FROM TABULA RASA TO EMERGENT ABILITIES: DISCOVERING ROBOT SKILLS VIA RESET-FREE UNSUPERVISED QUALITY-DIVERSITY

Luca Grillotti, **Maxence Faldor,** **Antoine Cully**
Department of Computing
Imperial College London
United Kingdom
{luca.grillotti16, m.faldor22, a.cully}@imperial.ac.uk

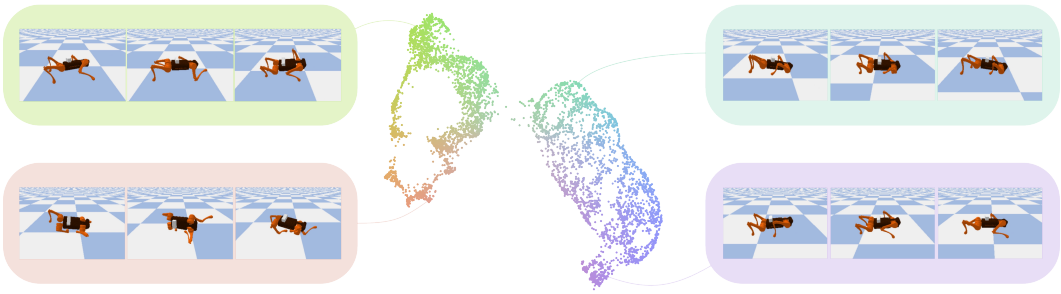

Figure 1: We propose Unsupervised Reset-free Skill Acquisition (URSA), a framework for discovering diverse unsupervised skills in reset-free environments. The visualization shows UMAP embeddings (Sainburg et al., 2021) of state trajectories from different skills discovered by our approach, demonstrating the diversity of learned behaviors.

ABSTRACT

Autonomous skill discovery in robotics seeks to enable robots to acquire diverse behaviors without explicit human guidance. However, learning such behaviors directly in the real world remains challenging due to the high number of interactions required. Existing approaches typically rely either on learning in simulated environments before real-world deployment, or on carefully designed heuristics. While the former face challenges when transferring to real robots due to the reality gap, the latter may require domain expertise to design effective heuristics. The recent algorithm Quality-Diversity Actor-Critic (QDAC) has shown promise in discovering diverse high-performing behaviors, yet its application to reset-free robotics remains limited due to safety concerns and the requirement for skills to be manually defined beforehand. Here, we propose Unsupervised Reset-free Skill Acquisition (URSA), an extension of QDAC that enables robots to autonomously discover and master diverse skills directly in reset-free environments, without prior knowledge of the skill space. URSA manages to discover diverse velocity and unsupervised skills on a Unitree A1 quadruped robot in simulation. These results establish a new framework for reset-free robot learning that enables continuous skill discovery with a small amount of human intervention, representing a significant step toward more autonomous and adaptable robotic systems.

1 INTRODUCTION

Autonomous skill discovery in robotics represents a promising direction that could transform how robots learn and adapt, paving the way for versatile and general-purpose robotic systems. While specialized robots have demonstrated remarkable proficiency in specific tasks like PCB insertion (Luo et al., 2024) and legged locomotion (Kostrikov et al., 2023), they are mostly limited to learning a single

skill or behavior. This narrow specialization makes them vulnerable - if the robot sustains unexpected damage, it may be unable to adapt and find alternative ways to complete its task. Therefore, developing systems that can autonomously learn a repertoire of diverse behaviors is crucial for creating more robust and adaptable robots (Cully et al., 2015), similar to how humans naturally develop multiple walking gaits and can adapt by switching to alternative movement patterns when injured. This capability would allow robots to adapt to unexpected situations and discover alternative strategies when their primary approach fails — a key characteristic of human intelligence and adaptability.

Traditional approaches to skill discovery have primarily relied on carefully engineered heuristics to guide the learning process (Sharma et al., 2020; Smith et al., 2023). While these hand-crafted approaches provide structure, they often overly restrict the robot's exploration, preventing it from discovering creative behaviors. Moreover, simulation-based approaches (Margolis & Agrawal, 2023; Lim et al., 2022a; Hoeller et al., 2024) require accurate robot models, which can be challenging to obtain and maintain, especially for complex systems.

In this work, we present Unsupervised Reset-free Skill Acquisition (URSA), a novel Reinforcement Learning (RL) framework for unsupervised robot skill discovery in reset-free environments, addressing the challenges of sample efficiency and reset-free learning. Our approach builds upon the Quality-Diversity Actor-Critic (Grillotti et al., 2024) (QDAC) algorithm and extends the DayDreamer (Wu et al., 2022) paradigm of alternating between reset-free data collection and model-based training. While DayDreamer focuses on learning a single policy for a specific task, we adapt it to enable open-ended skill discovery in an unsupervised manner. Furthermore, we incorporate safety constraints to ensure that the robot explores and learns diverse skills without risking damage.

Our key contributions are as follows:

- We introduce a novel formulation of the unsupervised skill discovery problem, which includes the discovery of the reachable skill space and the learning of a policy that can achieve a diverse set of skills from this space.
- We propose URSA, an unsupervised extension of QDAC that addresses this problem by utilizing a diversity-aware repertoire and a variational autoencoder to learn meaningful skill representations directly from state observations.
- We introduce a novel sampling method for target skills, using a gaussian kernel density estimator to focus on the reachable skill space and promote the discovery of distinct and diverse behaviors.
- We adapt the DayDreamer paradigm to unsupervised skill discovery, enabling efficient reset-free learning through alternating phases of data collection and model-based training.
- We demonstrate the effectiveness of our approach through extensive experiments on a Unitree A1 quadruped robot in simulation, showcasing autonomous skill discovery and robustness to actuators failures.

Our work contributes to enabling robots to autonomously discover and master a range of skills in reset-free settings, helping advance the development of more versatile robotic systems. By combining unsupervised learning and safety considerations, we provide a comprehensive framework for reset-free robot skill discovery that addresses key challenges in the field of autonomous robotics.

2 Problem Statement: A New Perspective on Skill Discovery

We consider a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, r, p)$ (Sutton & Barto, 2018). At each timestep t, the agent is in a state $\boldsymbol{s}_t \in \mathcal{S}$ and chooses an action $\boldsymbol{a}_t \in \mathcal{A}$ leading to a new state $\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$, and providing the agent with a reward $r_t = r(\boldsymbol{s}_t, \boldsymbol{a}_t)$. Consider we have access to a feature function $\phi : \mathcal{S}, \mathcal{A} \to \mathcal{Z}$, which can be either learned from data or provided by the user. These features capture instantaneous properties of the agent's behavior at each timestep, such as the robot's velocity or its foot contact configuration with the ground. To characterize the agent's behavior over an extended period of time, we introduce the concept of skill, which we formally define as follows.

Definition 2.1 (Skill)**.** The skill $\boldsymbol{z} \in \mathcal{Z}$ of a policy π is defined as the expected feature vector $\mathbb{E}_\pi\left[\phi(\boldsymbol{s}, \boldsymbol{a})\right]$ under the policy's stationary distribution.

This expectation captures the characteristic behavior that emerges when executing the policy. For example, consider a quadrupedal robot where the features $\boldsymbol{\phi}_t$ characterize which feet are in contact

with the ground at each timestep t, with $\boldsymbol{\phi}_t[i] = 1$ if the i-th foot is touching the ground and 0 otherwise. In this case, the i-th component of the skill $\boldsymbol{z}$ represents the proportion of time during which the i-th foot is in contact with the ground (i.e., the foot contact rate). The skill space thus characterizes the diverse ways the robot can locomote by capturing how frequently each leg is used. For instance, achieving a skill $\boldsymbol{z} = [0.8 \quad 0.3 \quad 0.8 \quad 0.3]^\intercal$ requires the robot to maintain the left feet in contact with the ground 80% of the time while using the right feet only 30% of the time over a trajectory of multiple timesteps, potentially corresponding to a limping gait.

Within the skill space $\mathcal{Z}$, we can distinguish two subspaces that characterize the capabilities of our system: the reachable skill space, which encompasses all theoretically attainable skills, and the achieved skill space, which represents the skills actually mastered by our policy.

Definition 2.2 (Reachable Skill Space)**.** The reachable skill space $\mathcal{Z}_p \subseteq \mathcal{Z}$ is the set of all skills $\boldsymbol{z} \in \mathcal{Z}$ for which some policy π can achieve them:

$$\mathcal{Z}_p = \{\boldsymbol{z} \in \mathcal{Z} \mid \exists \pi, \mathbb{E}_\pi \left[\phi(\boldsymbol{s}, \boldsymbol{a})\right] = \boldsymbol{z}\}$$

Definition 2.3 (Achieved Skill Space of a Policy)**.** Consider a skill-conditioned policy π, and for all skill $\boldsymbol{z} \in \mathcal{Z}$, we write $\pi_{\boldsymbol{z}}$ the skill-conditioned policy $\pi(\cdot|\cdot, \boldsymbol{z})$. A skill $\boldsymbol{z} \in \mathcal{Z}$ is said to be *achieved* by π if and only if $\mathbb{E}_{\pi_{\boldsymbol{z}}} \left[\phi(\boldsymbol{s}, \boldsymbol{a})\right] = \boldsymbol{z}$. The achieved skill space $\mathcal{Z}_\pi \subseteq \mathcal{Z}$ is the set of all achieved skills:

$$\mathcal{Z}_\pi = \{\boldsymbol{z} \in \mathcal{Z} \mid \mathbb{E}_{\pi_{\boldsymbol{z}}} \left[\phi(\boldsymbol{s}, \boldsymbol{a})\right] = \boldsymbol{z}\}$$

By construction, it follows that $\mathcal{Z}_\pi \subseteq \mathcal{Z}_p \subseteq \mathcal{Z}$. In this work, we assume that both $\mathcal{Z}_p$ and $\mathcal{Z}_\pi$ are bounded, though their exact boundaries are unknown. We aim at discovering and characterizing these spaces through learning, with the ultimate goal of maximizing both the diversity and quality of skills mastered by the robot.

Specifically, our work has two key objectives: maximizing the volume of the achieved skill space $\mathrm{vol}\left(\mathcal{Z}_\pi\right)$ to enable diverse behaviors, while simultaneously maximizing the expected return for each mastered skill to ensure high performance. More formally, we intend to solve the following problem, where $\mathrm{vol}\left(\cdot\right)$ denotes the Lebesgue measure, i.e. the n-dimensional volume function:

$$\text{maximize } \mathrm{vol}\left(\mathcal{Z}_\pi\right) \quad \text{and} \quad \forall t, \forall \boldsymbol{z} \in \mathcal{Z}_\pi, \text{ maximize } \mathbb{E}_{\pi_{\boldsymbol{z}}} \left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right]$$

To address the tractability of this optimization problem, we learn a surrogate probability distribution $q(\cdot)$ defined over the skill space $\mathcal{Z}$, which approximates a uniform distribution over the reachable skill space $\mathcal{Z}_p$. The above problem can be separated into two tractable subproblems to be solved simultaneously:

- maximizing the entropy of the skill distribution q over the reachable skill space $\mathcal{Z}_p$.
- learning a skill-conditioned policy π that maximizes the expected return while achieving each sampled skill $\boldsymbol{z} \sim q$:

$$\text{maximize} \quad \mathbb{E}_{\pi_{\boldsymbol{z}}} \left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right] \quad \text{subject to} \quad \mathbb{E}_{\pi_{\boldsymbol{z}}} \left[\phi\left(\boldsymbol{s}, \boldsymbol{a}\right)\right] = \boldsymbol{z} \tag{P1}$$

In summary, our approach aims to discover and master a diverse set of skills directly in a reset-free manner, without requiring any predefined skill space or manual feature engineering. By learning a surrogate uniform distribution q over the reachable skill space and maximizing the performance for each sampled skill $\boldsymbol{z} \sim q$, we enable physical robots to autonomously build a repertoire of diverse and useful behaviors through reset-free interactions.

3 Background

3.1 Quality-Diversity Actor-Critic (QDAC)

This work builds upon the QDAC algorithm (Grillotti et al., 2024), and extends it to (1) a reset-free setting and (2) unsupervised skill discovery. QDAC aims at finding a skill-conditioned policy $\pi_{\boldsymbol{z}} = \pi(\cdot|\cdot, \boldsymbol{z})$ that maximizes the reward while following a given skill $\boldsymbol{z}$.

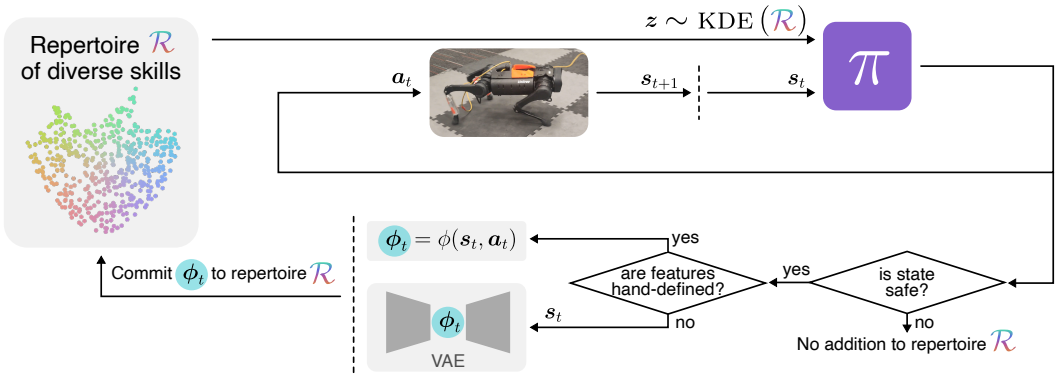

Figure 2: Overview of the Unsupervised Reset-free Skill Acquisition (URSA) framework. The system first checks if the current state $\boldsymbol{s}_t$ is safe, and if so, encodes it into low-dimensional features $\boldsymbol{\phi}_t$. These features are collected to build a repertoire $\mathcal{R}$ of diverse skills. Using this repertoire as input to a Kernel Density Estimator (KDE), the system periodically samples new skills $\boldsymbol{z}$ uniformly from the safe and reachable skill space. Finally, the skill-conditioned policy π maximizes its expected return while performing behaviors that match the sampled skill $\boldsymbol{z}$.

QDAC assumes a feature function $\phi(\cdot)$ is provided: for every state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$, the agent's feature is $\boldsymbol{\phi}_t = \phi(\boldsymbol{s}_t, \boldsymbol{a}_t)$. We consider the *value function* V (Sutton & Barto, 2018) and *successor features* $\boldsymbol{\psi}$ (Barreto et al., 2017), respectively defined as the discounted sum of rewards r_t and features $\boldsymbol{\phi}_t$ departing from $\boldsymbol{s}$ and following $\boldsymbol{z}$ with policy $\pi_{\boldsymbol{z}}$: $V(\boldsymbol{s}, \boldsymbol{z}) = \mathbb{E}_{\pi_{\boldsymbol{z}}}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| \boldsymbol{s}_t = \boldsymbol{s}\right]$ and $\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z}) = \mathbb{E}_{\pi_{\boldsymbol{z}}}\left[\sum_{i=0}^{\infty} \gamma^i \boldsymbol{\phi}_{t+i} \middle| \boldsymbol{s}_t = \boldsymbol{s}\right]$.

In addition, QDAC aims at solving problem P1 by making two approximations: (1) the expected state is approximated by the discounted average $(1-\gamma)\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z})$, and (2) the strict equality constraint from problem P1 is replaced by an inequality constraint forcing the policy to stay close to the target skill $\boldsymbol{z}$. The problem then becomes:

$$\forall \boldsymbol{z} \sim \mathcal{U}(\mathcal{Z}_p), \quad \text{maximize } V(\boldsymbol{s}, \boldsymbol{z}) \quad \text{subject to } \left\|(1-\gamma)\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z}) - \boldsymbol{z}\right\|_2 \leq \delta \tag{P2}$$

where δ is a hyperparameter that determines the maximal acceptable distance between the expected state and the skill.

Our approach addresses two key limitations of the original QDAC algorithm. Rather than requiring a predefined reachable skill space $\mathcal{Z}_p$ and fixed constraint threshold δ, our approach learns both the structure of the skill space and how to efficiently sample from it during training, while adaptively tuning δ. Additionally, instead of relying on pre-defined features, our approach learns them directly from unsupervised state encodings.

3.2 Real-world Robot Learning with DayDreamer

DayDreamer (Wu et al., 2022) is a world model-based reinforcement learning algorithm that enables efficient robot learning directly in the real world without simulators. At its core is a world model that learns to predict environment dynamics through an encoder-decoder architecture and a recurrent state-space model (RSSM). This model enables planning through imagined rollouts in a learned latent space, making the learning process more sample efficient.

The algorithm employs an actor-critic architecture that maximizes performance by planning within the learned world model's latent space. The algorithm uses a parallel training structure where data collection and model learning happen simultaneously: a learner thread continuously trains the world model and policy while an actor thread computes actions for reset-free interaction. This asynchronous architecture makes DayDreamer particularly suitable for physical robot learning where data collection is costly and time-consuming, as it can efficiently learn from limited reset-free interaction by leveraging imagination. Our approach extends DayDreamer's asynchronous architecture by incorporating unsupervised skill discovery into both the learner and actor threads.

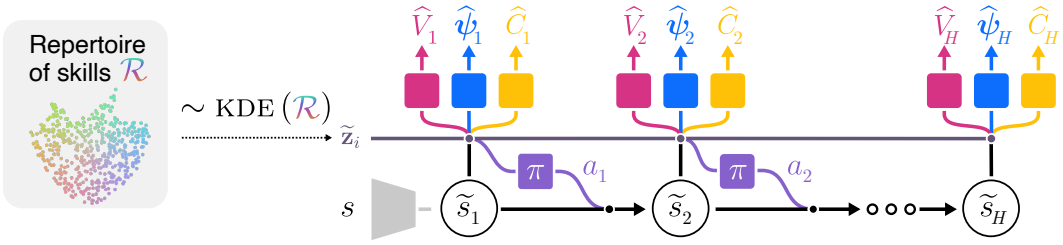

Figure 3: Imagination rollout performed within the world model. Each individual imagination rollout generates transitions following a target skill $\boldsymbol{z}$, starting from an initial state $\boldsymbol{s}$ for a fixed number of steps H. The world model predicts the reward r, feature vector $\boldsymbol{\phi}$; and uses them to train the networks parameterizing the value function V, the successor features $\boldsymbol{\psi}$, the cost function C, and the policy π.

4 Reset-Free Unsupervised Skill Discovery

In this work, we propose Unsupervised Reset-free Skill Acquisition (URSA), an RL-based approach that enables robots to autonomously discover and master diverse skills directly in reset-free environments, without prior knowledge of the skill space. To achieve this goal, URSA extends QDAC (Grillotti et al., 2024) with three key components. First, it incorporates safety constraints that prevent the robot from executing potentially dangerous skills. Second, it maintains a skill repertoire $\mathcal{R}$ that stores discovered and validated skills. Third, it includes an optional learnable feature function $\phi(\cdot)$ that automatically constructs a compact representation of the skill space from raw state observations.

4.1 Ensuring Safe Skill Discovery

In the context of reset-free robotics, some skills can be potentially dangerous for the robot, such as falling over. We are primarily interested in avoiding these skills since they are neither useful for the robot's operation nor worth learning, as they could potentially damage the robot. To address this concern, we introduce a safety mechanism following the same formalism as constrained reinforcement learning (Altman, 1999).

We define a safety set $\mathcal{S}_{\text{safe}} \subseteq \mathcal{S}$, which represents the subset of states that are considered safe for the robot. Our goal is to ensure that the agent avoids states outside of this safety set. To achieve this, we consider a cost function $c : \mathcal{S} \to \mathbb{R}$ that: $\boldsymbol{s} \in \mathcal{S}_{\text{safe}}$ if and only if $c_t = c(\boldsymbol{s}_t) \leq 0$. We then ensure that the associated critic $C(\boldsymbol{s}, \boldsymbol{z}) = \mathbb{E}_{\pi_{\boldsymbol{z}}}\left[\sum_{i=0}^{\infty} \gamma^i c_{t+i} | \boldsymbol{s}_t = \boldsymbol{s}\right]$ remains non-positive. This constraint helps the agent learn to avoid unsafe states during optimization. The incorporation of these safety constraints leads to a modified optimization problem:

$$\text{maximize } V(\boldsymbol{s}, \boldsymbol{z}) \quad \text{subject to } \|(1-\gamma)\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z}) - \boldsymbol{z}\|_2 \leq \delta \quad \text{and } C(\boldsymbol{s}, \boldsymbol{z}) \leq 0 \tag{P3}$$

To solve this problem, we consider the min-max optimization of the Lagrangian:

$$\max_{\pi} \min_{\lambda_1, \lambda_2 \geq 0} V(\boldsymbol{s}, \boldsymbol{z}) - \lambda_1(\|(1-\gamma)\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z}) - \boldsymbol{z}\|_2 - \delta) - \lambda_2 C(\boldsymbol{s}, \boldsymbol{z})$$

where λ_1 and λ_2 are the Lagrange multipliers associated with the distance to skill and safety constraints, respectively.

In practice, we found that optimizing the following actor objective with $0 \leq \lambda_1, \lambda_2 \leq 1$ leads to more stable training:

$$J_\pi = (1-\lambda_1)(1-\lambda_2)V(\boldsymbol{s}, \boldsymbol{z}) - \lambda_1(1-\lambda_2)(\|(1-\gamma)\boldsymbol{\psi}(\boldsymbol{s}, \boldsymbol{z}) - \boldsymbol{z}\|_2 - \delta) - \lambda_2 C(\boldsymbol{s}, \boldsymbol{z})$$

When λ_2 increases, the agent places greater emphasis on safety, even at the expense of reward maximization and skill execution. When λ_2 is low, increasing λ_1 causes the agent to focus more on executing the target skill rather than maximizing rewards. With this objective, the agent will first learn to satisfy safety constraints, then learn how to reach target skills $\boldsymbol{z}$, and finally learn how to maximize the reward.

Both Lagrange multipliers λ_1 and λ_2 are parameterized as neural networks that take the current state $\boldsymbol{s}$ and target skill $\boldsymbol{z}$ as inputs, since different state-skill pairs may require different trade-offs between the competing objectives. During training, the parameters of these networks are continuously optimized through gradient descent to balance between the competing objectives - increasing emphasis on safety when constraints are violated, focusing on skill execution when the agent deviates from the target behavior, and prioritizing reward maximization when both safety and skill objectives are satisfied. For instance, λ_1 increases when the agent struggles to execute the desired skill $\boldsymbol{z}$, putting more emphasis on skill execution, and decreases when the agent successfully executes the skill, shifting focus to reward maximization. Similarly, λ_2 increases when safety constraints are violated and decreases when the agent maintains safe operation.

To optimize for these objectives and learn all these critics, URSA uses the same approach as the model-based variant of QDAC (Grillotti et al., 2024), which uses world models to optimize policies efficiently. In this work, we adopt DayDreamer's (Wu et al., 2022) architecture that runs two asynchronous processes: one process interacts with the reset-free environment to collect data (Algorithm 1), while the other process continuously trains the world model and optimizes the policy and critics through imagination (Algorithm 2, Figure 3). This separation allows us to maximize learning efficiency by training continuously.

4.2 Efficient Skill Sampling from the Reachable Space

To achieve effective skill sampling in an unbounded skill space, URSA addresses four key challenges: (1) providing a flexible parameterization of the surrogate sampling distribution, (2) maximizing the entropy of this distribution to ensure uniform sampling, (3) considering only safe and reachable skills, and (4) adaptively tuning the threshold δ to find the optimal balance between strict and relaxed skill execution constraints. Figure 2 and Algorithm 1 provide high-level overviews of the skill collection and sampling process.

Adaptive Skill Distribution via Non-Parametric Density Estimation To sample skills from the reachable skill space $\mathcal{Z}_p$, we use a surrogate distribution q that focuses sampling on physically achievable skills. We model this surrogate distribution q using a Gaussian Kernel Density Estimator (KDE) (Parzen, 1962; Rosenblatt, 1956), which provides a flexible, non-parametric way to capture the structure of the reachable skill space. For a fixed-size repertoire of skills $\mathcal{R} = \{\boldsymbol{z}_i\}_{i=1}^{N_\mathcal{R}}$, the surrogate distribution is given by:

$$q = \text{KDE}\left(\mathcal{R}\right) = \frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma}$ represents the repertoire's skill covariance matrix, with a scaling factor of $N_\mathcal{R}^{-\frac{1}{D+4}}$ (where D denotes the dimensionality of the skill space), following the approach of Scott (1992).

Maximizing Entropy for Uniform Skill Sampling To approximate a uniform sampling of the reachable skill space, we maximize the entropy of the sampling distribution q. This entropy can be approximated via Monte-Carlo sampling with the skills from the repertoire:

$$\text{H}\left(q\right) = -\int q(x) \log q(x) dx \quad \approx \quad -\frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} q(\boldsymbol{z}_i) \log q(\boldsymbol{z}_i) = \widehat{\text{H}}(q)$$

This estimated entropy, written as $\widehat{\text{H}}(q)$, has a lower bound that only depends on the covariance matrix $\boldsymbol{\Sigma}$ and the distances between the skills in the repertoire $\boldsymbol{z}_i$ and their nearest neighbors $\boldsymbol{z}_i^{\text{nn}}$. Using the Mahalanobis distance $d_{\boldsymbol{\Sigma}}(\cdot, \cdot)$ with respect to the covariance matrix $\boldsymbol{\Sigma}$, we can derive the following lower bound, where k is a constant:

$$\widehat{\text{H}}(q) \geq -\frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} \log\left(1 + (N_\mathcal{R} - 1) e^{-\frac{1}{2} d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i, \boldsymbol{z}_i^{\text{nn}})^2}\right) + \frac{1}{2} \log\left(\det(\boldsymbol{\Sigma})\right) + k \tag{1}$$

This lower bound, whose derivation is provided in Appendix A, provides a tractable objective for maximizing the entropy of our sampling distribution q through the Mahalanobis distances between skills. To maximize this bound, we continuously update the repertoire by replacing low-diversity skills with newly discovered ones that increase the distribution's spread.

Algorithm 1 URSA – Data Collection

input Parameters θ_π ▷ Initial parameters for the actor
  $\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay buffer
  repeat
    $\boldsymbol{z} \sim \text{KDE}(\mathcal{R})$
    for T steps **do**
      $\boldsymbol{a}_t \sim \pi(\cdot|\boldsymbol{s}_t, \boldsymbol{z})$
      $\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$
      $r_t \leftarrow r(\boldsymbol{s}_t, \boldsymbol{a}_t)$
      $\boldsymbol{\phi}_t \leftarrow \phi(\boldsymbol{s}_t, \boldsymbol{a}_t)$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_t, \boldsymbol{s}_{t+1})\}$
      if $\boldsymbol{s}_t \in \mathcal{S}_{\text{safe}}$ **then**
        commit $\phi(\boldsymbol{s}_t, \boldsymbol{a}_t)$ to repertoire $\mathcal{R}$
    Fetch θ_π and θ_ϕ (if unsupervised) from Algo. 2
  until convergence

All highlighted components are specific to URSA

Algorithm 2 URSA – Training

input Parameters $\theta_{(\cdot)}$ ▷
  Initial parameters for the actor π, critics V, $\boldsymbol{\psi}$ and C, Lagrange multipliers λ_i, world model $\mathcal{W}$, and feature extractor ϕ (if unsupervised)
  repeat
    Fetch $\mathcal{D}$ and $\mathcal{R}$ from Algo. 1
    $\widetilde{\boldsymbol{z}}_i \sim \text{KDE}(\mathcal{R})$ for $i \in \{1, \ldots, N\}$
    $\theta_{\mathcal{W}} \leftarrow \theta_{\mathcal{W}} - \alpha_{\mathcal{W}} \nabla J_{\mathcal{W}}(\theta_{\mathcal{W}})$
    ▷Training from rollouts in $\mathcal{W}$ with skills $\widetilde{\boldsymbol{z}}_i$ (Fig. 3)
    $\theta_{\lambda_i} \leftarrow \theta_{\lambda_i} - \alpha_\lambda \nabla J_{\lambda_i}(\theta_{\lambda_i})$ for $i \in \{1, 2\}$
    $\theta_V \leftarrow \theta_V - \alpha_V \nabla J_V(\theta_V)$
    $\theta_{\boldsymbol{\psi}} \leftarrow \theta_{\boldsymbol{\psi}} - \alpha_{\boldsymbol{\psi}} \nabla J_{\boldsymbol{\psi}}(\theta_{\boldsymbol{\psi}})$
    $\theta_C \leftarrow \theta_C - \alpha_C \nabla J_C(\theta_C)$
    $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \nabla J_\pi(\theta_\pi)$
    if unsupervised **then**
      $\theta_\phi \leftarrow \theta_\phi - \alpha_\phi \nabla J_\phi(\theta_\phi)$ with $\mathcal{R}$
  until convergence

Filling the Repertoire with Safe and Reachable Skills As skills are defined as the expected feature vector $\mathbb{E}_{\pi_{\boldsymbol{z}}}[\boldsymbol{\phi}_t]$ (see Definition 2.1), any observed feature vector $\boldsymbol{\phi}_t$ can be treated as a potential new skill. Since each skill in the repertoire is derived from observations, URSA ensures that the sampling distribution q concentrates on regions of the skill space that are reachable by the robot. This construction helps focus exploration on the reachable skill space $\mathcal{Z}_p$ rather than wasting efforts on unreachable skills.

We take an additional precautionary measure by excluding unsafe states — i.e. states $\boldsymbol{s}$ which are not in the safety set $\mathcal{S}_{\text{safe}}$ — from the diversity-aware repertoire to ensure that the agent will not execute or attempt to learn unsafe skills. For example, if the robot finds itself upside down, we exclude any features observed in this unsafe configuration to prevent the system from learning skills that could damage the robot. By implementing these safety measures, we can guide the robot towards discovering and mastering a diverse set of skills while maintaining safe operation. The features resulting from safe states are all committed to the repertoire to maximize diversity within the reachable space.

Every time a feature $\boldsymbol{\phi}_t$ is committed to the repertoire, we compute all distances $d_{\boldsymbol{\Sigma}}(\boldsymbol{z}, \boldsymbol{z}^{\text{nn}})$ for all $\boldsymbol{z} \in \mathcal{R} \cup \{\boldsymbol{\phi}_t\}$, and remove the skill with the smallest distance. This process maintains a constant repertoire size while continuously increasing its diversity through the discovery of new skills. Assuming the impact of those updates on $\boldsymbol{\Sigma}$ is negligible, this process maximizes the lower bound on the approximated entropy of the skill-sampling distribution q (see Equation 1). This process ensures that the repertoire maintains a uniform coverage of the safe and reachable skill space by continuously incorporating features discovered during exploration.

Dynamic Threshold As the repertoire expands, the increasing distance between skills and their nearest neighbors requires adjusting the threshold δ. This threshold, which bounds the maximum distance between the expected state $(1-\gamma)\boldsymbol{\psi}$ and skill $\boldsymbol{z}$ (see Problem P3), must be dynamically updated to maintain achievable constraints. To that end, we define an hyperparameter $N_{\boldsymbol{z}}$ that controls the number of distinct skills the robot can execute in practice, such that $N_{\boldsymbol{z}} \leq N_{\mathcal{R}}$. The threshold δ is dynamically set by computing what the mean nearest-neighbor distance would be if the repertoire $\mathcal{R}$ contained exactly $N_{\boldsymbol{z}}$ skills that were uniformly distributed in the skill space. To compute this threshold, we first calculate the mean pairwise distance between each skill $\boldsymbol{z}_i$ and its nearest neighbor $\boldsymbol{z}_i^{\text{nn}}$ in $\mathcal{R}$, then apply a scaling factor based on the target number of distinct skills $N_{\boldsymbol{z}}$.

$$\delta = \left(\frac{N_{\mathcal{R}}}{N_{\boldsymbol{z}}}\right)^{1/D} \frac{1}{N_{\mathcal{R}}} \sum_{i=1}^{N_{\mathcal{R}}} d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i, \boldsymbol{z}_i^{\text{nn}})$$

This dynamic threshold helps ensure that we can choose $N_{\boldsymbol{z}}$ skills from $\mathcal{R}$ without overlap: when the agent executes a skill $\boldsymbol{z}$, it is unlikely that the constraint will be satisfied for other skills in $\mathcal{R}$.

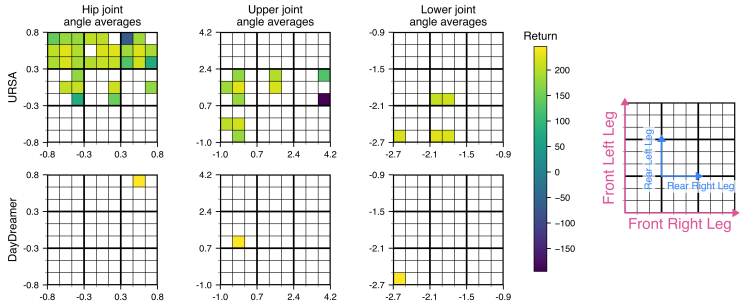

Figure 4: Visualization of how our learned skills cover different joint angles of the robot. These heatmaps visualize the range of motion achieved by each joint type (Hip, Upper and Lower) across all four legs of the robot. Each colored cell represents a region of joint angles that the algorithm can achieve through skills in its repertoire $\mathcal{R}$. For each skill, we compute the average joint angles during its execution, and color the corresponding cell if at least one skill's averages fall within that region.

4.3 Unsupervised Skill Discovery

In scenarios where a feature function is not provided, URSA learns one automatically through unsupervised learning. Specifically, we employ a feature function $\phi(\cdot)$ that transforms raw state observations into a compact representation, which then defines our skill space $\mathcal{Z}$. We implement this feature function using a variational autoencoder (VAE) Kingma & Welling (2014). The VAE takes as input the robot's zeroth-order kinematics variables from $\mathcal{S}$, such as the joints angles and the robot height, at each timestep t. These instantaneous state observations are encoded into a lower-dimensional latent space, and similar to Definition 2.1, a skill $\boldsymbol{z}$ is defined as the expected latent encoding under the policy's stationary distribution. This architecture enables us to obtain a meaningful and compressed representation of skills that naturally emerges from the robot's state observations over time.

The VAE is trained using the diverse collection of states accumulated in our repertoire $\mathcal{R}$. This unsupervised approach enables the robot to autonomously discover and acquire diverse behaviors without requiring any manual skill definition or human supervision.

5 Results

5.1 Experimental Setup

All experiments were conducted on the A1 robot from Unitree in a simulated PyBullet environment. This simulated environment is reset-free, meaning that the environment is never reset to a specific state. The state space is defined by the joint angles and velocities of the robot, and the action space is defined by the joint target positions. We use a discount factor of $\gamma = 0.995$ and maintain a repertoire size of $N_{\mathcal{R}} = 5000$ skills. The algorithm samples new skills every $T = 250$ timesteps and runs for a total of 2M timesteps. The robot's actuators are controlled by PD controllers operating at 20Hz in position control mode. For efficient learning, we run the Data Collection Loop and Training Loop concurrently on a single NVIDIA RTX 6000 Ada GPU.

5.2 Unsupervised Skill Discovery for Forward Locomotion

Reward, Feature and Cost Functions In this experiment, we intend to learn a diverse behaviours for moving forward in an unsupervised manner. As such, the feature function $\phi(\cdot)$ is implemented as a VAE encoder network that takes as input the robot's joint angles and height at each timestep t, and encodes them into a latent skill space $\mathcal{Z}$ of dimension $D = 3$. The reward function $r(s_t, a_t)$ is designed to promote forward movement while maintaining stability:

$$r(s_t, a_t) = r_{\text{upr}} + \mathbb{1}_{r_{\text{upr}} > 0.7} \cdot (5r_{\text{vel}_x} - 0.5r_{\text{vel}_y} - 0.5r_{\text{yaw}}) - 0.001(r_{\text{speed}} + r_{\text{work}} + r_{\text{smooth}})$$

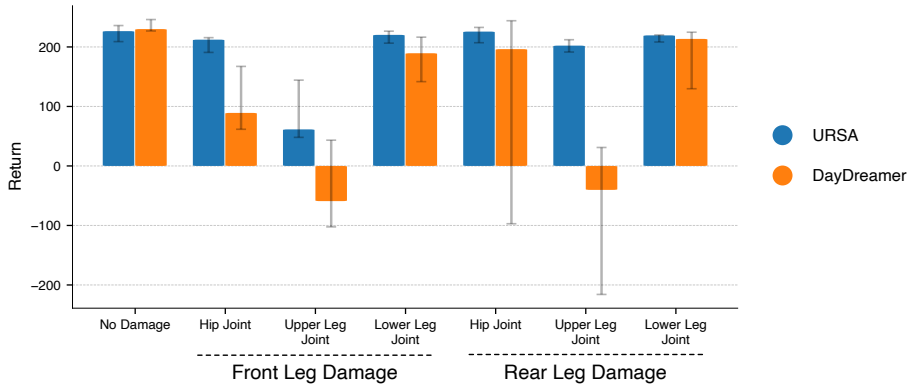

Figure 5: Comparison of returns between URSA and DayDreamer across various joint damage scenarios on the robot. Results show the median return and range (minimum to maximum) across 3 independent runs per algorithm.

where r_{upr} encourages upright posture, r_{vel_x} rewards forward velocity, and penalties are applied for lateral motion (r_{vel_y}), yaw velocity (r_{yaw}), and smoothness ($r_{\text{speed}}, r_{\text{work}}, r_{\text{smooth}}$). Taking inspiration from (Wu et al., 2022), velocity rewards are only applied when the robot is upright ($r_{\text{upr}} > 0.7$).

The safe state space is defined by all the states where $r_{\text{upr}} > 0.7$, which ensures stability by maintaining an upright posture. The cost function $c(s_t, a_t) = 0.7 - r(s_t, a_t)$ naturally penalizes unstable states and undesirable motions while promoting forward progression when upright. We aim to learn a diverse repertoire of $N_{\boldsymbol{z}} = 100$ distinct skills that capture different locomotion behaviors.

Evaluating Behavioral Diversity We first evaluate the behavioral diversity of the discovered skills by analyzing the coverage of the average joint angles achieved by each skill. Figure 4 shows that URSA learns a diverse set of behaviors, with an extended range of average joint angles achieved for different skills. The joint angle variations result in a diverse set of behaviors, as shown in Figure 1. This figure illustrates how URSA discovers a diverse range of movement patterns spanning multiple motion categories - from behaviors where the torso maintains ground contact to more sophisticated walking-like gaits. In contrast, DayDreamer focuses on learning a single skill that maximizes the expected return, limiting the diversity of its behavioral repertoire.

Damage Adaptation Building on this diverse skill repertoire, we evaluate how well these learned behaviors enable adaptation to damage. We compare how well URSA and DayDreamer perform when the robot undergoes actuator failures and needs to adapt. In particular, we test scenarios where one actuator stops working, stays fixed in position, and does not respond to any control commands. We consider damage scenarios where either one front leg or one back leg has a frozen actuator, with the frozen actuator being either the hip, upper leg, or lower leg joint. We then evaluate the return of the robot in each of these scenarios.

We find that URSA consistently outperforms DayDreamer across all damage scenarios tested (Fig. 5). This superior performance can be attributed to the diverse behavioral repertoire $\mathcal{R}$ learned by URSA, which provides multiple alternative movement strategies when the primary locomotion pattern becomes infeasible due to damage. The performance gap is particularly pronounced in scenarios involving front upper leg joint damage, where the robot needs to fundamentally alter its movement strategy. These results demonstrate that the behavioral diversity encouraged by URSA not only leads to more interesting behaviors but also provides benefits in terms of robustness and adaptability.

5.3 Heuristic-Based Skill Discovery

Reward, Feature and Cost Functions For the heuristic-based skill discovery experiments, we use a reward function that encourages stable posture and smooth motion:

$$r(s_t, a_t) = r_{\text{upr}} + r_{\text{hip}} + r_{\text{upper}} + r_{\text{lower}} - 0.001(r_{\text{speed}} + r_{\text{work}} + r_{\text{smooth}})$$

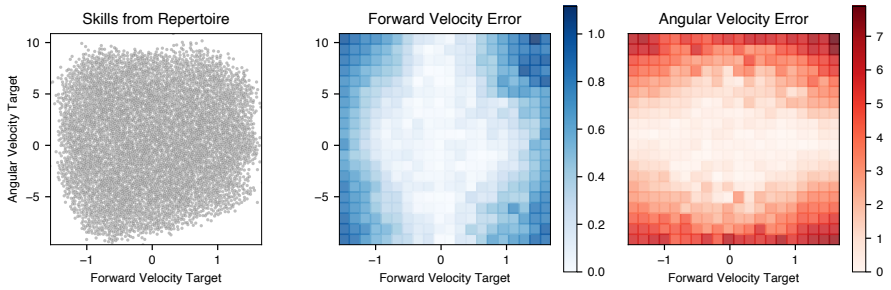

Figure 6: Velocity tracking errors during skill execution. We evaluate how accurately the robot follows target velocity commands, measuring both forward and angular velocity tracking errors across the entire reachable space discovered by URSA. Lower values indicate better velocity control.

where r_{upr}, r_{hip}, r_{upper}, and r_{lower} encourage proper posture for the torso and each joint type respectively. Each posture reward is only active when the previous joint in the kinematic chain has achieved a good pose (e.g., r_{hip} only activates when $r_{\text{upr}} > 0.7$), encouraging the robot to build stable poses from the ground up. Similar to the velocity experiments, we define safe states as those where $r_{\text{upr}} > 0.7$, $r_{\text{hip}} > 0.7$, $r_{\text{upper}} > 0.7$, and $r_{\text{lower}} > 0.7$, ensuring the robot maintains a proper standing posture. The cost function $c(s_t, a_t) = \max(0.7 - r_{\text{upr}}, 0) + \max(0.7 - r_{\text{hip}}, 0) + \max(0.7 - r_{\text{upper}}, 0) + (0.7 - r_{\text{lower}})$ penalizes states where hip, shoulder, knee, or upright posture rewards fall below their target thresholds of 0.7.

Evaluating Skill Reachability We evaluate the reachability of the discovered skills by analyzing how well the robot responds to target velocity commands. For each target velocity, we evaluate the tracking error between the robot's velocity and the target velocity. Figure 6 shows how well the robot tracks different target velocities using the discovered skills. The results demonstrate that URSA learns skills spanning a wide range of velocities, enabling precise control over the robot's movement. This reachability analysis confirms that our approach not only discovers diverse behaviors but also learns skills that are practically useful for controlled robot movement.

6 Related Work

Unsupervised skill discovery: The field of unsupervised skill discovery has seen significant advances in recent years, with various approaches aiming to learn diverse behaviors without explicit task rewards. Information-theoretic methods like DIAYN (Eysenbach et al., 2018) and DADS (Sharma et al., 2019) have pioneered this direction by maximizing the mutual information between skills and states, enabling the emergence of distinct behaviors. SMERL (Kumar et al., 2020) and DOMiNO (Zahavy et al., 2022) extended this approach by learning multiple solutions for each task to improve robustness to environmental changes. In the Quality-Diversity domain, algorithms like TAXONS (Paolo et al., 2020), STAX (Paolo et al., 2024), AURORA (Cully, 2019; Grillotti & Cully, 2022), and IMGEP-UGL (Péré et al., 2018) demonstrated that meaningful skills could emerge without hand-designed behavioral descriptors, instead learning these descriptors from data. While these methods have shown impressive results in simulation, they typically require extensive interaction with the environment and don't address the unique challenges of real-world learning. Our approach builds upon these foundations but differs in two key aspects: first, we introduce safety constraints and efficient sampling methods that make unsupervised skill discovery feasible in a reset-free environment; second, our algorithm leverages imagination-based planning through world models to reduce the required amount of real-world interaction.

Real-world robot learning: Learning directly in the real world presents unique challenges due to limited data collection, safety concerns, and the absence of reset mechanisms. Several approaches have tackled these challenges from different angles. Reset-free Quality-Diversity (Lim et al., 2022b; Smith et al., 2023) introduced a novel way to learn without manual resets by intelligently selecting behaviors that can serve as automatic resets, demonstrating successful learning of locomotion skills. The work of Laversanne-Finot et al. (2021) demonstrated successful learning of diverse

robotic arm behaviors for ball manipulation through intrinsically motivated goal exploration, though it still relied on episodic learning in controlled environments. off-DADS (Sharma et al., 2020) showed that unsupervised skill emergence is possible on real quadrupeds through efficient off-policy reinforcement learning, though it required careful selection of the discriminator's observation space. Recent work like DayDreamer (Wu et al., 2022) demonstrated that world models could enable efficient real-world learning by allowing policy optimization to occur primarily in imagination. Similarly, SERL (Luo et al., 2024) and A Walk in the Park (Kostrikov et al., 2023) achieved successful real-world learning through careful system design and algorithmic innovations focused on sample efficiency. DayDreamer's success in teaching a quadruped to walk in just one hour and enabling visual manipulation tasks with robotic arms established a new standard for sample-efficient real-world learning. Our work combines the strengths of these approaches while addressing their limitations. Like DayDreamer, we leverage world models for efficient learning, but we extend this to the discovery of diverse skills rather than focusing on a single task. We build upon Reset-free Quality-Diversity's insights about autonomous learning but incorporate safety constraints and more sophisticated skill selection mechanisms. Unlike off-DADS, our approach doesn't require careful reward engineering, instead discovering meaningful skills through unsupervised learning while maintaining safety and efficiency.

7 Discussion and Future Work

In this work, we presented URSA, demonstrating safe and efficient unsupervised skill discovery in reset-free robotics. While our results are promising, several key challenges remain. For example, learning safety boundaries from data rather than using predefined constraints could make the system more robust to changing environments. Also, developing more sophisticated sampling methods that target the agent's zone of proximal development - where learning progress potential is highest - could significantly improve skill acquisition efficiency. Finally, a key next step would be to validate our approach through real-world deployment on physical robotic systems. These advances would represent important steps toward more autonomous and capable robotic systems that can continuously learn and adapt in real-world settings.

Acknowledgments

References

Eitan Altman. *Constrained Markov decision processes*. Routledge, 1999.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4058–4068, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 81–89, 2019.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, May 2015. ISSN 1476-4687. doi: 10.1038/nature14422. URL https://doi.org/10.1038/nature14422.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is All You Need: Learning Skills without a Reward Function. September 2018. URL https://openreview.net/forum?id=SJx63jRqFm.

Luca Grillotti and Antoine Cully. Unsupervised behavior discovery with quality-diversity optimization. *IEEE Transactions on Evolutionary Computation*, 26(6):1539–1552, 2022. doi: 10.1109/TEVC.2022.3159855.

Luca Grillotti, Maxence Faldor, Borja González León, and Antoine Cully. Quality-diversity actor-critic: Learning high-performing and diverse behaviors via value and successor features critics. In *International Conference on Machine Learning*. PMLR, 2024.

David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Ilya Kostrikov, Laura M. Smith, and Sergey Levine. Demonstrating A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.056. URL https://doi.org/10.15607/RSS.2023.XIX.056.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: few-shot extrapolation via structured MaxEnt RL. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pp. 8198–8210, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-71382-954-6.

Adrien Laversanne-Finot, Alexandre Péré, and Pierre-Yves Oudeyer. Intrinsically motivated exploration of learned goal spaces. *Frontiers in neurorobotics*, 14:555271, 2021.

Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. Dynamics-Aware Quality-Diversity for Efficient Learning of Skill Repertoires. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5360–5366, Philadelphia, PA, USA, May 2022a. IEEE Press. doi: 10.1109/ICRA46639.2022.9811559. URL https://doi.org/10.1109/ICRA46639.2022.9811559.

Bryan Lim, Alexander Reichenbach, and Antoine Cully. Learning to walk autonomously via reset-free quality-diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 86–94, New York, NY, USA, July 2022b. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528715. URL https://doi.org/10.1145/3512290.3528715.

Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16961–16969, 2024. URL https://api.semanticscholar.org/CorpusID:267311834.

Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pp. 22–31. PMLR, 2023.

Giuseppe Paolo, Alban Laflaquiere, Alexandre Coninx, and Stephane Doncieux. Unsupervised Learning and Exploration of Reachable Outcome Space. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2379–2385, Paris, France, May 2020. IEEE. ISBN 978-1-72817-395-5. doi: 10.1109/ICRA40945.2020.9196819. URL https://ieeexplore.ieee.org/document/9196819/.

Giuseppe Paolo, Miranda Coninx, Alban Laflaquière, and Stephane Doncieux. Discovering and Exploiting Sparse Rewards in a Learned Behavior Space. *Evolutionary Computation*, 32(3):275–305, September 2024. ISSN 1063-6560. doi: 10.1162/evco_a_00343. URL https://doi.org/10.1162/evco_a_00343.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

Alexandre Péré, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. *arXiv preprint arXiv:1803.00781*, 2018.

Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832 – 837, 1956. doi: 10.1214/aoms/1177728190. URL https://doi.org/10.1214/aoms/1177728190.

Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

David W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley & Sons, Inc., New York, 1992.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-Aware Unsupervised Discovery of Skills. September 2019. URL https://openreview.net/forum?id=HJgLZR4KvH.

Archit Sharma, Michael Ahn, Sergey Levine, Vikash Kumar, Karol Hausman, and Shixiang Gu. Emergent Real-World Robotic Skills via Unsupervised Off-Policy Reinforcement Learning. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. ISBN 978-0-9923747-6-1. doi: 10.15607/RSS.2020.XVI.053. URL http://www.roboticsproceedings.org/rss16/p053.pdf.

Simón C. Smith, Bryan Lim, Hannah Janmohamed, and Antoine Cully. Quality-diversity optimisation on a physical robot through dynamics-aware and reset-free learning. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion, pp. 171–174, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701207. doi: 10.1145/3583133.3590625. URL https://doi.org/10.1145/3583133.3590625.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03924-6.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World Models for Physical Robot Learning, June 2022. URL http://arxiv.org/abs/2206.14176. arXiv:2206.14176 [cs].

Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

A MATHEMATICAL DERIVATIONS

We provide here the technical details behind the lower bound on the approximate entropy of the KDE:

$$\begin{aligned}\widehat{\mathrm{H}}(\textsc{kde}\,(\mathcal{R})) &= -\frac{1}{N_{\mathcal{R}}}\sum_{i=1}^{N_{\mathcal{R}}}\log\left(\frac{1}{N_{\mathcal{R}}}\sum_{j=1}^{N_{\mathcal{R}}}\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{z}_j,\boldsymbol{\Sigma})\right)\\ &\geq -\frac{1}{N_{\mathcal{R}}}\sum_{i=1}^{N_{\mathcal{R}}}\log\left(1+(N_{\mathcal{R}}-1)e^{-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_i^{\mathrm{nn}})^2}\right)+\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)+\frac{D}{2}\log(2\pi)+\log N_{\mathcal{R}}\end{aligned}$$

where $d_{\boldsymbol{\Sigma}}(\cdot,\cdot)$ is the Mahalanobis distance with respect to the covariance matrix $\boldsymbol{\Sigma}$.

Proof. For all $i \in \{1, 2, \ldots, N_{\mathcal{R}}\}$:

$$\begin{aligned}\log\sum_{j=1}^{N_{\mathcal{R}}}\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{z}_j,\boldsymbol{\Sigma}) &= \log\left(\frac{1}{\sqrt{(2\pi)^D\det(\boldsymbol{\Sigma})}}\sum_{j=1}^{N_{\mathcal{R}}}\exp\left(-\frac{1}{2}(\boldsymbol{z}_i-\boldsymbol{z}_j)^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i-\boldsymbol{z}_j)\right)\right)\\ &= \log\left(\sum_{j=1}^{N_{\mathcal{R}}}\exp\left(-\frac{1}{2}(\boldsymbol{z}_i-\boldsymbol{z}_j)^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i-\boldsymbol{z}_j)\right)\right)-\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)-\frac{D}{2}\log(2\pi)\\ &= \log\left(\sum_{j=1}^{N_{\mathcal{R}}}\exp\left(-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_j)^2\right)\right)-\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)-\frac{D}{2}\log(2\pi)\\ &= \mathrm{LSE}\left(-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_1)^2,\ldots,-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_{N_{\mathcal{R}}})^2\right)-\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)-\frac{D}{2}\log(2\pi)\\ &\leq \log\left(1+(N_{\mathcal{R}}-1)\exp\left(-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_i^{\mathrm{nn}})^2\right)\right)-\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)-\frac{D}{2}\log(2\pi)\end{aligned}$$

where LSE refers to the log-sum-exp function.

Then we have:

$$\begin{aligned}\widehat{\mathrm{H}}(\textsc{kde}\,(\mathcal{R})) &= -\frac{1}{N_{\mathcal{R}}}\sum_{i=1}^{N_{\mathcal{R}}}\left(\log\sum_{j=1}^{N_{\mathcal{R}}}\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{z}_j,\boldsymbol{\Sigma})\right)+\log N_{\mathcal{R}}\\ &\geq -\frac{1}{N_{\mathcal{R}}}\sum_{i=1}^{N_{\mathcal{R}}}\log\left(1+(N_{\mathcal{R}}-1)e^{-\frac{1}{2}d_{\boldsymbol{\Sigma}}(\boldsymbol{z}_i,\boldsymbol{z}_i^{\mathrm{nn}})^2}\right)+\frac{1}{2}\log\left(\det(\boldsymbol{\Sigma})\right)+\frac{D}{2}\log(2\pi)+\log N_{\mathcal{R}}\end{aligned}$$

□