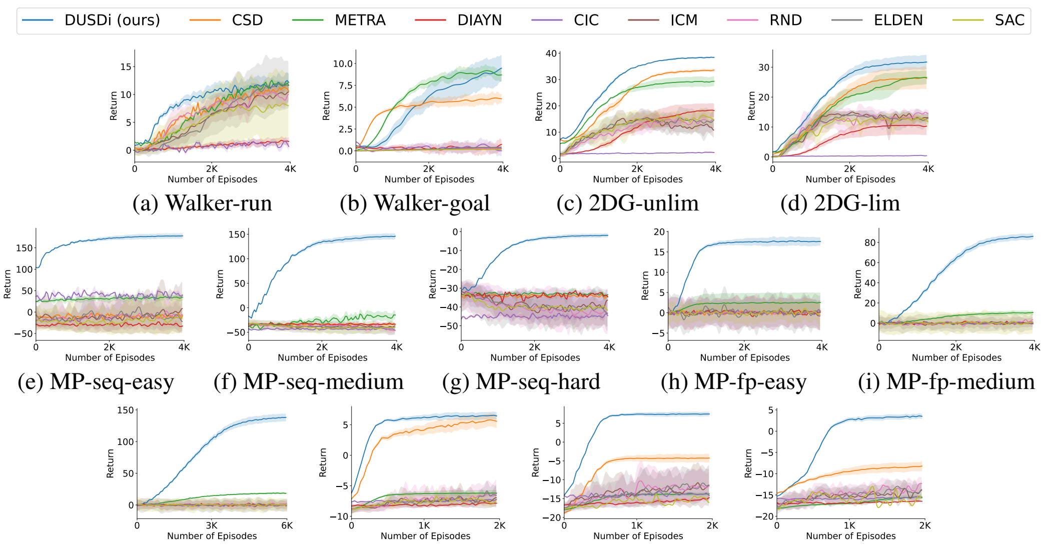
To examine the importance of Q-decomposition (**Q2**), we measure the performance of optimizing the DUSDi objective during skill learning with and without a decomposed Q network. We compare the classification accuracy of the skill discriminators $q^i_\phi(z^i|s^i)$, averaged over all skill channels, which indicates progress towards discovering diverse and distinguishable skills, with higher accuracy being better. We depict our results in Fig. 3. We observe that Q-decomposition has a similar performance to the regular Q network in the simplest 2D gunner domain, but significantly outperforms the regular Q network in domains with more state factors (Multi-Particle) and more complex dynamics (iGibson), suggesting that Q-decomposition is necessary for scaling towards complex domains.

4.4 Evaluating Downstream Task Learning

The promise of DUSDi is to incorporate disentanglement into skills so that the skills can be effectively used in downstream task learning. Therefore, the most critical evaluation of our work focuses on comparing the performance of different unsupervised RL methods on task learning (**Q3**). We compare against existing state-of-the-art unsupervised reinforcment learning algorithms, including **DIAYN** [12], **CIC** [22], **CSD** [32], **METRA** [33], **ICM** [34], **RND** [4], **ELDEN** [16], and **Vanilla RL** [13], where these baselines are further explained in Appendix E.

Similar to the evaluation setting in the URLB benchmark [21], we allow each method to train for 4 million steps without access to reward (i.e., pretraining phase) before the reward is revealed to the agent and the downstream learning takes place. During the pre-training phase, all methods use soft actor-critic (SAC) [13] to optimize the intrinsic reward. For all skill discovery methods (i.e., DUSDi, DIAYN, CIC, CSD, METRA), a skill-conditioned policy, $\pi_\theta(a|s,z)$, is learned during the pretraining phase. During downstream learning, the skill network is fixed, whereas an upper policy, $\pi_{\text{high}}(z|s)$, is trained using proximal policy optimization (PPO) [39] to optimize the task reward. Similar to previous works [12, 40], we omit proprioceptive states from the MI optimization for all skill discovery methods. For exploration methods (i.e., RND, ICM, ELDEN), a policy $\pi_\theta(a|s)$ is learned during the pretraining phase on intrinsic reward and fine-tuned using the task reward during the downstream learning phase. The hyperparameters are specified in Appendix G.

We evaluate all methods in four environments and 13 downstream tasks, detailed in Appendix D. The results are depicted in Fig. 4. As expected, DUSDi performs similarly to previous unsupervised RL methods in the DMC walker environment due to the simplicity in terms of its downstream objectives (all related to center-of-mass locomotion), but significantly outperforms all previous methods on domains where downstream tasks require coordinative control of multiple state factors. The most crucial comparison is between DUSDi and DIAYN. DIAYN is a special case of DUSDi where there is only one state factor (consisting of the entire state) and one skill component. Therefore comparing against DIAYN offers a straightforward examination of the effect of disentangled skills for downstream task learning. DUSDi significantly outperforms DIAYN in all downstream tasks, demonstrating the effectiveness of using disentangled skills. In general, we found exploration-based methods to be less capable than skill discovery methods, possibly due to their lack of temporal abstraction. CIC performs very poorly, likely because the CIC objective does not explicitly encourage distinguishable skills and instead generates the intrinsic reward solely based on state entropy, making it very hard for the upper policy to select the right skill. This result again shows the importance of having a proper skill representation. DUSDi also outperforms CSD and METRA on most downstream tasks, especially on the more complex and high-dimensional domains, like Multi-Particle. This superiority is



(a) Walker-run (b) Walker-goal (c) 2DG-unlim (d) 2DG-lim

(e) MP-seq-easy (f) MP-seq-medium (g) MP-seq-hard (h) MP-fp-easy (i) MP-fp-medium

(j) MP-fp-hard (k) IG-look (l) IG-housekeep (m) IG-inspect

Figure 4: Training curves of DUSDi and baselines on multiple downstream tasks (reward supervised second phase). The plots depict the mean and standard deviation of the return of each method over 3 random seeds. DUSDi outperforms all baselines that learn entangled skills, converging faster and to higher returns.

perhaps surprising considering that in our experiments, DUSDi only relies on the simple DIAYN-style intrinsic reward for skill discovery, but further demonstrates the importance of learning a disentangled skill space. It is important to notice that many techniques proposed to improve skill discovery quality (e.g., Baumli et al. [2], Zhao et al. [50]), can be seamlessly incorporated into DUSDi. Therefore, we expect our method to perform even better as new advances are made in unsupervised skill discovery.

4.5 Extending DUSDi to Image Space

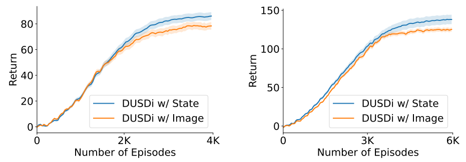
Although this paper primarily focuses on applying DUSDi to factored state space, we can straightforwardly extend it to image space through existing works in factored / object-centric representation learning [27, 18, 46, 26, 48] (**Q4**). We empirically illustrate this capability in the Multi-Particle environment, where we replace the low-dimensional state observation with 64×64 image observations. Specifically, we first pretrain an object-centric encoder following Yang et al. [48], and then use our method on top of the extracted representation to learn disentangled skills. Hence, essentially, the skill policy uses images as observation. As shown in Fig. 5, when learning from image observation, DUSDi achieves similar performance to learning from state space, whereas the baseline methods are unable to learn these two tasks even when learning from the low-dimensional state space as in Fig. 4.

4.6 Leveraging Structure of DUSDi Skills

While DUSDi can already learn downstream tasks quite efficiently, it is possible to further improve the sample efficiency of downstream task learning through leveraging the structured skill space of DUSDi (**Q5**), as described in the second paragraph of Sec.3.3. Specifically, we apply Causal Policy Gradient [15] to the Multi-Particle domain, where the causal dependencies between state factors and reward terms are easy to identify. We present our results in Fig. 6, where the sample efficiency of downstream task learning is greatly improved thanks to the structured skill space of DUSDi.

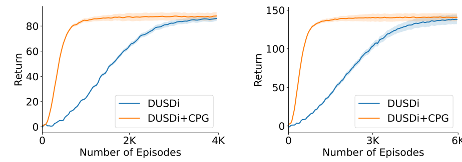
5 Related Work

Unsupervised Skill Discovery In unsupervised skill discovery, the goal of an agent is to learn task-agnostic skills without external rewards. To learn such skills, previous methods propose various forms of intrinsic reward: (1) maximizing the mutual information between visited states and the skill variables [12, 40, 5, 22], (2) maximizing the traveled distance along the direction specified by the skill variables [31–33], (3) learning to reach a diverse set of goals [45, 37, 35]. These skills can be



(a) MP-fp-medium (b) MP-fp-hard

Figure 5: Performance of DUSDi with image observations on two multi-particle downstream tasks over three random seeds. With the help of disentangled representation learning, DUSDi effectively learns skills based only on image observations and leverages the skills to solve challenging downstream tasks where baseline methods fail.



(a) MP-fp-medium (b) MP-fp-hard

Figure 6: Performance of DUSDi in two multi-particle downstream tasks when combined with Causal Policy Gradient (CPG, orange). The disentangled skills of DUSDi provide opportunities for leverage structure and speed up downstream task learning, greatly improving the sample efficiency when learning downstream tasks.

used to boost the sample efficiency of downstream task learning, for example, (1) using hierarchical RL where a high-level policy learns to select which skill to execute [12], or (2) using the skill policy to initialize the task solving policy and then fine-tuning it [22].

State Space Factorization in RL In RL, there is a long history of leveraging state factorization, including learning a world model between state factors for planning [20, 44], augmenting data [36], and providing intrinsic rewards [38, 16]. Relevant to our work are skill discovery methods that learn to either reach a goal for each controllable object [17, 8] or achieve interactions between a pair of specified objects [7]. Though these methods achieve disentanglement by influencing one or a pair of objects during a skill, they do not apply to tasks that require controlling multiple objects simultaneously, like driving where we need to control the car's speed and heading directions at the same time. In contrast, our method can combine disentangled skill components into concurrent skills [9] to solve a wide range of tasks.

Disentanglement in Skill Learning There are a few works investigating disentanglement in unsupervised skill discovery. Lee et al. [23] consider a special case of disentangled skills — for a multi-arm robot, learning independent skills for each arm. However, they rely on manually factored action spaces which is an assumption that often limits the behavior of the agent. Kim et al. [19] encourage the disentanglement between different dimensions of the skill variable by regularizing it with β-VAE objective [14], but Locatello et al. [25] point out such regularization is impossible to achieve disentanglement. To learn disentangled skills, Song et al. [41] learns a decoder from skill variables to state trajectories and their generation factors, which is then used to train the skill policy through imitation learning. However, their training of the decoder requires pre-collected trajectories and corresponding generation factors, whereas our method is fully unsupervised with no expert data.

6 Conclusion

We present DUSDi, an unsupervised skill discovery method for learning disentangled skills by leveraging the factorization of the state space. DUSDi designs a skill space that exploits the factorization of the state space and learns a skill-conditioned policy where each sub-skill affects only one state factor. DUSDi enforces disentanglement through an intrinsic reward based on mutual information, and shows superior performance on a set of downstream tasks with naturally factored state spaces compared to baselines and state-of-the-art unsupervised RL methods.

One limitation of DUSDi is the assumption of access to a factored state space. While a factored state space is naturally available in many existing RL environments, and can be extracted from images as we have shown in our experiment (Sec. 4.5), we believe that future advances in disentangled representation learning will greatly broaden the applicability of DUSDi. Secondly, DUSDi primarily focuses on learning a structured skill space for more efficient downstream learning, and its exploration capability during skill learning is largely determined by the specific algorithm used to optimize for our mutual information objective. While we used DIAYN [12] in this work due to its simplicity, it would be interesting to examine extending the idea of learning disentangled skills to other skill discovery methods, e.g., Zhao et al. [50], Laskin et al. [22], including those that are not based on mutual information [32, 49].

References

[1] David Barber and Felix Agakov. Information maximization in noisy channels: A variational approach. *Advances in Neural Information Processing Systems*, 16, 2003.

[2] Kate Baumli, David Warde-Farley, Steven Hansen, and Volodymyr Mnih. Relative variational intrinsic control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6732–6740, 2021.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[4] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018.

[5] Víctor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pages 1317–1327. PMLR, 2020.

[6] Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational empowerment as representation learning for goal-based reinforcement learning. *arXiv preprint arXiv:2106.01404*, 2021.

[7] Jongwook Choi, Sungtae Lee, Xinyu Wang, Sungryull Sohn, and Honglak Lee. Unsupervised object interaction learning with counterfactual dynamics models. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[8] Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger-causal hierarchical skill discovery. *arXiv preprint arXiv:2306.09509*, 2023.

[9] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey, 2022.

[10] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419, 2022.

[11] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*, 2018.

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

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

[15] Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Causal policy gradient for whole-body mobile manipulation. *arXiv preprint arXiv:2305.04866*, 2023.

[16] Jiaheng Hu, Zizhao Wang, Peter Stone, and Roberto Martin-Martin. Elden: Exploration via local dependencies. *arXiv preprint arXiv:2310.08702*, 2023.

[17] Xing Hu, Rui Zhang, Ke Tang, Jiaming Guo, Qi Yi, Ruizhi Chen, Zidong Du, Ling Li, Qi Guo, Yunji Chen, et al. Causality-driven hierarchical structure discovery for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20064–20076, 2022.

[18] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

[19] Jaekyeom Kim, Seohong Park, and Gunhee Kim. Unsupervised skill discovery with bottleneck option learning. *arXiv preprint arXiv:2106.14305*, 2021.

[20] Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[21] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark, 2021.

[22] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

[23] Youngwoon Lee, Jingyun Yang, and Joseph J Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International conference on learning representations*, 2019.

[24] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks, 2021.

[25] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[26] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

[27] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020.

[28] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

[29] Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. *arXiv preprint arXiv:2306.00600*, 2023.

[30] Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. *Advances in Neural Information Processing Systems*, 27, 2014.

[31] Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. *arXiv preprint arXiv:2202.00914*, 2022.

[32] Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. *arXiv preprint arXiv:2302.05103*, 2023.

[33] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

[34] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[35] Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 7750–7761. PMLR, 2020.

[36] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

[37] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

[38] Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious exploration via structured world models yields zero-shot object manipulation. *Advances in Neural Information Processing Systems*, 35:24170–24183, 2022.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[40] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

[41] Wonil Song, Sangryul Jeon, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. Learning disentangled skills for hierarchical reinforcement learning through trajectory autoencoder with weak labels. *Expert Systems with Applications*, page 120625, 2023.

[42] Thomas Spooner, Nelson Vadori, and Sumitra Ganesh. Factored policy gradients: Leveraging structure for efficient learning in momdps, 2021.

[43] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.

[44] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. *arXiv preprint arXiv:2206.13452*, 2022.

[45] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

[46] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

[47] Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896): 223–228, 2022.

[48] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9593–9602, 2021.

[49] Rushuai Yang, Chenjia Bai, Hongyi Guo, Siyuan Li, Bin Zhao, Zhen Wang, Peng Liu, and Xuelong Li. Behavior contrastive learning for unsupervised skill discovery, 2023.

[50] Rui Zhao, Yang Gao, Pieter Abbeel, Volker Tresp, and Wei Xu. Mutual information state intrinsic control. *arXiv preprint arXiv:2103.08107*, 2021.

A Pseudo-code for DUSDi Skill Learning

Algorithm 1 DUSDi Skill Learning

```
Initialize skill policy $\pi_\theta$, discriminators $q^i_\phi$, $q^i_\psi$ and value function $Q^i$ for each state factor $\mathcal{S}^i$.
for each skill training episode do
    Sample skill $z \sim p(z)$.
    Collect state transitions with actions from $\pi_\theta(a|s, z)$.
    Sample a batch of $(s, a, z)$ from the replay buffer.
    for $i = 1, \dots, N$ do
        Update $q^i_\phi(z^i|s^i)$ and $q^i_\psi(z^i|s^{\neg i})$ with discrimination losses.
        Update $Q^i(s, a, z)$ with reward $r^i$ using SAC.
    end for
    Update $\pi_\theta$ with $Q = \sum_{i=1}^N Q^i$ using SAC.
end for
```

B Entangled vs. disentangled components for Policy Learning

Compared to entangled skills, the advantages of using disentangled components mainly reside in an easier exploration in the skill space. For skill spaces of equivalent capacity, the DIAYN latent skill variable is a *single* integer between 1 and k^N, and the DUSDi skill variable is a N-dimensional vector with each dimension representing a disentangled component with k possible values. In this section, we analyze the benefits and search complexity of DUSDi's space over DIAYN's for two main cases: when there are no dynamical dependencies between state factors (optimal case for disentangled components) and where there are intrinsic dependencies between state factors.

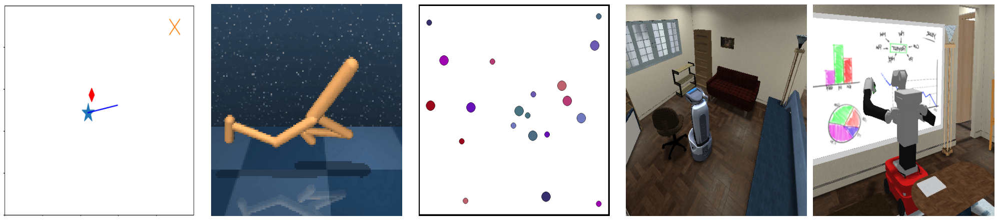
State Factors without Dynamical Dependencies: In this case, for DIAYN to find the correct skill to execute at the current time step, in the worst case, it needs to iterate through all skills, resulting in 1-step exploration sample-efficiency of $O(k^N)$. In contrast, for DUSDi, as disentangled components are independent of each other, with one skill trial, the agent can simultaneously observe the effects of setting each disentangled component as $\mathcal{Z}^i = z^i$. Hence, for an intelligent agent, to understand the effects of each disentangled component at the current state, it only needs to sweep through each disentangled component space with k trials (e.g., setting all disentangled components $\mathcal{Z}^i = 1, \dots, k$). After that, as the effects of each disentangled component are independent, by compositing disentangled components in novel ways, the agent has the ability to imagine the effects of all skills, leading to $O(k)$ exploration efficiency.

State Factors with Dynamical Dependencies: When there are dynamical dependencies, we denote PA^i as *parent* indices of state factors that $\mathcal{S}^i$ depends on, e.g., when moving a mouse ($\mathcal{S}^i$), $\mathcal{S}^{\text{PA}^i}$ denotes the hand. In such cases, the effect of $\mathcal{Z}^i$ is conditioned on the value of $\mathcal{Z}^{\text{PA}^i}$, and we need to iterate through all $(\mathcal{Z}^i, \mathcal{Z}^{\text{PA}^i})$ pairs to observe all possible influences on S^i. As a result, the exploration is constrained by the state factor with the largest number of parents. Denoting $|\text{PA}^i|$ as the number of parent factors for $\mathcal{S}^i$, the exploration sample-efficiency is $O(k^{1+\max_i |\text{PA}^i|})$. We can see that the $O(k)$ efficiency when there is no dynamical dependencies is a special case of $\max_i |\text{PA}^i| = 0$. Despite lower efficiency than $O(k)$, in many environments, the dynamics of each state factor only depend on a small number of other factors, i.e., $\max_i |\text{PA}^i| \ll N$. Hence, exploration with disentangled components is still more sample-efficient than using entangled skills.

C Environment Details

We test DUSDi on four environments, where a visualization of each of the environments is presented in Fig. 7.

2D Gunner: Shown in Fig. 7 (a), the blue star marks the position of the agent, the blue line marks its shooting direction, the red diamond marks ammo location, and the orange cross marks the target



(a) 2D Gunner (b) DMC Walker (c) Multi-Particle (d) iGibson

Figure 7: Environments Visualization

position. The agent has a 7-dimensional observation space, consisting of 3 state factors: [Agent Position, Ammo State, Target State]. The action is 5-dimensional, 2 for agent movement, 2 for ammo pickup, and 1 for shooting direction.

DMC-Walker: Shown in Fig. 7 (b), a 6 degree-of-freedom robot can locomote on a 2D plane through joint motions. The agent has a 26-dimensional observation space consisting of 3 state factors: [Body Position, Body Velocity, Robot Proprioception].

Multi-Particle: Shown in Fig. 7 (c), the agents are marked by small circles, while the stations are marked by large circles. Only stations and agents of the same color can interact with each other. The Multi-Particle environment has a 70-dimensional observation space, consisting of 20 state factors. The state factors include states for each landmark and states for each agent. The action space is 50-dimensional, with 5 dimensions per agent that control their motions and interactions with the landmarks.

iGibson: Shown in Fig. 7 (d), iGibson has 42-dimensional observation space consisting of 4 state factors, including [Agent Location, Electric Appliances State, Object(s) in View, Robot Proprioception]. The action space is 11-dimensional, consisting of base velocity (2D), head motion (2D), arm motion (6D), and gripper motion (1D).

D Downstream Tasks

DMC-Walker (Walker):

- **Run**: In this task, the walker agent is rewarded for moving forward at a particular velocity.
- **Goal Reaching**: In this downstream task, the agent has to reach randomly generated goal positions.

2D Gunner (2DG):

- **Unlimited Ammo (unlim)**: In this downstream task, a set of targets will randomly appear, where the agent needs to navigate to a position close to the target and shoot them in order to score. The ammo is unlimited so the agent does not need to worry about picking up ammo.
- **Limited Ammo (lim)**: This downstream task is different from the "unlimited ammo" in that the agent starts with no ammo and needs to pick up ammo in order to shoot. Everything else is identical.

Multi-Particle (MP):

- **Sequential interaction (seq)** (easy, medium, hard): In this task, agents need to sequentially interact with their corresponding station following an instruction sequence given at the start of each episode. Interacting with stations in the wrong order will be penalized. The easy version of this task has a sequence length of 2, while medium and hard have a sequence length of 5 and 8 respectively.
- **Food-poison (fp)** (easy, medium, hard): In this downstream task, each station will offer either food or poison to the corresponding agent. Each agent needs to decide whether to interact with its corresponding station based on a sequence of binary indicators provided to the agents. The difficulty level has the same meaning as in the sequential interaction task.

iGibson (IG):

- **Look around**: In this task, the robot needs to look at objects in the room sequentially.
- **Appliances inspection**: In this task, the robot needs to navigate to different electric appliances, and test whether each of them is working correctly by pointing a remote control towards it.

- **Housekeeping**: In this task, the robot needs to manage the electric appliances intelligently. Specifically, the robot needs to first look at a screen to receive instructions. Depending on the instruction, the robot needs to turn on / off certain electric appliances using the remote control.

E Baseline Methods

During downstream task evaluation, we compared against the following state-of-the-art unsupervised RL methods:

- **DIAYN** [12] represents skill variable z as an integer between 1 to k^N and learns skills by maximizing $I(\mathcal{S}; \mathcal{Z})$, the MI between $\mathcal{Z}$ and all state factors $\mathcal{S}$.
- **CIC** [22] learns a state representation with contrastive learning and learns skills by maximizing transition entropy in the representation space.
- **CSD** [32] learns skills maximizing distance traveled along the direction of z in the state space, where distance is measured in a controllability-aware manner.
- **METRA** [33] learn a set of behaviors that collectively cover as much of the state space as possible through optimizing a Wasserstein variant of the state-skill Mutual Information.
- **ICM** [34]: encourages visiting novel states by using prediction errors of action consequences as intrinsic rewards.
- **RND** [4] encourages visiting novel states by using prediction errors of features computed from a randomly initialized network as intrinsic rewards.
- **ELDEN** [16] operates in a factored state space similar to our approach, and encourages visiting states that induce novel factor dependencies.
- **SAC** [13] where no pretraining is used, and vanilla RL is directly applied to tackle the downstream tasks.

F Evaluating Skill Disentanglement Details

The DCI metric consists of three terms, namely **disentanglement**, **completeness**, and **informativeness**. In the context of this work, disentanglement (↑) measures, on average, to what extent each skill component only affects a single state factor. Completeness score (↑) measures, on average, to what extent each state factor is only influenced by a single skill component. Informativeness score (↑) measures the repeatability of learned skills: given the skill z, how accurately we can predict which states will be visited. We refer the reader to the work by Eastwood and Williams [11] for a detailed discussion of these metrics and how they are calculated.

In the original work, measuring DCI requires knowing the ground truth generative factors. In our case, the generative factors are simply the state factors, and we only need to discretize the value of each state factor to make it compatible for evaluation. For each method on each domain, we collect $100K$ rollout steps using the learned skill policy, $\pi(s, z)$, where the skill is (re)sampled from the uniform prior distribution, $p(z)$, every 50 steps. These (state, skill) pairs are then used to calculate DCI.

G Hyperparameters

Skill Dimensions: For all skill learning methods with discrete skills (i.e. DUSDi, DIAYN), we make sure that they have equivalent capacity. Specifically, for igibson and 2D gunner, each DUSDi skill consists of 3 skill components, each component with 5 possible values. As a result, DIAYN skill is an integer between 1 to 125 in these two domains. The only exception is Multi-Particle, where DUSDi has ten sub-skills, each with 5 possible values. Since skill as an integer between 1 and $5^{10} = 9765625$ is obviously challenging for DIAYN to converge, we set the number of discrete skills to be 4096 for DIAYN. For continuous skills (i.e. CSD, CIC, METRA), we follow the skill dimensions specified in the original papers (64D for CIC, 3D for CSD and METRA), which were shown to be effective for the respective methods.

Skill Learning Parameters: All skill learning methods in our baselines use SAC to optimize for the intrinsic reward, with the same policy and value network architecture. DUSDi applies additional decomposition and masking to the value networks, as described in Section. 3.2, which is not applicable to the baseline methods. Due to Q-decomposition, when using the same value network architecture, DUSDi's value network capacity is N times of the capacity of other methods' value networks (including when comparing the variations of DUSDi, i.e., no decomposition). For a fair comparison, we also tried to increase value network capacity for other methods to match the capacity for DUSDi, but found that their skill/task learning performances do not improve significantly. This suggests (1) that, for skill learning, reward variance, rather than network capacity, is the key reason for no Q-composition variation of DUSDi to converge slowly, and (2) that, for task learning, disentangled skills, rather than network capacity, is what make DUSDi significantly outperform baselines.

We present the hyperparameters for SAC in Table. 2. All methods use a low-level step size of $L = 50$.

Table 2: Hyperparameters of Skill Learning.

	Name	**Value**
SAC	optimizer	Adam
	activation functions	ReLu
	learning rate	1×10^{-4}
	batch size	1024
	critic target τ	0.01
	MLP size	[1024, 1024]
	steps per update	2
	# of environments	4
	Temperature α	0.02
	log std bounds	[-10, 2]

Downstream Hierarhical Learning: For all skill discovery methods, downstream learning of the skill selection policy is implemented with PPO. We used the same hyperparameters for all methods across all tasks, as specified in Table. 3.

Table 3: Hyperparameters of Downstream Learning.

	Name	**Value**
PPO	optimizer	Adam
	activation functions	Tanh
	learning rate	1×10^{-4}
	batch size	32
	clip ratio	0.1
	MLP size	[128, 128]
	GAE λ	0.98
	target steps	250
	n steps	20
	# of environments	4
	# of low-level steps L	50

Downstream Finetuning: For all non-skill discovery methods, downstream learning is done using the same hyperparameters as pretraining (table. 2), replacing the intrinsic reward with the task reward.

H Proof of Q Decomposition

Proof.

$$\begin{aligned}
Q_\pi(s,a,z) &= \mathbb{E}_\theta[\sum_{t=0}^{\infty}\gamma^t r_t] \\
&= \mathbb{E}_\theta[\sum_{t=0}^{\infty}\gamma^t \sum_{i=1}^{N} q_\phi^i(z^i|s^i) - \lambda q_\psi^i(z^i|s^{\neg i})] \\
&= \sum_{i=1}^{N}\mathbb{E}_\theta[\sum_{t=0}^{\infty}\gamma^t (q_\phi^i(z^i|s^i) - \lambda q_\psi^i(z^i|s^{\neg i}))] \\
&= \sum_{i=1}^{N} Q^i(s,a,z)
\end{aligned}$$

□

I Behavior Restriction of Skills via Domain Knowledge

Due to the decomposable nature of the intrinsic reward of DUSDi, we can conveniently restrict the behavior of skills by constraining the skill predictor $q_\phi^i(z^i|s^i)$ for a particular state factor i. For example, if we want s^i to stay within a certain range, we can set $q_\phi^i(z^i|s^i)$ to be a uniform distribution for all s^i not within this range, effectively discouraging the agent from going out of range. In the extreme case, we can fully specify the mapping between z^i and s^i, essentially resulting in performing goal-conditioned RL for state i (as pointed out in [6]) while performing DUSDi for the rest of the state factors.

We qualitatively examine this idea in the iGibson domain. By restricting a mobile manipulator to only locomote in regions that are close to a whiteboard, our robot successfully learns diverse board-wiping behaviors which are otherwise extremely hard to learn. Visualizations of the learned skills can be seen at `https://sites.google.com/view/dusdi`.