# THE INFORMATION GEOMETRY OF UNSUPERVISED REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

How can a reinforcement learning (RL) agent prepare to solve downstream tasks if those tasks are not known a priori? One approach is unsupervised skill discovery, a class of algorithms that learn a set of policies without access to a reward function. Such algorithms bear a close resemblance to representation learning algorithms (e.g., contrastive learning) in supervised learning, in that both are pretraining algorithms that maximize some approximation to a mutual information objective. While prior work has shown that the set of skills learned by such methods can accelerate downstream RL tasks, prior work offers little analysis into whether these skill learning algorithms are optimal, or even what notion of optimality would be appropriate to apply to them. In this work, we show that unsupervised skill discovery algorithms based on mutual information maximization do not learn skills that are optimal for every possible reward function. However, we show that the distribution over skills provides an optimal initialization minimizing regret against adversarially-chosen reward functions, assuming a certain type of adaptation procedure. Our analysis also provides a geometric perspective on these skill learning methods.

## 1 INTRODUCTION

The high sample complexity of reinforcement learning (RL) algorithms has prompted a large body of prior work to study pretraining of RL agents. During the pretraining stage, the agent collects *unsupervised* experience from the environment that is not labeled with any rewards. Prior methods have used this pretraining stage to learn representations of the environment that might assist the learning of downstream tasks. For example, some methods learn representations of the observations [31, 51] or representations of the dynamics model [11, 21, 52]. In this work, we focus on methods that learn a set of potentially-useful policies, often known as *skills* [1, 12, 19, 40, 50]. That is, the learned representation corresponds to a reparametrization of policies. The aim of this unsupervised pretraining is to learn skills that, when a reward function is given, can quickly be combined or composed to maximize this reward. Prior work has demonstrated that this general approach does accelerate learning downstream RL [1, 12, 19]. However, prior work offers little analysis about when and where such methods are provably effective. Even simple questions, such as what it means for a set of skills to be optimal, remain unanswered.

Algorithms for unsupervised skill learning are conceptually related to the representation learning methods used to improve supervised learning [4, 20, 25, 27, 43, 59]. Both typically maximize a lower bound on mutual information, and the learned representations are often combined linearly to solve downstream tasks [27, 43]. However, whereas prior work in supervised learning has provided thorough analysis of when and where these representation learning methods produce useful features [30, 36, 55], there has been comparatively little analysis into when unsupervised skill learning methods produce skills that are useful for solving downstream RL tasks.

In this paper, we analyze when and where existing skill learning methods based on mutual information maximization are (or are not) optimal for preparing to solve unknown, downstream tasks. On the one hand, we show that the skills learned by these methods are not complete, in that they cannot be used to represent the solution to every RL problem. This result implies that using the learned skills for hierarchical RL may result in suboptimal performance, and suggests new opportunities for better skill learning algorithms. One the other hand, we show that existing methods acquire the

optimal initialization for learning downstream tasks, under some assumptions about how adaptation is performed. To the best of our knowledge, this is the first result showing that unsupervised skill learning methods are optimal in any sense.

Our analysis also illuminates a number of properties of these methods. For example, we show that every skill is optimal for some reward function, and we provide a nontrivial upper bound on the number of unique skills learned. This result implies that these methods cannot learn an infinite number of unique skills, and instead will learn duplicate copies of some skills. The key to our analysis is to view RL algorithms and skill learning algorithms as geometric operations (see Fig. 1). Points correspond to distributions over states, and the set of all possible distributions is a convex polytope that lies on a probability simplex. We show that all reward-maximizing policies lie at vertices of this polytope and that maximizing mutual information corresponds to solving a facility assignment problem on the simplex.

The main contribution of this paper is a proof that skill learning algorithms based on mutual information are optimal for minimizing regret against unknown reward functions, assuming that adaptation is performed using a certain procedure. Our proof of optimality relies on certain problem assumptions, leaving the door open for future skill learning algorithms to perform better under different problem assumptions. This contribution provides a rigorous notion of what it means for an unsupervised RL algorithm to be optimal, and also answers additional questions about unsupervised skill learning algorithms, such as whether the skills correspond to reward-maximizing policies and how many unique skills will be learned.

## 2 Prior Work

In the *unsupervised RL* setting, an agent interacts with the environment without access to a reward function, with the aim of learning some representation of the environment that will assist in learning downstream tasks. Prior work has proposed many approaches for this problem, including learning a dynamics model [11, 21, 52], learning compact representations of observations [31, 51], performing goal-conditioned RL without hand-specified rewards [7, 51], doing pure exploration [24, 34, 40, 44, 50, 53] or learning collections of skills [1, 6, 12, 19, 22, 54, 58]. These prior methods are similar in that they all learn *representations*, with different methods learning representation of observations, dynamics, or policies. While each approach works well in some setting (see Laskin et al. [32] for a recent benchmark), the precise connection between these pretraining methods and success on downstream tasks remains unclear. We will focus on unsupervised skill learning methods.

The problem of unsupervised pretraining for RL has also been studied in the RL theory community [2, 38, 39], where it is referred to as the *reward-free* setting. The algorithms proposed in these prior works use a version of goal-conditioned RL for pretraining. However, the aim of these methods is different from ours: these methods aim to collect a sufficient breadth of data, whereas we focus on learning good representations.

Our analysis uses the dual of the RL problem, which is written in terms of the policy's state distribution [45, Eq. 6.9.2]. In contrast to prior work that proposes algorithms for solving the dual problem [41, 42, 57], we use the dual problem purely as an analytical tool. Our paper is similar to prior work that uses a geometric perspective to analyze RL algorithms [5, 10]. Whereas those prior works study the geometry of value functions and representations, ours will study the geometry of policies.

## 3 Preliminaries

We focus on infinite-horizon MDPs with discrete states $\mathcal{S}$ and actions $\mathcal{A}$, initial state distribution $p_0(\mathbf{s_0})$, dynamics $p(\mathbf{s_{t+1}} \mid \mathbf{s_t}, \mathbf{a_t})$, and discount factor $\gamma \in [0, 1]$. We assume a Markovian policy $\pi(\mathbf{a} \mid \mathbf{s})$, and define the discounted state occupancy measure of this policy as $\rho^\pi(\mathbf{s}) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P_t^\pi(\mathbf{s})$, where $P_t^\pi(\mathbf{s})$ is the probability that policy $\pi$ visits state $\mathbf{s}$ at time $t$. The environment dynamics dictate which state marginal distributions are feasible. The RL objective can be expressed using the reward function $r(\mathbf{s})$ and the state marginal distribution:

$$(1-\gamma)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s_t})\right] = \mathbb{E}_{\rho^\pi(\mathbf{s})}\left[r(\mathbf{s})\right].$$

The factor of $1-\gamma$ accounts for the fact that the sum of discount factors $1+\gamma+\gamma^2+\cdots=\frac{1}{1-\gamma}$. Without loss of generality, we focus on *state-dependent reward functions*; action-dependent reward functions can be handled by modifying the state to include the previous action. While our analysis will ignore function approximation error, it applies to any MDP, including MDPs where observations correspond to features (e.g., activations of a frozen neural network) rather than original states.

### 3.1 Unsupervised Skill Learning Algorithms

Prior skill learning algorithms learn a policy $\pi_\theta(\mathbf{a} \mid \mathbf{s}, \mathbf{z}_{\text{input}})$ with parameters $\theta$ and conditioned on an additional input $\mathbf{z}_{\text{input}} \sim p(\mathbf{z}_{\text{input}})$. Let $\rho^{\pi_\theta}(\mathbf{s} \mid \mathbf{z}_{\text{input}})$ denote the state marginal distribution of policy $\pi_\theta(\mathbf{a} \mid \mathbf{s}, \mathbf{z}_{\text{input}})$. We will focus on a simple method, DIAYN [12], which maximizes the mutual information between the representation $\mathbf{z}_{\text{input}}$ and the states $\mathbf{s}$ visited by policy $\pi_\theta(\mathbf{a} \mid \mathbf{s}, \mathbf{z}_{\text{input}})$:

$$\max_{\theta, p(\mathbf{z}_{\text{input}})} I(\mathbf{s}; \mathbf{z}_{\text{input}}) \triangleq \mathbb{E}_{p(\mathbf{z}_{\text{input}})} \underbrace{\mathbb{E}_{\rho^{\pi_\theta}(\mathbf{s}|\mathbf{z}_{\text{input}})}[\log \rho^{\pi_\theta}(\mathbf{s} \mid \mathbf{z}_{\text{input}}) - \log \rho^{\pi_\theta}(\mathbf{s})]}_{D_{\mathrm{KL}}(\rho^{\pi_\theta}(\mathbf{s}|\mathbf{z}_{\text{input}}) \,\|\, \rho(\mathbf{s}))}. \tag{1}$$

Because these methods have two learnable components, $\theta$ and $p(\mathbf{z}_{\text{input}})$, analyzing them directly can be challenging. Instead, we will use a simplified abstract model of skill learning where both the policy parameters $\theta$ and the latent variable $\mathbf{z}_{\text{input}}$ are part of a single representation, $\mathbf{z} = (\theta, \mathbf{z}_\theta)$. Then, we can define a skill learning algorithm as learning a single distribution $p(\mathbf{z})$:

$$\max_{p(\mathbf{z})} I(\mathbf{s}; \mathbf{z}) \triangleq \mathbb{E}_{p(\mathbf{z})} \underbrace{\mathbb{E}_{\rho(\mathbf{s}|\mathbf{z})}[\log \rho(\mathbf{s} \mid \mathbf{z}) - \log \rho(\mathbf{s})]}_{D_{\mathrm{KL}}(\rho(\mathbf{s}|\mathbf{z}) \,\|\, \rho(\mathbf{s}))}. \tag{2}$$

We can recover Eq. 1 by selecting representation distributions that factor as $p(\mathbf{z} = (\theta, \mathbf{z}_{\text{input}}) = \delta(\theta) p(\mathbf{z}_{\text{input}})$. Whereas skills learning algorithms are often viewed as optimizing individual skills, this simplified abstract model lets us think about these algorithms as assigning different probabilities to different policies. It will turn out that these algorithms assign a probability of 0 to almost every skill. We will refer to the small number of policies that are given non-zero probability as the "skills". Our main result will be that the state distribution averaged over the average skills, $\rho(\mathbf{s}) = \mathbb{E}_{p(\mathbf{z})}[\rho(\mathbf{s} \mid \mathbf{z})]$, is a good initialization for solving downstream tasks using a particular adaptation procedure.

### 3.2 Adapting to New Reward Functions

To study whether one pretraining algorithms is better than another, we must describe how the agent will adapt when a reward function is provided. An agent can employ many different methods to adapt to a reward function. For example, the agent may finetune a policy model using gradient descent, or it may select one among a set of pretrained policies. Below, we describe one particular adaptation procedure.

In the adaptation step, we assume that the agent is given a reward function $r(\mathbf{s})$ and an initialization, which is expressed as distribution over states $\rho(\mathbf{s})$. It is this initialization that we learn during pretraining. We will focus on the following objective for adaptation, corresponds to maximizing reward (i.e., minimizing regret) plus a regularization term:

$$\textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) \triangleq \min_{\rho^*(\mathbf{s}) \in \mathcal{C}} \underbrace{\max_{\rho^+(\mathbf{s}) \in \mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \mathbb{E}_{\rho^*(\mathbf{s})}[r(\mathbf{s})]}_{\text{regret}} + D_{\mathrm{KL}}(\rho^*(\mathbf{s}) \,\|\, \rho(\mathbf{s})), \tag{3}$$

where $\mathcal{C}$ corresponds to the (convex) state marginal polytope. The regularization term instantiates our intuition that adaptation should happen quickly, so the policy should not change its state distribution too much. The regularization can also be motivated as preventing overfitting to a limited number of samples of a noisy reward function.

Solving this adaptation objective is equivalent to performing one step of natural gradient descent [3, 29, 35], where the underlying metric is the state marginal distribution [47, Theorem 1]. This natural gradient is different from the standard natural policy gradient [29], which does not depend on the environment dynamics and is much easier to estimate. This equivalence suggests that our notion of adaptation is idealized: it assumes that this adaptation step can be performed exactly, without considering how data is collected, how state marginal distributions are estimated, or how the underlying policy parameters should be updated. Nonetheless, this paper is (to the best of our knowledge) the first to provide any formal connection between the mutual information objective optimized by unsupervised skill learning algorithms and performance on downstream tasks.

**Finding the optimal initialization.** In the unsupervised pretraining step, we attempt to find the initialization that will be best for the adaptation step. While there are many different ways an agent may perform this unsupervised pretraining, our analysis will focus on unsupervised skill learning. If the reward function were known, then the optimal initialization would simply be the optimal policy for that reward function. We assume that the reward function is not known, and further assume that we do not have any prior information about the reward function (e.g., a prior over reward functions). We therefore employ a worst-case approach, assuming that an adversary will select the reward function to make the adaptation step as difficult as possible. We can express the pretraining objective as follows:

Table 1: Notation for analysis.

| | |
|---|---|
| $r(\mathbf{s})$ | Reward function. |
| $\rho(\mathbf{s})$ | Initialization. |
| $\rho^+(\mathbf{s})$ | Optimal state marginal distribution for reward $r(\mathbf{s})$. Used to define regret. |
| $\rho^*(\mathbf{s})$ | Optimal state marginal distribution after adaptation, which maximizes reward and minimizes information cost. |

$$\min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})). \tag{4}$$

## 4 Overview of Main Results

Our key technical contribution is an analysis of when and where unsupervised RL algorithms learn optimal skills. Our main result is that maximizing mutual information is equivalent to learning an optimal initialization for solving the hardest downstream tasks.

**Theorem 4.1.** *The optimal initialization for adapting to an unknown reward function is the average state marginal given by maximizing mutual information (Eq. 2):*

$$\min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) = \max_{p(\mathbf{z})} I(\mathbf{s};\mathbf{z}).$$

Moreover, there is a close relationship between the solutions to these two optimization problems. After finding the distribution over policy representations that maximizes mutual information, $p^*(\mathbf{z})$, the state distribution that minimizes the worst-case cost of adaptation can be written as $\rho^*(\mathbf{s}) = \mathbb{E}_{p^*(\mathbf{z})}[\rho(\mathbf{s} \mid \mathbf{z})]$. This average state distribution of the skills is not hard to compute: it corresponds to sampling one of the skills $\mathbf{z} \sim p^*(\mathbf{z})$ and then execute the corresponding policy, $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$. To the best of our knowledge, this is the first result relating unsupervised skill learning algorithms to their performance on downstream tasks.

We then show that such methods, even when learning a very large number of skills, will not in general acquire all possible optimal policies. The number of unique skills learned by these unsupervised skill learning algorithms is upper bounded by the number of states (Lemma 6.3). Because the number of policies that are optimal for RL tasks can be much larger than the number of states, skills learned by maximizing mutual information are not sufficient for representing the solution to every RL problem (Lemma 6.4). Therefore, these skill learning algorithms are not optimal in the sense that the learned skills are not sufficient for solving every task. In short, our analysis shows that the *set of skills* are not optimal for *representing* optimal policies, but the average state distribution of the skills provides an optimal initialization for *learning* optimal policies using a particular adaptation procedure.

## 5 RL as Geometry on a Simplex

In order to prove our main result and further analyze the kinds of policies learned by unsupervised skill learning methods, we will first introduce a geometric perspective on RL. This perspective will allow us to better understand how unsupervised skill learning relates to the space of potentially optimal policies (for state-dependent reward functions).

We visualize policies by their discounted state occupancy measure, $|\mathcal{S}|$-dimensional point lying on the probability simplex (see Fig. 1). Note that this is a distribution over states, not actions, so a policy that chooses actions uniformly will not necessarily lie at the center of the simplex. Of course, not all state marginals are feasible, as the MDP dynamics might preclude certain distributions. We will be interested in where skill learning algorithms place skills on this probability simplex, and whether these skills are optimal for learning downstream tasks. Note that not all points on the simplex correspond to realizable policies, and multiple policies could map to the same state occupancy measure.

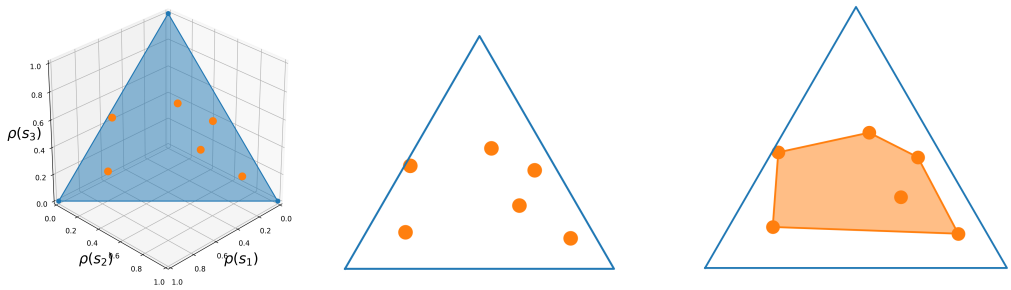



Figure 1: **Polytope of achievable policies**: *(Left)* For a 3-state MDP, the set of feasible state marginal distributions is described by a triangle $[(1,0,0),(0,1,0),(0,0,1)]$ in $\mathbb{R}^3$. We plot the state occupancy measure of 6 different policies. *(Center)* We plot those same policies, but remove the coordinate axes to improve visual clarity. *(Right)* For any convex combination of valid state occupancy measures, there exists a Markovian policy that has this state occupancy measure. Thus, policy search happens inside a convex polytope.

### 5.1 Which State Marginals are Achievable?

Understanding which state marginals are achievable is important for analyzing what unsupervised skill learning algorithms do, and how the learned skills relate to downstream *state-dependent* reward functions. While the set of achievable state marginal distributions can be described by a set of linear equations [45, Eq. 6.9.2] (see Appendix A.1), we offer a different and complementary description based on the following property [60, Theorem 2.8]:

**Proposition 1.** *Given a set of feasible state marginals, any convex combination is also feasible.*

This property is illustrated in Fig. 1 (right): if we plot the state marginals for six policies, then any point inside the convex hull of those state marginals is also feasible. That is, there exists a (Markovian) policy that achieves that state marginal distribution. In general, we can obtain the set of *all* feasible state marginals by taking the convex hull of the state marginals of all deterministic policies. We will refer to this set of achievable state marginals as a *polytope*. Every vertex of the state marginal polytope will *contain* a deterministic policy,[1] and the state marginal of every policy can be represented as a convex combination of the state marginals at the vertices.

### 5.2 Which Policies are Optimal?

We now analyze where reward-maximizing policies lie within this state marginal polytope. Again taking a geometric perspective, we visualize reward functions as vectors starting from the origin, so that the expected return objective can be expressed as the inner product between the state marginal distribution and the reward vector (see Fig. 2). This visualization helps illustrate two facts about the state marginal distributions of reward-maximizing policies:

**Fact 1.** *For every state-dependent reward function, among the set of policies that maximize that reward function is one that lies at a vertex of the state marginal polytope.*

**Fact 2.** *For every vertex of the state marginal polytope, there exists a reward function for which that vertex corresponds to a reward-maximizing policy.*

Fact 1 is an application of the maximum principle [48, Chpt. 32]; Fact 2 is an application of the strong separation theorem [26, Theorem 3.2.2]. These observations suggest

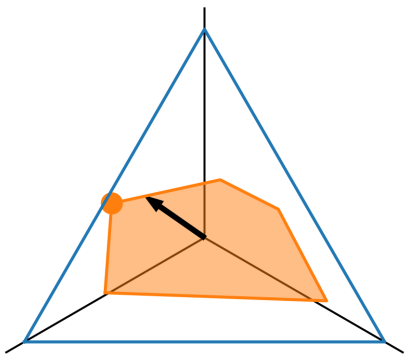
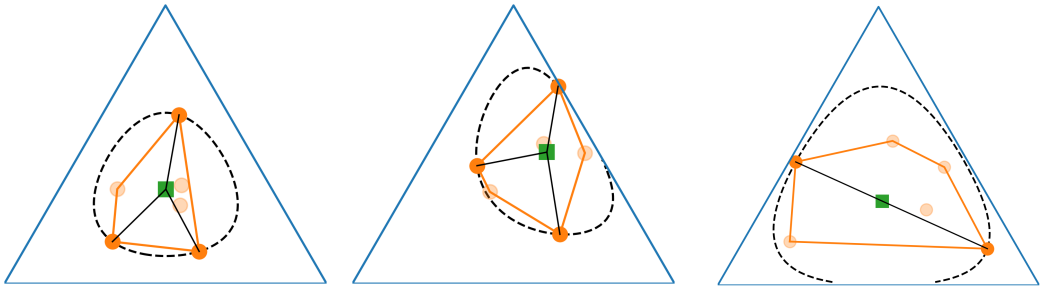


Figure 2: **Maximizing Rewards:** We visualize reward functions as vectors starting at the origin. Maximizing expected return corresponds to maximizing the dot product between the state marginal distribution and this reward vector.

[1] Note, however, that some deterministic policies may not lie at vertices of this polytope.



(a) Examples where DIAYN recovers $3 = |\mathcal{S}|$ skills.

(b) Example where DIAYN learns only two unique skills.

Figure 3: **Information Geometry of Skill Learning**: Maximizing mutual information is equivalent to minimizing the *maximum* divergence between a prior distribution over states $p(\mathbf{s})$ (green square) and any achievable state marginal distribution (Lemma 6.5). The opaque circles are skills discovered by DIAYN, and the dashed line denotes state marginals that are equidistant from the green square.

that, for the purpose of RL, it is sufficient to search among policies whose state marginals lie at vertices.[2] These observations also suggests an objective for unsupervised skill learning:

**Definition 5.1** (Vertex discovery problem)**.** Given a controlled Markov process (i.e., an MDP without a reward function), find the smallest set of policies such that every vertex of the state marginal polytope contains at least one policy.

While we are unaware of any tractable algorithm for exactly solving this problem, the next section discusses how a prior skill learning algorithm provides an efficient *approximation* to this problem.

## 6 The Geometry of Skill Learning

We now analyze unsupervised skill discovery algorithms using the tools discussed in the previous section. We view the skills learned by these methods as a set of points on the state marginal polytope. We will show that the average state distribution of the skills provides an optimal initialization for solving downstream tasks using a certain adaptation procedure. However, we then show that the skills do not cover every vertex, and are thus insufficient for representing the solution to every downstream RL task. Before presenting these results, we introduce a classic result from information geometry, which is the keystone to our analysis.

### 6.1 Where do Skills Learned by Mutual Information Fall on the Polytope?

We analyze where the skills learned by DIAYN lie on the state marginal polytope by dissecting the mutual information objective. The mutual information can be written as the average divergence between each policy's state distribution $\rho(\mathbf{s} \mid \mathbf{z})$ and the average state distribution, $\rho(\mathbf{s})$:

$$I(\mathbf{s}; \mathbf{z}) = \mathbb{E}_{p(\mathbf{z})} \left[ D_{\mathrm{KL}}(\rho(\mathbf{s} \mid \mathbf{z}) \parallel \rho(\mathbf{s})) \right].$$

To optimize this mutual information, a skill learning algorithm would choose to assign higher probability to some policies and lower probability (perhaps zero) to other policies. In particular, this mutual information suggests that skill learning algorithms should assign higher probabilities to policies whose state distributions are further from the average state distribution. In fact, if one policy has a state distribution that is further to the average state distribution than another policy, then the skill learning algorithm should assign a probability of zero to this second policy. Thus, the net result of optimizing mutual information is to place probability only on policies that are as far away from their average as possible:

**Lemma 6.1.** *The DIAYN skills must lie at vertices of the state marginal polytope, and thus are optimal for some reward functions.*

[2] While it is well known that any reward function has a deterministic optimal policy [45, Sec. 4.4], our visualization adds nuance to the story, suggesting that searching in the space of *all* deterministic policies can be wasteful, at least in the case of rewards that depend only on state.

This result answers the question of where the skills learned by mutual information fall on the polytope. Recalling that the vertices of the state marginal polytope correspond to optimal policies (Fact 1), this result implies that the skills learned by DIAYN are a subset of the set of optimal policies. Thus,

**Corollary 6.1.1.** *DIAYN skills are all optimal for some downstream state-dependent reward functions.*

### 6.2 How Many Unique Skills are Learned?

A simple way to use skills acquired during pretraining to solve a downstream RL task is to simply search among the skills to find the one that achieves the highest reward. So, it is natural to ask whether skill learning algorithms learn skills that are optimal for every possible reward function. Indeed, this is a very natural notion of what it might mean for a skill learning algorithm to be optimal. Prior work motivates the use of larger dimensional $\mathbf{z}$ or continuous-valued $\mathbf{z}$, arguing that these design decisions will allow the method to learn a larger (perhaps infinite) number of skills [12, 23], based on the intuition that these skills may be optimal for solving downstream tasks. However, it has remained unclear whether such design decisions are useful and whether the goal of acquiring an infinite or exponential number of skills is even possible. Among prior skill learning methods, we are unaware of prior work that characterizes the *number* of skills that these methods will learn. In this section we show that existing skill learning algorithms based on mutual information are not optimal in this sense. Our proof will follow a simple counting argument based on the number of unique skills.

To count the number of unique skills, we start by analyzing the distance between each skill and the state distribution averaged over skills, measured as a KL divergence between state marginals. We can use the KKT conditions to prove that all skills with non-zero support (i.e., for which $p(\mathbf{z}) > 0$) are equidistant[3] from the average state marginal:

**Lemma 6.2.** *Let $\rho(\mathbf{s} \mid \mathbf{z})$ be given. Let $p(\mathbf{z})$ be the solution to maximizing mutual information (Eq. 2) and let $\rho(\mathbf{s}) = \mathbb{E}_{p(\mathbf{z})}[\rho(\mathbf{s} \mid z)]$ be the average state marginal. Then the following holds:*

$$p(\mathbf{z}) > 0 \implies D_{\mathrm{KL}}(\rho(\mathbf{s} \mid \mathbf{z}) \| \rho(\mathbf{s})) = \max_{\mathbf{z}^*} D_{\mathrm{KL}}(\rho(\mathbf{s} \mid \mathbf{z}^*) \| \rho(\mathbf{s})) \triangleq d_{max}.$$

We visualize this result in Fig. 3b: the dashed line denotes a state marginals $\rho(\mathbf{s} \mid \mathbf{z})$ that are equidistant from the average state marginal (green square). Note that all skills where DIAYN places non-zero probability mass (denoted by solid orange dots instead of transparent orange dots) lie on this dashed circle. This Lemma allows us to count how many unique skills DIAYN acquires, under a certain regularity condition.

**Lemma 6.3.** *Assume that the vertices of the state marginal polytope are not concyclic, as defined using the KL divergence. Then DIAYN will recover at most $|\mathcal{S}|$ distinct skills.*

*Proof.* The proof follows a simple counting argument. Without loss of generality, we only consider vertices of the state marginal polytope as candidate skills (Lemma 6.1). The location of the average state marginal ($\rho(\mathbf{s})$) on the probability simplex is determined by the location of the skills and the distance to the skills. The average state marginal is a vector in $|\mathcal{S}|$-dimensional space with $|\mathcal{S}| - 1$ degrees of freedom; we subtract one to account for the constraint that the state marginal sum to 1.

Consider adding skills one by one. When we add a new skill (i.e., set $p(\mathbf{z}) > 0$ for some $\mathbf{z}$), we add an additional constraint to the location of this average state marginal; namely, that $D_{\mathrm{KL}}(\rho(\mathbf{s} \mid \mathbf{z}) \parallel p(\mathbf{z})) = d_{\max}$. The value of $d_{\max}$ adds one additional degree of freedom. Thus, after adding $(|\mathcal{S}| - 1) + 1$ skills, the average state marginal is fully specified. Adding additional skills would make the average state marginal ill-defined, as it would have to violate one of the distance constraints. Thus, we can have at most $|\mathcal{S}|$ unique skills. □

DIAYN need not learn exactly $|\mathcal{S}|$ skills. In some MDPs, such as the one illustrated in Fig. 3b, DIAYN acquires only 2 unique skills for a 3-state MDP. If an additional skill (at a vertex) were added, then the average distance from the skills to the average state marginal would decrease, worsening the mutual information objective. Additionally, when the regularity assumption is violated, there could be more than $|\mathcal{S}|$ skills that all have a KL divergence of $d_{\max}$ from the initialization $\rho(s)$, in which case DIAYN could learn more than $|\mathcal{S}|$ skills.

[3] While the KL divergence is not a true "distance" metric, the following result is precise.

To summarize, our analysis shows that DIAYN only acquires optimal policies (i.e., those at vertices of the state marginal polytope. However, the fact that there can be up to $|\mathcal{A}|^{|\mathcal{S}|}$ vertices (see Appendix A.6) and the fact that DIAYN acquires at most $|\mathcal{S}|$ skills leads us to conclude that DIAYN does not solve the vertex discovery problem:

**Theorem 6.4.** *In general, skill learning algorithms that maximize the mutual information $I(\mathbf{s}; \mathbf{z})$ will not recover the set of all optimal policies.*

This result is important for determining how to use the output of an unsupervised skill learning algorithm to solve downstream tasks. One approach to adaptation is to find whichever skill receives the highest return. This result indicates that this approach will produce suboptimal performance on a large number of reward functions. In fact, for every reward maximizing policy which a skill learning algorithm can represent, there can be an exponential number that the skill learning algorithm can fail to represent. This result casts doubt on the use of the skills themselves for solving downstream tasks, at least those skills learned by the existing generation of skill learning algorithms.

Might other simple algorithms for skill discovery recover the entire set of optimal policies? We are currently unaware of any algorithm that does solve the vertex discovery problem. In Appendix A.4, we show that *(1)* including actions in the mutual information objective or *(2)* maximizing a collection of indicator reward functions also fail to discover all optimal policies.

Unsupervised skill learning algorithms learn a full distribution over skills. Instead of just looking at the set of policies that are given non-zero probability (i.e., skills), do the numerical values of these probabilities give us useful information to help solve downstream tasks? Our next section shows that this distribution over skills is provably useful for solving downstream tasks.

### 6.3 An Optimal Initialization for an Unknown Reward Function

The aim of this section is to provide intuition and a proof sketch of our main result (Theorem 4.1), which says the distribution over skills provides an optimal initialization for adapting to downstream tasks (under some assumptions). Intuitively, the difficulty of solving downstream tasks depends on how much the agent must adapt its initialization to learn the optimal policy for downstream tasks. Different initializations will make some tasks easier and some tasks more difficult.

It may seem counterintuitive that unsupervised skill learning algorithms would learn a good initialization. Maximizing mutual information *pushes* skills away from one another, not as finding a skill or policy that is close to every other policy. We now discuss a different perspective, thinking about how the *average* state distribution over the skills changes during optimization. It turns out that maximizing mutual information corresponds to *pulling* the average state distribution closer to other policies. Precisely, the average state distribution is pulled closer to the *furthest* policy:

**Lemma 6.5** (Theorem 13.11 from Cover and Thomas [9], based on Gallager [17], Ryabko [49])**.** *Let $\rho(\mathbf{s} \mid \mathbf{z})$ be given. Maximizing mutual information is equivalent to minimizing the divergence between the average state distribution $\rho(\mathbf{s})$ and the state distribution of the furthest skill:*

$$\max_{p(\mathbf{z})} I(\mathbf{s}; \mathbf{z}) = \min_{\rho(\mathbf{s})} \max_{\mathbf{z}} D_{\mathrm{KL}}(\rho(\mathbf{s} \mid \mathbf{z}) \parallel \rho(\mathbf{s}))$$

This characterization can be understood as a *facility location problem* [13] or a *1-center problem* [18, 37], where the distance metric is taken to be the KL divergence (which is not itself a metric). Since maximizing mutual information learns an average state distribution that is close to other policies, we expect that this average state distribution will be a good initialization for learning these other policies. This result provides the intuition behind our main result, which we present in Appendix A.5.

**Relationship with meta-learning.** We now discuss a close connection between this result and meta-learning. The pretraining objective (Eq. 4) bears a resemblance to the objective for meta-learning algorithms [16, 33, 46]: both aim to find a parameter initialization for which an inner optimization problem can be efficiently solved. Whereas these methods assume access to a training distribution, skill learning algorithms used for pretraining do not have information about the downstream task. Moreover, meta-learning typically studies the average performance at test time, our objective measures the worst-case performance at test time.

### 6.4 Limitations of Analysis

The primary limitation of our analysis is that we used a simplified model of skill learning algorithms, one that viewed policy parameters and latent codes as one monolithic representation. This simplified model is more expressive than the practical implementations of skill learning algorithms introduced in prior work. While these practical implementations are less expressive, it remains to be seen whether they actually learn more unique skills. Likewise, it remains how these expressivity constraints affect the average state distribution. The geometry of the state marginal polytope may also change for these practical implementations. In fact, the set of policies achievable by some restricted family of parametric policies may no longer be convex.

### 6.5 Implications for Future Skill Learning Algorithms

Our analysis also sheds light on fundamental limitations of skill learning algorithms based on mutual information. First, as explained in Sec. 6.2, existing algorithms based on mutual information do not learn skills that are sufficient for representing every policy. Future skill learning algorithms that can represent every policy may prove better at adapting to new tasks.

The second limitation is the adaptation procedure for which these methods produce the optimal initialization, under an assumption of how adaptation is performed. The natural gradient used for adaptation is not the standard natural policy gradient [29], but rather the natural gradient with respect to the state marginal distribution. This gradient implicitly depends on the environment dynamics; computing this gradient would likely require exploring the environment to determine how changes in a policy's parameters affect changes in the states visited. Said in other words, our analysis suggests that existing unsupervised skill learning methods are optimal for a somewhat unrealistic model of how adaptation happens. This suggests an opportunity to develop significantly more powerful algorithms that serve as effective initializations for more realistic adaptation procedures, such as those studied in meta-RL [15, 28, 56].

## 7 Conclusion

We have used a geometric perspective to answer a number of open questions in the study of skill discovery algorithms and unsupervised RL. We also showed that the skills learned by unsupervised skill learning algorithms are not sufficient for maximizing every downstream reward function. However, we do show that the average state distribution of the skills provides an initialization that is optimal for solving unknown downstream tasks using a particular adaptation procedure. Our analysis showed that the number of skills is bounded by the number of states, suggesting that scaling these methods to learn more skills will hit a theoretical upper bound. Taken together, we believe that this analysis sheds light on what prior methods are doing, when and where they should work, and what limitations could be addressed in the next generation of skill learning algorithms.

## References

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

[2] Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020). Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*.

[3] Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

[4] Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. (2018). Mutual information neural estimation. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR.

[5] Bellemare, M., Dabney, W., Dadashi, R., Ali Taiga, A., Castro, P. S., Le Roux, N., Schuurmans, D., Lattimore, T., and Lyle, C. (2019). A geometric perspective on optimal representations for reinforcement learning. *Advances in neural information processing systems*, 32:4358–4369.

[6] Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i Nieto, X., and Torres, J. (2020). Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pages 1317–1327. PMLR.

[7] Chebotar, Y., Hausman, K., Lu, Y., Xiao, T., Kalashnikov, D., Varley, J., Irpan, A., Eysenbach, B., Julian, R., Finn, C., et al. (2021). Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*.

[8] Co-Reyes, J., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. (2018). Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In *International Conference on Machine Learning*, pages 1009–1018. PMLR.

[9] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA.

[10] Dadashi, R., Taiga, A. A., Roux, N. L., Schuurmans, D., and Bellemare, M. G. (2019). The value function polytope in reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1486–1495. PMLR.

[11] Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., and Levine, S. (2018). Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*.

[12] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*.

[13] Farahani, R. Z. and Hekmatfar, M. (2009). *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media.

[14] Feinberg, E. A. and Shwartz, A. (2012). *Handbook of Markov decision processes: methods and applications*, volume 40. Springer Science & Business Media.

[15] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

[16] Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR.

[17] Gallager, R. G. (1979). Source coding with side information and universal coding.

[18] Garfinkel, R., Neebe, A., and Rao, M. (1977). The m-center problem: Minimax facility location. *Management Science*, 23(10):1133–1142.

[19] Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

[20] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

[21] Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

[22] Hansen, S., Dabney, W., Barreto, A., Warde-Farley, D., de Wiele, T. V., and Mnih, V. (2020). Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*.

[23] Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. (2018). Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*.

[24] Hazan, E., Kakade, S., Singh, K., and Van Soest, A. (2019). Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR.

[25] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

[26] Hiriart-Urruty, J.-B. and Lemaréchal, C. (2004). *Fundamentals of convex analysis*. Springer Science & Business Media.

[27] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

[28] Humplik, J., Galashov, A., Hasenclever, L., Ortega, P. A., Teh, Y. W., and Heess, N. (2019). Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*.

[29] Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14.

[30] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.

[31] Laskin, M., Srinivas, A., and Abbeel, P. (2020). Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR.

[32] Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., and Abbeel, P. (2021). Urlb: Unsupervised reinforcement learning benchmark.

[33] Lee, K., Maji, S., Ravichandran, A., and Soatto, S. (2019a). Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665.

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

[35] Martens, J. (2020). New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21:1–76.

[36] McAllester, D. and Stratos, K. (2020). Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR.

[37] Minieka, E. (1970). The m-center problem. *Siam Review*, 12(1):138–139.

[38] Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. (2020). Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR.

[39] Modi, A., Chen, J., Krishnamurthy, A., Jiang, N., and Agarwal, A. (2021). Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*.

[40] Mohamed, S. and Jimenez Rezende, D. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems*, 28:2125–2133.

[41] Nachum, O. and Dai, B. (2020). Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*.

[42] Ng, A. Y., Parr, R., and Koller, D. (1999). Policy search via density estimation. In *NIPS*, pages 1022–1028.

[43] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[44] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR.

[45] Puterman, M. L. (1990). Markov decision processes. *Handbooks in operations research and management science*, 2:331–434.

[46] Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32:113–124.

[47] Raskutti, G. and Mukherjee, S. (2015). The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457.

[48] Rockafellar, R. T. (2015). *Convex analysis*. Princeton university press.

[49] Ryabko, B. Y. (1979). Coding of a source with unknown but ordered probabilities. *Problems of Information Transmission*, 15(2):134–138.

[50] Salge, C., Glackin, C., and Polani, D. (2014). Empowerment–an introduction. In *Guided Self-Organization: Inception*, pages 67–114. Springer.

[51] Schwarzer, M., Rajkumar, N., Noukhovitch, M., Anand, A., Charlin, L., Hjelm, D., Bachman, P., and Courville, A. (2021). Pretraining representations for data-efficient reinforcement learning. *arXiv preprint arXiv:2106.04799*.

[52] Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. (2020). Planning to explore via self-supervised world models. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592. PMLR.

[53] Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., and Lee, K. (2021). State entropy maximization with random encoders for efficient exploration. *arXiv preprint arXiv:2102.09430*.

[54] Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. (2019). Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*.

[55] Song, J. and Ermon, S. (2019). Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*.

[56] Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

[57] Wang, T., Bowling, M., and Schuurmans, D. (2007). Dual representations for dynamic programming and reinforcement learning. *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 44–51.

[58] Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. (2018). Unsupervised control through non-parametric discriminative rewards. In *International Conference on Learning Representations*.

[59] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

[60] Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.

# A Proofs and Additional Analysis

## A.1 Mathematical Definition of the State Marginal Polytope

We provide a formal description of the constraint set. Precisely, the set of achievable state marginal distributions are those for which the *state-action* marginal distribution $\rho(\mathbf{s}, \mathbf{a})$ satisfies the following linear constraints [45, Eq. 6.9.2]:

$$\sum_{\mathbf{a}' \in \mathcal{A}} \rho(\mathbf{s}', a') = (1-\gamma) p_1(\mathbf{s}') + \gamma \sum_{\mathbf{s} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}} p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \rho(\mathbf{s}, \mathbf{a}) \quad \forall \mathbf{s}' \in \mathcal{S}$$
$$\rho(\mathbf{s}, \mathbf{a}) \geq 0 \quad \forall \mathbf{s} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}.$$

The constraint that $\rho$ be a valid probability distribution (i.e., $\sum_{\mathbf{s},\mathbf{a}} \rho(\mathbf{s}, \mathbf{a}) = 1$) is redundant with the constraints above. These constraints can also be written in matrix form as $(\tilde{I} - \gamma P)\vec{\rho}_{s,a} = (1-\gamma)\vec{p}_1$, where $\tilde{I}$ is a binary matrix that sums the action probabilities for each state (i.e., $\tilde{I}\vec{\rho}_{s,a} = \vec{\rho}_s$), $P$ is the transition matrix, and $\vec{p}_1$ is the initial state distribution.

## A.2 Additional Intuition for the State Marginal Polytope

First, let two policies $\pi_1(\mathbf{a} \mid \mathbf{s})$ and $\pi_2(\mathbf{a} \mid \mathbf{s})$ be given and consider their corresponding state marginals, $\rho_1(\mathbf{s})$ and $\rho_2(\mathbf{s})$. We illustrate these marginals in Fig. 1 (center). Any state marginal on the line between the two state marginals $\rho_1$ and $\rho_2$ is also achievable. We can express all such state marginals as $\rho_\lambda = \lambda\rho_1 + (1-\lambda)\rho_2$. One way to construct the policy that achieves the state marginal $\rho_\lambda$ is to form a *non-Markovian* mixture policy: sample $i \sim \text{Bernoulli}(\lambda)$ at the start of each episode and then sample actions $\mathbf{a} \sim \pi_i(\mathbf{a} \mid \mathbf{s})$ for each step of that episode. By construction, this mixture policy will have a state occupancy measure of $\rho_\lambda$.

It turns out that there also exists a *Markovian* policy $\pi_\lambda$ that achieves the state marginal $\rho_\lambda$:

$$\pi_\lambda(\mathbf{a} \mid \mathbf{s}) = \lambda(\mathbf{s})\pi_1(\mathbf{a} \mid \mathbf{s}) + (1 - \lambda(\mathbf{s}))\pi_2(\mathbf{a} \mid \mathbf{s}) \quad \text{where} \quad \lambda(\mathbf{s}) \triangleq \frac{\lambda\rho_1(\mathbf{s})}{\lambda\rho_1(\mathbf{s}) + (1-\lambda)\rho_2(\mathbf{s})} \tag{5}$$

While the existence proof is somewhat involved [60, Theorem 2.8][14, Theorem 6.1], the construction of the policy is surprisingly simple. First, note that it is sufficient to ensure $\rho_\lambda(\mathbf{s}, \mathbf{a}) = \lambda\rho_1(\mathbf{s}, \mathbf{a}) + (1-\lambda)\rho_2(\mathbf{s}, \mathbf{a})$. Then, we apply Bayes' rule to determine the corresponding policy:

$$\begin{aligned}
\pi_\lambda(\mathbf{a} \mid \mathbf{s}) = \frac{\rho_\lambda(\mathbf{s}, \mathbf{a})}{\rho_\lambda(\mathbf{s})} &= \frac{\lambda\rho_1(\mathbf{s}, \mathbf{a}) + (1-\lambda)\rho_2(\mathbf{s}, \mathbf{a})}{\lambda\rho_1(\mathbf{s}) + (1-\lambda)\rho_2(\mathbf{s})} \\
&= \frac{\lambda\pi_1(\mathbf{a} \mid \mathbf{s})\rho_1(\mathbf{s}) + (1-\lambda)\pi_2(\mathbf{a} \mid \mathbf{s})\rho_2(\mathbf{s})}{\lambda\rho_1(\mathbf{s}) + (1-\lambda)\rho_2(\mathbf{s})} \\
&= \lambda(\mathbf{s})\pi_1(\mathbf{a} \mid \mathbf{s}) + (1-\lambda(\mathbf{s}))\pi_2(\mathbf{a} \mid \mathbf{s}) \quad \text{where} \quad \lambda(\mathbf{s}) \triangleq \frac{\lambda\rho_1(\mathbf{s})}{\lambda\rho_1(\mathbf{s}) + (1-\lambda)\rho_2(\mathbf{s})}.
\end{aligned}$$

By induction, one can also show that convex combinations of *multiple* state marginals also yields achievable state marginal distributions:

**Fact 3.** *The convex hull of achievable state marginal distributions is also achievable.*

We illustrate this observation in Fig. 1 (right). The construction of the corresponding policy is analogous to Eq. 5.

### A.3 Proof of Lemma 6.1

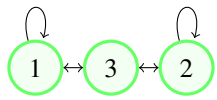
We now prove Lemma 6.1:

*Proof.* If $p(\mathbf{z})$ were non-zero for a policy where the state marginal $\rho(\mathbf{s} \mid \mathbf{z})$ did not lie at a vertex, then the mutual information objective could be improved by shifting $p(\mathbf{z})$ to place probability mass on the vertex rather than interior point $\rho(\mathbf{s} \mid \mathbf{z})$. □

### A.4 Alternative Strategies Also Fail to Discovery All Vertices

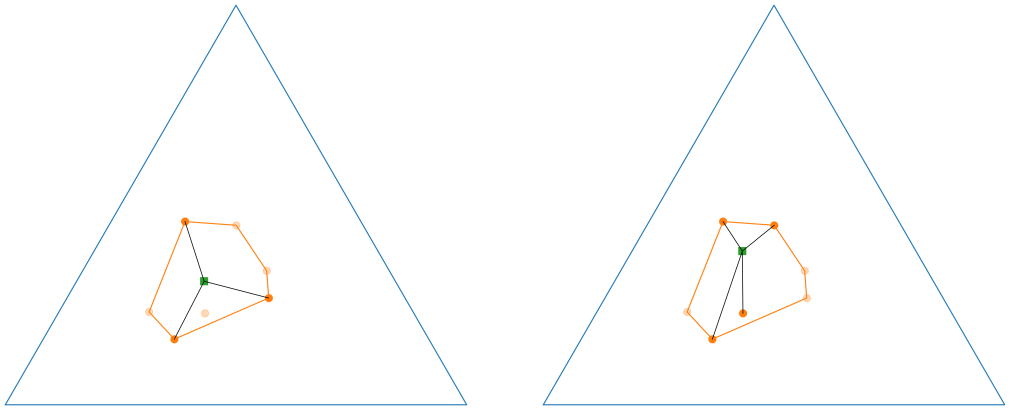
**Indicator reward functions.** Another simple method for skill discovery, based in prior work [38, Sec. 3.1], is to define a sparse reward function for visiting a particular state and action: $r_{s,a}(\mathbf{s}', \mathbf{a}') \triangleq \mathbb{1}(\mathbf{s}' = s, \mathbf{a}' = a)$. This method also fails to learn all vertices of the state marginal polytope, as we show in the following worked example.

We define the dynamics for a 3-state, 2-action MDP, set $\gamma = 0.5$, and define the initial state distribution to be uniform over the three states. This counterexample has deterministic dynamics, represented by arrows below:




None of the policies that are optimal for reward functions $\{r_{\mathbf{s},\mathbf{a}}\}$ are optimal for the reward $r(\mathbf{s}, a) = \mathbb{1}(\mathbf{s} \in \{1, 2\})$. The intuition behind this counterexample is that there are multiple ways to maximize this reward function: remaining at state 1 and remaining at state 2. However, none of the optimal policies for $\{r_{s,a)}\}$ employ this strategy; rather, all these policies deterministically attempt to repeat a single transition.

**Including actions in the mutual information ($I((\mathbf{s}, a); z)$).** Prior work on skill learning has considered many alternative definitions of the mutual information objective, experimenting with conditioning on actions [12], initial states [19], and entire trajectories [8]. We refer the reader to Achiam et al. [1] for a survey of these methods. Consider, for instance, the objective $I((\mathbf{s}, \mathbf{a}); \mathbf{z})$. Our reasoning from Lemma 6.3 suggests that this alternative objective would result in learning at most $|\mathcal{S}| \cdot |\mathcal{A}|$ skills, more than the $|\mathcal{S}|$ skills learned when maximizing $I(\mathbf{s}; \mathbf{z})$. This objective results



(a) $I(\mathbf{s};\mathbf{z})$ (b) $I((\mathbf{s},\mathbf{a});\mathbf{z})$

Figure 4: **Counterexample 1**: Maximizing $I((\mathbf{s},\mathbf{a});\mathbf{z})$ does not result in discovering more vertices of the state marginal polytope.

in discovering vertices of the *state-action* marginal polytope. However, these additional skills do not necessarily cover additional vertices of the *state* marginal polytope. We now present a worked example where maximizing $I((\mathbf{s},\mathbf{a});\mathbf{z})$ only covers 3 of 6 vertices, the same number as covered by maximizing $I(\mathbf{s};\mathbf{z})$.

We describe Counterexample 1 from Sec. 6 in more detail. This example has 3 states, 4 actions, and 7 skills. The matrix describing the state-action marginals $p(\mathbf{s},\mathbf{a} \mid \mathbf{z})$ is given as follows:

$$P_{sa} = \begin{bmatrix} 0.0716829 & 0.18424628 & 0.1795034 & 0.13672623 & 0.00295358 & 0.09510334 \\ 0.03845856 & 0.05906211 & 0.08669371 & 0.01741351 & 0.02568876 & 0.10246764 \\ 0.11750427 & 0.14607 & 0.11480248 & 0.00287273 & 0.00954288 & 0.10701075 \\ 0.02102705 & 0.02261918 & 0.14991443 & 0.23005368 & 0.06343457 & 0.01514798 \\ 0.13386153 & 0.06171002 & 0.09080011 & 0.16650628 & 0.05625721 & 0.06271223 \\ 0.10353768 & 0.09575361 & 0.02849315 & 0.00634354 & 0.13433545 & 0.05968919 \\ 0.05926209 & 0.10606773 & 0.04541879 & 0.05518233 & 0.14630736 & 0.12098995 \\ 0.05692926 & 0.0745452 & 0.04428352 & 0.12954613 & 0.0261651 & 0.13530253 \\ 0.13704367 & 0.01333569 & 0.10392952 & 0.04119388 & 0.11826303 & 0.12777541 \\ 0.12092377 & 0.0703527 & 0.08563638 & 0.08155578 & 0.04348018 & 0.05650998 \\ 0.06520597 & 0.00354202 & 0.11542116 & 0.09079959 & 0.14099273 & 0.05765374 \\ 0.04477772 & 0.03235234 & 0.07636344 & 0.00824066 & 0.09351822 & 0.27113241 \\ 0.20516233 & 0.19498386 & 0.03909264 & 0.11229717 & 0.06929778 & 0.09957651 \\ 0.02888635 & 0.08654659 & 0.00456412 & 0.10995985 & 0.0328871 & 0.0167457 \end{bmatrix}$$

The matrix describing the state marginals is computed as follows:

$$P_a = P_{sa} \begin{bmatrix} 1. & 0. & 0. \\ 1. & 0. & 0. \\ 1. & 0. & 0. \\ 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 1. & 0. \\ 0. & 1. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \\ 0. & 0. & 1. \\ 0. & 0. & 1. \\ 0. & 0. & 1. \end{bmatrix}$$

We illustrate the skills found by maximizing $I(\mathbf{s};\mathbf{z})$ and $I((\mathbf{s},\mathbf{a});\mathbf{z})$ in Fig. 4. Note that including the actions in the mutual information does *not* result in discovering more vertices of the state marginal polytope.

### A.5 Proof of Theorem 4.1

*Proof.* Our proof proceeds in two steps. First, we will solve the adaptation problem, assuming that both the reward function $r(\mathbf{s})$ and the initialization $\rho(\mathbf{s})$ are given. We then substitute the optimal policy $\rho^*(\mathbf{s})$ into the pretraining problem and solve for the optimal initialization $\rho(\mathbf{s})$.

First, we solve for the optimal policy $\rho^*(\mathbf{s})$ of the adaptation step. Using calculus of variations, the optimal policy can be written as

$$\rho^*(\mathbf{s}) = \frac{\rho(\mathbf{s})e^{r(\mathbf{s})}}{\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds'}.$$

We can then express the cost of adaptation as follows:

$$\begin{aligned}
&\textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) \\
&\quad= \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \mathbb{E}_{\rho^*(\mathbf{s})}[r(\mathbf{s})] + D_{\mathrm{KL}}(\rho^*(\mathbf{s}) \parallel \rho(\mathbf{s})) \\
&\quad= \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \frac{\int r(\mathbf{s})\rho(\mathbf{s})e^{r(\mathbf{s})}ds}{\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds'} + \frac{\int \rho(\mathbf{s})e^{r(\mathbf{s})}}{\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds'}\left(\cancel{\log\rho(\mathbf{s})} + r(\mathbf{s}) - \log\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds' - \cancel{\log\rho(\mathbf{s})}\right)ds \\
&\quad= \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \cancel{\frac{\int r(\mathbf{s})\rho(\mathbf{s})e^{r(\mathbf{s})}ds}{\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds'}} + \cancel{\frac{\int \rho(\mathbf{s})e^{r(\mathbf{s})}}{\int \rho(\mathbf{s}')e^{r(\mathbf{s}')}ds'}r(\mathbf{s})ds} - \log\int \rho(\mathbf{s})e^{r(\mathbf{s})}ds \\
&\quad= \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \log\int \rho(\mathbf{s})e^{r(\mathbf{s})}ds.
\end{aligned}$$

We can note write the pretraining problem (Eq. 4) as follows:

$$\min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) = \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \log\int \rho(\mathbf{s})e^{r(\mathbf{s})}ds.$$

We now aim to solve this optimization problem for $\rho(\mathbf{s})$. We start by noting that the $\max$ operator is commutative, so we can swap the order of the maximization of $r(\mathbf{s})$ and $\rho^+(\mathbf{s})$. That is, instead of the adversary first choosing an reward function and then choosing an optimal policy for that reward function, we can consider the adversary first choosing an optimal policy and then choosing a reward function for which that policy is optimal.

$$\min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) = \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \mathbb{E}_{\rho^+(\mathbf{s})}[r(\mathbf{s})] - \log\int \rho(\mathbf{s})e^{r(\mathbf{s})}ds.$$

Using calculus of variations, we determine that the optimal reward function satisfies

$$r(\mathbf{s}) = \log\rho^+(\mathbf{s}) - \log\rho(\mathbf{s}) + b,$$

where $b \in \mathbb{R}$ is a constant scalar. There are many optimal reward functions, one for each choice of $b$. However, as the scalar $b$ will cancel out in the next step of our proof, this multiplicity is not a concern. Substituting this worst-case reward function, we can write the overall pretraining objective as

$$\begin{aligned}
&\min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{r(\mathbf{s})\in\mathbb{R}^{|\mathcal{S}|}} \textsc{CostOfAdaptation}(\rho(\mathbf{s}), r(\mathbf{s})) \\
&\quad= \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[\log\rho^+(\mathbf{s}) - \log\rho(\mathbf{s}) + b] - \log\int \cancel{\rho(\mathbf{s})}\frac{\rho^+(\mathbf{s})}{\cancel{\rho(\mathbf{s})}}e^b ds \\
&\quad= \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[\log\rho^+(\mathbf{s}) - \log\rho(\mathbf{s}) + \cancel{b}] - \cancel{\log e^b} \\
&\quad= \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} \mathbb{E}_{\rho^+(\mathbf{s})}[\log\rho^+(\mathbf{s}) - \log\rho(\mathbf{s})] \\
&\quad= \min_{\rho(\mathbf{s})\in\mathcal{C}} \max_{\rho^+(\mathbf{s})\in\mathcal{C}} D_{\mathrm{KL}}(\rho^+(\mathbf{s}) \parallel \rho(\mathbf{s})).
\end{aligned}$$

Applying Lemma 6.5 completes the proof. □

### A.6 Example where all deterministic policies are unique vertices

## B Additional Plots

Fig. 6 plots the same experiment as Fig. 3 in the main text, repeated for different randomly-sampled dynamics.

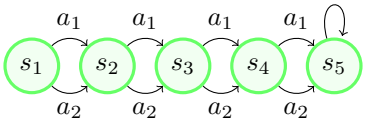



Figure 5

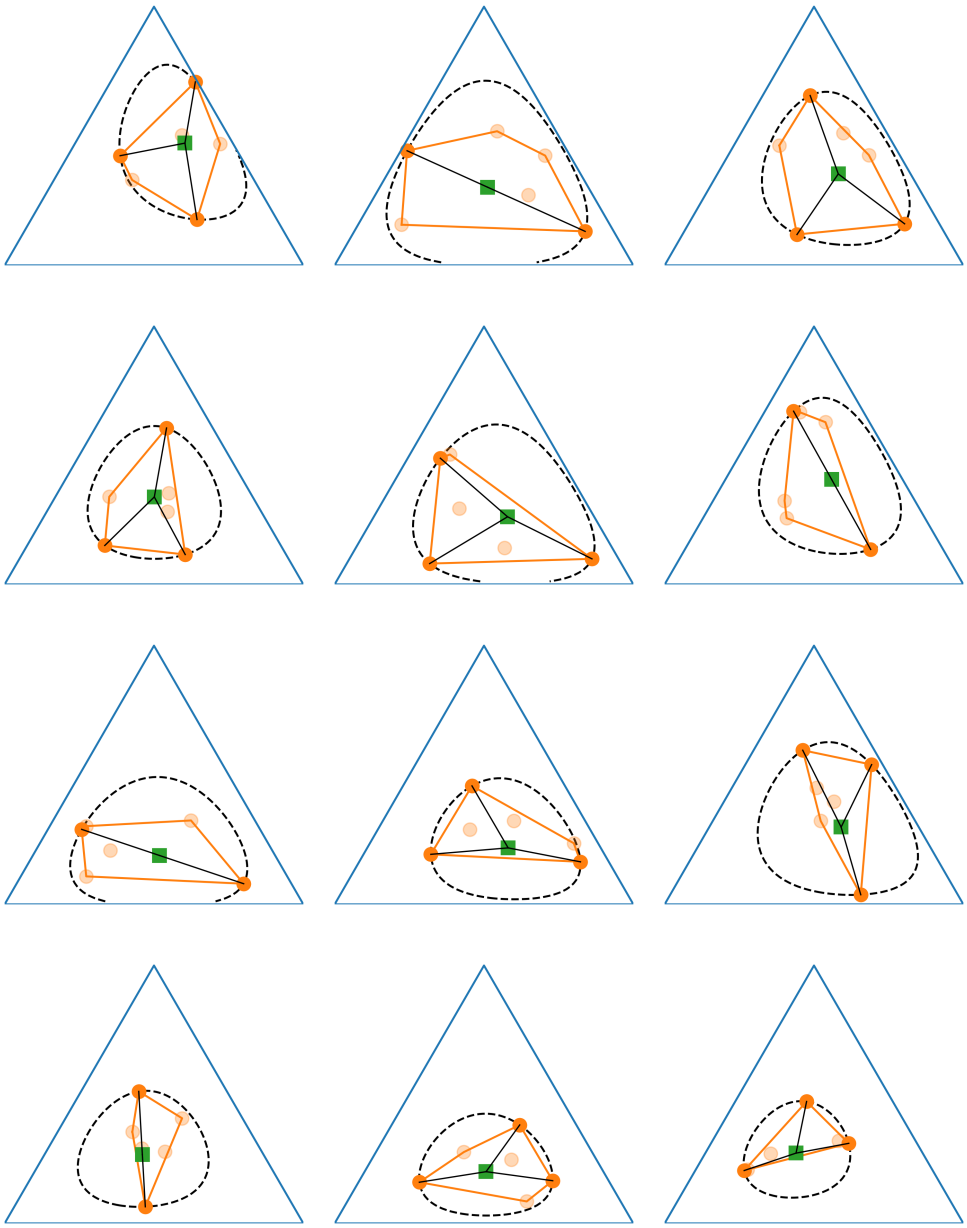


Figure 6: **Information Geometry of Skill Learning**: More examples of the skills learned by DIAYN (Fig. 3).