Discovering Mixture Skills for Unsupervised Reinforcement Learning

Nelson Ma, Junyu Xuan, Guangquan Zhang, Jie Lu
Australian Artificial Intelligence Institute (AAII)
University of Technology Sydney
Sydney, NSW Australia 2000
nelson.y.ma@student.uts.edu.au, junyu.xuan@uts.edu.au,
guangquan.zhang@uts.edu.au, jie.lu@uts.edu.au

Abstract

Skill discovery has emerged as a popular route for unsupervised reinforcement learning (URL), offering agents a diverse, reusable set of behaviours learned before any task-specific reward is experienced. However, existing methodologies tend to favour either categorical codes or unimodal skill priors, which simplifies training at the cost of limiting the variety of behaviours they can represent. We introduce Discovery of Mixture Skills (DiMS), a URL algorithm that learns a latent Gaussian mixture by training a Gaussian Mixture Variational Autoencoder (GMVAE) in tandem with the unsupervised policy. In DiMS, a hierarchical GMVAE simultaneously discovers clusters of skills, while an auxiliary macro-latent dynamically positions mixture components to prevent mode collapse. A joint loss term combining log-likelihood and curiosity rewards enables simultaneous updates of representation and policy while improving exploration. Experiments on the Unsupervised Reinforcement Learning Benchmark (URLB) show that DiMS consistently outperforms a wide range of state-of-the-art baselines. Ablation studies confirm that the mixture prior is critical to these gains, and that DiMS is robust to alternative exploration bonuses. Overall, our results suggest that Gaussian mixture skill priors offer a compelling foundation for future unsupervised RL.

1 Introduction

Reinforcement learning (RL) has proven remarkably effective in tackling a variety of complex tasks such as autonomous driving (Sallab et al. 2017) and refining large language models (Brown et al. 2020). However, traditional RL often depends on extensive, task-specific training procedures, necessitating retraining whenever new objectives or environments are introduced. This reliance on task-specific data not only increases computational costs, but also hinders the efficient reuse of learned behaviours across different domains. Unsupervised RL offers a promising solution: in the absence of explicit objectives or task supervision, agents seek to explore and acquire broadly applicable policies that can potentially streamline and accelerate downstream task-focused training.

Within the broader landscape of unsupervised RL, *skill discovery* has emerged as a potent approach wherein agents learn to condition their policies on a latent skill that is trained to correspond to a unique behaviour. By structuring agents in this manner, skill discovery methods emphasise building a versatile repertoire of stable, reproducible behaviours in a manner that alternative unsupervised RL methods that prioritise state space coverage cannot. Such diversity in learned skills not only facilitates effective exploration but also provides robust behavioural priors that can be readily finetuned when adapting to downstream tasks.

Despite these potential advantages, effective skill discovery is challenging. Most existing algorithms structure their latent representation as either a finite set of discrete codes or a single continuous random variable (Ladosz et al. 2022). The former restricts the agent to a small number of skills, limiting granularity and forcing unrelated behaviours to share a label once the budget is exhausted, while the latter draws every latent from the same spherical cloud, so qualitatively different behaviours drift and overlap unless strong regularisers are imposed. Fixed, task-agnostic priors therefore leave the method caught between under-expressiveness and mode collapse, leading to learned skill sets that transfer unreliably to downstream rewards.

A learned Gaussian mixture prior offers a principled solution. By coupling a categorical latent that selects distinct behavioural clusters with a continuous latent that specifies exact behaviour, a broad spectrum of control is possible; the categorical variable first partitions the behavioural space, while the Gaussian component then controls specific behaviours. Because the mixture parameters are learned, each component migrates to a behaviourally meaningful region of latent space, turning the prior into a powerful inductive bias. Similar mixture representations have proven effective in multi-modal image generation (Dilokthanakul et al. 2016), disentangled text modelling (Shi et al. 2019), and regime-switching time-series analysis (Guo et al. 2018). When applied to skill discovery, mixture representations enable non-overlapping groups of skills with smooth intra-group variation, and a structured prior that downstream policies can exploit for rapid adaptation and robust exploration.

In this work, we build on these insights by proposing a skill discovery framework that utilises a Gaussian mixture latent. Our central idea is to learn, rather than prescribe, a hierarchical latent that provides the agent with a rich behavioural prior capable of generating a broader repertoire of skills. Such flexibility can lead to more expressive skill em-

beddings than conventional discrete or single-vector methods. By pretraining agents on these hierarchical mixture-skills, we lay a stronger foundation for downstream tasks, improving exploration and learning more robust and adaptable priors.

The primary contributions of this paper are as follows:

1. We introduce a latent skill mixture model that adapts hierarchical Gaussian-Mixture Variational Auto-Encoder (GMVAE) architectures to unsupervised RL. This model learns compact state encodings while clustering them into well-separated latent skill modes, providing a principled bridge between discrete and continuous skill spaces
2. Building on this GMVAE, we develop **Di**scovery of **M**ixture **S**kills (DiMS), which utilises skill-agnostic exploration bonuses to simultaneously gather informative data and train the GMVAE, before learning a policy capable of effectively categorising and reaching diverse regions of the state space.
3. We demonstrate that our method outperforms existing unsupervised RL algorithms in the standard Unsupervised Reinforcement Learning Benchmark (URLB) (Laskin et al. 2021) and achieves a new state-of-the-art result.

2 Problem Statement and Notation

As is typical in RL literature, we view the RL problem through the lens of a Markov Decision Process (MDP) $M = \{S, A, P, R\}$ consisting of state space S, action space A, state transition function $P = p(s'|s, a)$ and reward function $R = r(s, a, s')$. An agent interacts with an RL environment via its policy $\pi_\theta(a|s)$ with parameters θ, learning it in such a fashion that maximises expected cumulative reward $\mathbb{E}_\pi[r(\tau)]$, where $\tau = \{s_n, a_n, s'_n\}_{n\in[0,N]}$ is the trajectory induced by the agent policy over an episode. Then, Unsupervised RL describes a reward-free setting where, in the absence of a typical extrinsic reward r^{ext}, the agent instead seeks to maximise some intrinsic reward $r^{int} = r^{int}(s, a, s')$. If constructed appropriately, r^{int} should provide an effective exploratory signal which, at a later stage, can be used for efficient finetuning with some r^{ext}.

Skill discovery augments the above MDP by introducing a latent skill $\tilde{z} \in Z$, which can be either discrete or continuous. The overarching goal therein is to learn a skill-conditioned policy $\pi_\theta(a|s, \tilde{z})$ so that distinct values of $\tilde{z}$ induce useful and meaningfully different behaviours, with agents being guided by intrinsic reward $r^{int}(s, a, s', \tilde{z})$. Subsequently, during the finetuning phase, a specific skill $\tilde{z}^*$ is chosen from a small interaction budget and $\pi_\theta(a|s, \tilde{z}^*)$ is further optimised with extrinsic reward r^{ext}. For the remainder of this work, π_θ and r^{int} will refer to their skill-conditioned counterparts for clarity, unless otherwise specified. Furthermore, we use a tilde to notationally distinguish the usual skill discovery latent $\tilde{z}$ from any latent decompositions, such as $\tilde{z} = (w, y, z)$.

3 Related Work

Unsupervised Reinforcement Learning. Using the terminology from Laskin et al. (2021), URL methods as a collective are commonly grouped into three broad categories: knowledge-based, data-based, and competence-based approaches. Knowledge-based methods (Pathak et al. 2017; Burda et al. 2019; Pathak, Gandhi, and Gupta 2019; Fickinger et al. 2021; Rajeswar et al. 2023), reward the agent for actions that yield high prediction error or uncertainty in learned models, encouraging agents to explore unfamiliar regions of the state space. However, they tend to suffer from detachment and can over-explore inherently noisy locations. Data-based approaches (Liu and Abbeel 2021b; Seo et al. 2021; Zhang et al. 2021; Yang et al. 2024; Ying et al. 2025) instead maximise the entropy of visited states, promoting wide state space coverage and uniform exploration. Both methodologies, while effective at exploration, do not prioritise the learnability or reuse of structured behaviours, making it difficult to recover coherent skills that can support downstream adaptation.

Unsupervised Skill Discovery. Competence-based approaches, also known as unsupervised skill discovery, seek to address the limitations of previous methods by learning a set of distinguishable and reusable behaviours, as defined by a latent skill vector $\tilde{z}$. For each $\tilde{z}$, an effective skill-conditioned policy $\pi(a|s, \tilde{z})$ must generate distinct trajectories while also maximising diversity in states visited by different $\tilde{z}$.

One common approach towards achieving diversity is to maximise the mutual information between latent skill and state, $I(s, z)$, which can collectively be described as MI approaches (Xie et al. 2022). Reverse-MI approaches (Gregor, Rezende, and Wierstra 2017; Eysenbach et al. 2019; Strouse et al. 2022; Dave and Rueckert 2025) directly maximise skill discriminability, $I(s, \tilde{z}) = H(\tilde{z}) - H(\tilde{z}|s)$. Although these methods can produce clearly distinct skills, they are vulnerable to a lack of exploration, as maximising $H(\tilde{z})$ does not necessarily imply high state entropy $H(s)$. Like our method DiMS, VALOR (Achiam et al. 2018) frames skill discovery as a variational inference problem using an autoencoder-like structure, but does not learn a mixture prior over skills, nor does it explicitly incorporate maximum-entropy exploration to improve behavioural diversity and state space coverage. Recently, some methods have worked to improve state space coverage using skill clustering (Qing and Zhu 2024), recurrent training (Jiang, Gao, and Chen 2022) and guidance heuristics (Kamienny et al. 2022; Kim et al. 2023).

Alternatively, Forward-MI approaches directly maximise state entropy $H(s)$ with the alternate decomposition $I(s, \tilde{z}) = H(s) - H(s|\tilde{z})$. In doing so, these approaches enjoy diversity in behaviour and strong state space exploration. However, approximating state entropy is a difficult challenge; likelihood modelling (Sharma et al. 2020) can suffer from model inaccuracies, while density estimates and pseudo-counts (Lee et al. 2019; Liu and Abbeel 2021a) can scale poorly in high-dimension state spaces. Nevertheless, these methods have enjoyed strong benchmark results by pairing the state-entropy objective with contrastive skill learning (Laskin et al. 2022), ensemble policies (Zhao et al. 2022; Bai et al. 2024) and additional regularisers (Yang et al. 2023; Qing et al. 2025).

Of particular relevance to our work is the Explore-Discover-Learn (EDL) framework (Campos et al. 2020), which adopts a generative strategy for approximating the forward mutual-information objective by training a VAE to model the conditional state likelihood $p_\psi(s|\tilde{z})$. This approach

shares strong similarity to DiMS due to its generative nature, but we considerably build upon these ideas by improving exploration, learning the generative model simultaneously to simplify training phases, and utilising a Gaussian mixture latent to improve representation expressiveness.

Finally, other methods do not directly fall into the forward- and reverse-MI taxonomy. Goal-conditioned RL (Andrychowicz et al. 2017; Pong et al. 2020) can also be interpreted as a special case of skill discovery where the goal state $s = \tilde{z}$. Another line of work directly maximises distance traveled in the state space, seeking to induce dynamic behaviour without directly estimating MI or state densities (Park et al. 2022, 2023; Park, Rybkin, and Levine 2024).

Learning Mixture Representations. Mixture modelling offers a principled way to represent multi-modal data, rather than forcing a unimodal fit. This can be achieved through a variety of approaches, including the classic Mixture-Density Network (Bishop 1994), and recent high-capacity models such as the Transformer (Vaswani et al. 2017) and Diffusion models (Kingma et al. 2021) which learn implicit mixtures directly in output space. These latter approaches, however, rely on very large architectures whose internal structure is hard to reuse in downstream tasks. In comparison, Mixture Variational Auto-Encoders (Alemi et al. 2017; Dilokthanakul et al. 2016) strike a practical balance: an information bottleneck produces a compact latent code, enabling controllable sampling and sample-efficient transfer—properties that are particularly attractive for reinforcement learning (RL).

GMVAE structures in literature can be split broadly between *flat*, versus *hierarchical* structure. Flat GMVAEs employ a single categorical latent $y \sim \mathrm{Cat}(K)$ to select a Gaussian representation $p(z|y)$ (Jiang et al. 2016). Hierarchical designs add an additional discrete (or continuous) layer above this mixture, so each top-level choice owns its own sub-posterior, yielding greater expressiveness at the cost of extra KL terms and heavier compute (Kviman et al. 2023). We adopt the hierarchical GMVAE of Dilokthanakul et al. (2016) for our method as the additional flexibility offered by its high-level *macro* latent w enables mixtures to adapt to diverse regions of latent space.

4 Latent Skill Mixture Model

Mixture-based latent skills provide a principled way to cluster an agent's behaviour into distinct modes in a similar fashion to discrete skills, while retaining the range of behavioural expression of continuous single-latent skill discovery. A naive implementation would consider the latent skill $\tilde{z} = (y, z)$, where $y \in \{1, \dots, K\}$ represents a broad categorical *cluster index* for the Gaussian *skill* $z|y \sim \mathcal{N}\big(\mu(y), \sigma^2(y)\big)$.

However, adapting this latent decomposition to skill discovery presents a significant challenge. Typically, skill discovery methods fix the sampling distribution of z and focus on learning a suitable conditional policy (Ladosz et al. 2022). In contrast, sampling from a particular mixture component requires knowledge of the relevant mixture parameters, such as mean and covariance. Although it is possible to follow the lead of existing methodologies and choose arbitrary and equidistant cluster centres, learning optimal parameters during the pretraining process yields not only behaviourally meaningful clusters, but also coherently organises them in latent space.

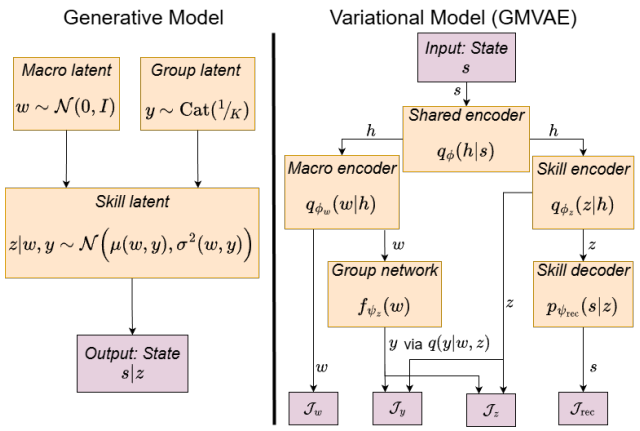

Figure 1: **DiMS Overview.** Outline of the generative latent model used in DiMS (left), as well as the variational model used to train the GMVAE encoder. Formulating the variational model in this manner enables the use of a low-variance estimator for the categorical posterior $q(y|w, z)$.

To achieve this, we learn the mixture parameters jointly with the unsupervised policy by means of a *Gaussian Mixture variational auto-encoder* (GMVAE). Given input states s sampled from a replay buffer B, our GMVAE learns an encoder with parameters $\phi = \{\phi_w, \phi_z\}$ and a decoder with parameters $\psi = \{\psi_z, \psi_{rec}\}$. A structural diagram of the GMVAE used can be found in Figure 1. For the remainder of this section, we describe the model components and training.

Generative model. Following Dilokthanakul et al. (2016), we consider a three-stage generative process where separate latent variables govern global variation, skill clusters, and local behaviour. Concretely, given the number of skill clusters (mixture components) K, for each state s:

1. We draw a *macro latent* $w \sim \mathcal{N}(0, I)$, which is forwarded through a network $f_{\psi_z}(w)$ to produce K sets of means and variances $\{(\mu_{\psi_z}(w,k), \sigma^2_{\psi_z}(w,k))\}_{k=1}^K$ corresponding to each mixture component
2. Then, the categorical cluster index $y \sim \mathrm{Cat}\big(\frac{1}{K}\big)$ is sampled
3. Finally, the continuous skill z is drawn from $z \mid w, y \sim \mathcal{N}\big(\mu(w,y),\, \sigma^2(w,y)\big)$

Intuitively, the macro latent w is responsible for arranging where skill clusters are centred, before y and z specify behaviour. Without this adaptive capacity, mixture parameters would be required to simultaneously fit trajectories scattered across the entire state space, risking clusters centring on uninformative midpoints or collapsing entirely. Conditioning the prior on an additional latent w enables mixture centres to more dynamically adjust to the space the agent is currently exploring, preserving expressiveness and keeping behavioural clusters y well separated.

Encoder. The purpose of the encoder is to approximate the joint posterior $q_\phi(w, y, z|s) = q_{\phi_w}(w|s)q_{\phi_z}(z|s)q(y|w, z)$ that assigns conditional densities over macro w, probability over the cluster latent y, and corresponding skill z. Architecturally, the encoder receives a state s as input and outputs Gaussian posteriors $q_{\phi_w}(w|s) = \mathcal{N}\big(\mu_{\phi_w}(s), \sigma^2_{\phi_w}(s)\big)$ and $q_{\phi_z}(z|s) = \mathcal{N}\big(\mu_{\phi_z}(s), \sigma^2_{\phi_z}(s)\big)$. Then, it is possible to calculate the categorical posterior analytically,

$$q(y = k|w, z) = \frac{\mathcal{N}(z \mid \mu(w, k), \sigma^2(w, k))}{\sum_{j=1}^{K} \mathcal{N}(z \mid \mu(w, j), \sigma^2(w, j))},$$

Decoder. The decoder seeks to reconstruct the joint distribution $p_\psi(s, w, y, z) = p_{\psi_{rec}}(s|z)p_{\psi_z}(z|w, y)p(w)p(y)$. The reconstruction term $p_{\psi_{rec}}(s|z)$ maps skills back to the state space, while

$$p_{\psi_z}(z|w, y) = \mathcal{N}\big(z\ ;\ \mu_{\psi_z}(w, y), \sigma^2_{\psi_z}(w, y)\big)$$

is the Gaussian mixture prior whose parameters are produced by the network $f_{\psi_z}(w)$. We use a standard normal prior for $p(w) = \mathcal{N}(0, I)$ and a uniform categorical prior $p(y) = \frac{1}{K}$. Together, these terms define a learned Gaussian mixture prior $p_{\psi_z}(w, y, z)$ that keeps components well separated in latent space and adapts to the regions of state space the agent actually visits - properties crucial for skill discovery and sampling.

Objective. We maximise the evidence lower bound (ELBO)

$$\mathcal{J}_{\text{ELBO}} = \mathbb{E}_{q_\phi(w,y,z|s)}\Big[\alpha_{\text{rec}}\mathcal{J}_{\text{rec}} + \alpha_w\mathcal{J}_w + \alpha_y\mathcal{J}_y + \alpha_z\mathcal{J}_z\Big], \tag{1}$$

where

$$\mathcal{J}_{\text{rec}} = \log p_{\psi_{\text{rec}}}(s|z), \tag{2}$$

$$\mathcal{J}_w = -\text{KL}\big[q_{\phi_w}(w|s) \parallel p(w)\big], \tag{3}$$

$$\mathcal{J}_y = -\text{KL}\big[q(y|w, z) \parallel p(y)\big], \tag{4}$$

$$\mathcal{J}_z = -\text{KL}\big[q_{\phi_z}(z|s) \parallel p_{\psi_z}(z|w, y)\big]. \tag{5}$$

The reconstruction objective $\mathcal{J}_{\text{rec}}$ ensures z captures behaviourally relevant information about the state, while the last objective $\mathcal{J}_z$ ensures the amortised encoder $q_{\phi_z}(z|s)$ matches the true conditional distribution of z under the generative model, $p_{\psi_z}(z|w, y)$. Finally, the regularisation terms $\mathcal{J}_w$ and $\mathcal{J}_y$ constrain the relevant approximate posteriors, preventing posterior collapse.

From an information-theoretic standpoint, maximising the generative likelihood $\log p_{\psi_{\text{rec}}}(s|z)$ decreases the conditional state entropy $H(S|Z)$, ensuring that skills produce consistent, reliable trajectories. Conversely, maximising the variational posterior $\log q_\phi(z|s)$ decreases the conditional entropy $H(Z|S)$, generating states with highly discriminable latent representations. When combined with a high-entropy exploration policy (such as APT (Liu and Abbeel 2021b)), optimising both terms concurrently serves to maximise both forward- and reverse decompositions of MI, thereby balancing behavioural diversity with latent discriminability.

Algorithm 1: DiMS: Discovery of Mixture Skills

Input: env, training steps N_{steps}, skill update cadence K
Initialise replay buffer $\mathcal{B}$, policy parameters θ, exploration parameters η, GMVAE parameters (ϕ, ψ)
for step $= 1$ **to** N_{steps} **do**
  if step $\bmod K == 0$ **then**
    Sample $\tilde{z} = (w, y, z) \sim p_{\psi_z}(z|w, y)\, p(w)\, p(y)$
  end if
  Execute $a \sim \pi_\theta(\,\cdot \mid s, w, y, z)$, observe next observation s'
  Store transition $(s, a, s', \tilde{z})$ in $\mathcal{B}$
  // GMVAE training
  Sample batch $\mathcal{B}_{\text{GMVAE}}$ from $\mathcal{B}$
  Update (ϕ, ψ) on $\mathcal{B}_{\text{GMVAE}}$ using ELBO (Equation 1)
  // Policy training
  Sample batch $\mathcal{B}_{\text{Policy}}$ from $\mathcal{B}$
  Update η on $\mathcal{B}_{\text{Policy}}$ (Equation 6)
  Compute r_{DiMS} for $(s, w, y, z) \in \mathcal{B}_{\text{Policy}}$ (Equation 8)
  Update θ with r_{DiMS} using base RL algorithm
end for

5 DiMS: Discovery of Mixture Skills

In this section we present details on our method **Di**scovery of **M**ixture **S**kills (DiMS), an unsupervised RL algorithm that leverages the latent-mixture model of Section 4 to learn a latent hierarchical skill representation and an RL policy capable of reproducing each skill. In addition, we detail key implementation considerations. Pseudocode for DiMS can be found in Algorithm 1.

Exploration. Effective skill discovery requires broad state space coverage in order to gather sufficient data for learning diverse behaviours. Recalling the forward-MI decomposition $I(S, \tilde{Z}) = H(S) - H(S|\tilde{Z})$, while the GMVAE decoder in DiMS can optimise the conditional term $H(S|\tilde{Z})$, the marginal entropy $H(S)$ can only be increased by visiting a wide variety of states. Therefore, it is particularly important for DiMS to populate a replay buffer with a high-entropy state distribution.

Although there are a wide range of exploratory signals available in the literature, we choose *Random Network Distillation* (RND) (Burda et al. 2019) as our exploration bonus, which utilises the prediction error of a trainable network $f_{\text{pred},\eta}$ against a fixed, randomly initialised target f_{target} as a proxy for curiosity:

$$\mathcal{L}_{\text{explore}} = r_{\text{explore}}(s) = \big\| f_{\text{pred},\eta}(s) - f_{\text{target}}(s)\big\|_2^2. \tag{6}$$

RND is a natural fit for DiMS as it is lightweight, independent of any latent $\tilde{z}$, stable on continuous-control benchmarks, and empirically outperforms alternative exploration signals (see ablations in Section 6.3). It is particularly important that the exploration bonus remains skill-agnostic, as introducing a term conditional on a separate latent (e.g. a CIC (Laskin et al. 2022) discriminator) purely for exploration would increase computational overhead and blur the semantics of the learned skills. Notationally, we use η to represent the parameters involved with this exploration module.

Skill Discovery. While exploration proceeds, DiMS continuously updates its GMVAE on mini-batches from the replay buffer. The training batches are sampled independently of those used for policy updates, in order to decorrelate GMVAE optimisation from policy learning. The decision to train the GMVAE online serves two key purposes: firstly, it lets the mixture prior associated with each cluster latent y track the state distribution induced by the agent, and it also removes the need for a separate, sample-heavy exploration phase. Crucially, the joint updates prevent distribution mismatch between the data the GMVAE sees and the states visited by the policy. Decoder likelihoods - and hence intrinsic rewards - are therefore evaluated only on states that are in support, resulting in a well-defined reward landscape.

Policy Learning. After initial exploration, DiMS trains a skill-conditioned policy $\pi_\theta(a|s,\tilde{z}) = \pi_\theta(a|s,w,y,z)$. The guidance signal r_{skill} is provided by the encoder $q_\phi(z|s)$, which encourages the policy to arrive at states that are associated with the latent skill, and is defined as:

$$r_{\text{skill}}(s,z) = -\log q_{\phi_z}(z|s). \tag{7}$$

The DiMS intrinsic reward is calculated in latent-space, rather than using the native state space (e.g. $p_\psi(s|\tilde{z})$, as is done in EDL). The latent co-ordinates are scale-invariant and shaped by an information bottleneck, making the signal more robust to irrelevant features and noise in state observations. Furthermore, we emphasise z for skill reward rather than cluster y or macro latent w because the more granular latent governs the exact behavioural target, as opposed to w and y which are more associated with defining high-level groupings.

Combined with the exploration bonus discussed in Equation 6, we arrive at the full intrinsic reward

$$\begin{aligned} r_{\text{DiMS}}(s,w,y,z) &= r_{\text{skill}}(s,w,y,z) + \beta r_{\text{explore}}(s) \\ &= -\log q_{\phi_z}(z|s,w,y) \\ &\quad + \beta \left\| f_{\text{pred}}(s) - f_{\text{target}}(s) \right\|_2^2, \end{aligned} \tag{8}$$

where β is annealed over time to enable agents to explore early, before specialising into skills later in training. Although maintaining a separate exploration term adds an extra hyperparameter, it provides considerable advantages during training - the agent is encouraged to reach bottleneck states that may not be easily accessible with the skill reward alone, thereby increasing both the number and diversity of discovered skills.

As a whole, DiMS retains the conceptual EDL cycle (Campos et al. 2020): *explore* with RND, *discover* skills via a generative model, and *learn* a skill-conditioned policy. However, among other differences, DiMS condenses these distinct stages into a single, self-contained process, considerably improving consistency.

We base our implementation of DiMS on the URLB framework (Laskin et al. 2021). This includes using DDPG (Lillicrap et al. 2015) as a base RL algorithm to provide a fair frame of comparison against peer methods in Section 6. Our implementations of exploration bonuses are also derived from those within URLB.

6 Experiments and Results

To understand the impact of DiMS, in this section we describe experiments that aim to answer the following questions: 1) What behavioural patterns emerge from DiMS and how does its mixture-latent structure manifest compared with existing methods? 2) How does DiMS compare to peer URL methodologies in terms of adaptation efficiency and downstream RL performance? 3) What are the benefits of the latent skill mixture model, compared to other latent parameterisations? 4) How robust is DiMS to the particular choice of exploration bonus?

6.1 Behavioural Analysis

Experiment Setup. To illustrate how the hierarchical mixture-latent model of DiMS shapes behaviour, we begin with a visual exploratory study in the two-dimensional *Bottleneck Maze* environment. This benchmark features a narrow central corridor that severely limits exploration, with many URL agents failing to effectively find a way past the bottleneck (Campos et al. 2020; Kim et al. 2023). Each episode begins with the agent at the bottom-left corner $(0,0)$.

For simplicity, we set the macro latent w to a standard Gaussian $N(0,1)$ and train DiMS without reward for 1M steps. In Figure 2 we select five equally-spaced values of w across the CDF. For each value, we sample 25 skills z per cluster latent y, yielding 100 trajectories per plot.

Results. First, it is clear that different values of w heavily influence cluster centre location, and therefore agent behaviour. With a simple single-dimension w, cluster centres systematically move across the maze area as we traverse the macro-latent space, producing skills that reliably reach the otherwise hard-to-access upper-left and upper-right rooms. Without

We also note that increasing the dimensions in w lead to more varied cluster patterns, which range between spreading out across all parts of the maze to being clustered in a particular room. Those additional runs are included in the supplementary material.

6.2 Unsupervised Reinforcement Learning Benchmark

Experiment Setup. The Unsupervised Reinforcement Learning Benchmark (URLB) (Laskin et al. 2021) is a well-known evaluation suite that couples a reward-free pretraining phase with twelve downstream control tasks drawn from three MuJoCo domains (*Walker*, *Quadruped*, and *Jaco*).

We compare DiMS with a wide variety of unsupervised RL methodologies:

- Representative knowledge-based and data-based methods chosen for their performance standard (RND (Burda et al. 2019), APT (Liu and Abbeel 2021b))
- Traditional skill discovery benchmarks (DIAYN (Eysenbach et al. 2019), EDL (Campos et al. 2020), APS (Liu and Abbeel 2021a), CSD (Park et al. 2023))
- Recent high-performance skill discovery algorithms (CIC (Laskin et al. 2022), BeCL (Yang et al. 2023), Metra (Park, Rybkin, and Levine 2024), CeSD (Bai et al. 2024)).

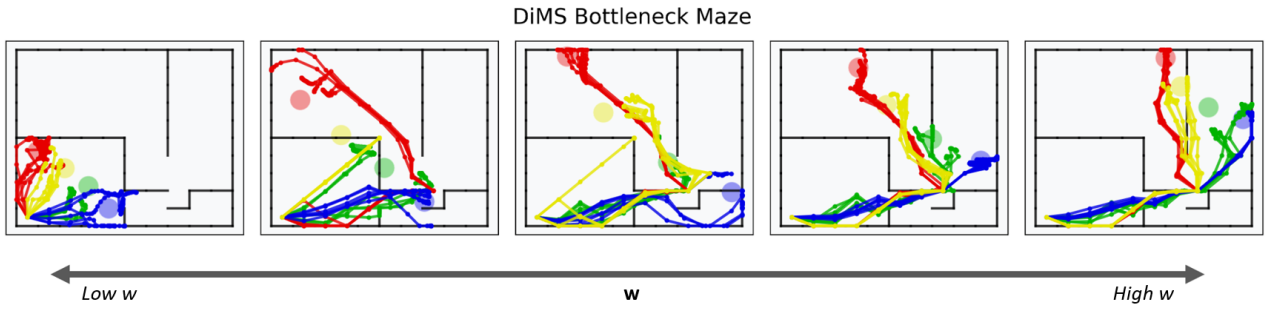

Figure 2: **DiMS Behaviour in 2D Maze.** Behaviour of DiMS agent on a 2D maze environment, across a variety of macro latents w ranging from low (left) to high (right). Reconstructed centres for each index of y are highlighted for visibility. The DiMS macro latent w enables learned cluster centres y to cover the entire state space. Cluster centre radius is indicative and not scaled.

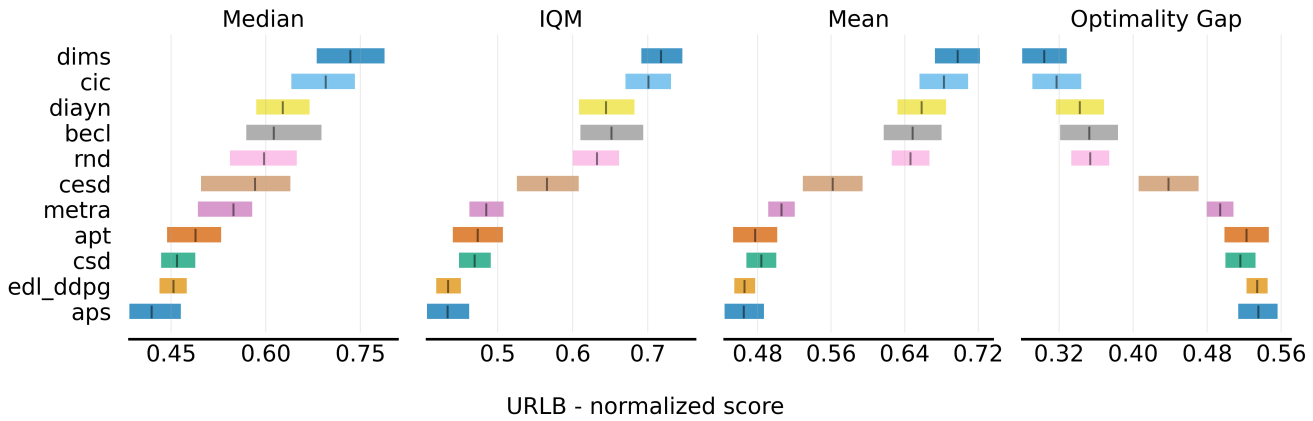

Figure 3: **URLB Results.** Performance of a variety of unsupervised RL methodologies on 12 downstream tasks from the URLB, using stratified bootstrap intervals from Agarwal et al. (2021). DiMS achieves state-of-the-art results in the benchmark.

We utilise original URLB implementations where available, and otherwise replicate the original work in the URLB framework (CSD, Metra, EDL). In addition, all agents, including DiMS, share the same base RL algorithm so differences arise solely from the intrinsic reward module. This particularly affects EDL, whose original implementation uses Sibling Rivalry (Trott et al. 2019) to improve base policy performance.

Every method is pretrained for 2M environment steps with intrinsic reward only, then finetuned for 100K steps on each downstream task using extrinsic reward. Similar to the protocol of Laskin et al. (2022), for skill discovery methods we allocate the first 4K finetuning steps for skill selection, where the agent tests a randomly-sampled skill every 100 steps, before fixing the highest-return skill for the remaining 96K finetuning steps. During this process, we reset the environment for every skill tested, ensuring that skills are tested from the starting state s_0.

As is standard, we run 10 random seeds per task-method combination (10 seeds $*$ 12 tasks $*$ 11 methods $=$ 1210 total runs), normalise results against an expert DDPG policy trained from scratch for 2M steps (sourced from Laskin et al. (2022)), and report aggregate performance with the stratified-bootstrap statistics recommended by RLIABLE (Agarwal et al. 2021). In particular, IQM discards the top and bottom 25% of seeds and calculates the mean of the remainder, and is recommended by the authors as the primary measure.

Results. Overall results on URLB are shown in Fig. 3. DiMS outperforms peer methodologies in aggregate across all 12 downstream tasks. Compared to its closest skill-discovery baseline, CIC, DiMS generally outperforms in *Jaco* manipulation ($p = 0.03$), while underperforming slightly on the locomotive *Walker* domain. We hypothesise that the more expressive latents provided by the GMVAE allow the agent to distinguish the various modes of manipulation in *Jaco*. However, *Walker* has a comparatively smaller set of potential skills, where the high-capacity latent space may hamper downstream finetuning.

When considering other baselines, we noticed that DIAYN scored noticeably higher than in the original URLB work (Laskin et al. 2021); the improvement stems from our skill-selection protocol which consistently evaluates all DIAYN skills from the initial state s_0 during the first 4k finetuning steps, as opposed to changing skills mid-trajectory. In contrast, EDL (with DDPG as a base) appears to underperform: without the Sibling Rivalry policy used in the original work exploration is significantly more difficult. Complete numeric

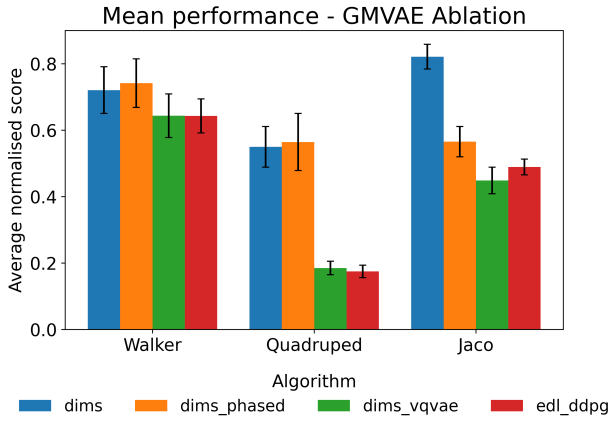

Figure 4: **GMVAE Structure Ablation.** Cumulative impact of key structural and training decisions. GMVAE vastly improves performance in *Quadruped*, while online training stabilises *Jaco* performance.

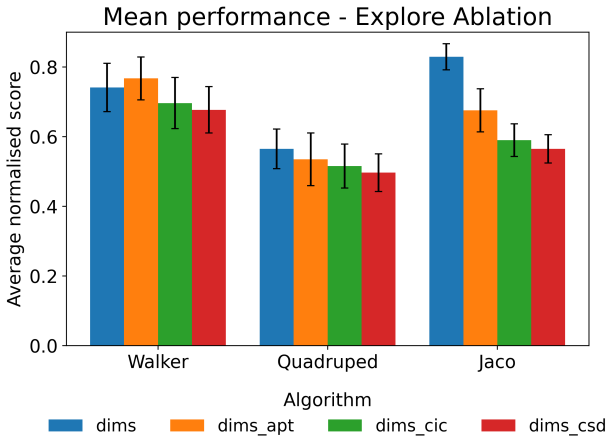

Figure 5: **Explore Bonus Ablation.** Performance sensitivity to exploration bonus in each domain. Skill-agnostic exploration bonuses (RND, APT) lead to best performance.

results are provided in the supplementary material.

6.3 Ablation Studies

GMVAE. The key features of DiMS lie in its hierarchical mixture-latent GMVAE and its online training scheme that trains the GMVAE jointly with the policy. Our first study aims to isolate the benefits of each component, comparing the performance of four variants on the URLB: (i) full DiMS, (ii) DiMS with phased exploration-discover-learn training stages, (iii) DiMS with a discrete VQ-VAE in place of the GMVAE, and (iv) a staged VQ-VAE version, which is essentially EDL (Campos et al. 2020) with a DDPG backbone.

In Figure 4, the full DiMS variant (*dims*) yields the strongest returns, confirming that both the GMVAE architecture and the online updates improve overall performance. When training is phased into explore–discover–learn stages (*dims_phased*), performance suffers particularly on the difficult Jaco manipulation tasks. This is likely due to distribution mismatch between the DiMS policy and the fixed buffer used to learn the GMVAE prior. Replacing the GMVAE with a discrete VQ-VAE (*dims_vqvae*) degrades performance further. Ultimately, these ablations show that the mixture latents introduced by DiMS give the expressiveness needed to embed complex tasks, while online updates minimise distribution mismatch with exploration data.

Exploration Bonus. During pretraining, DiMS augments its skill-conditioned intrinsic reward with a separate exploration bonus to maximise marginal entropy $H(S)$ and provide the GMVAE with a broad, balanced replay buffer for learning. In Figure 5 we analyse the sensitivity of DiMS performance with regards to four exploration signals: (i) RND (paper version) and (ii) APT represent skill-agnostic exploration bonuses that reward novelty, while (iii) CIC and (iv) CSD train an independent skill latent purely for exploration.

Across all three URLB domains, skill-agnostic bonuses (*dims*, *dims_apt*) clearly outperform exploration strategies that introduce a second latent (*dims_cic*, *dims_csd*). The first two options keep exploration independent of the skill variable, providing a clear separation between "where to go" and "how to behave", while in contrast learning a separate exploration latent creates competing objectives — one latent tries to maximise coverage while the GMVAE latent seeks discriminability. Notably, substituting RND with APT primarily sacrifices performance in *Jaco*, suggesting that the curiosity-based RND is more suited towards fine control tasks.

7 Conclusion

In this paper we presented *Discovery of Mixture Skills* (DiMS), a novel unsupervised-RL framework that pairs a hierarchical GMVAE with online skill learning and a skill-agnostic exploration bonus. By learning a Gaussian mixture prior conditioned on the macro latent, DiMS bridges the gap between discrete and continuous skill spaces and, empirically, yields state-of-the-art performance on the URLB suite while remaining robust across various ablations.

There are several interesting avenues to extend our work further. First, although DiMS has focused on mixture latents, there is a wide variety of variational structures to investigate for further improvement. One promising direction is the stick-breaking VAE (Nalisnick and Smyth 2017), whose non-parametric prior can adapt the number of mixture components online instead of fixing K. Additionally, our GMVAE - like with most URL approaches - seeks to reconstruct individual states (or transitions). Exploring recurrent VAEs to encode longer state trajectories or joint state-action context could capture more sophisticated behaviours and discover skills with explicit temporal structures, further broadening the repertoire of behaviours that can be learned in reward-free environments.

Mixture-aware skill discovery thus proves both practical and powerful, and the success of DiMS shows that agents can self-organise behaviourally diverse skill sets while they explore — an encouraging milestone on the road to fully autonomous competence discovery.

Acknowledgements

This work is supported by the Australian Research Council under Australian Laureate Fellowships FL190100149 and Discovery Early Career Researcher Award DE200100245.

References

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational Option Discovery Algorithms. *arXiv preprint arXiv:1807.10299*.

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A.; and Bellemare, M. G. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Advances in Neural Information Processing Systems*.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*. Toulon, France.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; and Zaremba, W. 2017. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*. Long Beach, USA.

Bai, C.; Yang, R.; Zhang, Q.; Xu, K.; Chen, Y.; Xiao, T.; and Li, X. 2024. Constrained Ensemble Exploration for Unsupervised Skill Discovery. *arXiv preprint arXiv:2405.16030*.

Bishop, C. M. 1994. Mixture Density Networks. Technical report, Neural Computing Research Group, Aston University.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*. Vancouver, Canada.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by random network distillation. In *International Conference on Learning Representations*. New Orleans, USA.

Campos, V.; Trott, A.; Xiong, C.; Socher, R.; Giro-I-Nieto, X.; and Torres, J. 2020. Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills. In *International Conference on Machine Learning*. Virtual Conference.

Dave, V.; and Rueckert, E. 2025. Skill Disentanglement in Reproducing Kernel Hilbert Space. In *AAAI Conference on Artificial Intelligence*. Philadelphia, USA.

Dilokthanakul, N.; Mediano, P. A.; Garnelo, M.; Lee, M. C.; Salimbeni, H.; Arulkumaran, K.; and Shanahan, M. 2016. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*. New Orleans, USA.

Fickinger, A.; Jaques, N.; Parajuli, S.; Chang, M.; Rhinehart, N.; Berseth, G.; Russell, S.; and Levine, S. 2021. Explore and Control with Adversarial Surprise. *arXiv preprint arXiv:2107.07394*.

Gregor, K.; Rezende, D. J.; and Wierstra, D. 2017. Variational Intrinsic Control. In *International Conference on Learning Representations*. Toulon, France.

Guo, Y.; Liao, W.; Wang, Q.; Yu, L.; Ji, T.; and Li, P. 2018. Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach. In *Asian Conference on Machine Learning*, 97–112.

Jiang, Z.; Gao, J.; and Chen, J. 2022. Unsupervised skill discovery via recurrent skill training. *Advances in Neural Information Processing Systems*, 35: 39034–39046.

Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; and Zhou, H. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*.

Kamienny, P.-A.; Tarbouriech, J.; Lazaric, A.; and Denoyer, L. 2022. Direct then Diffuse: Incremental Unsupervised Skill Discovery for State Covering and Goal Reaching. In *International Conference on Learning Representations*. Virtual Conference.

Kim, H.; Lee, B. K.; Lee, H.; Hwang, D.; Park, S.; Min, K.; and Choo, J. 2023. Learning to Discover Skills through Guidance. In *Advances in Neural Information Processing Systems*, volume 36, 28226–28254. New Orleans, USA.

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 34, 21696–21707. Virtual Conference.

Kviman, O.; Molén, R.; Hotti, A.; Kurt, S.; Elvira, V.; and Lagergren, J. 2023. Cooperation in the Latent Space: The Benefits of Adding Mixture Components in Variational Autoencoders. In *International Conference on Machine Learning*, 18008–18022. Hawaii, USA.

Ladosz, P.; Weng, L.; Kim, M.; and Oh, H. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85: 1–22.

Laskin, M.; Liu, H.; Peng, X. B.; Yarats, D.; Rajeswaran, A.; and Abbeel, P. 2022. Unsupervised Reinforcement Learning with Contrastive Intrinsic Control. In *Advances in Neural Information Processing Systems*. New Orleans, USA.

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Virtual Conference.

Lee, L.; Eysenbach, B.; Parisotto, E.; Xing, E. P.; Levine, S.; and Salakhutdinov, R. 2019. Efficient Exploration via State Marginal Matching. *arXiv preprint arXiv:1906.05274*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, H.; and Abbeel, P. 2021a. APS: Active Pretraining with Successor Features. In *International Conference on Machine Learning*. Virtual Conference.

Liu, H.; and Abbeel, P. 2021b. Behavior From the Void: Unsupervised Active Pre-Training. In *Advances in Neural Information Processing Systems*. Virtual Conference.

Nalisnick, E.; and Smyth, P. 2017. Stick-Breaking Variational Autoencoders. In *International Conference on Learning Representations*. Toulon, France.

Park, S.; Choi, J.; Kim, J.; Lee, H.; and Kim, G. 2022. Lipschitz-constrained Unsupervised Skill Discovery. In *International Conference on Learning Representations*. Virtual Conference.

Park, S.; Lee, K.; Lee, Y.; and Abbeel, P. 2023. Controllability-Aware Unsupervised Skill Discovery. In *International Conference on Machine Learning*. Hawaii, USA.

Park, S.; Rybkin, O.; and Levine, S. 2024. METRA: Scalable Unsupervised RL with Metric-Aware Abstraction. In *International Conference on Learning Representations*. Vienna, Austria.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning*. Sydney, Australia.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-Supervised Exploration via Disagreement. In *International Conference on Machine Learning*. Long Beach, USA.

Pong, V.; Dalal, M.; Lin, S.; Nair, A.; Bahl, S.; and Levine, S. 2020. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. In *International Conference on Machine Learning*. Virtual Conference.

Qing, S.; Sun, Y.; Ding, K.; Zhang, H.; and Zhu, F. 2025. CoSD: Balancing behavioral consistency and diversity in unsupervised skill discovery. *Neural Networks*, 182: 106889.

Qing, S.; and Zhu, F. 2024. Refine to the essence: Less-redundant skill learning via diversity clustering. *Engineering Applications of Artificial Intelligence*, 133: 107981.

Rajeswar, S.; Mazzaglia, P.; Verbelen, T.; Piché, A.; Dhoedt, B.; Courville, A.; and Lacoste, A. 2023. Mastering the unsupervised reinforcement learning benchmark from pixels. In *International Conference on Machine Learning*, 28598–28617. Honolulu, USA.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*.

Seo, Y.; Chen, L.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2021. State Entropy Maximization with Random Encoders for Efficient Exploration. In *International Conference on Machine Learning*. Virtual Conference.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *International Conference on Learning Representations*. Virtual Conference.

Shi, Y.; N, S.; Paige, B.; and Torr, P. 2019. Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models. In *Advances in Neural Information Processing Systems*. Vancouver, Canada.

Strouse, D.; Baumli, K.; Warde-Farley, D.; Mnih, V.; and Hansen, S. S. 2022. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*. Virtual Conference.

Trott, A.; Zheng, S.; Xiong, C.; and Socher, R. 2019. Keeping Your Distance: Solving Sparse Reward Tasks Using Self-Balancing Shaped Rewards. In *Advances in Neural Information Processing Systems*, volume 32. Vancouver, Canada.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Xie, Z.; Lin, Z.; Li, J.; Li, S.; and Ye, D. 2022. Pretraining in Deep Reinforcement Learning: A Survey. *arXiv preprint arXiv:2211.03959*.

Yang, R.; Bai, C.; Guo, H.; Li, S.; Zhao, B.; Wang, Z.; Liu, P.; and Li, X. 2023. Behavior Contrastive Learning for Unsupervised Skill Discovery. In *International Conference on Machine Learning*. Hawaii, USA.

Yang, Y.; Zhou, T.; Han, L.; Fang, M.; and Pechenizkiy, M. 2024. Automatic Curriculum for Unsupervised Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 2002–2010. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems.

Ying, C.; Chen, H.; Zhou, X.; Hao, Z.; Su, H.; and Zhu, J. 2025. Exploratory Diffusion Model for Unsupervised Reinforcement Learning. *arXiv preprint arXiv:2502.07279*.

Zhang, C.; Cai, Y.; Huang, L.; and Li, J. 2021. Exploration by Maximizing Renyi Entropy for Reward-Free RL Framework. In *AAAI Conference on Artificial Intelligence*. Virtual Conference.

Zhao, A.; Lin, M.; Li, Y.; Liu, Y.-j.; and Huang, G. 2022. A Mixture Of Surprises for Unsupervised Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 35, 26078–26090. New Orleans, USA.