SKILL-CONDITIONED POLICY OPTIMIZATION WITH SUCCESSOR FEATURES REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A key aspect of intelligence is the ability to exhibit a wide range of behaviors to adapt to unforeseen situations. Designing artificial agents that are capable of showcasing a broad spectrum of skills is a long-standing challenge in Artificial Intelligence. In the last decade, progress in deep reinforcement learning has enabled to solve complex tasks with high-dimensional, continuous state and action spaces. However, most approaches return only one highly-specialized solution to a single problem. We introduce a **S**kill-**C**onditioned **OP**timal **A**gent (SCOPA) that leverages successor features representations to learn a continuous range of skills that solve a task. We extend the generalized policy iteration framework with a policy skill improvement update based on successor features that is analogous to the classic policy improvement update. This novel skill improvement update enables to efficiently learn executing skills. From this result, we develop an algorithm that seamlessly unifies value function and successor features policy iteration with constrained optimization to (1) maximize performance, while (2) executing the desired skills. Compared with other skill-conditioned reinforcement learning methods, SCOPA reaches significantly higher performance and skill space coverage on challenging continuous control locomotion tasks with various types of skills. We also demonstrate that the diversity of skills is useful in five downstream adaptation tasks. Videos of our results are available at: `bit.ly/scopa`.

1 INTRODUCTION

Reinforcement Learning (RL) has enabled groundbreaking achievements like mastering discrete games (Mnih et al., 2013; Silver et al., 2016) but also continuous control domains for locomotion (Haarnoja et al., 2019; Heess et al., 2017) and manipulation (OpenAI et al., 2019). These milestones have showcased the extraordinary potential of RL algorithms in solving specific problems.

In contrast, human intelligence is beyond mastering a single task, and adapts to unforeseen environments by combining skills. Empowering artificial agents with diverse skills was shown to improve exploration (Gehring et al., 2021), to facilitate knowledge transfer (Eysenbach et al., 2018), to enable hierarchical problem-solving (Allard et al., 2022), to enhance robustness and adaptation (Kumar et al., 2020; Cully et al., 2015) and finally, to foster creativity (Zahavy et al., 2023; Lehman et al., 2020).

Following this observation, methods have been developed to make agents more versatile, including Goal-Conditioned Reinforcement Learning (GCRL) (Liu et al., 2022), Unsupervised Reinforcement Learning (URL) (Eysenbach et al., 2018; Sharma et al., 2019), Quality-Diversity (QD) optimization (Pugh et al., 2016; Cully & Demiris, 2018; Zahavy et al., 2022), and reward design (Margolis & Agrawal, 2022). However, designing algorithms to learn large ranges of skills that are useful to solve downstream tasks remains a challenge.

In this work, we focus on learning a policy conditioned on a continuous range of pre-defined skills while also maximizing performance. Approaches like DOMINO (Zahavy et al., 2022) or QD optimization share the objective of finding diverse and optimal skills but they only discover a finite set of solutions. Specifically, DOMINO learns N diverse policies that are different from one another, but these policies are not trained to execute pre-defined skills. Other methods like GCRL and URL try to reach goals or execute skills while disregarding other objectives like safety or efficiency, leaving a gap in our quest for machines that can execute expressive and optimal skills to solve complex tasks.

We introduce a Skill-Conditioned Optimal Agent (SCOPA) that leverages successor features representations to learn a continuous range of expressive skills while maximizing performance. Our contributions are as follows: First, we derive a policy skill improvement update analogous to the classic policy (performance) improvement update. Second, we use the policy skill improvement update to develop a theoretically justified algorithm that combines successor features with universal function approximators to learn an infinite range of expressive skills efficiently. Third, we seamlessly unify value function and successor features policy iteration with constrained optimization to (1) maximize performance, while (2) executing desired skills.

We evaluate our approach on multiple challenging continuous control tasks combined with four different skills and show that SCOPA manages to execute more skills and achieves higher performance than other baselines. Finally, we show that the skills can be used to adapt to downstream tasks in a few shots or through hierarchical learning.

2 BACKGROUND

2.1 REINFORCEMENT LEARNING

We consider the standard reinforcement learning framework (Sutton & Barto, 2018) where an agent sequentially interacts with a *Markov Decision Process* (MDP) in order to maximize the expected sum of rewards. At each time step t, the agent observes a *state* $s_t \in \mathcal{S}$ and takes an *action* $a_t \in \mathcal{A}$, which causes the environment to transition to a next state $s_{t+1} \in \mathcal{S}$ and to give a reward $r_t \in \mathbb{R}$, sampled from the dynamics $p(s_{t+1}, r_t \mid s_t, a_t)$. We denote $\rho^\pi(s) = \lim_{t\to\infty} P(s_t = s|s_0, \pi)$ the stationary distribution of states under π, which we assume exists and is independent of s_0 (Sutton et al., 1999) . Additionally, we focus on the case where each state s_t is associated with observable features $\boldsymbol{\phi}_t = \boldsymbol{\phi}(s_t) \in \Phi \subset \mathbb{R}^d$. In other cases, $\boldsymbol{\phi}$ can be learned with a neural network (Grillotti & Cully, 2022b), but that is not the focus of the present work.

The objective of the agent is to find a policy π that maximizes the expected discounted sum of rewards, or expected return $\mathbb{E}_\pi\left[\sum_t \gamma^t r_t\right]$. The so-called value-based method in RL rely on the concept of *value function* $v^\pi(s)$, defined as the expected return obtained when starting from state s and following policy π thereafter (Puterman, 1994): $v^\pi(s) = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| s_t = s\right]$.

In addition to the value function, we also leverage the concept of *successor features* $\boldsymbol{\psi}^\pi(s)$, which is the expected discounted sum of features obtained when starting from state s and following policy π thereafter (Barreto et al., 2017): $\boldsymbol{\psi}^\pi(s) = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty} \gamma^i \boldsymbol{\phi}_{t+i} \middle| s_t = s\right]$. The successor features captures the expected feature occupancy under a given policy, offering insights into the agent's future behavior and satisfies a Bellman equation in which $\boldsymbol{\phi}_t$ plays the role of the reward $\boldsymbol{\psi}^\pi(s) = \boldsymbol{\phi}_t + \gamma\mathbb{E}_\pi\left[\boldsymbol{\psi}^\pi(s_{t+1})|s_t = s\right]$, and can be learned with any RL methods (Dayan, 1993).

In practice, we make use of a universal value function approximator $v^\pi(s, \mathbf{z})$ (Schaul et al., 2015) and of a universal successor features approximator $\boldsymbol{\psi}^\pi(s, \mathbf{z})$ (Borsa et al., 2018) that depend on state s but also on the skill $\mathbf{z}$ conditioning the policy. The value function quantifies the performance while the successor features characterizes the behavior of the agent, both starting from state s and conditioned on skill $\mathbf{z}$. For conciseness, we omit π from the notations ρ^π, v^π and $\boldsymbol{\psi}^\pi$ in the following sections.

2.2 WORLD MODELS

Learning a skill-conditioned function approximator is challenging because in general, the agent will only see a small subset of possible $(s, \mathbf{z})$ combinations (Schaul et al., 2015; Borsa et al., 2018). In that case, a world model can be used to improve sample efficiency. One key advantage of model-based methods is to learn a compressed spatial and temporal representation of the environment to train a simple policy that can solve the required task (Ha & Schmidhuber, 2018). World models are particularly valuable for conducting simulated rollouts in imagination which can subsequently inform the optimization of the agent's behavior, effectively reducing the number of environment interactions required for learning (Hafner et al., 2019a). Moreover, world models enable to compute straight-through gradients, which backpropagate directly through the learned dynamics (Hafner et al., 2023). Most importantly, the small memory footprint of imagined rollouts enables to sample thousands of on-policy trajectories in parallel (Hafner et al., 2023), making possible to learn skill-conditioned function approximators with massive skill sampling in imagination.

In this work, we use a Recurrent State Space Model (RSSM) from Hafner et al. (2019b). At each iteration, the world model $\mathcal{W}$ is trained to learn the transition dynamics, and to predict the observation, reward, and termination condition. An *Imagination* MDP $(\widetilde{\mathcal{S}}, \mathcal{A}, \widehat{p}, \gamma)$, can then be defined from the latent states $\widetilde{s} \in \widetilde{\mathcal{S}}$ and from the dynamics $\widehat{p}$ of $\mathcal{W}$. In parallel, DREAMERV3 trains a critic network $\widehat{v}(\widetilde{s}_t)$ to regress the λ-return $V_\lambda(\widetilde{s}_t)$ (Sutton & Barto, 2018). Then, the actor is trained to maximize V_λ, with an entropy regularization for exploration: $J(\theta_\pi) = \mathbb{E}_{\substack{a \sim \pi(.|\widetilde{s}) \\ \widetilde{s}' \sim \widehat{p}(.|\widetilde{s},a)}} \left[\sum_{t=1}^{H} V_\lambda(\widetilde{s}_t)\right]$.

3 PROBLEM STATEMENT

In this work, we define a skill $\mathbf{z} \in \mathcal{Z}$ executed by a policy as the expected features under the policy's stationary distribution, $\mathbf{z} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\phi}_t = \mathbb{E}_{s\sim\rho}[\boldsymbol{\phi}(s)]$, that we simply note $\mathbb{E}_\pi[\boldsymbol{\phi}(s)]$. Given this definition, we intend to learn a skill-conditioned policy $\pi(\cdot|\cdot, \mathbf{z})$ that (1) maximizes the expected return, and (2) is subject to the expected features under the policy $\pi(\cdot|\cdot, \mathbf{z})$ converges to the desired skill $\mathbf{z}$, which corresponds to the following constrained optimization problem:

$$\forall \mathbf{z} \in \mathcal{Z}, \quad \text{maximize } \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right] \quad \text{subject to } \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\boldsymbol{\phi}(s)\right] = \mathbf{z} \tag{P1}$$

For example, consider a robot whose objective is to minimize energy consumption, and where the features characterize the velocity of the robot $\boldsymbol{\phi}_t = \mathbf{v}_t = [v_x(t) \quad v_y(t)]^\intercal$ and the skill space $\mathcal{Z} = [-v_{\text{max}}, v_{\text{max}}]^2$. For each desired velocity $\mathbf{z} \in \mathcal{Z}$, $\pi(\cdot|\cdot, \mathbf{z})$ is expected to (1) minimize energy consumption, while (2) following the desired velocity $\mathbf{z}$ in average, $\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{v}_t = \mathbf{z}$.

Now consider another example, where the objective is to go forward as fast as possible, and the features characterize which foot is in contact with the ground at each time step. For example, $\boldsymbol{\phi}_t = [1 \quad 0]^\intercal$ for a biped robot that is standing on its first leg and with the second leg not touching the ground at time step t. With these features, the i-th component of the skill $\mathbf{z}$ will be the proportion of time during which the i-th foot of the robot is in contact with the ground over a trajectory, denoted as feet contact rate. In that case, the skill space is continuous and characterizes the myriad of ways the robot can walk and specifically, how often each leg is being used. In comparison with the skill space, the feature space is finite and contains only four elements, $\Phi = \{0, 1\}^2 \subset \mathcal{Z} = [0, 1]^2$. In order to achieve a feet contact of $\mathbf{z} = [0.1 \quad 0.6]^\intercal$, the robot needs to use 10% of the time the first foot and 60% of the time the second foot, and necessarily needs multiple time steps to execute the skill. Notice that in that case, the feature space is smaller than the skill space $\Phi \subsetneq \mathcal{Z}$.

Problem P1 amounts to maximizing the return *while* executing a desired skill. If we only consider the constraint, we can show that our problem is related to GCRL. In GCRL, a common assumption is to consider that achieving a desired goal $\mathbf{g}$ is equivalent to reaching a state that is in the subset $\{s \in \mathcal{S} | \, \boldsymbol{\phi}(s) = \mathbf{g}\}$ (Liu et al., 2022). In contrast, our constraint is to execute a desired skill $\mathbf{z}$, i.e. to sample a trajectory that is in the subset $\{(s_t)_{t\geq 0} \in \mathcal{S}^{\mathbb{N}} | \, \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\phi}(s_t) = \mathbf{z}\}$. Therefore, skills do not depend on a single time step, but are rather defined at trajectory-level. Although closely related, we can show that the feature space Φ is a subset of the skill space $\mathcal{Z}$, which forms the convex hull of the feature space. Therefore, the skill space is larger than the goal space, $\Phi \subset \mathcal{Z} = \text{Conv}(\Phi)$.

4 METHODS

In this section, we introduce a Skill-Conditioned Optimal Agent (SCOPA), an approximate solution to Problem P1. In the following section, we define the objective functions SCOPA considers for optimization. Then, we describe the design choices made to efficiently optimize the objectives.

---

[1]

Algorithm 1 SCOPA

```
Initialize dataset $\mathcal{D}$ with samples collected from random exploration, and set up parameters $\theta$
while not converged do
    Sample $\mathbf{z} \sim \mathcal{U}(\mathcal{Z})$
    for $T$ steps do
        ▷ Environment exploration steps
        Sample action $a_t \sim \pi(\cdot|\widetilde{s}_t, \mathbf{z})$        ▷ $\widetilde{s}_t$ denotes the hidden state of the world model $\mathcal{W}$
        $r_t, s_{t+1} \leftarrow$ perform action $a_t$ in environment
        Add transition $(s_t, a_t, r_t, \boldsymbol{\phi}_t, s_{t+1})$ to its corresponding sequence in dataset $\mathcal{D}$        ▷ $\boldsymbol{\phi}_t := \boldsymbol{\phi}(s_t)$
        ▷ Training steps
        Train world model parameters $\theta_{\mathcal{W}}$ to reconstruct observations, rewards $\widehat{r}$ and features $\widehat{\boldsymbol{\phi}}$ using $\mathcal{D}$
        Sample $N$ skills $\widetilde{\mathbf{z}}_i \sim \mathcal{U}(\mathcal{Z})$ and states $s_i \in \mathcal{D}$; compute latent states $\widetilde{s}_i \leftarrow \mathcal{W}(s_i)$
        For each $i$, perform a rollout of horizon $H$ in imagination using $\mathcal{W}$ starting from $\widetilde{s}_i$ with $\pi(\cdot|\cdot, \widetilde{\mathbf{z}}_i)$
        $\theta_\lambda \leftarrow \theta_\lambda - \alpha_\lambda \nabla L(\theta_\lambda)$        ▷ Minimize Lagrange multiplier loss (3)
        $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \nabla J(\theta_\pi)$        ▷ Maximize actor's objective (4)
        Update parameters $\theta_v$ and $\theta_{\boldsymbol{\psi}}$ of value and successor features networks
```

4.1 ACTOR OPTIMIZATION

First, we relax the constraint from Problem P1 using the L^2 norm and δ, a threshold that quantifies the maximum acceptable distance between the desired skill and the expected features,

$$\forall t, \forall \mathbf{z} \in \mathcal{Z}, \quad \text{maximize } \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right] \quad \text{subject to } \left\|\mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\boldsymbol{\phi}(s)\right] - \mathbf{z}\right\|_2 \le \delta \tag{P2}$$

Second, we derive an upper bound for the distance between the desired skill and the expected features, whose proof is provided in Appendix B. A similar proposition is proven in a more general case in Appendix B.1. The goal is to minimize the bound so that the constraint in Problem P2 is satisfied.

Proposition. *Consider an infinite horizon, finite MDP with observable features in Φ. Let π be a policy and let $\boldsymbol{\psi}$ be the discounted successor features. Then, for all skills $\mathbf{z} \in \mathcal{Z}$, we can derive an upper bound for the distance between $\mathbf{z}$ and the expected features under π:*

$$\left\|\mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\boldsymbol{\phi}(s)\right] - \mathbf{z}\right\|_2 \le \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2\right] \tag{1}$$

Third, we derive a new Problem P3 by replacing the intractable constraint of Problem P2 with the tractable upper bound in Equation 1. Note that if the more restrictive constraint in Problem P3 is satisfied, then the constraint in Problem P2 is necessarily satisfied as well.

$$\forall \mathbf{z} \in \mathcal{Z}, \quad \text{maximize } \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[v(s,\mathbf{z})\right] \quad \text{subject to } \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2\right] \le \delta \tag{P3}$$

Finally, we solve Problem P3 by combining the objective and the constraint using Lagrangian relaxation as described by Abdolmaleki et al. (2018; 2023). For all states s, and all skills $\mathbf{z} \in \mathcal{Z}$, we maximize the following objective:

$$(1 - \lambda_{\mathcal{L}}(s,\mathbf{z}))\, v(s,\mathbf{z}) - \lambda_{\mathcal{L}}(s,\mathbf{z}) \|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2 \quad \text{subject to } 0 \le \lambda_{\mathcal{L}}(s,\mathbf{z}) \le 1 \tag{2}$$

The first term in red aims at maximizing the return, while the second term in blue aims at executing the desired skill. We extend generalized policy iteration to successor features by alternating policy evaluation for both critics $v(s,\mathbf{z})$ and $\boldsymbol{\psi}(s,\mathbf{z})$, and policy improvement based on Equation 2. Thus, we introduce a policy (skill) improvement with successor features that is analogous to the classic policy (performance) improvement with value function.

The coefficient $\lambda_{\mathcal{L}}$ is analogous to a Lagrange multiplier

and is optimized to balance the quality-diversity trade-off. If $\|(1-\gamma)\boldsymbol{\psi}(s_1,\mathbf{z}_1) - \mathbf{z}_1\|_2 \le \delta$ is satisfied for $(s_1, \mathbf{z}_1)$, we expect $\lambda_{\mathcal{L}}(s_1, \mathbf{z}_1)$ to decrease to encourage maximizing the return. On the

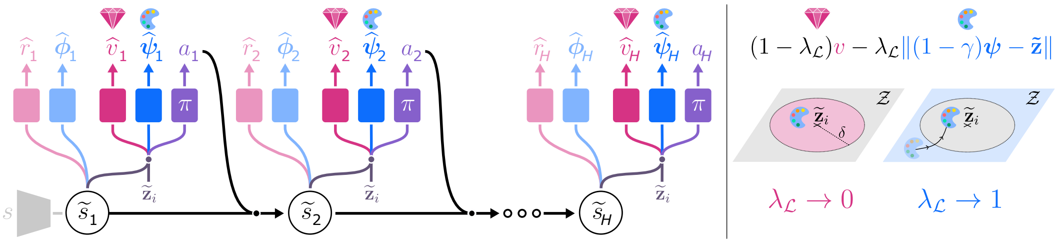


Figure 1: **Left** Imagination rollout performed within the world model $\mathcal{W}$. Each individual rollout i generates on-policy transitions following skill $\widetilde{\mathbf{z}}_i$, starting from a state $\widetilde{s}_1$ for a fixed number of steps H. The world model predicts $\widehat{r}_i$ and $\widehat{\boldsymbol{\phi}_i}$ that enable to compute $\widehat{v}_i$ and $\widehat{\boldsymbol{\psi}}_i$ respectively. **Right** Actor's objective function computed with the predicted $\widehat{v}$ and $\widehat{\boldsymbol{\psi}}$. The Lagrangian multiplier is optimized to balance the quality-diversity trade-off, see Equation 3.

contrary, if the constraint is not satisfied for $(s_2, \mathbf{z}_2)$, we expect $\lambda_{\mathcal{L}}(s_2, \mathbf{z}_2)$ to increase to encourage satisfying the constraint.

We represent the function $\lambda_{\mathcal{L}}$ by a neural network parameterized by θ_λ. We optimize the parameters θ_λ so that $\lambda_{\mathcal{L}}(s, \mathbf{z}) \to 1$ when the actor is unable to solve the constraint, in order to focus on satisfying the constraint. Conversely, $\lambda_{\mathcal{L}}(s, \mathbf{z}) \to 0$ when the constraint is satisfied, to focus primarily on maximizing the return. In practice, we use a cross-entropy loss to optimize those parameters:

$$L(\theta_\lambda) = \mathbb{E}_{\substack{s\sim\rho \\ \mathbf{z}\sim\mathcal{U}(\mathcal{Z})}} \left[-(1-y)\log\left(1-\lambda_{\mathcal{L}}(s,\mathbf{z})\right) - y\log\left(\lambda_{\mathcal{L}}(s,\mathbf{z})\right)\right]$$
$$\text{where } y = \begin{cases} 0 & \text{if } \|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2 \leq \delta \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

4.2 TRAINING

The main challenge to train skill-conditioned function approximators like our policy or the value function/successor features is the need to sample a large number of $(s, \mathbf{z})$ combinations (Schaul et al., 2015; Borsa et al., 2018), see Section 2.2. To solve this problem, we use the model-based RL algorithm DREAMERV3 (Hafner et al., 2023) that enables to generate thousands of on-policy trajectories simultaneously in imagination.

A pseudo-code of SCOPAis provided in Algorithm 1. At each iteration, a skill $\mathbf{z}$ is uniformly sampled and for T steps, the agent interacts with the environment following skill $\mathbf{z}$ with $\pi(\cdot|\cdot, \mathbf{z})$. At each step, the transition is stored in a dataset $\mathcal{D}$, which is used to perform a world model training step. Then, N skills are uniformly sampled to perform rollouts in imagination, and those rollouts are used to (1) train the two critics $v(s, \mathbf{z})$, $\boldsymbol{\psi}(s, \mathbf{z})$ and (2) train the actor π.

World model training The dataset is used to train the world model $\mathcal{W}$ according to DreamerV3. In addition to the reward $\widehat{r}_t$, we extend the model to estimate the features $\widehat{\boldsymbol{\phi}}_t$, like shown on Figure 1.

Critic training The estimated rewards $\widehat{r}_t$ and features $\widehat{\boldsymbol{\phi}}_t$ predicted by the world model are used to predict the value function $\widehat{v}$ and the successor features $\widehat{\boldsymbol{\psi}}$ respectively. Then, similarly to DREAMERV3, the value function $\widehat{v}$ and successor features $\widehat{\boldsymbol{\psi}}$ are trained to regress the λ-returns, V_λ and Ψ_λ respectively. The successor features target is defined recursively as follows:

$$\Psi_\lambda(\widetilde{s}_t, \widetilde{\mathbf{z}}) = \widehat{\boldsymbol{\phi}}_t + \gamma\widehat{c}_t \left((1-\lambda)\,\widehat{\boldsymbol{\psi}}(\widetilde{s}_{t+1}, \widetilde{\mathbf{z}}) + \lambda\Psi_\lambda(\widetilde{s}_{t+1}, \widetilde{\mathbf{z}})\right) \quad \text{and} \quad \Psi_\lambda(\widetilde{s}_H, \widetilde{\mathbf{z}}) = \widehat{\boldsymbol{\psi}}(\widetilde{s}_H, \widetilde{\mathbf{z}})$$

Actor training For each actor training step, we sample N skills $\widetilde{\mathbf{z}}_1 \dots \widetilde{\mathbf{z}}_N \in \mathcal{Z}$. We then perform N rollouts of horizon H in imagination using the world model and policies $\pi(\cdot|\cdot, \widetilde{\mathbf{z}}_i)$. Those rollouts are used to train the critic v, the successor features network $\boldsymbol{\psi}$, and the actor by backpropagating through the dynamics of the model. The actor maximizes the following objective, with an entropy regularization for exploration, where $\text{sg}(\cdot)$ represents the *stop gradient* function.

$$J(\theta_\pi) = \mathbb{E}_{\substack{\widetilde{s}_{1:H}\sim\mathcal{W},\pi \\ \widetilde{\mathbf{z}}\sim\mathcal{U}(\mathcal{Z})}} \left[\sum_{t=1}^{H} (1-\text{sg}(\lambda_{\mathcal{L},t}))\, \underbrace{V_\lambda(\widetilde{s}_t,\widetilde{\mathbf{z}})}_{\text{Performance}} - \text{sg}(\lambda_{\mathcal{L},t})\, \underbrace{\|(1-\gamma)\Psi_\lambda(\widetilde{s}_t,\widetilde{\mathbf{z}}) - \widetilde{\mathbf{z}}\|_2}_{\text{Distance to desired skill } \widetilde{\mathbf{z}}}\right] \quad (4)$$

5 EXPERIMENTS

5.1 TASKS

We compare our method to prior techniques on a range of challenging continuous control tasks using Google Brax (Freeman et al., 2021) physics engine. We consider three classic locomotion environments called Walker, Ant and Humanoid that we combine with four different feature functions that we call *feet contact*, *jump*, *velocity* and *angle*. In these traditional locomotion tasks, the objective is to go forward as fast as possible while minimizing energy consumption.

The feet contact is a vector indicating for each foot of the agent, if the foot is in contact or not with the ground, exactly as defined by Cully et al. (2015) and in DCG-ME (Faldor et al., 2023). For example, if the Ant only touches the ground with its second foot at time step t, then $\boldsymbol{\phi}(s_t) = [0 \quad 1 \quad 0 \quad 0]^\intercal$. The feet contact have been extensively studied in Quality-Diversity (Cully et al., 2015; Nilsson & Cully, 2021; Faldor et al., 2023) to learn diverse locomotion behaviors with different gaits. The diversity of feet contact found by such QD algorithms has been demonstrated to be very useful in downstream tasks such as damage recovery (Cully et al., 2015). Note that to execute a specific skill, the agent needs multiple time steps, see Section 3 for a longer discussion.

The jump feature is a one-dimensional vector corresponding to the height of the lowest foot. For example, if the left foot of the humanoid is 10 cm above the ground and if its right foot is 3.5 cm above the ground, then the features $\boldsymbol{\phi}(s_t) = [0.035]$. The skills derived from the jump features are also challenging to execute because to maintain an average $\mathbf{z} = \frac{1}{T}\sum_{i=0}^{T-1} \boldsymbol{\phi}_{t+i}$, the agent is forced to oscillate around that value $\mathbf{z}$ because of gravity.

The velocity feature is a two-dimensional vector giving the velocity of the agent in the xy-plane, $\boldsymbol{\phi}(s_t) = [v_x(t) \quad v_y(t)]^\intercal$. The velocity feature space has been extensively studied in GCRL (Liu et al., 2022; Zhu et al., 2021; Finn et al., 2017). We evaluate on the velocity features to show that our method works on classic GCRL tasks. The velocity features are also interesting because satisfying a velocity goal that is negative on the x-axis is directly opposite to maximizing the return that is defined as forward velocity minus energy consumption.

Finally, the angle is a two-dimensional vector representing the angle α of the main body about the z-axis, $\boldsymbol{\phi}(s_t) = [\cos(\alpha) \quad \sin(\alpha)]^\intercal$. You can find the feature spaces summarized in Appendix C.1.

5.2 BASELINES

We compare SCOPA against four baselines SMERL (Kumar et al., 2020), REVERSE SMERL (Zahavy et al., 2022), DCG-ME (Faldor et al., 2023) and UVFA (Schaul et al., 2015) that all return a skill-conditioned policy balancing a quality-diversity trade-off.

SMERL learns a repertoire of policies that maximize the expected return while also producing distinct behaviors by encouraging the trajectories of different latent variables $\mathbf{z}$ to be distinct. To that end, it uses the following reward: $r_{\text{SMERL}} = r + \alpha\mathbb{1}(R \geq R^* - \epsilon)\tilde{r}$. Similarly, REVERSE SMERL focuses on finding diverse behaviours, and only optimizes for performance if the policy is not close to the optimal policy: $r_{\text{REVERSE}} = \mathbb{1}(R < R^* - \epsilon)r + \alpha\tilde{r}$. For a fair comparison with our method, we compute the diversity reward with continuous DIAYN (Choi et al., 2021) so that SMERL learns a continuous (infinite) subspace instead of a discrete (finite) subset of policies (Kumar et al., 2020). Finally, we use DIAYN+prior (Eysenbach et al., 2018; Chalumeau et al., 2022) with $q(\mathbf{z}_{\text{DIAYN}}|s_t)$ and feature priors to guide SMERL and REVERSE SMERL towards relevant skills as explained in DIAYNoriginal paper. More details about the baselines are available in Appendix D.1. DCG-ME is a state-of-the-art Quality-Diversity algorithm for challenging continuous control tasks that evolves a population of both high-performing and diverse solutions, and simultaneously distills those solutions into a single skill-conditioned policy. We also evaluate the performance of SCOPA against a standard

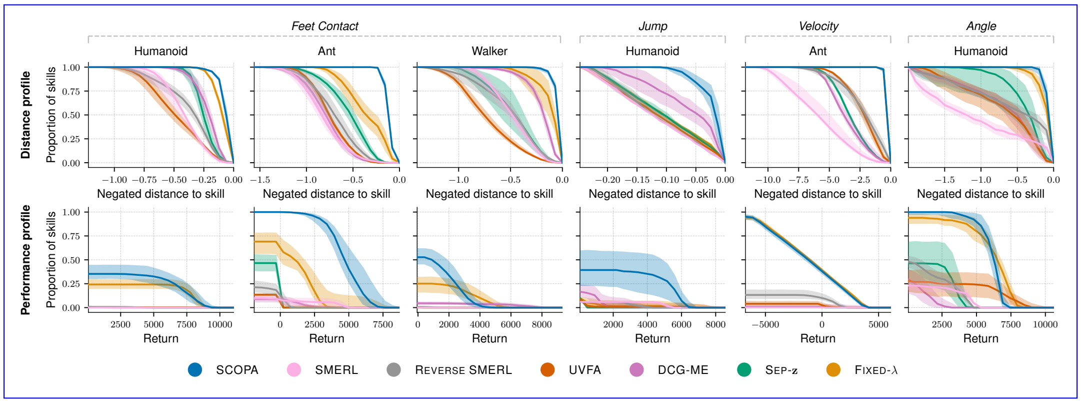


Figure 2: (top) Distance profiles and (bottom) performance profiles for each task defined in Section 5.3. The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI. Figure F.22 illustrates how to read distance and performance profiles.

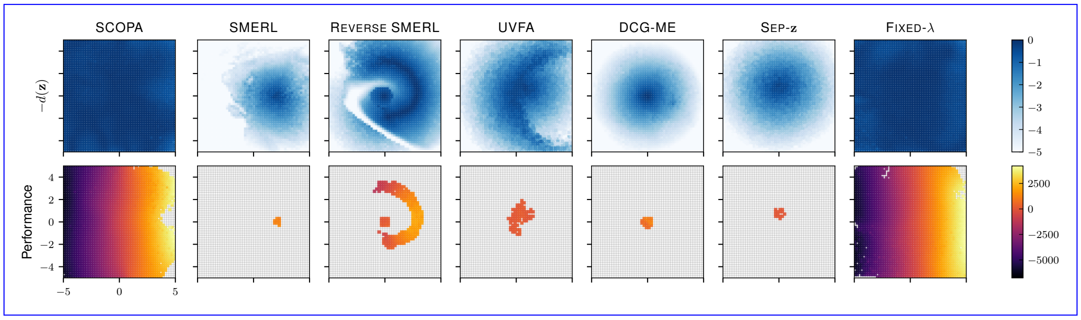


Figure 3: Heatmaps of (top) negative distance to skill, (bottom) performance defined in Section 5.3, for Ant Velocity. The heatmap represents the skill space $\mathcal{Z} = [-5 \text{ m/s}, 5 \text{ m/s}]^2$, of target velocities. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = [v_x \quad v_y]^\intercal$. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

GCRL algorithm, which rely on Universal Value Function Approximators (UVFA). UVFA is trained similar to SCOPA but the Lagrangian multiplier $\lambda_{\mathcal{L}}$ is fixed, and maximizes a reward that balances between task reward and following the skill at each time step: $(1 - \lambda_{\mathcal{L}})r_t - \lambda_{\mathcal{L}} \left\| \boldsymbol{\phi}_t - \mathbf{z} \right\|_2$.

We perform two additional ablation experiments, that we call Separable-**z** (Sep-**z**) and Fixed-λ. For Sep-**z**, we remove the successor features representation and use a naive distance to skill instead $\sum_t \gamma^t \left\| \boldsymbol{\phi}_t - \mathbf{z} \right\|_2$. For Fixed-$\lambda$, we remove the Lagrangian multiplier, so the value of $\lambda_{\mathcal{L}}$ remains constant. Thus, Sep-**z** and Fixed-λ bridge the gap between SCOPA and UVFA. A summarized description of all variants under study is provided in Table D.3.

5.3 EVALUATION METRICS

We evaluate our method using two types of metrics: (1) the distance to skill metrics that evaluate the ability of an agent to execute target skills, and (2) the performance metrics that quantify the ability of an agent to maximize return while executing target skills. Each experiment is replicated 10 times with random seeds independently and trains all components from scratch. We report the Inter-Quartile Mean (IQM) value for each metric, with the estimated 95% Confidence Interval (CI) (Agarwal et al., 2021). The statistical significance of the results is evaluated using the Mann-Whitney U test (Mann & Whitney, 1947) and the probabilities of improvement are reported in Appendix A.1.

Distance to skill metrics To evaluate the ability of a policy to achieve a given skill $\mathbf{z}$, we estimate the *expected distance to skill*, denoted $d(\mathbf{z})$, by averaging the euclidean distance between the desired skill $\mathbf{z}$ and the observed skill over 10 rollouts. First, we use $d(\mathbf{z})$ to compute *distance profiles* on Figure 2, which quantify for a given distance d, the proportion of skills in the skill space that have an expected distance to skill smaller than d, computed with the function $d \mapsto \frac{1}{N_{\mathbf{z}}} \sum_{i=1}^{N_{\mathbf{z}}} \mathbb{1}(d(\mathbf{z}_i) < d)$. Second, we summarize the ability of a policy to execute skills with the *distance score*, $\frac{1}{N_{\mathbf{z}}} \sum_{i=1}^{N_{\mathbf{z}}} -d(\mathbf{z}_i)$.

Performance metrics To evaluate the ability of a policy to solve a task given a skill $\mathbf{z}$, we estimate the *expected undiscounted return*, denoted $R(\mathbf{z})$, by averaging the return over 10 rollouts. First, we use $R(\mathbf{z})$ to compute *performance profiles* on Figure 2, which quantify for a given return R, the proportion of skills in the skill space that have an expected return larger than R, after filtering out the skills that are not achieved by the policy. To this end, we compute the expected distance to skill $d(\mathbf{z})$, and discard skills with an expected distance to skill that is larger than a predefined threshold, $d(\mathbf{z}) > d_{\text{eval}}$. More precisely, the performance profile is the function $R \mapsto \frac{1}{N_{\mathbf{z}}} \sum_{i=1}^{N_{\mathbf{z}}} \mathbb{1}(d(\mathbf{z}_i) < d_{\text{eval}}, R(\mathbf{z}_i) > R)$. Second, we summarize the ability of a policy to maximize return while executing skills, with the *performance score*, $\frac{1}{N_{\mathbf{z}}} \sum_{i=1}^{N_{\mathbf{z}}} R(\mathbf{z}) \mathbb{1}(d(\mathbf{z}_i) < d_{\text{eval}})$.

5.4 RESULTS

The goal of our experiments is to answer two questions: (1) Is SCOPA capable of solving problem P1? (2) Are the high-performing and diverse skills useful to adapt to new MDPs? Our results show that SCOPA achieves significantly better distance and performance scores compared to other baselines (Fig. 4). Empirical evaluations also show that SCOPA can adapt to out-of-distribution changes, in a few shots or with hierarchical learning. Videos of our results are available at: `bit.ly/scopa`.

Distance to skill SCOPA outperforms all baselines in executing a wide range of skills (Fig. 2). SMERL, REVERSE SMERLand UVFAminimize the distance between features and skills. However, for challenging skill spaces like feet contact, this approach cannot achieve most of the skills, see Section 3. This is why they can only execute skills in the corners of the skill space where $\mathbf{z} = \phi_t$, as shown in Figure A.9. Notably, SCOPAis capable of achieving skills that are contrary to the task reward, as illustrated by the velocity features in Figure 2 and Figure 3, which is not the case for SMERLor REVERSE SMERL. SCOPAoutperforms SMERLin all tasks since SMERLprimarily optimizes for the task reward rather than optimizing to acquire skills. Finally, our approach outperforms DCG-ME that fails to explicitly minimize the expected skill distance, a common issue among QD algorithms (Flageat & Cully, 2023). These comparisons highlight the significance of the policy skill improvement term in blue in Equation 4.

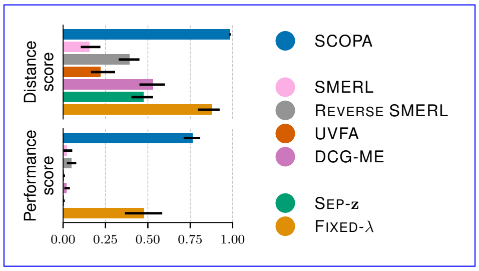


Figure 4: Distance and performance scores normalized and aggregated across all tasks. The values correspond to the IQM while the error bars represent IQM 95% CI.

Performance SCOPA outperforms most baselines in maximizing return (Fig. 4), as they don't achieve many skills in the first place, and the performance score only evaluates the performance of skills successfully executed by the policy. FIXED-λ is a variation of our method that does not adapt the Lagrange multiplier and is the only baseline that gets performance scores and profiles comparable to SCOPA. However, SCOPA is capable to learn to hop on one foot in Humanoid feet contact. In contrast, FIXED-λ covers a smaller range of skills, as evidenced by the edges of the skill space on Figure A.9. Additionally, SCOPA significantly outperforms FIXED-λ on the challenging jump task, as demonstrated in Figure 2, due to the necessity of a strong weight on the constraint. Ultimately, using an adaptive $\lambda_{\mathcal{L}}$ proves advantageous for our approach.

Using skills for adaptation We evaluate on five downstream tasks, namely Humanoid - Hurdles, Humanoid - Motor Failure, Ant - Gravity, Walker - Friction and Ant - Obstacle. On Ant - Obstacle, we hierarchically learn a meta-controller to choose the skills of a pre-trained policy. On all tasks except Ant - Obstacle, we perform few-shot adaptation experiments where no re-training is allowed and where we evaluate the top-performing skills for each method, see Appendix C.2 for more details. SCOPAis consistently as robust or more robust than other baselines on all downstream tasks except on Walker - Friction where DCG-MEis better, see Figure 5. On the hurdles task,

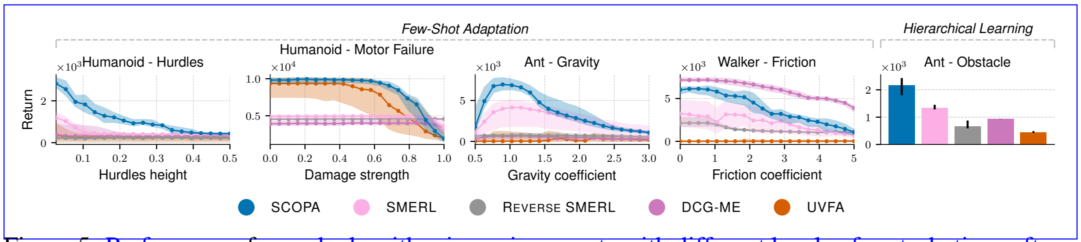


Figure 5: Performance for each algorithm in environments with different levels of perturbations after few-shot adaptation or after hierarchical learning. The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI.

SCOPA significantly outperforms other baselines by consistently jumping over higher hurdles. On the motor failure task, SCOPA and UVFA show great robustness on a wide range of damage strengths but perform worse than SMERLand REVERSE SMERL for maximal damage. Finally, on the obstacle task, SCOPA significantly outperforms other baselines.

6 RELATED WORK

Learning Diverse Optimal Policies Quality-Diversity (QD) optimization is a family of evolutionary algorithms that generate large collections of solutions, such as policies, that are both diverse and high-performing (Pugh et al., 2016; Cully & Demiris, 2018). A feature space (also called descriptor space) is defined to evaluate the diversity of the solutions (Cully et al., 2015). QD algorithms have been combined with RL-based methods (Pierrot et al., 2022; Nilsson & Cully, 2021; Faldor et al., 2023; Tjanaka et al., 2022; Fontaine et al., 2020) and have been shown to be effective for skill discovery (Chalumeau et al., 2022). QD approaches and our framework both solve the same problem, however most QD algorithms output a large number of policies instead of a single skill-conditioned one, which can be difficult to deal with in practice in downstream tasks (Faldor et al., 2023). Finally, QD algorithms rely on Genetics Algorithms which can cause slow convergence in challenging environments with high-dimensional state spaces (Pierrot et al., 2022). Similarly to QD methods, DOMiNO (Zahavy et al., 2022) learns a set of diverse and near-optimal policies, but contrary to SCOPA, the policies discovered by DOMiNO are not trained to execute specific target skills $\mathbf{z}$.

Unsupervised Reinforcement Learning Most URL approaches discover diverse skills by maximizing an intrinsic reward that is defined in terms of the discriminability of the trajectories induced by the different skills. Usually, the methods maximize the Mutual Information (MI) between the trajectories and the skills $I(\tau, \mathbf{z})$ (Gregor et al., 2016; Sharma et al., 2019; Mazzaglia et al., 2022), which is usually reduced to a MI-maximization between skills and states with the following lower bound: $I(\tau, \mathbf{z}) \geq \sum_{t=1}^{T} I(s_t, \mathbf{z})$. Moreover, those methods may rely on the use of priors to focus the exploration on useful skills (Eysenbach et al., 2018; Sharma et al., 2019). In addition to priors, some approaches like SMERL (Kumar et al., 2020) or DIAYN+REWARD and DADS+REWARD (Chalumeau et al., 2022) maximize a trade-off between the extrinsic reward from the task, and an intrinsic reward for diversity, further reducing the gap between those methods and our algorithm. Other approaches, such as (Park et al., 2022; 2023), manage to discover diverse dynamic skills without any prior on ϕ; nonetheless, these approaches do not maximize any extrinsic reward.

Goal-Conditioned Reinforcement Learning GCRL trains agents to achieve different goals, which are commonly defined as a subspace of the states, $\mathbf{g} = \phi(s)$. However, some approaches don't ignore the reward of the original problem to learn useful policies (Sutton et al., 2023). Additionally, GCRL can be reformulated in a maximum mutual information setting, further reducing the gap between GCRL and skill discovery (Choi et al., 2021; Gu et al., 2021). In our method, if we ignore the extrinsic reward and optimize only to satisfy the constraint, we can show that our problem is related to GCRL. Moreover, in Goal-Augmented MDP (GA-MDP), the reward $r(s_t, a_t, \mathbf{g}) = \mathbb{1}(\|\boldsymbol{\phi}_{t+1} - \mathbf{g}\|_2 \leq \delta)$ is sparse (Andrychowicz et al., 2017) and reshaped dense rewards like $r(s_t, a_t, \mathbf{g}) = -d(\boldsymbol{\phi}_{t+1}, \mathbf{g})$ cause local optima challenges (Trott et al., 2019). which justifies the successor features policy skill improvement introduced in Section 4. Another line of work learns diverse exploring options, which can then be used in a temporally-extended manner to improve the efficiency of RL algorithms (Klissarov & Machado, 2023).

7 CONCLUSION

In this work, we present a quality-diversity reinforcement learning algorithm, SCOPA , that learns to execute a continuous range of skills while maximizing return. We extend the generalized policy iteration framework with a policy skill improvement update that is analogous to the classic policy improvement update. Our approach seamlessly unifies value function and successor features policy iteration with constrained optimization to (1) maximize performance, while (2) executing skills.

Our empirical evaluations suggest that SCOPA outperforms some skill-conditioned reinforcement learning and quality-diversity methods on a variety of continuous control locomotion tasks. Quantitative results demonstrate that SCOPAcan extrapolate to out-of-distribution test conditions, while qualitative analyses showcase a range of impressive behaviors.

In this paper, the feature functions are defined in advance, and provide the user the opportunity to give the algorithm prior information about potential downstream tasks, as well as more control and interpretability for the skills. However, an exciting direction for future work would be to learn the feature functions in an unsupervised manner, broadening the scope to include task-agnostic skills.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=S1ANxQW0b`.

Abbas Abdolmaleki, Sandy H. Huang, Giulia Vezzani, Bobak Shahriari, Jost Tobias Springenberg, Shruti Mishra, Dhruva TB, Arunkumar Byravan, Konstantinos Bousmalis, Andras Gyorgy, Csaba Szepesvari, Raia Hadsell, Nicolas Heess, and Martin Riedmiller. On Multi-objective Policy Optimization as a Tool for Reinforcement Learning: Case Studies in Offline RL and Finetuning, August 2023. URL `http://arxiv.org/abs/2106.08199`. arXiv:2106.08199 [cs].

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html`.

Maxime Allard, Simón C. Smith, Konstantinos Chatzilygeroudis, and Antoine Cully. Hierarchical quality-diversity for online damage recovery. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 58–67, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528751. URL `https://dl.acm.org/doi/10.1145/3512290.3528751`.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf`.

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.

Sumeet Batra, Bryon Tjanaka, Matthew C. Fontaine, Aleksei Petrenko, Stefanos Nikolaidis, and Gaurav Sukhatme. Proximal Policy Gradient Arborescence for Quality Diversity Reinforcement

Learning, May 2023. URL `http://arxiv.org/abs/2305.13795`. arXiv:2305.13795 [cs].

Diana Borsa, Andre Barreto, John Quan, Daniel J. Mankowitz, Hado van Hasselt, Remi Munos, David Silver, and Tom Schaul. Universal Successor Features Approximators. September 2018. URL `https://openreview.net/forum?id=S1VWjiRcKX`.

Felix Chalumeau, Raphael Boige, Bryan Lim, Valentin Macé, Maxime Allard, Arthur Flajolet, Antoine Cully, and Thomas Pierrot. Neuroevolution is a Competitive Alternative to Reinforcement Learning for Skill Discovery. September 2022. URL `https://openreview.net/forum?id=6BHlZgyPOZY`.

Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational Empowerment as Representation Learning for Goal-Conditioned Reinforcement Learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 1953–1963. PMLR, July 2021. URL `https://proceedings.mlr.press/v139/choi21b.html`. ISSN: 2640-3498.

Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2018. doi: 10.1109/TEVC.2017.2704781.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14422. URL `http://arxiv.org/abs/1407.3501`. arXiv:1407.3501 [cs, q-bio].

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613.

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`.

Maxence Faldor, Félix Chalumeau, Manon Flageat, and Antoine Cully. MAP-Elites with Descriptor-Conditioned Gradients and Archive Distillation into a Single Policy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, pp. 138–146, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590503. URL `https://dl.acm.org/doi/10.1145/3583131.3590503`.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1126–1135, Sydney, NSW, Australia, August 2017. JMLR.org.

Manon Flageat and Antoine Cully. Uncertain quality-diversity: Evaluation methodology and new methods for quality-diversity in uncertain domains. *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2023. doi: 10.1109/TEVC.2023.3273560.

Manon Flageat, Bryan Lim, Luca Grillotti, Maxime Allard, Simón C. Smith, and Antoine Cully. Benchmarking Quality-Diversity Algorithms on Neuroevolution for Reinforcement Learning, November 2022. URL `http://arxiv.org/abs/2211.02193`. arXiv:2211.02193 [cs].

Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, pp. 94–102, New York, NY, USA, June 2020. Association for Computing Machinery. ISBN 978-1-4503-7128-5. doi: 10.1145/3377930.3390232. URL `https://dl.acm.org/doi/10.1145/3377930.3390232`.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation. June 2021. URL `https://openreview.net/forum?id=VdvDlnnjzIN`.

Jonas Gehring, Gabriel Synnaeve, Andreas Krause, and Nicolas Usunier. Hierarchical Skills for Efficient Exploration. November 2021. URL `https://openreview.net/forum?id=NbaEmFm2mUW`.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational Intrinsic Control, November 2016. URL `http://arxiv.org/abs/1611.07507`. arXiv:1611.07507 [cs].

Luca Grillotti and Antoine Cully. Relevance-guided unsupervised discovery of abilities with quality-diversity algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 77–85, New York, NY, USA, July 2022a. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528837. URL `https://dl.acm.org/doi/10.1145/3512290.3528837`.

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

Luca Grillotti, Manon Flageat, Bryan Lim, and Antoine Cully. Don't Bet on Luck Alone: Enhancing Behavioral Reproducibility of Quality-Diversity Solutions in Uncertain Domains. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, pp. 156–164, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590498. URL `https://dl.acm.org/doi/10.1145/3583131.3590498`.

Shixiang Shane Gu, Manfred Diaz, Daniel C. Freeman, Hiroki Furuta, Seyed Kamyar Seyed Ghasemipour, Anton Raichuk, Byron David, Erik Frey, Erwin Coumans, and Olivier Bachem. Braxlines: Fast and Interactive Toolkit for RL-driven Behavior Engineering beyond Reward Maximization, October 2021. URL `http://arxiv.org/abs/2110.04686`. arXiv:2110.04686 [cs].

David Ha and Jürgen Schmidhuber. World Models. March 2018. doi: 10.5281/zenodo.1207631. URL `http://arxiv.org/abs/1803.10122`. arXiv:1803.10122 [cs, stat].

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications, January 2019. URL `http://arxiv.org/abs/1812.05905`. arXiv:1812.05905 [cs, stat].

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. September 2019a. URL `https://openreview.net/forum?id=S1lOTC4tDS`.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565. PMLR, May 2019b. URL `https://proceedings.mlr.press/v97/hafner19a.html`. ISSN: 2640-3498.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, and David Silver. Emergence of Locomotion Behaviours in Rich Environments. July 2017.

Martin Klissarov and Marlos C. Machado. Deep Laplacian-based Options for Temporally-Extended Exploration, June 2023. URL `http://arxiv.org/abs/2301.11181`. arXiv:2301.11181 [cs].

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.

Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J. Bentley, Samuel Bernard, Guillaume Beslon, David M. Bryson, Nick Cheney, Patryk Chrabaszcz, Antoine Cully, Stephane Doncieux, Fred C. Dyer, Kai Olav Ellefsen, Robert Feldt, Stephan Fischer, Stephanie Forrest, Antoine Fŕenoy, Christian Gagńe, Leni Le Goff, Laura M. Grabowski, Babak Hodjat, Frank Hutter, Laurent Keller, Carole Knibbe, Peter Krcah, Richard E. Lenski, Hod

Lipson, Robert MacCurdy, Carlos Maestre, Risto Miikkulainen, Sara Mitri, David E. Moriarty, Jean-Baptiste Mouret, Anh Nguyen, Charles Ofria, Marc Parizeau, David Parsons, Robert T. Pennock, William F. Punch, Thomas S. Ray, Marc Schoenauer, Eric Schulte, Karl Sims, Kenneth O. Stanley, François Taddei, Danesh Tarapore, Simon Thibault, Richard Watson, Westley Weimer, and Jason Yosinski. The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities. *Artificial Life*, 26 (2):274–306, May 2020. ISSN 1064-5462. doi: 10.1162/artl_a_00319. URL `https://doi.org/10.1162/artl_a_00319`.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-Conditioned Reinforcement Learning: Problems and Solutions. volume 6, pp. 5502–5511, July 2022. doi: 10.24963/ijcai.2022/770. URL `https://www.ijcai.org/proceedings/2022/770`. ISSN: 1045-0823.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.

Gabriel B. Margolis and Pulkit Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior, December 2022. URL `http://arxiv.org/abs/2212.03238`. arXiv:2212.03238 [cs, eess].

Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid Locomotion via Reinforcement Learning, May 2022. URL `http://arxiv.org/abs/2205.02824`. arXiv:2205.02824 [cs].

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Alexandre Lacoste, and Sai Rajeswar. Choreographer: Learning and Adapting Skills in Imagination. October 2022. URL `https://openreview.net/forum?id=BxYsP-7ggf`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, December 2013. URL `http://arxiv.org/abs/1312.5602`. arXiv:1312.5602 [cs].

Olle Nilsson and Antoine Cully. Policy gradient assisted MAP-Elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, pp. 866–875, New York, NY, USA, June 2021. Association for Computing Machinery. ISBN 978-1-4503-8350-9. doi: 10.1145/3449639.3459304. URL `https://dl.acm.org/doi/10.1145/3449639.3459304`.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving Rubik's Cube with a Robot Hand, October 2019. URL `http://arxiv.org/abs/1910.07113`. arXiv:1910.07113 [cs, stat].

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=BGvt0ghNgA`.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245. PMLR, 2023. URL `https://proceedings.mlr.press/v202/park23h.html`.

Thomas Pierrot, Valentin Macé, Felix Chalumeau, Arthur Flajolet, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin-Gilbert. Diversity policy gradient for sample efficient quality-diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 1075–1083, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528845. URL `https://dl.acm.org/doi/10.1145/3512290.3528845`.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI*, 3, 2016. ISSN 2296-9144. URL `https://www.frontiersin.org/articles/10.3389/frobt.2016.00040`.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 978-0-471-61977-2.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal Value Function Approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1312–1320. PMLR, June 2015. URL `https://proceedings.mlr.press/v37/schaul15.html`. ISSN: 1938-7228.

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`.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961. URL `https://www.nature.com/articles/nature16961`. Number: 7587 Publisher: Nature Publishing Group.

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.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL `https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf`.

Richard S. Sutton, Marlos C. Machado, G. Zacharias Holland, David Szepesvari, Finbarr Timbers, Brian Tanner, and Adam White. Reward-Respecting Subtasks for Model-Based Reinforcement Learning. *Artificial Intelligence*, 324:104001, November 2023. ISSN 00043702. doi: 10.1016/j.artint.2023.104001. URL `http://arxiv.org/abs/2202.03466`. arXiv:2202.03466 [cs].

Bryon Tjanaka, Matthew C. Fontaine, Julian Togelius, and Stefanos Nikolaidis. Approximating gradients for differentiable quality diversity in reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 1102–1111, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528705. URL `https://dl.acm.org/doi/10.1145/3512290.3528705`.

Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping Your Distance: Solving Sparse Reward Tasks Using Self-Balancing Shaped Rewards. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/hash/64c26b2a2dcf068c49894bd07e0e6389-Abstract.html`.

Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering Policies with DOMiNO: Diversity Optimization Maintaining Near Optimality. September 2022. URL `https://openreview.net/forum?id=kjkdzBW3b8p`.

Tom Zahavy, Vivek Veeriah, Shaobo Hou, Kevin Waugh, Matthew Lai, Edouard Leurent, Nenad Tomasev, Lisa Schut, Demis Hassabis, and Satinder Singh. Diversifying AI: Towards Creative Chess with AlphaZero, August 2023. URL `http://arxiv.org/abs/2308.09175`. arXiv:2308.09175 [cs].

Menghui Zhu, Minghuan Liu, Jian Shen, Zhicheng Zhang, Sheng Chen, Weinan Zhang, Deheng Ye, Yong Yu, Qiang Fu, and Wei Yang. MapGo: Model-Assisted Policy Optimization for Goal-Oriented Tasks, May 2021. URL `http://arxiv.org/abs/2105.06350`. arXiv:2105.06350 [cs].

A SUPPLEMENTARY RESULTS

A.1 QUANTITATIVE RESULTS

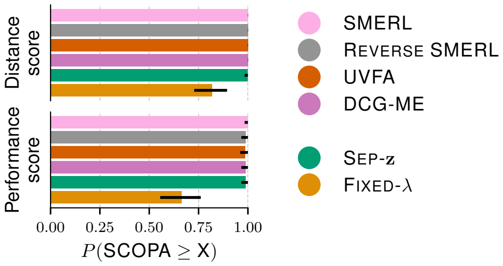


Figure A.6: Probabilities of improvement of SCOPA over all other baselines, aggregated across all tasks, as defined by Agarwal et al. (2021).

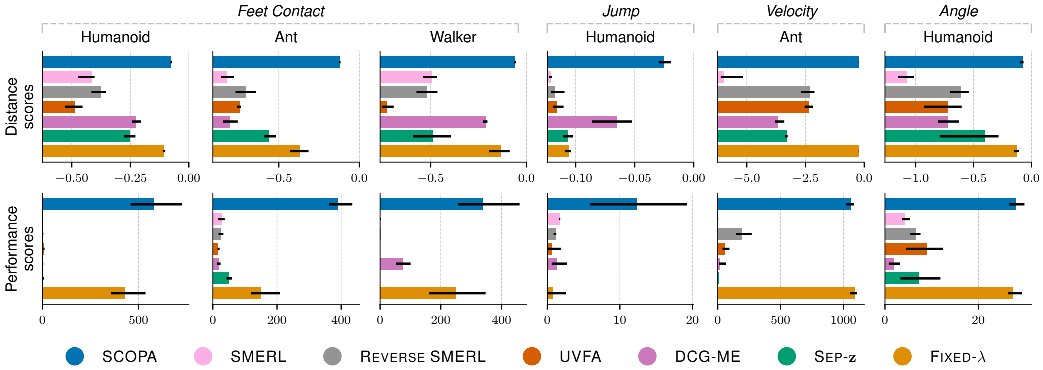


Figure A.7: IQM for distance and performance scores per task.

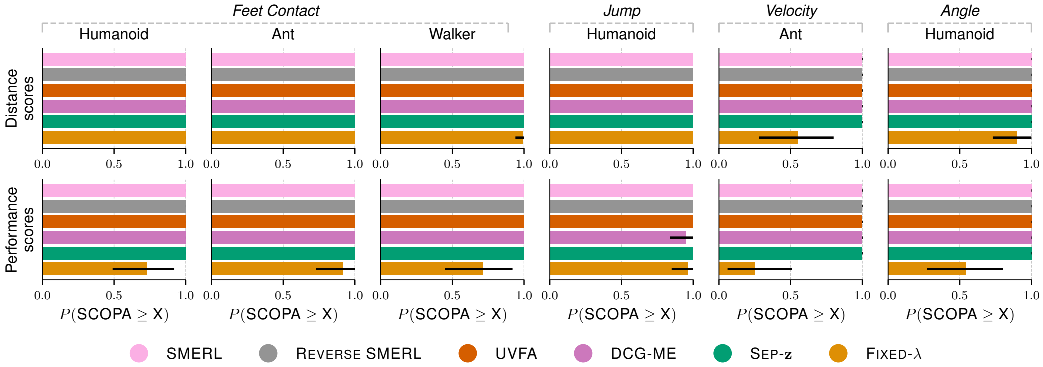


Figure A.8: Per-task probabilities of improvement (as defined by Agarwal et al. (2021)) of SCOPA over all other baselines.

A.2 HEATMAPS

In Figures A.9 to A.14, we report the profiles and heatmaps defined in Section 5.3. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

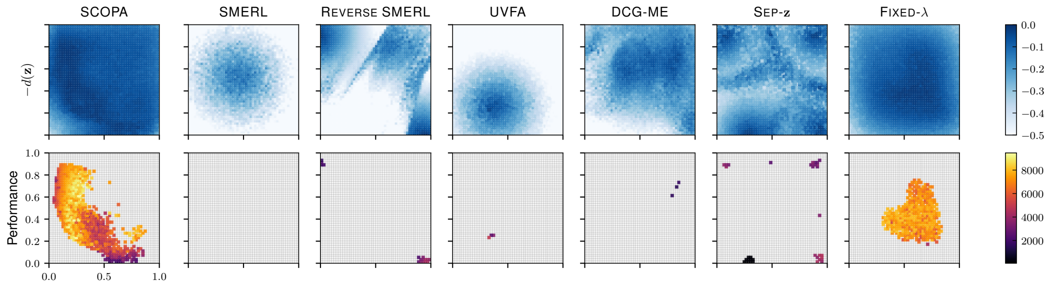


Figure A.9: **Humanoid Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

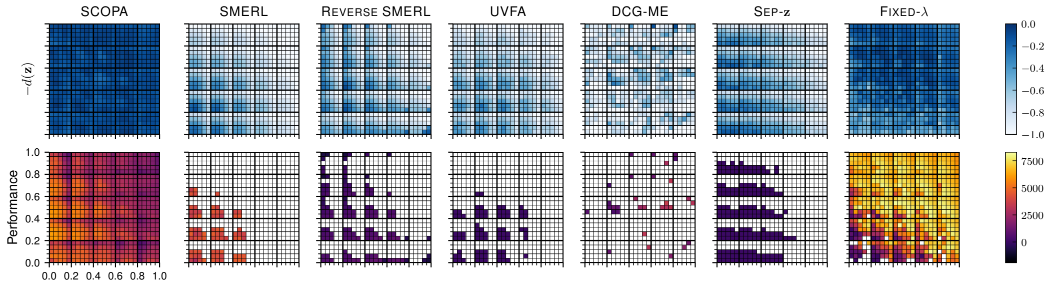


Figure A.10: **Ant Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

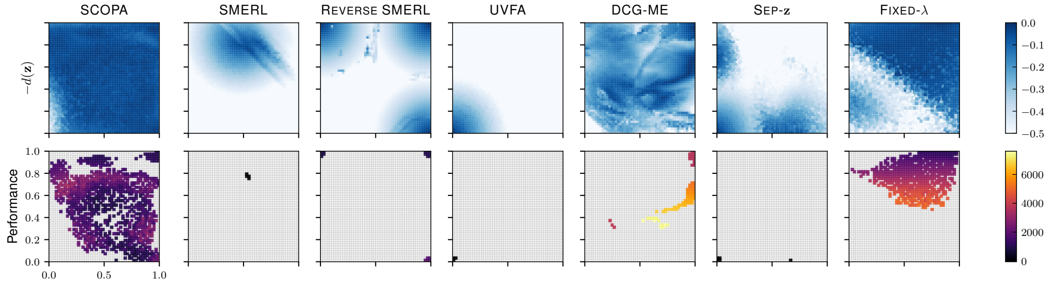


Figure A.11: **Walker Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

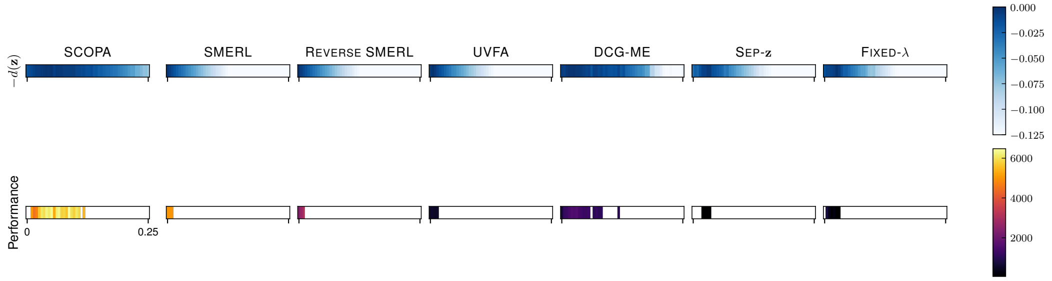


Figure A.12: **Humanoid Jump** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

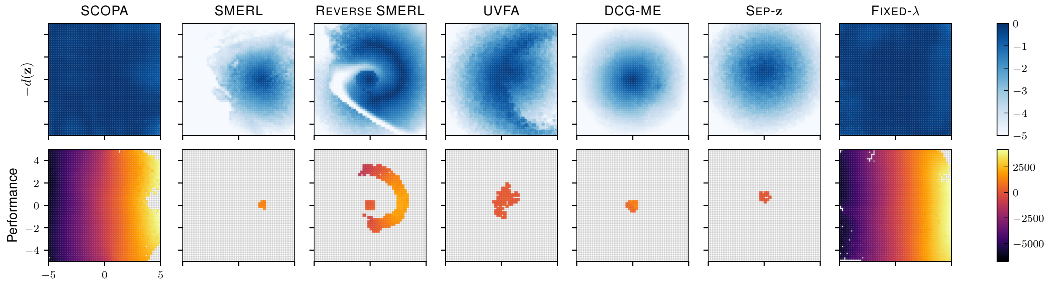


Figure A.13: **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

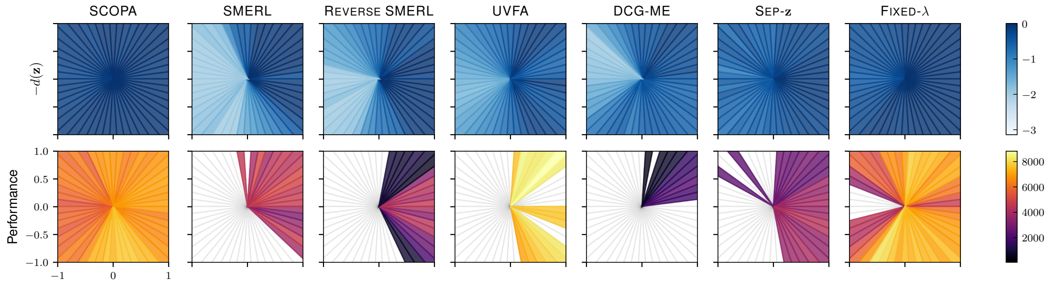


Figure A.14: **Humanoid Angle** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$).

A.3 RESULTS WITHOUT FILTERING WITH d_{EVAL}

In Figures A.15 to A.21, we report the profiles and heatmaps defined in Section 5.3 except that skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

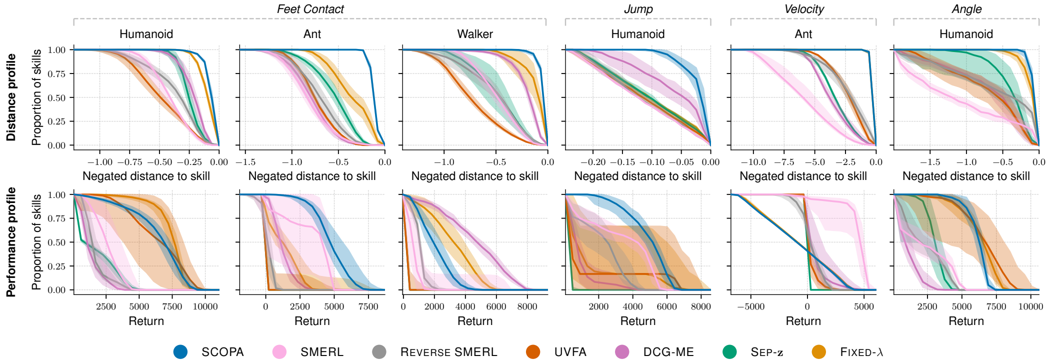


Figure A.15: **(top)** Distance profiles and **(bottom)** performance profiles for each task defined in Section 5.3 similar to Figure 2 except that skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out. The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI. Figure F.22 illustrates how to read distance and performance profiles.

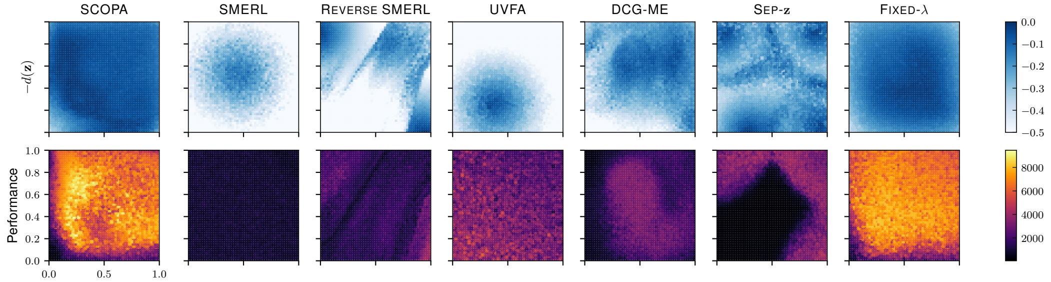


Figure A.16: **Humanoid Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

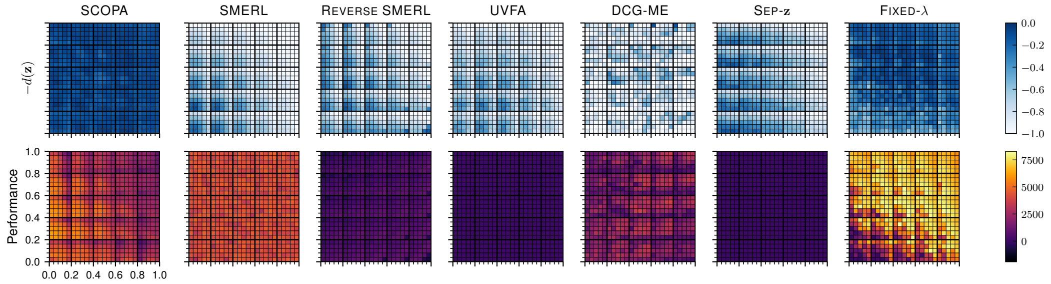


Figure A.17: **Ant Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

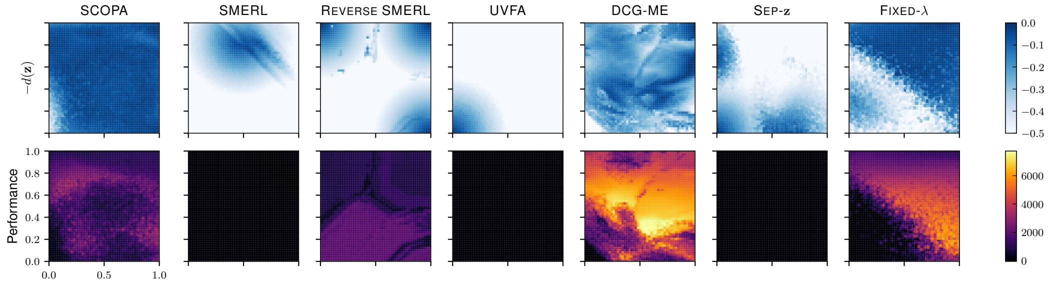


Figure A.18: **Walker Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

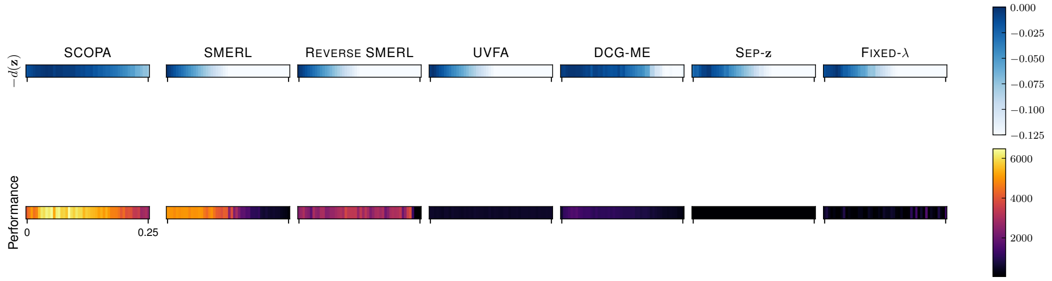


Figure A.19: **Humanoid Jump** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

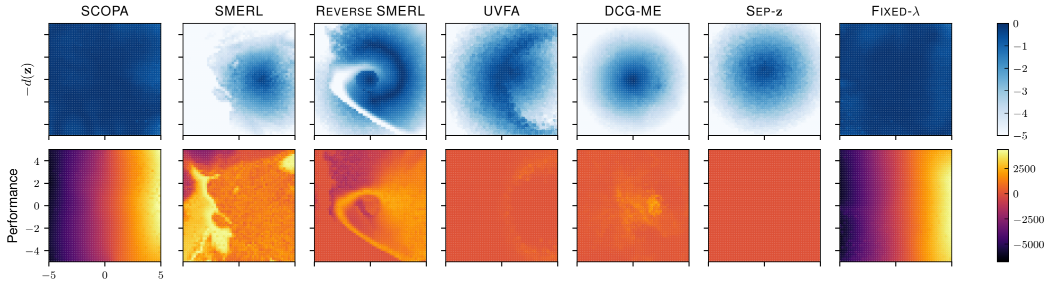


Figure A.20: **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

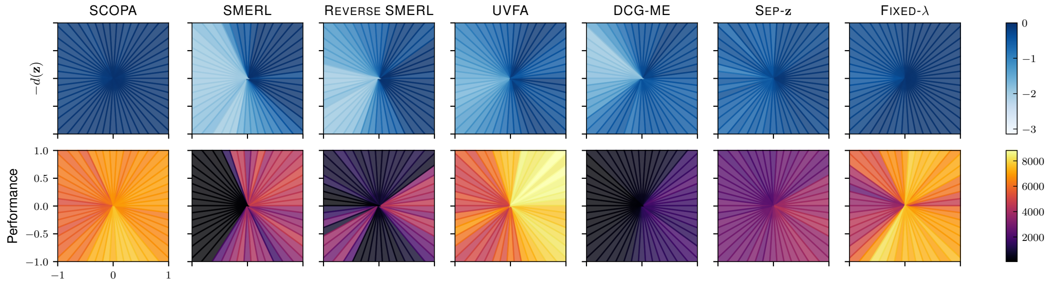


Figure A.21: **Humanoid Angle** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3 for all baselines. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

B THEORETICAL RESULTS

Proposition. *Consider an infinite horizon, finite MDP with observable features in Φ. Let π be a policy and let ψ be the discounted successor features. Then, for all skills $\mathbf{z} \in \mathcal{Z}$, we can derive an upper bound for the distance between $\mathbf{z}$ and the expected features under π:*

$$\left\|\mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\boldsymbol{\phi}(s)\right]-\mathbf{z}\right\|_2 \le \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\left\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z})-\mathbf{z}\right\|_2\right] \tag{5}$$

Proof. For all states $s \in \mathcal{S}$, the Bellman equation for ψ gives:

$$\boldsymbol{\psi}(s,\mathbf{z}) = \boldsymbol{\phi}(s) + \gamma \mathbb{E}_{\substack{a\sim\pi(.|s,\mathbf{z}) \\ s'\sim p(.|s,a)}}\left[\boldsymbol{\psi}(s',\mathbf{z})\right] \tag{6}$$

For all skills $\mathbf{z} \in \mathcal{Z}$ and for all sequences of T states $(s_0, \ldots, s_{T-1})$ sampled from $\pi(\cdot|\cdot,\mathbf{z})$, we have:

$$\begin{aligned}
&\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t-\mathbf{z}\right\|_2 \\
&=\left\|\frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{\phi}_t-\mathbf{z})\right\|_2 \\
&=\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\boldsymbol{\psi}(s_t,\mathbf{z})-\gamma\mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}}\left[\boldsymbol{\psi}(s_{t+1},\mathbf{z})\right]-\mathbf{z}\right)\right\|_2 && \text{(Equation 6)} \\
&=\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left((1-\gamma)\boldsymbol{\psi}(s_t,\mathbf{z})+\gamma\boldsymbol{\psi}(s_t,\mathbf{z})-\gamma\mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}}\left[\boldsymbol{\psi}(s_{t+1},\mathbf{z})\right]-\mathbf{z}\right)\right\|_2 \\
&=\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left((1-\gamma)\boldsymbol{\psi}(s_t,\mathbf{z})-\mathbf{z}\right)+\frac{\gamma}{T}\sum_{t=0}^{T-1}\left(\boldsymbol{\psi}(s_t,\mathbf{z})-\mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}}\left[\boldsymbol{\psi}(s_{t+1},\mathbf{z})\right]\right)\right\|_2 \\
&\le\frac{1}{T}\sum_{t=0}^{T-1}\left\|(1-\gamma)\boldsymbol{\psi}(s_t,\mathbf{z})-\mathbf{z}\right\|_2+\gamma\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\boldsymbol{\psi}(s_t,\mathbf{z})-\mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}}\left[\boldsymbol{\psi}(s_{t+1},\mathbf{z})\right]\right)\right\|_2 \\
& && \text{(triangular inequality)}
\end{aligned}$$

We denote $\rho(s) = \lim_{t\to\infty} P(s_t = s|s_0, \pi)$ the stationary distribution of states under π, which we assume exists and is independent of s_0. Thus, we can write:

$$
\begin{aligned}
&\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left(\boldsymbol{\psi}(s_t, \mathbf{z}) - \mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}} [\boldsymbol{\psi}(s_{t+1}, \mathbf{z})] \right) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}(s_t, \mathbf{z}) - \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left(\mathbb{E}_{\substack{a_t\sim\pi(.|s_t,\mathbf{z}) \\ s_{t+1}\sim p(.|s_t,a_t)}} [\boldsymbol{\psi}(s_{t+1}, \mathbf{z})] \right) \\
&= \mathbb{E}_{s\sim\rho} [\boldsymbol{\psi}(s, \mathbf{z})] - \mathbb{E}_{s\sim\rho} \left[\mathbb{E}_{\substack{a\sim\pi(.|s,\mathbf{z}) \\ s'\sim p(.|s,a)}} [\boldsymbol{\psi}(s', \mathbf{z})] \right] \\
&= \mathbb{E}_{s\sim\rho} [\boldsymbol{\psi}(s, \mathbf{z})] - \mathbb{E}_{\substack{s\sim\rho \\ a\sim\pi(.|s,\mathbf{z}) \\ s'\sim p(.|s,a)}} [\boldsymbol{\psi}(s', \mathbf{z})] \\
&= \mathbb{E}_{s\sim\rho} [\boldsymbol{\psi}(s, \mathbf{z})] - \mathbb{E}_{s'\sim\rho} [\boldsymbol{\psi}(s', \mathbf{z})] \\
&= 0
\end{aligned}
$$

Finally, by taking the inequality to the limit as $T \to \infty$, we get:

$$
\begin{aligned}
\left\| \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})} [\boldsymbol{\phi}(s)] - \mathbf{z} \right\|_2 &\le \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})} [\|(1-\gamma)\boldsymbol{\psi}(s, \mathbf{z}) - \mathbf{z}\|_2] + 0 \\
\left\| \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})} [\boldsymbol{\phi}(s)] - \mathbf{z} \right\|_2 &\le \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})} [\|(1-\gamma)\boldsymbol{\psi}(s, \mathbf{z}) - \mathbf{z}\|_2]
\end{aligned}
$$

□

Proposition B.1. *Consider a continuous MDP with a bounded feature space Φ, a skill $\mathbf{z} \in \mathcal{Z}$, and π a policy such that the sequence $\left(\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t\right)_{T\geq 1}$ almost surely*[1] *converges for trajectories sampled from $\pi(\cdot|\cdot,\mathbf{z})$. If we write $\epsilon := \sup_t \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}(s_t,\mathbf{z}) - \boldsymbol{\psi}(s_{t+1},\mathbf{z})\|_2\right]$, then:*

$$\mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\left\|\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2\right] \leq \sup_t \mathbb{E}_{\pi(\cdot|\cdot,\mathbf{z})}\left[\|(1-\gamma)\boldsymbol{\psi}(s_t,\mathbf{z}) - \mathbf{z}\|_2\right] + \epsilon \tag{7}$$

Furthermore, it is worth noting that if the MDP dynamics p and π are deterministic, then $\epsilon = 0$.

Proof. Let $\mathbf{z} \in \mathcal{Z}$.

To make the proof easier to read, we use the following notations:

$$\boldsymbol{\psi}_t := \boldsymbol{\psi}(s_t,\mathbf{z})$$
$$\pi_{\mathbf{z}} := \pi(\cdot|\cdot,\mathbf{z})$$

We define β as follows:

$$\beta := \sup_t \mathbb{E}_{\pi_{\mathbf{z}}}\left[\|(1-\gamma)\boldsymbol{\psi}_t - \mathbf{z}\|_2\right] \tag{8}$$

Then we have, for all t,

$$\mathbb{E}_{\pi_{\mathbf{z}}}\left[\|(1-\gamma)\boldsymbol{\psi}_t - \mathbf{z}\|_2\right] \leq \beta \tag{9}$$

$$\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}_t - \boldsymbol{\psi}_{t+1}\right\|_2\right] \leq \epsilon \tag{10}$$

The Bellman equation applied to Successor Features (SF) can be written:

$$\boldsymbol{\psi}_t = \boldsymbol{\phi}_t + \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right] \tag{11}$$

$$\text{or also: } \boldsymbol{\phi}_t = \boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right] \tag{12}$$

We can now derive an upper bound for $\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$. For all sequences of T states $s_{0:T-1}$ we have:

$$\begin{aligned}
\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2 &= \left\|\frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{\phi}_t - \mathbf{z})\right\|_2 \\
&= \left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right] - \mathbf{z}\right)\right\|_2 && \text{(from Equation 12)} \\
&= \left\|\frac{1}{T}\sum_{t=0}^{T-1}\left((1-\gamma)\boldsymbol{\psi}_t + \gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right] - \mathbf{z}\right)\right\|_2 \\
&= \left\|\frac{1}{T}\sum_{t=0}^{T-1}((1-\gamma)\boldsymbol{\psi}_t - \mathbf{z}) + \frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right]\right)\right\|_2 \\
&\leq \left\|\frac{1}{T}\sum_{t=0}^{T-1}((1-\gamma)\boldsymbol{\psi}_t - \mathbf{z})\right\|_2 + \left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right]\right)\right\|_2 \\
& && \text{(triangular inequality)}
\end{aligned}$$

---

[1] *almost sure* refers to the almost sure convergence from probability theory where rollouts are sampled from $\pi(\cdot|\cdot,\mathbf{z})$.

Thus,

$$\begin{aligned}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t-\mathbf{z}\right\|_2\right]\leq\mathbb{E}_{\pi_{\mathbf{z}}}&\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\right\|_2\right]\\&+\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\right)\right\|_2\right]\end{aligned}\tag{13}$$

We consider now the two terms on the right hand-side separately. First of all, we prove that the first term is lower than or equal to β:

$$\begin{aligned}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\right\|_2\right]&\leq\mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\|_2\right] &&\text{(triangular inequality)}\\&\leq\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\|((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\|_2\right]\\&\leq\frac{1}{T}\sum_{t=0}^{T-1}\beta &&\text{(from Equation 9)}\\&\leq\beta &&(14)\end{aligned}$$

Also, we can prove that the second term of the right-hand side in Equation 13 is lower than or equal to $\epsilon+\eta_T$ where $\lim_{T\to\infty}\eta_T=0$. For all sequences of T states $s_{0:T-1}$, we have:

$$\begin{aligned}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\right)&=\sum_{t=0}^{T-1}\gamma\boldsymbol{\psi}_t-\sum_{t=0}^{T-1}\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\\&=\gamma\boldsymbol{\psi}_0-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle|\,s_{T-1}\right]+\sum_{t=1}^{T-1}\gamma\boldsymbol{\psi}_t-\sum_{t=0}^{T-2}\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\\&=\gamma\boldsymbol{\psi}_0-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle|\,s_{T-1}\right]+\sum_{t=0}^{T-2}\gamma\boldsymbol{\psi}_{t+1}-\sum_{t=0}^{T-2}\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\\&=\gamma\boldsymbol{\psi}_0-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle|\,s_{T-1}\right]+\sum_{t=0}^{T-2}\left(\gamma\boldsymbol{\psi}_{t+1}-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\right)\end{aligned}$$

Thus, after dividing by T and applying the norm and expectation, we get:

$$\begin{aligned}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\right)\right\|_2\right]\leq\mathbb{E}_{\pi_{\mathbf{z}}}&\left[\left\|\frac{1}{T}\left(\gamma\boldsymbol{\psi}_0-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle|\,s_{T-1}\right]\right)\right\|_2\right]\\&+\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-2}\left(\gamma\boldsymbol{\psi}_{t+1}-\gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle|\,s_t\right]\right)\right\|_2\right]\\&\text{(triangular inequality)}\end{aligned}$$

Let $\eta_T := \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\left(\gamma\boldsymbol{\psi}_0 - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T \middle| s_{T-1}\right]\right)\right\|_2\right]$, we then have:

$$
\begin{aligned}
&\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right]\right)\right\|_2\right] \\
&\le \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-2}\left(\gamma\boldsymbol{\psi}_{t+1} - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right]\right)\right\|_2\right] && \text{(triangular inequality)} \\
&\le \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-2}\left\|\gamma\boldsymbol{\psi}_{t+1} - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t\right]\right\|_2\right] \\
&\le \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-2}\left\|\gamma\boldsymbol{\psi}_{t+1} + \boldsymbol{\phi}_t - \boldsymbol{\psi}_t\right\|_2\right] && \text{(from Equation 11)} \\
&\le \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\gamma\boldsymbol{\psi}_{t+1} + \boldsymbol{\phi}_t - \boldsymbol{\psi}_t\right\|_2\right] \\
&\le \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}_{t+1} - \boldsymbol{\psi}_t\right\|_2\right] \\
&\le \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\epsilon \\
&\le \eta_T + \frac{T-1}{T}\epsilon
\end{aligned}
\tag{15}
$$

After combining the two previously derived Equations 14 and 15, we get:

$$
\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2\right] \le \beta + \eta_T + \frac{T-1}{T}\epsilon \tag{16}
$$

Now we intend to prove that $\lim_{T\to\infty}\eta_T = 0$

$$
\begin{aligned}
\eta_T &= \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\left(\gamma\boldsymbol{\psi}_0 - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}\right]\right)\right\|_2\right] \\
&= \frac{\gamma}{T}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\left(\boldsymbol{\psi}_0 - \mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}\right]\right)\right\|_2\right] \\
&\le \frac{\gamma}{T}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_0\right\|_2 + \left\|\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}\right]\right\|_2\right] && \text{(triangular inequality)} \\
&\le \frac{\gamma}{T}\left(\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_0\right\|_2\right] + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}\right]\right\|_2\right]\right)
\end{aligned}
$$

As the space of features Φ is bounded, there exist a $\rho > 0$ such that for all $\boldsymbol{\phi} \in \Phi$, $\|\boldsymbol{\phi}\|_2 \le \rho$. Hence, for all t, $\|\boldsymbol{\psi}_t\|_2 = \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\sum_{i=0}^{\infty}\gamma^i\boldsymbol{\phi}_{t+i}\right\|_2 \middle| s_t\right] \le \mathbb{E}_{\pi_{\mathbf{z}}}\left[\sum_{i=0}^{\infty}\gamma^i\left\|\boldsymbol{\phi}_{t+i}\right\|_2 \middle| s_t\right] \le \frac{\rho}{1-\gamma}$. Hence,

$$
\begin{aligned}
\eta_T &\le \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}\right]\right\|_2\right]\right) \\
&\le \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_T\right\|_2\middle| s_{T-1}\right]\right]\right) && \text{(Jensen's inequality)} \\
&\le \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_T\right\|_2\right]\right) && \text{(law of total expectation)} \\
&\le \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \frac{\rho}{1-\gamma}\right) \\
&\le \frac{1}{T}\left(\frac{2\rho\gamma}{1-\gamma}\right)
\end{aligned}
\tag{17}
$$

Then, knowing that for all T, we have $0 \leq \eta_T$, the squeeze theorem ensures that $\lim_{T\to\infty} \eta_T = 0$.

Now we will prove that the left-hand side of Equation 13 converges, let $X_T := \left\| \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}_t - \mathbf{z} \right\|_2$. For all T,

$$\begin{aligned} |X_T| &\leq \frac{1}{T} \sum_{t=0}^{T-1} \|\boldsymbol{\phi}_t\|_2 + \|\mathbf{z}\|_2 && \text{(triangular inequality)} \\ &\leq \rho + \|\mathbf{z}\|_2 \end{aligned}$$

Moreover, $\mathbf{z}$ is a fixed variable, which means that $|X_T|$ is bounded. In addition, $\mathbb{E}_{\pi_\mathbf{z}} \left[\rho + \|\mathbf{z}\|_2\right] < \infty$, and the sequence $(X_T)_{T\geq 1}$ converges almost surely (since $\left(\frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}_t\right)_{T\geq 1}$ converges almost surely by hypothesis). The dominated convergence theorem then ensures that $(\mathbb{E}_{\pi_\mathbf{z}} [X_T])_{T\geq 1}$ converges and:

$$\lim_{T\to\infty} \mathbb{E}_{\pi_\mathbf{z}} [X_T] = \mathbb{E}_{\pi_\mathbf{z}} \left[\lim_{T\to\infty} X_T\right] \tag{18}$$

Finally, by taking the Equation 13 to the limit as $T \to \infty$, we get:

$$\lim_{T\to\infty} \mathbb{E}_{\pi_\mathbf{z}} \left[\underbrace{\left\| \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}_t - \mathbf{z} \right\|_2}_{X_T} \right] \leq \lim_{T\to\infty} \left(\beta + \eta_T + \frac{T-1}{T} \epsilon \right)$$

$$\mathbb{E}_{\pi_\mathbf{z}} \left[\left\| \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}_t - \mathbf{z} \right\|_2 \right] \leq \beta + \underbrace{\lim_{T\to\infty} \eta_T}_{=0} + \underbrace{\lim_{T\to\infty} \frac{T-1}{T}}_{=1} \epsilon$$

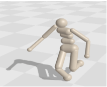
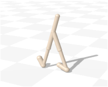
$$\boxed{\mathbb{E}_{\pi_\mathbf{z}} \left[\left\| \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}_t - \mathbf{z} \right\|_2 \right] \leq \beta + \epsilon}$$

□

C Tasks Details

C.1 Tasks

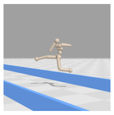
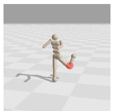
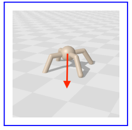
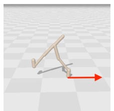
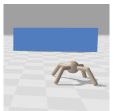
Table C.1: Tasks

	Feet Contact			Jump	Velocity	Angle
	Humanoid	Ant	Walker	Humanoid	Ant	Humanoid
						
State dim	244	27	17	244	27	244
Action dim	17	8	6	17	8	17
Features dim	2	4	2	1	1	1
Features space	$\{0,1\}^2$	$\{0,1\}^4$	$\{0,1\}^2$	$[0, 0.25]$	$[-5., 5]^2$	$]-\pi,\pi]$
Skill space	$[0,1]^2$	$[0,1]^4$	$[0,1]^2$	$[0, 0.25]$	$[-5., 5]^2$	$]-\pi,\pi]$
Episode length	1000	1000	1000	1000	1000	1000
Threshold δ	0.01	0.1	0.01	0.0025	0.1	0.06
Distance eval d_{eval}	0.05	0.5	0.05	0.0125	0.5	0.30

C.2 ADAPTATION TASKS

For all adaptation tasks, the reward stays the same but the dynamics of the MDP is changed. The goal is to leverage the diversity of skills to adapt to unforeseen situations.

Table C.2: Adaptation tasks

	HURDLES HUMANOID	MOTOR FAILURE HUMANOID	GRAVITY HUMANOID	FRICTION WALKER	OBSTACLE ANT
					
FEATURES	Jump	Feet Contact	Feet Contact	Feet Contact	Velocity
ADAPTATION	Few-shot	Few-shot	Few-shot	Few-shot	Hierarchy

C.2.1 FEW SHOT ADAPTATION

For all few-shot adaptation tasks, we evaluate all skills for each replication of each method and select the best one to solve the adaptation task. In Figure 5, the lines represent the IQM for the 10 replications and the shaded areas correspond to the 95% CI.

On **Humanoid - Hurdles**, we use the jump skills to jump over hurdles varying in height from 0 to 50 cm.

On **Humanoid - Motor Failure**, we use the feet contact skills to find the best way to continue walking forward despite the damage. In this experiment, we scale the action corresponding to the torque of the left knee (actuator 10) by the damage strength (x-axis of Figure 5) ranging from 0.0 (no damage) to 1.0 (maximal damage).

On **Ant - Gravity**, we use the feet contact skills to find the best way to continue walking forward despite the change in gravity. In this experiment, we scale the gravity by a coefficient ranging from 0.5 (low gravity) to 3.0 (high gravity).

On **Walker - Friction**, we use the feet contact skills to find the best way to continue walking forward despite the change in friction. In this experiment, we scale the friction by a coefficient ranging from 0.0 (low friction) to 5.0 (high friction).

C.2.2 HIERARCHICAL LEARNING

For the hierarchical learning task, we use each replication of each method to learn a meta-controller that selects the skills of the policy in order to adapt to the new task.

On **Ant - Obstacle**, the meta-controller is trained with SAC to select the velocity skills that enables to go around the obstacle and move forward as fast as possible in order to maximize performance.

D BASELINES DETAILS

Table D.3: Comparison of the main features for the different algorithms

Algorithm	Objective Function	Model-based	Lagrange Multiplier λ
SCOPA	$(1-\lambda)\sum\gamma^t r_t - \lambda\left\|(1-\gamma)\boldsymbol{\psi} - \mathbf{z}\right\|_2$	✓	Learned, function of $\mathbf{z}$
SMERL	$\sum\gamma^t\left(r_t - \lambda\mathbb{1}(R \geq R^* - \epsilon)\left\|\mu(\boldsymbol{\phi}_t) - \mathbf{z}_{\text{DIAYN}}\right\|_2^2\right)^*$	✗	Fixed
REVERSE SMERL	$\sum\gamma^t\left(\mathbb{1}(R < R^* - \epsilon)r_t - \lambda\left\|\mu(\boldsymbol{\phi}_t) - \mathbf{z}_{\text{DIAYN}}\right\|_2^2\right)^*$	✗	Fixed
DCG-ME **	$\sum\gamma^t r_t$	✗	—
UVFA	$(1-\lambda)\sum\gamma^t r_t - \lambda\sum\gamma^t\left\|\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	Fixed
SEP-$\mathbf{z}$	$(1-\lambda)\sum\gamma^t r_t - \lambda\sum\gamma^t\left\|\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	Learned, function of $\mathbf{z}$
FIXED-λ	$(1-\lambda)\sum\gamma^t r_t - \lambda\left\|(1-\gamma)\boldsymbol{\psi} - \mathbf{z}\right\|_2$	✓	Fixed

* see Section D.1 for a detailed explanation of the reward used in SMERLand REVERSE SMERL.

**DCG-ME learns diverse skills with a mechanism that is not visible in its objective function.

D.1 SMERL

To estimate the optimal return $R_{\mathcal{M}}(\pi^*_{\mathcal{M}})$ required by SMERL, we apply the same method as Kumar et al. (2020). We trained SAC on each environment and used SAC performance R_{SAC} as the the optimal return value for each environment. Similarly to Kumar et al. (2020), we choose $\lambda = 2.0$ by taking the best value when evaluated on HalfCheetah environment.

We use SMERLwith continuous DIAYNwith a Gaussian discriminator (Choi et al., 2021), so that the policy learns a continuous range of skills instead of a finite number of skills (Kumar et al., 2020) . We also use DIAYN+prior (Eysenbach et al., 2018; Chalumeau et al., 2022) with $q(\mathbf{z}_{\text{DIAYN}}|s)$ and feature priors to guide SMERLtowards relevant skills as explained in DIAYNoriginal paper.

With a Gaussian discriminator $q(\mathbf{z}_{\text{DIAYN}}|s) = \mathcal{N}(\mathbf{z}_{\text{DIAYN}}|\mu(s), \Sigma(s))$, the intrinsic reward is of the form $\tilde{r} = \log q(\mathbf{z}_{\text{DIAYN}}|s) - \log p(\mathbf{z}_{\text{DIAYN}}) \propto \|\mu(s) - \mathbf{z}_{\text{DIAYN}}\|_2^2$ up to an additive and a multiplicative constant, as demonstrated by Choi et al. (2021). Replacing the state s with the prior information $\boldsymbol{\phi}(s)$ in the discriminator gives $\tilde{r} \propto -\|\mu(\boldsymbol{\phi}(s)) - \mathbf{z}_{\text{DIAYN}}\|_2^2$. Consequently, we can see that the intrinsic reward from DIAYNcorresponds to executing a latent skill $\mathbf{z}_{\text{DIAYN}}$ (i.e. achieving a latent goal) in the unsupervised space defined by the discriminator $q(\mathbf{z}_{\text{DIAYN}}|\boldsymbol{\phi}(s))$. Indeed, the intrinsic reward is analogous to the reward used in GCRL of the form $r \propto -\|\boldsymbol{\phi}(s) - \mathbf{g}\|_2$ (Liu et al., 2022). Moreover, the bijection between the latent skills (i.e. latent goals) and the features (i.e. goals) is given by $\mathbf{z}_{\text{DIAYN}} \sim q(\mathbf{z}_{\text{DIAYN}}|\boldsymbol{\phi}(s))$.

Finally, we explain how we can use the previous analysis of DIAYN's intrinsic reward to calculate the distance to skill metric. Given a skill $\mathbf{z}$, we can use the discriminator to get the Maximum Likelihood (ML) of the distribution $\mathbf{z}_{\text{DIAYN}} = \text{ML}(q(\mathbf{z}_{\text{DIAYN}}|\mathbf{z})) = \text{ML}(\mathcal{N}(\mu(\mathbf{z}), \Sigma(\mathbf{z}))) = \mu(\mathbf{z})$. Given $\mathbf{z}_{\text{DIAYN}}$, we can rollout the policy $\pi(.|., \mathbf{z}_{\text{DIAYN}})$ in the environment and get a trajectory. From the trajectory, we can observe the achieved skill $\mathbf{z}_{\text{achieved}}$, and then calculate the distance to skill as $\|\mathbf{z} - \mathbf{z}_{\text{achieved}}\|_2$.

D.2 REVERSE SMERL

For REVERSE SMERL, we use exactly the same hyperparameters as SMERL.

The same discussion holds for SMERLand REVERSE SMERLholds, see Appendix D.1 for more details.

D.3 SEP-$\mathbf{z}$

We can show that the constraint in SCOPA's objective function is easier to satisfy than SEP-$\mathbf{z}$'s constraint.

For all skills $\mathbf{z} \in \mathcal{Z}$ and for all sequences of states $(s_t)_{t\geq 0}$, we have:

$$\begin{aligned}
\left\| (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}_t - \mathbf{z} \right\|_2 &= \left\| (1-\gamma) \left(\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}_t - \sum_{t=0}^{\infty} \gamma^t \mathbf{z} \right) \right\|_2 \\
&= \left\| (1-\gamma) \sum_{t=0}^{\infty} \gamma^t (\boldsymbol{\phi}_t - \mathbf{z}) \right\|_2 \\
&\leq (1-\gamma) \sum{}_{t=0}{}^{\infty} \gamma^t \left\| \boldsymbol{\phi}_t - \mathbf{z} \right\|_2
\end{aligned}$$

Thus, we have the following inequality:

$$\left\| (1-\gamma) \boldsymbol{\psi}(s, \mathbf{z}) - \mathbf{z} \right\|_2 \leq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \left\| \boldsymbol{\phi}_t - \mathbf{z} \right\|_2$$

At each timestep t, SEP-$\mathbf{z}$ tries to satisfy $\boldsymbol{\phi}_t = \mathbf{z}$, whereas SCOPA approximately tries to satisfy $\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \boldsymbol{\phi}_t = \mathbf{z}$, which is less restrictive.

E HYPERPARAMETERS

We provide here all the hyperparameters used for the baselines and SCOPA. SCOPA uses the same hyperparameters as the ones used by DREAMERV3 (Hafner et al., 2023), hence we provide here only the parameters mentioned in this work.

Our implementation is based on the implementation of DREAMERV3. Our successor features network is also implemented as a distributional critic. In our implementation, the Lagrangian network $\lambda_{\mathcal{L}}$ is only conditioned on the skill $\mathbf{z}$, as we noticed no difference in performance.

The hyperparameters for SMERL, REVERSE SMERLand DCG-MEare exactly the same as in their original papers, where they were fine-tuned on similar locomotion tasks. We also tried using hidden layers of size 512 for those baselines, in order to give them an architecture that is closer to SCOPA. We noticed this deteriorates the results of Reverse SMERL and DCG-ME, and this does not significantly increase (statistically speaking) the performance of SMERL.

Each algorithm is run for 10^7 environment steps, to ensure that they have enough iterations to converge.

Table E.4: SCOPA hyperparameters

Parameter	Value
Actor network	[512, 512, $\|\mathcal{A}\|$]
Critic network	[512, 512, 1]
Successor Feature network	[512, 512, $\|\mathcal{Z}\|$]
Lagrangian network	[8, 1]
Imagination batch size N	1024
Real environment exploration batch size	16
Total timesteps	1×10^7
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.997
Imagination horizon H	15
Target smoothing coefficient τ	0.005
Sampling period T	100
Lambda Return λ	0.95

Table E.5: SMERL hyperparameters

Parameter	Value
Actor network	[256, 256, $\|\mathcal{A}\|$]
Critic network	[256, 256, 1]
Batch size	256
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.99
Target smoothing coefficient τ	0.005
Skill distribution	Normal distribution
Diversity reward scale	10.0
SMERL target	R_{SAC}
SMERL margin	$0.1R_{\text{SAC}}$

Table E.6: REVERSE SMERL hyperparameters

Parameter	Value
Actor network	[256, 256, $\lvert\mathcal{A}\rvert$]
Critic network	[256, 256, 1]
Batch size	256
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.99
Target smoothing coefficient τ	0.005
Skill distribution	Normal distribution
Diversity reward scale	10.0
SMERL target	R_{SAC}
SMERL margin	$0.1 R_{\text{SAC}}$

Table E.7: DCG-ME hyperparameters

Parameter	Value
Number of centroids	1024
Evaluation batch size b	256
Policy networks	[128, 128, $\lvert\mathcal{A}\rvert$]
Number of GA variations g	128
GA variation param. 1 σ_1	0.005
GA variation param. 2 σ_2	0.05
Actor network	[256, 256, $\lvert\mathcal{A}\rvert$]
Critic network	[256, 256, 1]
TD3 batch size N	100
Critic training steps n	3000
PG training steps m	150
Optimizer	Adam
Policy learning rate	5×10^{-3}
Actor learning rate	3×10^{-4}
Critic learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.99
Actor delay Δ	2
Target update rate	0.005
Smoothing noise var. σ	0.2
Smoothing noise clip	0.5
lengthscale l	0.008
Descriptor sigma σ_d	0.0004

Table E.8: UVFA hyperparameters

Parameter	Value
Actor network	[512, 512, $\|\mathcal{A}\|$]
Critic network	[512, 512, 1]
Lagrangian network	[8, 1]
Imagination batch size N	1024
Real environment exploration batch size	16
Total timesteps	1×10^7
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.997
Imagination horizon H	15
Target smoothing coefficient τ	0.005
Sampling period T	100
Lambda Return λ	0.95
Lagrange multiplier $\lambda_{\mathcal{L}}$	0.66

Table E.9: SEP-$\mathbf{z}$ hyperparameters

Parameter	Value
Actor network	[512, 512, $\|\mathcal{A}\|$]
Critic network	[512, 512, 1]
Lagrangian network	[8, 1]
Imagination batch size N	1024
Real environment exploration batch size	16
Total timesteps	1×10^7
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.997
Imagination horizon H	15
Target smoothing coefficient τ	0.005
Sampling period T	100
Lambda Return λ	0.95

Table E.10: FIXED-λ hyperparameters

Parameter	Value
Actor network	[512, 512, $\lvert\mathcal{A}\rvert$]
Critic network	[512, 512, 1]
Successor Feature network	[512, 512, $\lvert\mathcal{Z}\rvert$]
Lagrangian network	[8, 1]
Imagination batch size N	1024
Real environment exploration batch size	16
Total timesteps	1×10^7
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	10^6
Discount factor γ	0.997
Imagination horizon H	15
Target smoothing coefficient τ	0.005
Sampling period T	100
Lambda Return λ	0.95
Lambda $\lambda_{\mathcal{L}}$	0.5

F EVALUATION METRICS DETAILS

In this section, we illustrate how to compute and read the distance and performance profiles in Figure 2. In the QD community, there is a consensus that a convenient evaluation metric for assessing the performance of algorithms is the ”distance/performance profile” (Flageat et al., 2022; Grillotti et al., 2023; Grillotti & Cully, 2022a; Batra et al., 2023), and is also being used in skill learning for robotics (Margolis et al., 2022). The profiles are favored because it effectively captures the essence of what QD algorithms aim to achieve: not just finding a single optimal solution but exploring a diverse set of high-quality solutions.

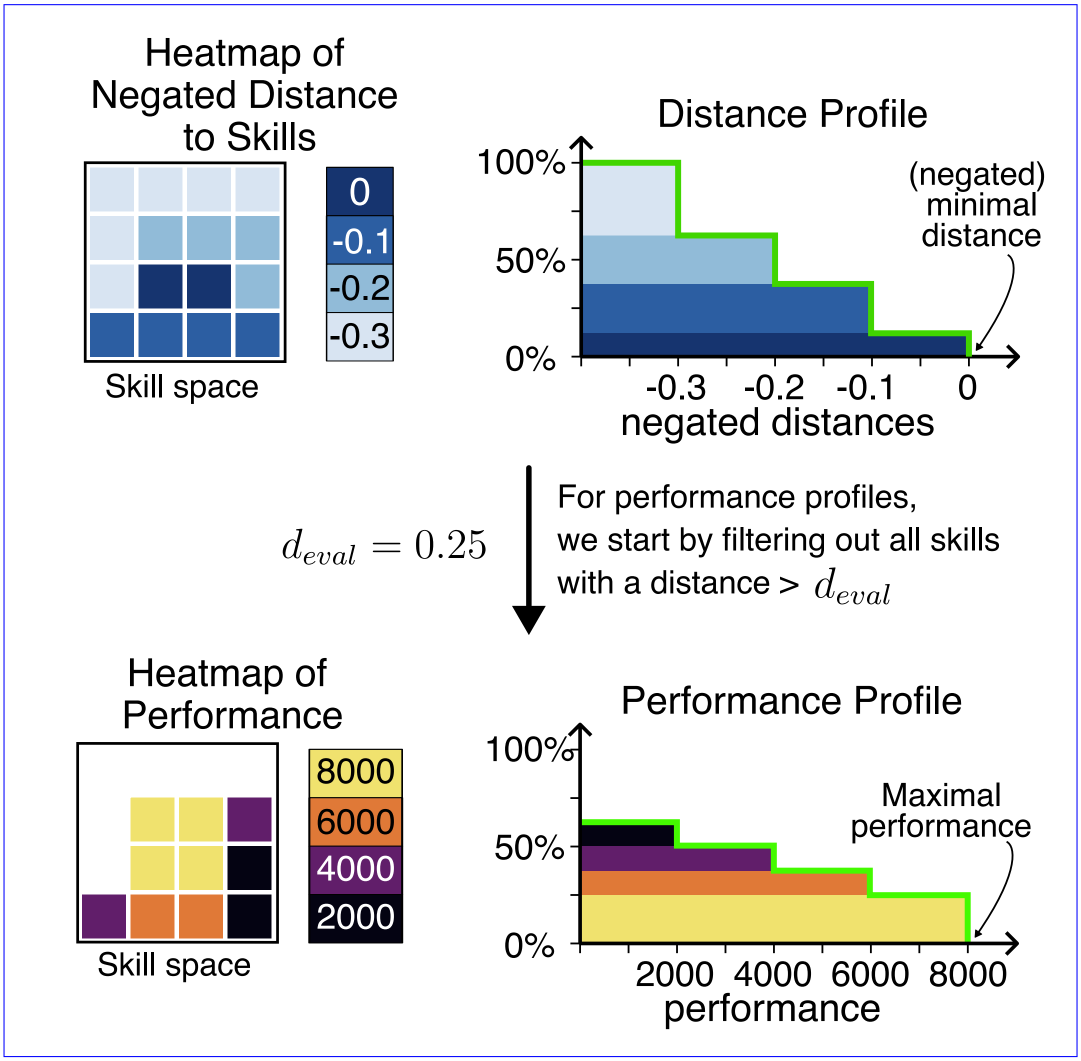


Figure F.22: **Right** *Distance* and *performance* profiles obtained from **Left** Heatmaps of negated distance to skills and performance. A skill $\mathbf{z}$ is considered successfully executed if $d(\mathbf{z}) < d_{\text{eval}}$, otherwise it is considered failed. All the skills that are not successfully executed by the policy are filtered out before computing the performance heatmap and profile. For a given distance d, the distance profile shows the proportion of skills with distance lower than d. For a given performance p, the performance profile shows the proportion of skills with performance higher than p. This figure also illustrates how to read the performance of the highest-performing skill (maximal performance of the agent) and the distance to skill of the best executed skill (minimal distance to skill of the agent).