Quality-Diversity Actor-Critic: Learning High-Performing and Diverse Behaviors via Value and Successor Features Critics

Luca Grillotti [*1] **Maxence Faldor** [*1] **Borja González León** [1 2] **Antoine Cully** [1]

Abstract

A key aspect of intelligence is the ability to demonstrate a broad spectrum of behaviors for adapting to unexpected situations. Over the past decade, advancements in deep reinforcement learning have led to groundbreaking achievements to solve complex continuous control tasks. However, most approaches return only one solution specialized for a specific problem. We introduce Quality-Diversity Actor-Critic (QDAC), an off-policy actor-critic deep reinforcement learning algorithm that leverages a value function critic and a successor features critic to learn high-performing and diverse behaviors. In this framework, the actor optimizes an objective that seamlessly unifies both critics using constrained optimization to (1) maximize return, while (2) executing diverse skills. Compared with other Quality-Diversity methods, QDAC achieves significantly higher performance and more diverse behaviors on six challenging continuous control locomotion tasks. We also demonstrate that we can harness the learned skills to adapt better than other baselines to five perturbed environments. Finally, qualitative analyses showcase a range of remarkable behaviors: adaptive-intelligent-robotics.github.io/QDAC.

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). These milestones have showcased the extraordinary potential of RL algorithms in solving specific problems.

[*]Equal contribution [1]Department of Computing, Imperial College London, London, United Kingdom [2]Iconic AI. Correspondence to: Luca Grillotti <luca.grillotti16@imperial.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

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), and reward design (Margolis & Agrawal, 2022). However, designing algorithms to learn expressive skills that are useful to solve downstream tasks remains a challenge. Reward design requires a lot of manual work and fine-tuning while being very brittle. GCRL and URL try to achieve 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.

Quality-Diversity (QD) optimization (Pugh et al., 2016) is a family of methods, originating from Evolutionary Algorithms, that generate a diverse population of high-performing solutions. QD algorithms have shown promising results in hard-exploration settings (Ecoffet et al., 2021), to recover from damage (Cully et al., 2015) or to reduce the reality gap (Chatzilygeroudis et al., 2018). In particular, QD algorithms have been scaled to challenging, continuous control tasks, by synergizing evolutionary methods with reinforcement learning (Faldor et al., 2023b; Pierrot et al., 2022). Other approaches like SMERL (Kumar et al., 2020) and DOMiNO (Zahavy et al., 2022) share the same objective of finding diverse and near-optimal policies and optimize a quality-diversity trade-off employing a pure reinforcement learning formulation. Most QD algorithms guide the diversity search towards relevant behaviors using a manually defined behavior descriptor function, that meaningfully characterizes solutions for the type of diversity desired (Cully & Demiris, 2018; Mouret & Clune, 2015). Two notable

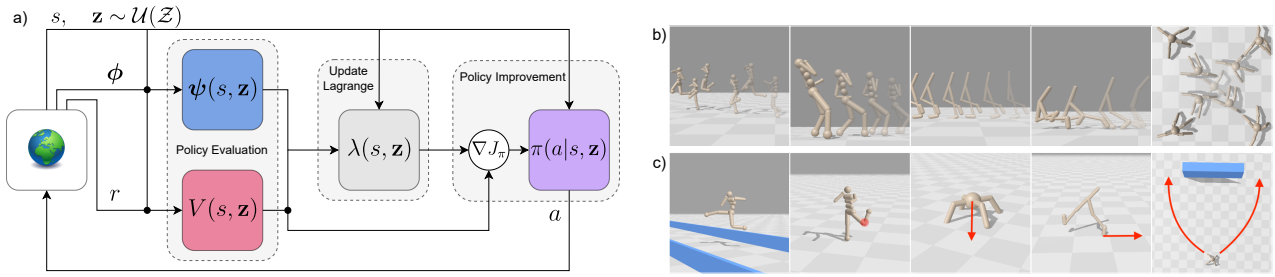


Figure 1. **a)** QDAC's architecture: the agent $\pi(a|s, \mathbf{z})$ learns high-performing and diverse behaviors with a dual critics optimization $V(s, \mathbf{z})$ and $\psi(s, \mathbf{z})$ which are balanced with a Lagrange multiplier $\lambda(s, \mathbf{z})$. **b)** Example of diverse behaviors on a set of challenging continuous control tasks. **c)** Few-shot adaptation tasks and hierarchical learning tasks using the diversity of skills learned by QDAC.

exceptions are AURORA (Grillotti & Cully, 2022b) and SMERL, that learn an unsupervised diversity measure, using an autoencoder architecture and DIAYN respectively.

In this work, we aim to solve the Quality-Diversity problem, i.e. to learn a large number of high-performing and diverse behaviors, where the diversity measure is given as part of the task, as a function of state-action occupancy (see Section 3 for a detailed problem statement). First, we introduce an approximate policy skill improvement update based on successor features, analogous to the classic policy improvement update based on value function (Section 4.1). Second, we show that the policy skill improvement update based on successor features enables the policy to efficiently learn to execute skills with a theoretical justification (see Proposition in Section 4.1). Third, we formalize the goal of Quality-Diversity into a problem that seamlessly unifies value function and successor features critics using constrained optimization to (1) maximize performance, while (2) executing desired skills (see Problem P3 in Section 4.1). Finally, we introduce Quality-Diversity Actor-Critic (QDAC), a practical algorithm that solves this problem by leveraging two independent critics — the value function criticizes the actions made by the actor to improve quality while the successor features criticizes the actions made by the actor to improve diversity (Section 4.2).

We evaluate our approach on six continuous control tasks and show that QDAC achieves 15% more diverse behaviors and 38% higher performance than other baselines (Section 5.4.1). Finally, we show that the skills can be used to adapt to downstream tasks in a few shots or via hierarchical learning (Section 5.4.2).

2. Background

We consider the reinforcement learning framework (Sutton & Barto, 2018) where an agent interacts with a *Markov Decision Process* (MDP) 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 $s_{t+1} \in \mathcal{S}$, sampled from the dynamics $p(s_{t+1} \mid s_t, a_t)$. Additionally, the agent receives a reward $r_t = r(s_t, a_t)$ and observes features $\boldsymbol{\phi}_t = \boldsymbol{\phi}(s_t, a_t) \in \Phi \subset \mathbb{R}^d$. In this work, we assume the features $\boldsymbol{\phi}_t$ are provided by the environment as part of the task, akin to the rewards, and are not learned by the agent. We denote $\rho^\pi(s) = \lim_{t\to\infty} P(s_t = s|s_0, \pi)$ the stationary distribution of states under a policy π, which we assume exists and is independent of s_0 (Sutton et al., 1999).

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 methods 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 this work, the value function is approximated via a neural network parameterized by θ_V. Similarly to Mnih et al. (2013), those parameters are optimized by minimizing the Bellman error:

$$J_V(\theta_V) = \mathbb{E}_\pi \left[\left(V_{\theta_V}(s_t) - r_t - \gamma V_{\theta'_V}(s_{t+1})\right)^2\right] \quad (1)$$

where θ'_V are the parameters of a target network, which are updated at a lower pace to improve training stability (Mnih et al., 2015).

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 features 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 $\boldsymbol{\psi}^\pi(s) = \mathbb{E}_\pi \left[\boldsymbol{\phi}_t + \gamma \boldsymbol{\psi}^\pi(s_{t+1}) | s_t = s\right]$, and can be learned with any RL methods (Dayan, 1993). In this work specifically, the successor features are approximated via a neural network parameterized by $\theta_{\boldsymbol{\psi}}$. Analogously to the value function network, $\theta_{\boldsymbol{\psi}}$ is optimized by minimizing

the Bellman error:

$$J_{\boldsymbol{\psi}}(\theta_{\boldsymbol{\psi}}) = \mathbb{E}_{\pi}\left[\left\|\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(s_t) - \boldsymbol{\phi}_t - \gamma\boldsymbol{\psi}_{\theta'_{\boldsymbol{\psi}}}(s_{t+1})\right\|_2^2\right] \quad (2)$$

where $\theta'_{\boldsymbol{\psi}}$ are the parameters of the corresponding target network.

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. For conciseness, we omit π from the notations ρ^{π}, V^{π}, $\boldsymbol{\psi}^{\pi}$ and we note $\pi_{\mathbf{z}}(a|s) := \pi(a|s, \mathbf{z})$.

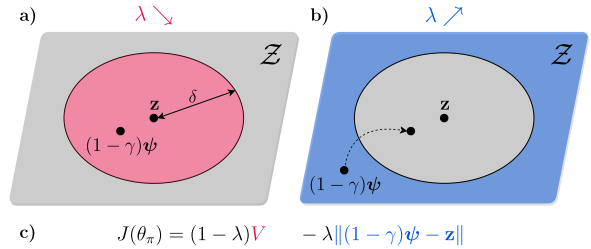


Figure 2. The Lagrange multiplier is optimized to balance the quality-diversity trade-off, see Eq. 5. **a)** If the expected features $(1-\gamma)\boldsymbol{\psi}(s, \mathbf{z})$ is in the neighborhood of $\mathbf{z}$, then $\lambda(s, \mathbf{z})$ decreases to focus on maximizing the return. **b)** Otherwise, $\lambda(s, \mathbf{z})$ increases to focus on executing $\mathbf{z}$. **c)** After the Lagrange multiplier is updated, the policy is optimized according to the objective.

3. Problem Statement

In this work, we aim to solve the Quality-Diversity problem, i.e. to learn a policy that can execute a large number of different and high-performing behaviors. In this section, we formalize this intuitive goal into a concrete optimization problem. The behavior of a policy π is characterized by the expected features under the policy's stationary distribution, $\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\phi}_t = \mathbb{E}_{\pi}\left[\boldsymbol{\phi}(s, a)\right]$ and we define the space of all possible behaviors to be the skill space $\mathcal{Z}$.

Given this definition, we intend to learn a skill-conditioned policy $\pi(a|s, \mathbf{z})$ that (1) maximizes the expected return, and (2) is subject to the expected features converge to the desired skill $\mathbf{z}$. In other words, we solve the following constrained optimization problem, for all $\mathbf{z} \in \mathcal{Z}$,

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

The feature function $\boldsymbol{\phi}$ can be any arbitrary function of the state of the MDP and of the action taken by the agent. Computing diversity based on the raw observations in high-dimensional environments (e.g., pixel observations) may not lead to interesting behaviors. Thus, the features can be thought of as relevant characteristics or events for the type of diversity desired, such as joint positions, contact with the ground, speed and so on. To illustrate the generality of this problem formulation, we now give two examples. Consider a robot whose objective is to minimize energy consumption, and where the features characterize the velocity of the robot $\boldsymbol{\phi}_t = \mathbf{v}_t = \begin{bmatrix} v_x(t) & v_y(t) \end{bmatrix}^{\intercal}$ and the skill space $\mathcal{Z} = \mathbb{R}^2$. For each desired velocity $\mathbf{z} \in \mathcal{Z}$, $\pi(a|s, \mathbf{z})$ 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 with a legged robot, 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, $\boldsymbol{\phi}_t = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\intercal}$ 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 $\mathbf{z}$ (i.e. average features) will be the proportion of time during which the i-th foot of the robot is in contact with the ground, denoted as feet contact rate. In that case, the skill space characterizes the myriad of ways the robot can walk and specifically, how often each leg is being used. Notice that to achieve a feet contact of $\mathbf{z} = \begin{bmatrix} 0.1 & 0.6 \end{bmatrix}^{\intercal}$, the robot needs to use 10% of the time the first foot and 60% of the time the second foot over a trajectory of *multiple* time steps.

4. Methods

In this section, we present Quality-Diversity Actor-Critic (QDAC), a quality-diversity reinforcement learning algorithm that discovers high-performing and diverse skills. First, we present a concrete optimization problem that optimizes for quality and diversity as defined in Section 3. Second, we define the notion of successor features policy iteration that we combine with value function policy iteration to derive a tractable objective for the actor, that solves problem P1 approximately. Third, we derive a practical algorithm that optimizes this objective.

4.1. Actor Objective

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. We solve the optimization problem, for all $\mathbf{z} \in \mathcal{Z}$,

$$\begin{aligned} &\text{maximize } \mathbb{E}_{\pi_{\mathbf{z}}}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right] \\ &\text{subject to } \left\|\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\phi}(s, a)\right] - \mathbf{z}\right\|_2 \leq \delta \end{aligned} \quad \text{(P2)}$$

Second, we derive an upper bound for the distance between desired skill and 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:*

$$\|\mathbb{E}_{\pi_{\mathbf{z}}}[\boldsymbol{\phi}(s,a)] - \mathbf{z}\|_2 \leq \mathbb{E}_{\pi_{\mathbf{z}}}[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2] \quad (3)$$

Third, we derive a new Problem P3 by replacing the intractable constraint from Problem P2 with the tractable upper bound in Equation 3. The constraint in Problem P3 is more restrictive than that of Problem P2. Indeed, the above proposition ensures that if the constraint in Problem P3 is satisfied, then the constraint in Problem P2 is necessarily satisfied as well. For all $\mathbf{z} \in \mathcal{Z}$,

$$\begin{aligned} &\text{maximize } \mathbb{E}_{\pi_{\mathbf{z}}}[V(s,\mathbf{z})] \\ &\text{subject to } \mathbb{E}_{\pi_{\mathbf{z}}}[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2] \leq \delta \end{aligned} \quad \text{(P3)}$$

Finally, we solve Problem P3 using the method of Lagrange multipliers as described by Abdolmaleki et al. (2018; 2023). For all states s, and all skills $\mathbf{z} \in \mathcal{Z}$, we maximize the Lagrangian function, subject to $0 \leq \lambda(s,\mathbf{z}) \leq 1$,

$$(1-\lambda(s,\mathbf{z}))\, V(s,\mathbf{z}) - \lambda(s,\mathbf{z})\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2 \quad (4)$$

The first term in red aims at maximizing the return, while the second term in blue aims at executing the desired skill. To optimize the actor to be high-performing while executing diverse skills, we use a generalized policy iteration method. The algorithm consists in (1) policy evaluation for both critics $V(s,\mathbf{z})$ and $\boldsymbol{\psi}(s,\mathbf{z})$, and (2) policy improvement via optimization of the Lagrangian function introduced in Equation 4. This formulation combines the classic policy improvement based on value function with a novel policy skill improvement based on successor features.

The Lagrange multiplier λ is optimized to balance the quality-diversity trade-off. If $\|(1-\gamma)\boldsymbol{\psi}(s_1,\mathbf{z}_1) - \mathbf{z}_1\|_2 \leq \delta$ is satisfied for $(s_1,\mathbf{z}_1)$, we expect $\lambda(s_1,\mathbf{z}_1)$ to decrease to encourage maximizing the return. On the contrary, if the constraint is not satisfied for $(s_2,\mathbf{z}_2)$, we expect $\lambda(s_2,\mathbf{z}_2)$ to increase to encourage satisfying the constraint.

4.2. Practical Algorithm

The objective in Equation 4 can be optimized with any reinforcement learning algorithm that implements generalized policy iteration. We give two variants of our method, one variant named QDAC, that is model-free and that builds on top of SAC, and one variant named QDAC-MB, that is model-based and that builds on top of DreamerV3. Additional details about QDAC-MB are provided in Appendix C.2. In this section, we detail the model-free variant.

QDAC's model-free pseudocode is provided in Algorithm 1. At each iteration, a skill $\mathbf{z}$ is uniformly sampled for an episode of length T, during which the agent interacts with the environment following skill $\mathbf{z}$ with $\pi(\cdot|s,\mathbf{z})$. At each time step t, the transition is stored in a replay buffer $\mathcal{D}$, augmented with the features $\boldsymbol{\phi}(s_t,a_t)$ and with the current desired skill $\mathbf{z}$.

Then, the Lagrange multiplier is updated to balance the quality-diversity trade-off. The parameters θ_λ are optimized so that $\lambda(s,\mathbf{z})$ increases when the actor is unable to execute the desired skill $\mathbf{z}$, to put more weight on executing the skill. Conversely, the parameters θ_λ are optimized so that $\lambda(s,\mathbf{z})$ decreases when the actor is able to execute the desired skill $\mathbf{z}$, to put more weight on maximizing the return. The update of the Lagrange multiplier and its role in the actor objective are depicted in Figure 2. In practice, we use a cross-entropy loss to optimize θ_λ:

$$\begin{aligned} J_\lambda(\theta_\lambda) = \mathbb{E}_{\substack{s\sim\rho \\ \mathbf{z}\sim\mathcal{U}(\mathcal{Z})}}[&-(1-y)\log(1-\lambda(s,\mathbf{z})) \\ &- y\log(\lambda(s,\mathbf{z}))] \\ \text{where } y = \begin{cases} 0 & \text{if } \|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2 \leq \delta \\ 1 & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

Finally, the critics V, $\boldsymbol{\psi}$ and the actor $\pi_{\mathbf{z}}$ are trained with a policy iteration step adapted from SAC and following Equation 4. The objective is optimized with stochastic gradient descent using a mini-batch of transitions sampled from the replay buffer. To improve sample efficiency, the transitions from the mini-batch are duplicated with new random skills sampled uniformly in the skill space. Additional information about QDAC's training procedure are provided in Appendix C.1.

5. Experiments

The goal of our experiments is twofold: (1) evaluate QDAC's ability to learn high-performing and diverse skills based on a wide range of features, (2) evaluate QDAC's ability to harness learned skills to solve downstream tasks.

5.1. Tasks

5.1.1. Learning Diverse High-Performing Skills

We evaluate our method on a range of challenging continuous control tasks using the Google Brax (Freeman et al., 2021) physics engine. We consider the three classic locomotion environments Walker, Ant and Humanoid that we combine with four different feature functions that we call *feet contact*, *velocity*, *jump* and *angle*. The first two features

Algorithm 1 QDAC

input Parameters θ_π, θ_V, $\theta_{\boldsymbol{\psi}}$, θ_λ ▷ Initial parameters for the actor, critics and Lagrange multiplier
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay buffer
repeat
  $\mathbf{z} \sim \mathcal{U}(\mathcal{Z})$ ▷ Sample skill uniformly from skill space
  for T steps **do**
    ▷ Environment steps
    $a_t \sim \pi(a_t|s_t, \mathbf{z})$ ▷ Sample action from policy
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t, \mathbf{z})$ ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), \boldsymbol{\phi}(s_t, a_t), s_{t+1}, \mathbf{z})\}$ ▷ Store transition in the replay buffer
    ▷ Training steps
    $\theta_\lambda \leftarrow \theta_\lambda - \alpha_\lambda \nabla J_\lambda(\theta_\lambda)$ ▷ Update Lagrange multiplier with Eq. 5
    $\theta_V \leftarrow \theta_V - \alpha_V \nabla J_V(\theta_V)$ ▷ Policy evaluation for value function with Eq. 1
    $\theta_{\boldsymbol{\psi}} \leftarrow \theta_{\boldsymbol{\psi}} - \alpha_{\boldsymbol{\psi}} \nabla J_{\boldsymbol{\psi}}(\theta_{\boldsymbol{\psi}})$ ▷ Policy evaluation for successor features with Eq. 2
    $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \nabla J_\pi(\theta_\pi)$ ▷ Policy improvement with Eq. 4
  end for
until convergence

are traditional benchmark tasks that have been extensively studied in the Quality-Diversity and GCRL literature (Cully et al., 2015; Faldor et al., 2023a; Nilsson & Cully, 2021; Zhu et al., 2021; Finn et al., 2017), while the two last ones are challenging tasks that we introduce in this work. In these locomotion tasks, the objective is to go forward as fast as possible while minimizing energy consumption.

Feet Contact features indicate for each foot of the agent, if the foot is in contact or not with the ground, exactly as defined in DCG-ME's original paper (Faldor et al., 2023a). For example, if the Ant only touches the ground with its second foot at time step t, then $\boldsymbol{\phi}(s_t, a_t) = \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^\intercal$. 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). The expected features correspond to the proportion of time each foot is in contact with the ground.

Velocity features are two-dimensional vectors indicating the velocity of the agent in the xy-plane, $\boldsymbol{\phi}(s_t, a_t) = \begin{bmatrix}v_x(t) & v_y(t)\end{bmatrix}^\intercal$. We evaluate on the velocity features to show that our method works on classic GCRL tasks. Moreover, the velocity features are interesting because satisfying a velocity that is negative on the x-axis is directly opposite to maximizing the forward velocity reward.

Jump features are one-dimensional vectors indicating 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 $\boldsymbol{\phi}(s_t, a_t) = \begin{bmatrix}0.035\end{bmatrix}$. 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.

Angle features are two-dimensional vectors indicating the angle α of the main body about the z-axis, $\boldsymbol{\phi}(s_t, a_t) = \begin{bmatrix}\cos(\alpha) & \sin(\alpha)\end{bmatrix}^\intercal$. The goal of this task is to go as fast as possible in the x-direction while facing any directions, forcing the agent to sidestep or moonwalk.

5.1.2. Harnessing Skills for Few-Shot Adaptation and Hierarchical Learning

We evaluate our method on few-shot adaptation scenarios with four types of perturbation and on one hierarchical learning task. For each task, the reward is the same but the MDP's dynamics is perturbed. Additional details are available in Appendix D.2.

In few-shot adaption tasks, no re-training is allowed and we evaluate the top-performing skills for each method while varying the perturbation to measure the robustness of the different algorithms, see Appendix D.2 for more details. *Humanoid - Hurdles* requires the agent to jump over hurdles of varying heights. *Humanoid - Motor Failure* requires the agent to adapt to different degrees of failure in the motor controlling its left knee. In *Ant - Gravity*, the agent needs to adapt to different gravity conditions. Finally, *Walker - Friction* requires the agent to adapt to varying levels of ground friction. Here, we evaluate the agent's ability to adjust its locomotion strategy to a new perturbed MDP.

In the hierarchical learning task, named *Ant - Wall*, the agent is faced with navigating around a wall. A meta-controller is trained with Soft Actor-Critic (SAC) to maximize forward movement. Here, we evaluate the ability to use the diversity of skills discovered by QDAC for hierarchical RL.

5.2. Baselines

We compare QDAC with two families of methods that both balance a quality-diversity trade-off. The first family consists in evolutionary algorithms that maintain a diverse population of high-performing individuals whereas the second family uses a pure reinforcement learning formulation. Additionally, we perform three ablation studies.

Quality-Diversity via Evolutionary Algorithms We compare our method with PPGA (Batra et al., 2023), DCG-ME (Faldor et al., 2023a;b) and QD-PG (Pierrot et al., 2022), three evolutionary algorithms that optimize a diverse population of high-performing individuals. PPGA is a state-of-the-art Quality-Diversity algorithm that mixes Proximal Policy Optimization (PPO) (Schulman et al., 2017) with CMA-MAEGA (Fontaine & Nikolaidis, 2023); it alternates between (1) estimating the performance-feature gradients with PPO and (2) maintaining a population of coefficients to linearly combine those performance-feature gradients, those coefficients are optimized to maximize archive improvement. DCG-ME is another state-of-the-art Quality-Diversity algorithm that evolves a population of both high-performing and diverse solutions, and simultaneously distills those solutions into a single skill-conditioned policy. QD-PG is a Quality-Diversity algorithm that uses quality and diversity policy gradients to optimize its population of policies.

Quality-Diversity via Reinforcement Learning We also compare our method with DOMiNO (Zahavy et al., 2022), SMERL (Kumar et al., 2020) and Reverse SMERL (Zahavy et al., 2022) that balance a quality-diversity trade-off using a pure reinforcement learning formulation. DOMiNO is a reinforcement learning algorithm designed to discover diverse behaviors while preserving near-optimality. Analogous to our method, it characterizes policies' behaviors using successor features. SMERL learns a latent-conditioned policy that maximizes the mutual information between states s and latent variables z, with a threshold to toggle the diversity reward in the objective $r + \alpha \mathbb{1}(R \geq R^* - \epsilon)\tilde{r}$. The diversity reward $\tilde{r}$ is measured from the likelihood of a discriminator $q(z|s)$ coming from DIAYN. In other words, SMERL maximizes a weighted combination of environment reward and diversity reward when the policy is near-optimal, and only the environment reward otherwise. Reverse SMERL maximizes a similar reward $\mathbb{1}(R < R^* - \epsilon)r + \alpha\tilde{r}$. In other words, Reverse SMERL maximizes a weighted combination of environment reward and diversity reward when the policy is not near-optimal, and only the diversity reward otherwise.

Ablations We perform three additional ablation studies, that we call No-SF, Fixed-λ and UVFA. For No-SF, we remove the successor features representation and use a naive distance to skill instead $\sum_t \gamma^t \|\boldsymbol{\phi}_t - \mathbf{z}\|_2$ to understand the contribution of the successor features critic to optimize diversity. For Fixed-λ, we remove the Lagrange multiplier and use a fixed trade-off instead to understand the contribution of constrained optimization. UVFA (Schaul et al., 2015) is an algorithm that corresponds to the combination of our two previous ablations, as such we consider it to be an ablation in this work. A summarized description of all baselines under study is provided in Table E.3.

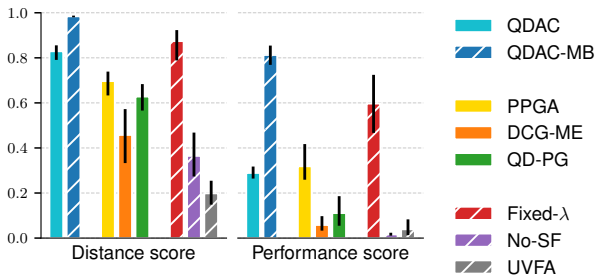


Figure 3. Distance and performance scores normalized and aggregated across all tasks. The values correspond to the IQM while the error bars represent IQM 95% CI.

5.3. Evaluation Metrics

We evaluate our method using two types of metrics from the Quality-Diversity literature that aim at evaluating the performance and the diversity of the discovered skills: (1) the distance to skill metrics, that evaluate the ability of an agent to execute desired skills, and (2) the performance metrics, that quantify the ability of an agent to maximize return while executing desired skills. Each experiment is replicated 10 times with random seeds. 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, as defined by Faldor et al. (2023a;b). First, we use $d(\mathbf{z})$ to compute *distance profiles* on Figure 4, 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, as defined by Flageat & Cully (2023); Grillotti et al. (2023). First, we use $R(\mathbf{z})$ to compute *performance profiles* on Figure 4, 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 ex-

Figure 4. (**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 D.33 illustrates how to read distance and performance profiles.

pected 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) Does QDAC solve the Quality-Diversity problem? (2) Can we harness the high-performing and diverse skills to adapt to perturbed MDP? In Section 5.4.1, we demonstrate that QDAC achieves significantly higher performance and more diverse behaviors on six challenging continuous control locomotion tasks (Fig. 3). In Section 5.4.2, we show that we can harness the learned skills to adapt better than other baselines to five perturbed MDP (Fig. 6). The code is available at: github.com/adaptive-intelligent-robotics/QDAC.

5.4.1. Learning Diverse High-Performing Skills

In this section, we evaluate QDAC with metrics coming from the Quality-Diversity literature, namely the distance to skill and performance metrics. However, DOMiNO and SMERL optimize for diversity, without the focus on executing specific skills. Consequently, the concept of 'distance to skill' does not apply and thus, the traditional QD metrics are not applicable. Nonetheless, we compare our approach with these baselines on adaptation tasks, for which they were initially designed, in Section 5.4.2.

QDAC and QDAC-MB outperform all baselines in executing a wide range of skills (Fig. 4), except on Humanoid Jump where DCG-ME and QD-PG achieve a better distance profile than our model-free variant. Yet, QDAC-MB outperforms those baselines, due to the representation capabilities of the world model. The jump features are challenging because of the $\min$ operator, and because the features are not explicitly available in the observations given to the agent.

Notably, QDAC and QDAC-MB are capable of achieving skills that are contrary to the task reward, as illustrated by the velocity features in Figure 4 and Figure 5, which is not the case for PPGA, DCG-ME and QD-PG. Finally, our approach outperforms DCG-ME that fails to explicitly minimize the expected distance to skill, a common issue among QD algorithms (Flageat & Cully, 2023).

QDAC-MB outperforms Fixed-λ on all tasks, showing the importance of using constrained optimization to solve the QD problem. Furthermore, QDAC-MB achieves better performance than No-SF on all tasks, showing the importance of the successor features critics to optimize diversity. Finally, QDAC significantly outperforms No-SF and UVFA on all tasks. On feet contact tasks, No-SF and UVFA can only execute skills in the corners of the skill space where $\mathbf{z} = \boldsymbol{\phi}_t$, as shown in Figure A.12. This is because these baselines employ a naive approach that consists in minimizing the distance between the features and the desired skill. Thus, they can only execute skills where the legs are always or never in contact with the ground. These comparisons supports the claim that QDAC is capable of accurately executing a diversity of skills and highlight the significance of the policy skill improvement term in blue in Equation 4.

QDAC and QDAC-MB outperform DCG-ME and QD-PG in maximizing return (Fig. 3), as the latter don't achieve many skills in the first place, and the performance score only evaluates the performance of skills successfully executed by the policy. While PPGA achieves a performance score comparable to QDAC, it does so by finding fewer robust policies, albeit with better performance (Fig. 4). Fixed-λ is the only

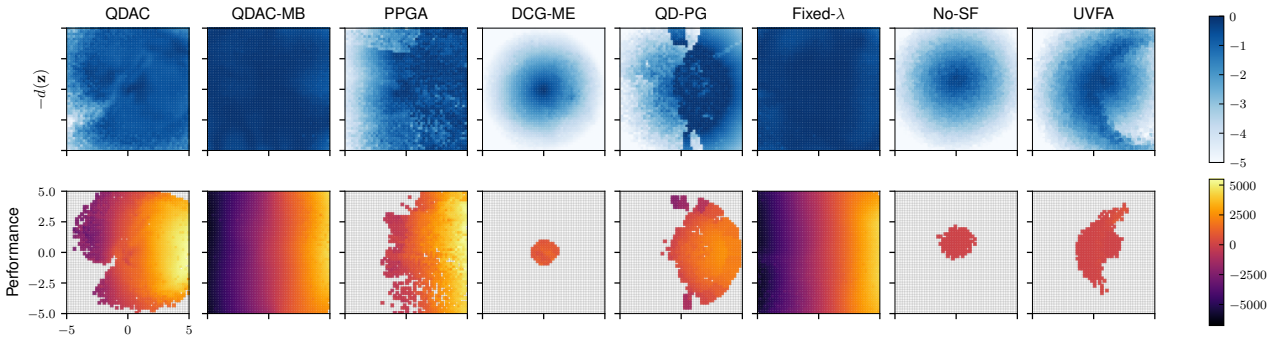


Figure 5. **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space $\mathcal{Z} = [-5 \text{ m/s}, 5 \text{ m/s}]^2$, of target velocities. This space is discretized into cells, with each cell representing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

baseline that gets performance scores and profiles comparable to QDAC-MB. However, Fixed-λ covers a smaller range of skills, as evidenced by the edges of the skill space on Figure A.12. Additionally, QDAC-MB outperforms Fixed-λ on the challenging jump task (Fig. 4), due to the necessity of a strong weight on the constraint. Ultimately, using an adaptive λ proves advantageous for our approach.

As summarized in Figure 3, QDAC and QDAC-MB achieve a better quality-diversity trade-off than other baselines, quantified by the distance score and the performance score.

5.4.2. Harnessing Skills for Few-Shot Adaptation and Hierarchical Learning

QDAC and QDAC-MB demonstrate competitive performance in few-shot adaptation and hierarchical RL tasks, see Figure 6. On the hurdles task, when considering hurdle heights strictly greater than 0, QDAC-MB significantly outperforms other baselines by consistently jumping over higher hurdles and showcases remarkable behaviors. On the motor failure task, although performing worse than PPGA, QDAC shows great robustness, especially in the high damage regime. QDAC-MB performs better than QDAC on low damage, but QDAC can adapt to 100% damage strength on the left knee, still achieving more than 5,000 in return. In other words, QDAC has found a way to continue walking despite not being able to control at all the left knee. QDAC does not seem able to adapt to gravity variations, but QDAC-MB shows competitive performance although performing slightly worse than PPGA and DOMiNO. On Walker - Friction, QDAC outperforms all baselines except PPGA and DCG-ME that achieve marginally better performance. Finally, QDAC's learned skills appear to be the best on the hierarchical RL task, as it achieves significantly higher performance than other baselines.

Our extensive experiments and analyses in Sections 5.4.1 and 5.4.2 firmly establish the efficacy of QDAC and QDAC-MB in addressing the dual challenges of learning high-performing and diverse skills. These methods not only surpass traditional Quality-Diversity algorithms in optimizing for specific pre-defined skills but also demonstrate remarkable adaptability and robustness in perturbed environments.

6. Related Work

QD optimization (Pugh et al., 2016; Cully & Demiris, 2018) is a family of algorithms that generate large collections of solutions, such as policies, that are both diverse and high-performing. Those methods originate from Novelty Search (Lehman & Stanley, 2011a;b) and Evolutionary Algorithms literature, where the diversity is defined across a population of solutions. Quality-Diversity algorithms have been shown to be competitive with skill discovery reinforcement learning methods (Chalumeau et al., 2022), and promising for adaptation to unforeseen situations (Cully et al., 2015). When considering large neural network policies, the sample efficiency of QD algorithms can be improved by using Evolution Strategies (Fontaine et al., 2020; Colas et al., 2020), RL-based methods (Pierrot et al., 2022; Nilsson & Cully, 2021; Faldor et al., 2023a; Tjanaka et al., 2022; Batra et al., 2023; Xue et al., 2024). The sample efficiency can be further improved by decomposing the policies into several parts and coevolving a sub-population for each part (Xue et al., 2024). However, most QD algorithms output a large number of policies, which can be difficult to deal with in downstream tasks. Similarly to QDAC, DCG-ME addresses that issue by optimizing a single skill-conditioned policy (Faldor et al., 2023a). Finally, approaches like SMERL (Kumar et al., 2020) or DOMiNO (Zahavy et al., 2022) also solve the QD objective employing a pure reinforcement learning formulation. Contrary to QDAC, the policies discovered by DOMiNO are not

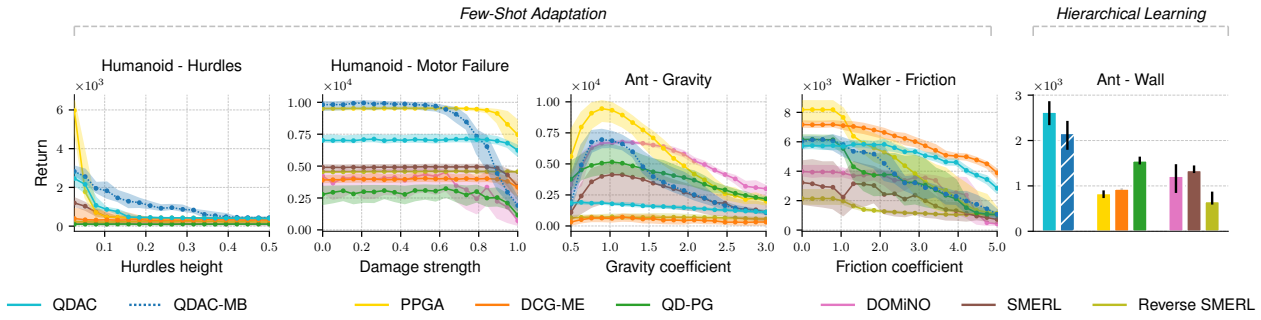


Figure 6. Performance for each algorithm in environments with different levels of perturbations after few-shot adaptation or after hierarchical learning. The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI.

trained to execute specific target skills.

Most Unsupervised Reinforcement Learning approaches discover diverse behaviors by maximizing an intrinsic reward defined in terms of the discriminability of the trajectories. 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), simplified to an MI-maximization between skills and states with the following lower bound: $I(\tau, \mathbf{z}) \geq \sum_{t=1}^{T} I(s_t, \mathbf{z})$. It has been shown that MI-based algorithms are equivalent to distance-to-skill minimization algorithms (Choi et al., 2021; Gu et al., 2021), and therefore present similarities with our work. However, most URL algorithms maximize an intrinsic reward while disregarding any other objective, making it difficult to discover useful and expressive behaviors.

While diversity can be achieved by maximizing a mutual information objective, it can also be explicitly defined as a distance between behavioral descriptors. Such descriptors can take the form of successor features (Zahavy et al., 2022; 2023) or of expected features obtained though entire episodes (Cully et al., 2015; Batra et al., 2023). In this work, we rely on this latter definition, as expressed in Problems P1 and P2. The features ϕ can be defined in different ways. First, they can be a subpart of the state of the agent such as the joint positions and velocities (Zahavy et al., 2022), torso velocity (Cheng et al., 2023), or feet contacts (Cully et al., 2015). In this case, the state of the agent may guide the search towards relevant notions of diversity; however, this requires expert knowledge about the task, and the choice of feature definition strongly influences the quality and diversity of the generated solutions (Tarapore et al., 2016). Second, to avoid hand-defining features, we could define ϕ as the full state (Kumar et al., 2020) or as an unsupervised low-dimensional encoding of it (Grillotti & Cully, 2022b; Mazzaglia et al., 2022; Liu & Abbeel, 2021). In this case, additional techniques can be used to ensure the learned behaviors are relevant, such as adding an extrinsic reward term (Chalumeau et al., 2022), promoting diversity in the neighborhood of relevant solutions (Grillotti & Cully, 2022a), or adding constraints for near-optimality (Zahavy et al., 2022; Kumar et al., 2020). Instead, QDAC constrains the agent's behavior to follow a given hand-defined skill $\mathbf{z}$, and maximizes the performance for all skills $\mathbf{z} \in \mathcal{Z}$.

7. Conclusion

In this work, we present QDAC, an actor-critic deep reinforcement learning algorithm, that leverages a value function critic and a successor features critic to learn high-performing and diverse behaviors. In this framework, the actor optimizes an objective that seamlessly unifies both critics using constrained optimization to (1) maximize return, while (2) executing diverse skills.

Empirical evaluations suggest that QDAC outperforms previous Quality-Diversity methods on six continuous control locomotion tasks. Quantitative results demonstrate that QDAC is competitive in adaptation tasks, while qualitative analyses reveal a range of diverse and remarkable behaviors.

In the future, we hope to apply QDAC to other tasks with different properties than the tasks from this paper. For example, it would be interesting to apply QDAC to non-ergodic environments such as Atari games.

Furthermore, like the vast majority of Quality-Diversity algorithms, QDAC uses a manually defined diversity measure to guide the diversity search towards relevant behaviors. An exciting direction for future work would be to learn the feature function in an unsupervised manner to discover task-agnostic skills.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. A. 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.

Abdolmaleki, A., Huang, S. H., Vezzani, G., Shahriari, B., Springenberg, J. T., Mishra, S., TB, D., Byravan, A., Bousmalis, K., Gyorgy, A., Szepesvari, C., Hadsell, R., Heess, N., and Riedmiller, M. 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].

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. 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.

Allard, M., Smith, S. C., Chatzilygeroudis, K., and Cully, A. 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.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H., and Silver, D. 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.

Batra, S., Tjanaka, B., Fontaine, M. C., Petrenko, A., Nikolaidis, S., and Sukhatme, G. S. Proximal policy gradient arborescence for quality diversity reinforcement learning. *CoRR*, abs/2305.13795, 2023. doi: 10.48550/ARXIV.2305.13795. URL https://doi.org/10.48550/arXiv.2305.13795.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D. J., Hasselt, H. v., Munos, R., Silver, D., and Schaul, T. Universal Successor Features Approximators. September 2018. URL https://openreview.net/forum?id=S1VWjiRcKX.

Chalumeau, F., Boige, R., Lim, B., Macé, V., Allard, M., Flajolet, A., Cully, A., and Pierrot, T. Neuroevolution is a Competitive Alternative to Reinforcement Learning for Skill Discovery. September 2022. URL https://openreview.net/forum?id=6BHlZgyPOZY.

Chatzilygeroudis, K., Vassiliades, V., and Mouret, J.-B. Reset-free trial-and-error learning for robot damage recovery. *Robotics and Autonomous Systems*, 100:236–250, 2018. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2017.11.010. URL https://www.sciencedirect.com/science/article/pii/S0921889017302440.

Cheng, J., Vlastelica, M., Kolev, P., Li, C., and Martius, G. Learning diverse skills for local navigation under multi-constraint optimality. In *Intrinsically-Motivated and Open-Ended Learning Workshop @NeurIPS2023*, 2023. URL https://openreview.net/forum?id=GGCb8ZA9Za.

Choi, J., Sharma, A., Lee, H., Levine, S., and Gu, S. S. 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.

Colas, C., Madhavan, V., Huizinga, J., and Clune, J. Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 67–75, 2020.

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

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. 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].

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

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. First return, then explore. *Nature*, 590(7847):580–586, February 2021. ISSN 1476-4687. doi: 10.1038/s41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is All You Need: Learning Skills without a Reward Function. September 2018. URL https://openreview.net/forum?id=SJx63jRqFm.

Faldor, M., Chalumeau, F., Flageat, M., and Cully, A. 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 2023a. Association for Computing Machinery. ISBN 9798400701191. doi: 10.1145/3583131.3590503. URL https://dl.acm.org/doi/10.1145/3583131.3590503.

Faldor, M., Chalumeau, F., Flageat, M., and Cully, A. Synergizing quality-diversity with descriptor-conditioned reinforcement learning, 2023b.

Finn, C., Abbeel, P., and Levine, S. 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.

Flageat, M. and Cully, A. 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.

Flageat, M., Lim, B., Grillotti, L., Allard, M., Smith, S. C., and Cully, A. Benchmarking Quality-Diversity Algorithms on Neuroevolution for Reinforcement Learning, November 2022. URL http://arxiv.org/abs/2211.02193. arXiv:2211.02193 [cs].

Fontaine, M. and Nikolaidis, S. Covariance matrix adaptation map-annealing. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 456–465, 2023.

Fontaine, M. C., Togelius, J., Nikolaidis, S., and Hoover, A. K. 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.

Freeman, C. D., Frey, E., Raichuk, A., Girgin, S., Mordatch, I., and Bachem, O. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation. June 2021. URL https://openreview.net/forum?id=VdvDlnnjzIN.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Gehring, J., Synnaeve, G., Krause, A., and Usunier, N. Hierarchical Skills for Efficient Exploration. November 2021. URL https://openreview.net/forum?id=NbaEmFm2mUW.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational Intrinsic Control, November 2016. URL http://arxiv.org/abs/1611.07507. arXiv:1611.07507 [cs].

Grillotti, L. and Cully, A. 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.

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

Grillotti, L., Flageat, M., Lim, B., and Cully, A. 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.

Gu, S. S., Diaz, M., Freeman, D. C., Furuta, H., Ghasemipour, S. K. S., Raichuk, A., David, B., Frey, E., Coumans, E., and Bachem, O. 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].

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

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft Actor-Critic Algorithms and Applications, January 2019. URL http://arxiv.org/abs/1812.05905. arXiv:1812.05905 [cs, stat].

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to Control: Learning Behaviors by Latent Imagination. September 2019a. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. 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.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., and Silver, D. Emergence of Locomotion Behaviours in Rich Environments. July 2017.

Kumar, S., Kumar, A., Levine, S., and Finn, C. 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.

Lehman, J. and Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Lehman, J. and Stanley, K. O. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 211–218, 2011b.

Lehman, J., Clune, J., Misevic, D., Adami, C., Altenberg, L., Beaulieu, J., Bentley, P. J., Bernard, S., Beslon, G., Bryson, D. M., Cheney, N., Chrabaszcz, P., Cully, A., Doncieux, S., Dyer, F. C., Ellefsen, K. O., Feldt, R., Fischer, S., Forrest, S., Fŕenoy, A., Gagńe, C., Le Goff, L., Grabowski, L. M., Hodjat, B., Hutter, F., Keller, L., Knibbe, C., Krcah, P., Lenski, R. E., Lipson, H., MacCurdy, R., Maestre, C., Miikkulainen, R., Mitri, S., Moriarty, D. E., Mouret, J.-B., Nguyen, A., Ofria, C., Parizeau, M., Parsons, D., Pennock, R. T., Punch, W. F., Ray, T. S., Schoenauer, M., Schulte, E., Sims, K., Stanley, K. O., Taddei, F., Tarapore, D., Thibault, S., Watson, R., Weimer, W., and Yosinski, J. 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.

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

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

Liu, M., Zhu, M., and Zhang, W. 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.

Mann, H. B. and Whitney, D. R. 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.

Margolis, G. B. and Agrawal, P. 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].

Margolis, G. B., Yang, G., Paigwar, K., Chen, T., and Agrawal, P. Rapid Locomotion via Reinforcement Learning, May 2022. URL http://arxiv.org/abs/2205.02824. arXiv:2205.02824 [cs].

Mazzaglia, P., Verbelen, T., Dhoedt, B., Lacoste, A., and Rajeswar, S. Choreographer: Learning and Adapting Skills in Imagination. October 2022. URL https://openreview.net/forum?id=BxYsP-7ggf.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with Deep Reinforcement Learning, December 2013. URL http://arxiv.org/abs/1312.5602. arXiv:1312.5602 [cs].

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Mouret, J.-B. and Clune, J. Illuminating search spaces by mapping elites, April 2015. URL http://arxiv.org/abs/1504.04909. arXiv:1504.04909 [cs, q-bio].

Nilsson, O. and Cully, A. 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.

Pierrot, T., Macé, V., Chalumeau, F., Flajolet, A., Cideron, G., Beguir, K., Cully, A., Sigaud, O., and Perrin-Gilbert, N. 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.

Pugh, J. K., Soros, L. B., and Stanley, K. O. 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.

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

Schaul, T., Horgan, D., Gregor, K., and Silver, D. 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.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-Aware Unsupervised Discovery of Skills. September 2019. URL https://openreview.net/forum?id=HJgLZR4KvH.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. 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.

Sutton, R. S. and Barto, A. G. *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.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., and Müller, K. (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.

Tarapore, D., Clune, J., Cully, A., and Mouret, J. How do different encodings influence the performance of the map-elites algorithm? In Friedrich, T., Neumann, F., and Sutton, A. M. (eds.), *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pp. 173–180. ACM, 2016. doi: 10.1145/2908812.2908875. URL https://doi.org/10.1145/2908812.2908875.

Tjanaka, B., Fontaine, M. C., Togelius, J., and Nikolaidis, S. 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.

Xue, K., Wang, R.-J., Li, P., Li, D., HAO, J., and Qian, C. Sample-efficient quality-diversity by cooperative coevolution. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JDud6zbpFv.

Zahavy, T., Schroecker, Y., Behbahani, F., Baumli, K., Flennerhag, S., Hou, S., and Singh, S. Discovering Policies with DOMiNO: Diversity Optimization Maintaining Near Optimality. September 2022. URL https://openreview.net/forum?id=kjkdzBW3b8p.

Zahavy, T., Veeriah, V., Hou, S., Waugh, K., Lai, M., Leurent, E., Tomasev, N., Schut, L., Hassabis, D., and Singh, S. Diversifying AI: Towards Creative Chess with AlphaZero, August 2023. URL http://arxiv.org/abs/2308.09175. arXiv:2308.09175 [cs].

Zhu, M., Liu, M., Shen, J., Zhang, Z., Chen, S., Zhang, W., Ye, D., Yu, Y., Fu, Q., and Yang, W. 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

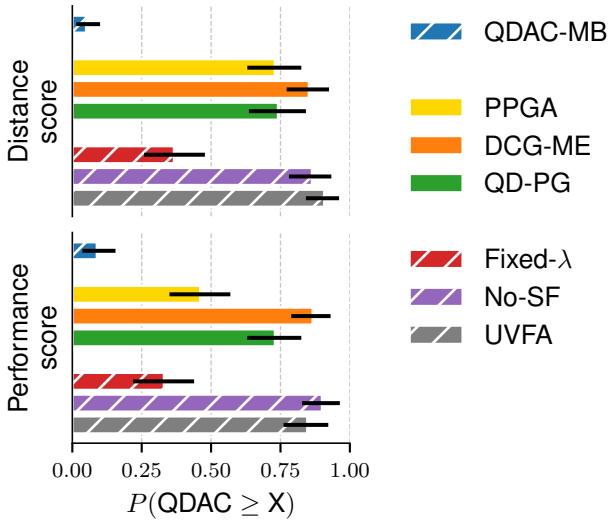


Figure A.7. Probabilities of improvement of QDAC over all other baselines, aggregated across all tasks, as defined by Agarwal et al. (2021).

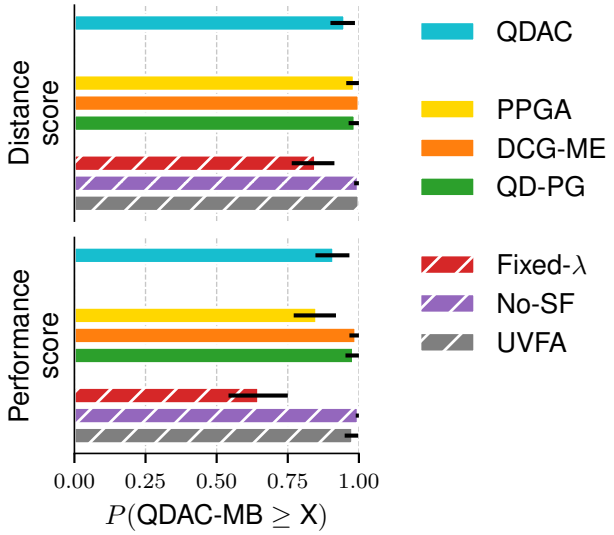


Figure A.8. Probabilities of improvement of QDAC-MB over all other baselines, aggregated across all tasks, as defined by Agarwal et al. (2021).

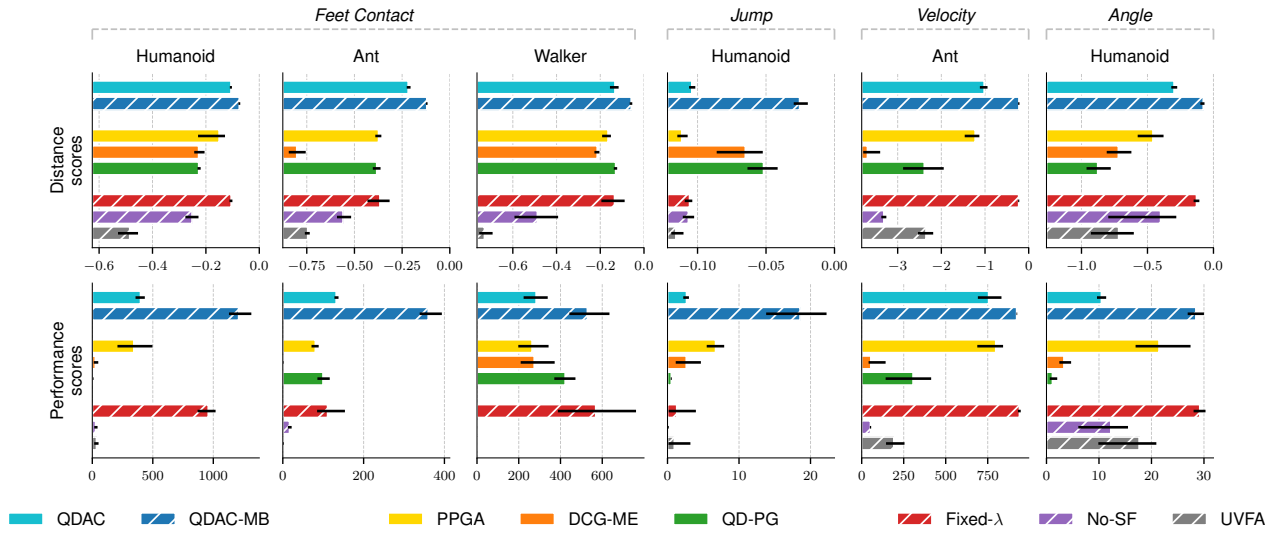


Figure A.9. IQM for distance and performance scores per task.

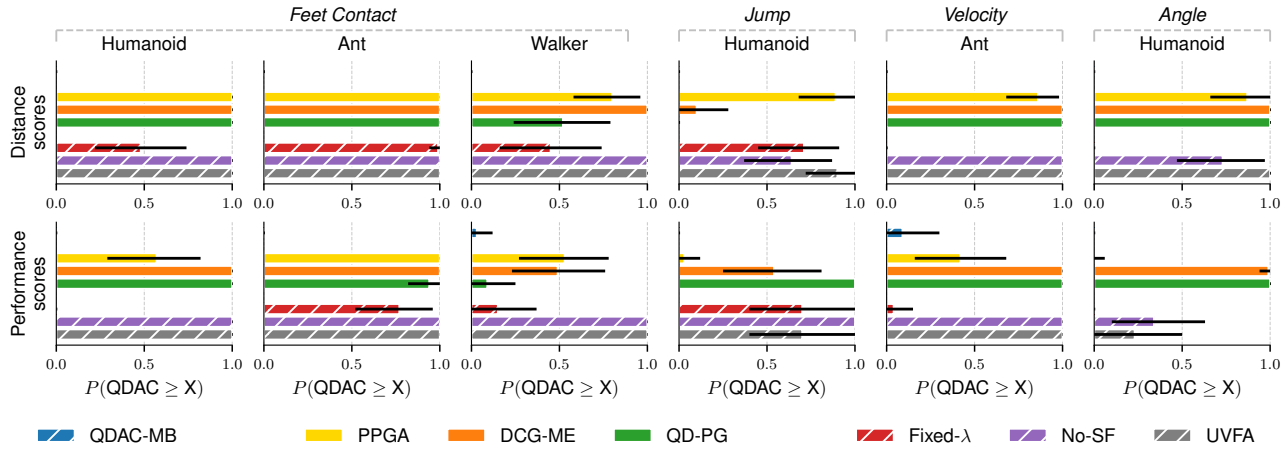


Figure A.10. Per-task probabilities of improvement (as defined by Agarwal et al. (2021)) of QDAC over all other baselines.

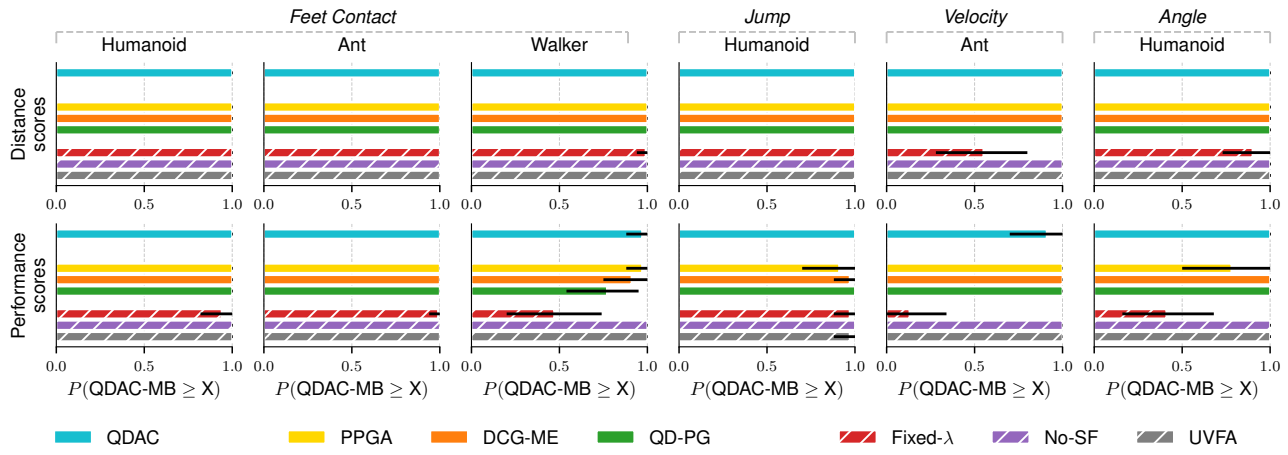


Figure A.11. Per-task probabilities of improvement (as defined by Agarwal et al. (2021)) of QDAC-MB over all other baselines.

A.2. Heatmaps

In Figures A.12 to A.17, we report the heatmaps for the metrics defined in Section 5.3: the negative distance to skill (in the top row) and the performance (in the bottom row). Each heatmap represents the skill space of the corresponding task. In the first row, the color of each cell represents the negated distance to the closest skill achieved by the policy (the darker the better). In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$); while colorized cells indicate the performance score (i.e. the return) achieved by the agent for the corresponding skill.

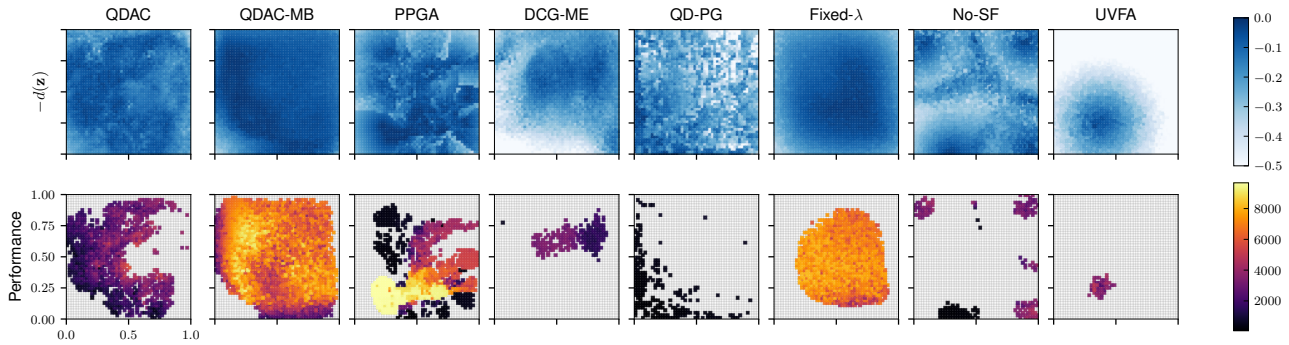


Figure A.12. **Humanoid Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^2$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

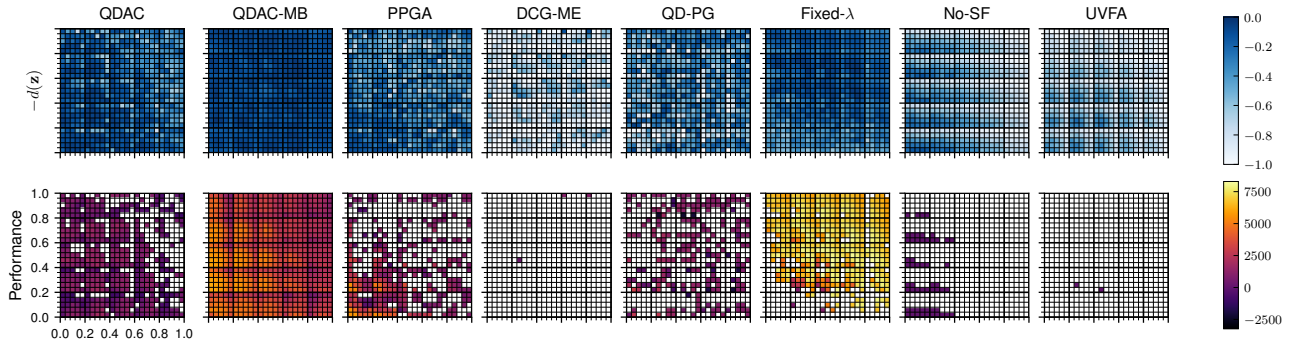


Figure A.13. **Ant Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^4$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

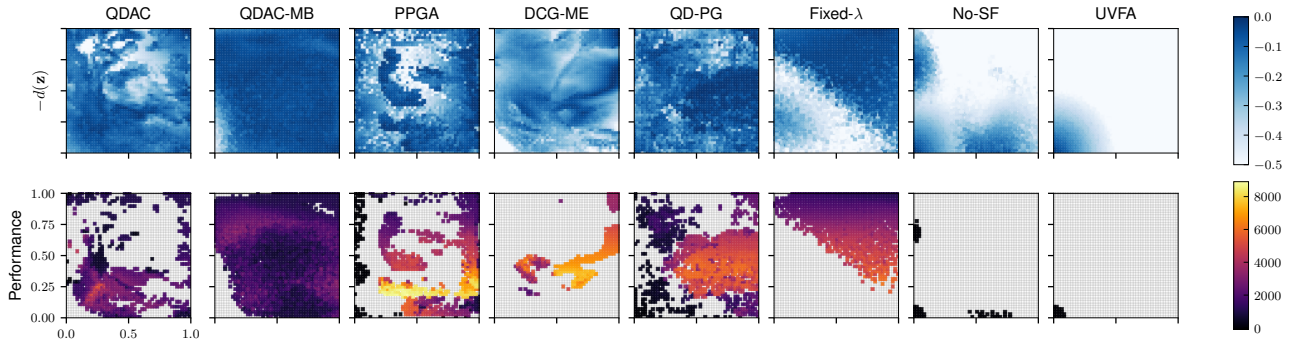


Figure A.14. **Walker Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^2$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

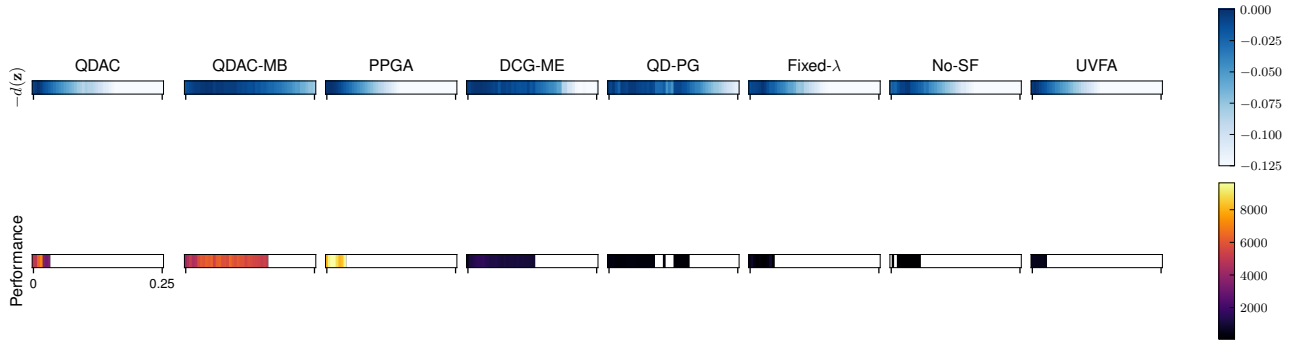


Figure A.15. **Humanoid Jump** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space of jumping skills $\mathcal{Z} = [0, 0.25]$. This space is discretized into cells, with each cell representing a distinct skill; in this task, the skills refer to the average of the lowest foot heights over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

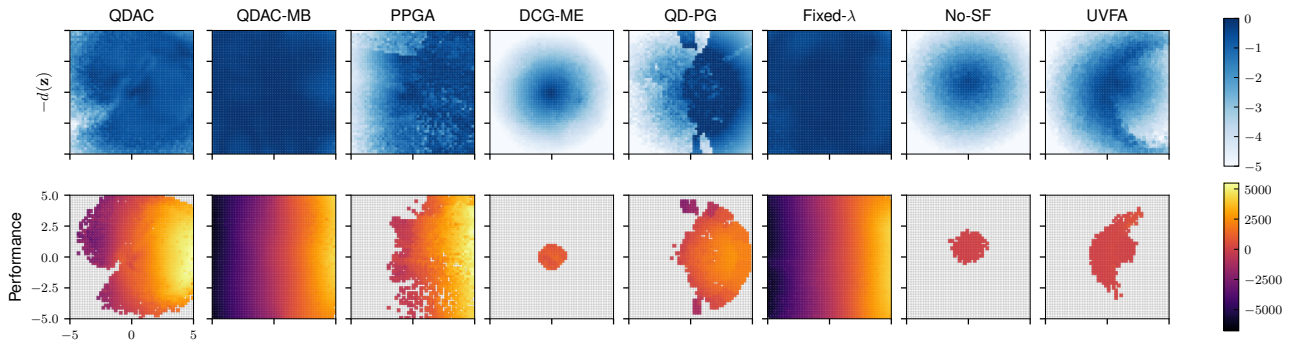


Figure A.16. **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space $\mathcal{Z} = [-5 \text{ m/s}, 5 \text{ m/s}]^2$, of target velocities. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} v_x & v_y \end{bmatrix}^\intercal$. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

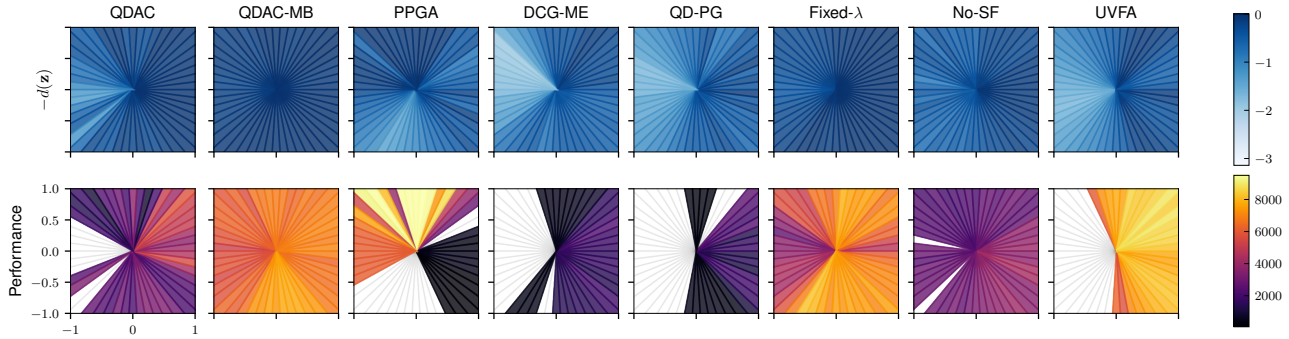


Figure A.17. **Humanoid Angle** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. The heatmap represents the skill space of body angles $\mathcal{Z} =]-\pi, \pi]$. This space is discretized into cells, with each cell representing a distinct skill; in this task, the skills refer to the angle of the humanoid body about the z-axis. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

A.3. Results without filtering with d_{eval}

In Figures A.18 to A.24, 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.

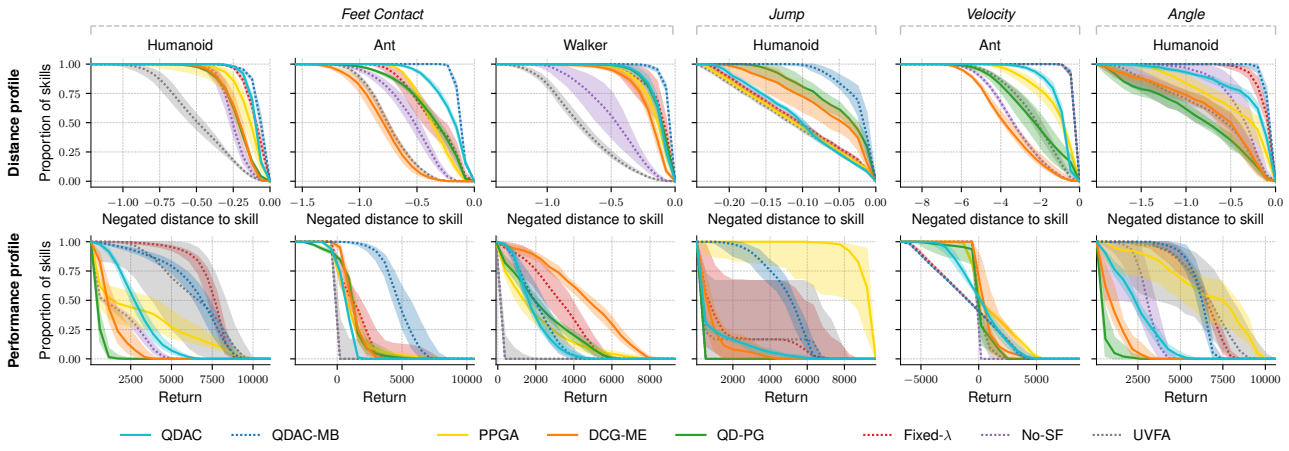


Figure A.18. (**top**) Distance profiles and (**bottom**) performance profiles for each task defined in Section 5.3 similar to Figure 4 except that skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out. The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI. Figure D.33 illustrates how to read distance and performance profiles.

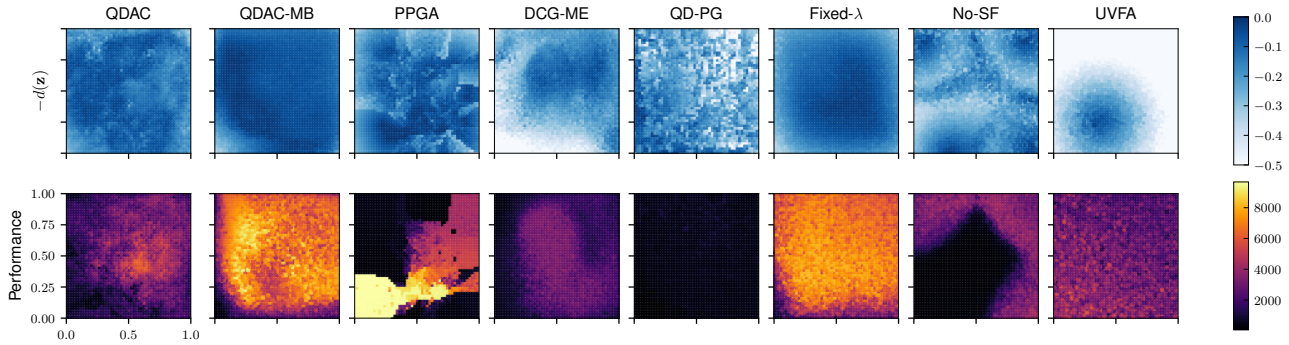


Figure A.19. **Humanoid Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

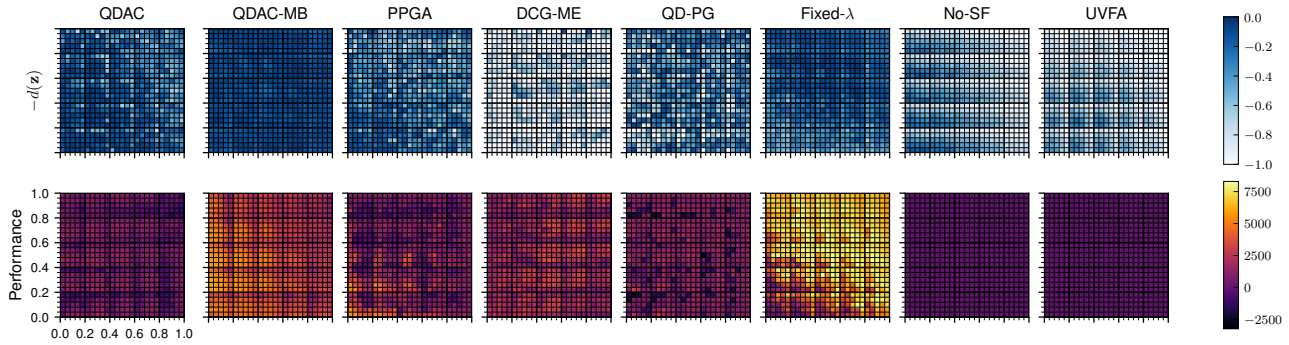


Figure A.20. **Ant Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

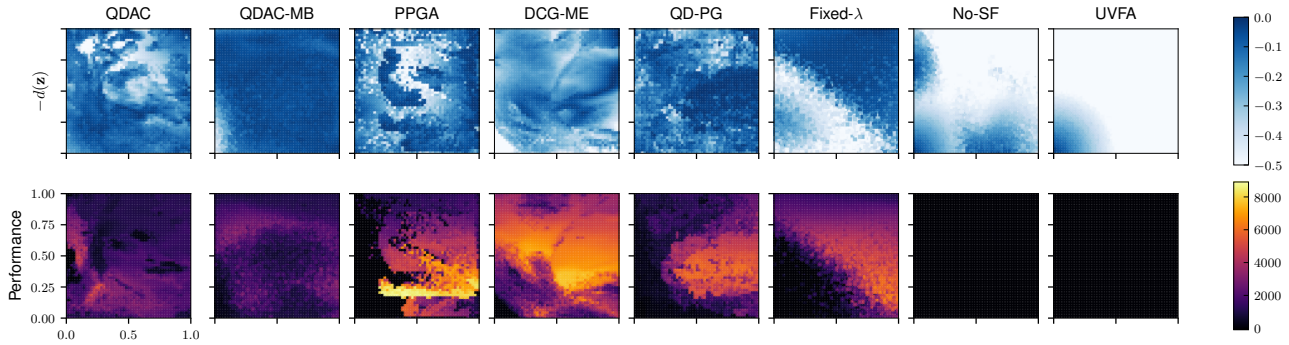


Figure A.21. **Walker Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

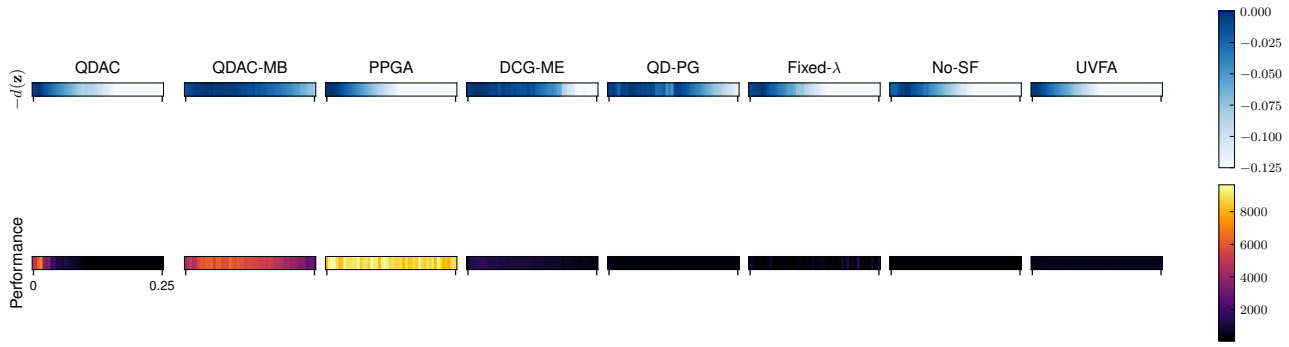


Figure A.22. **Humanoid Jump** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

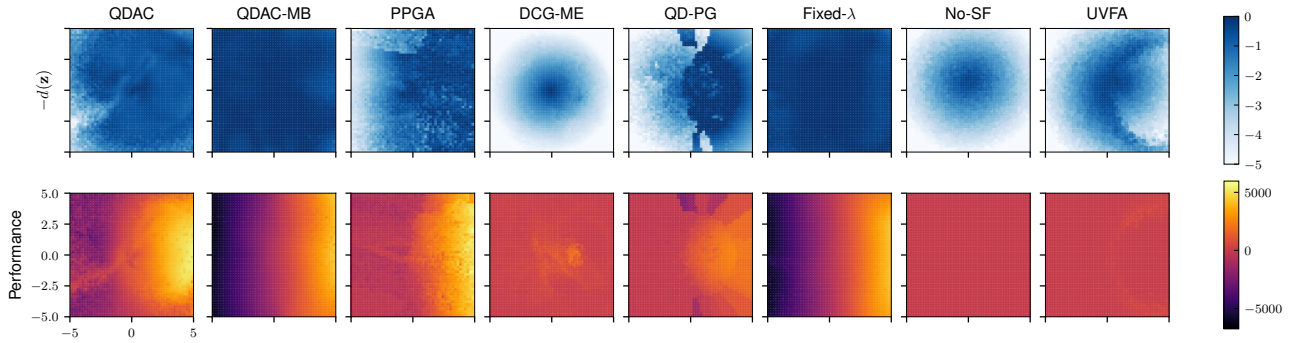


Figure A.23. **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

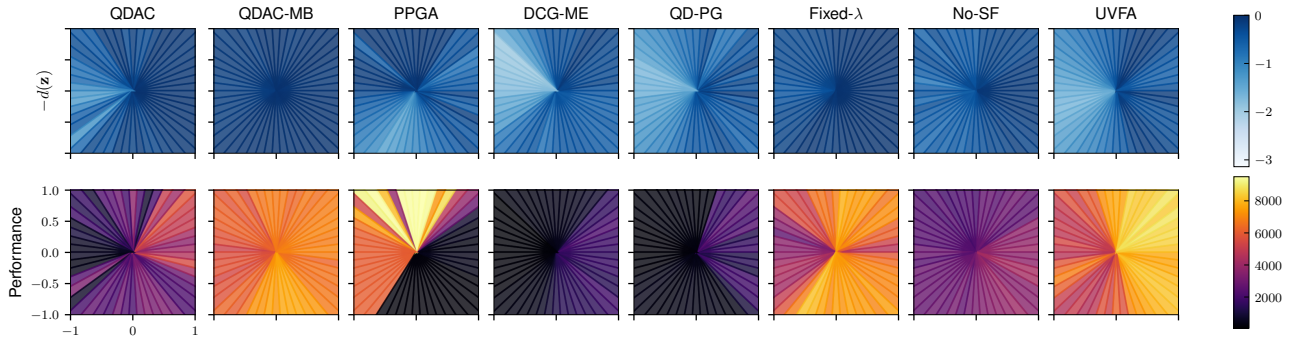


Figure A.24. **Humanoid Angle** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. In the bottom row, the skills that are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$) are **not** filtered out.

A.4. Archive Profiles and Heatmaps for DOMiNO, SMERL and Reverse SMERL

In Figures A.25 to A.31, we present the archive profiles and heatmaps achieved by DOMiNO, SMERL, and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022): we generate an archive from the intermediate policies encountered during training, and use this archive to compare against QDAC and QDAC-MB.

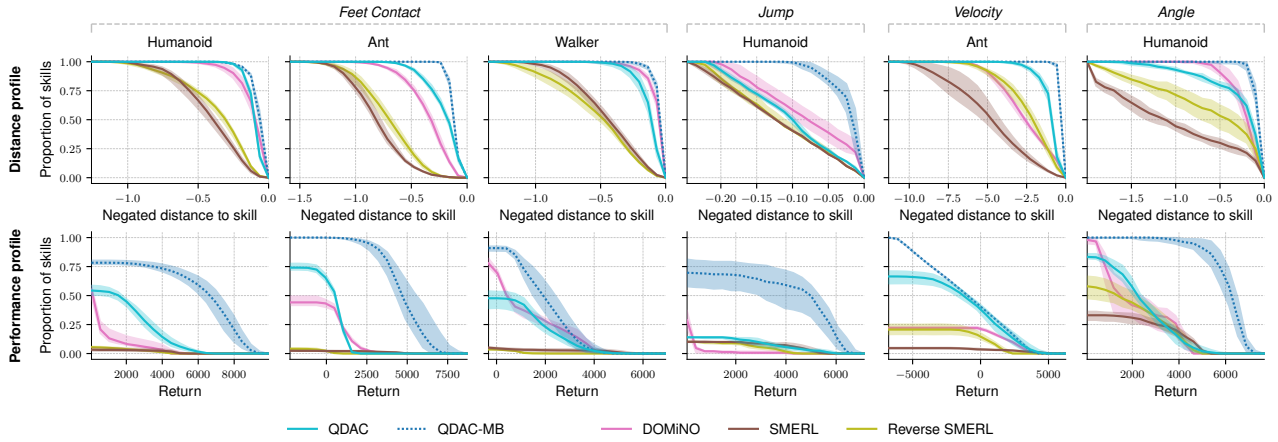


Figure A.25. (**top**) Distance profiles and (**bottom**) performance profiles for each task defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The lines represent the IQM for 10 replications, and the shaded areas correspond to the 95% CI. Figure D.33 illustrates how to read distance and performance profiles.

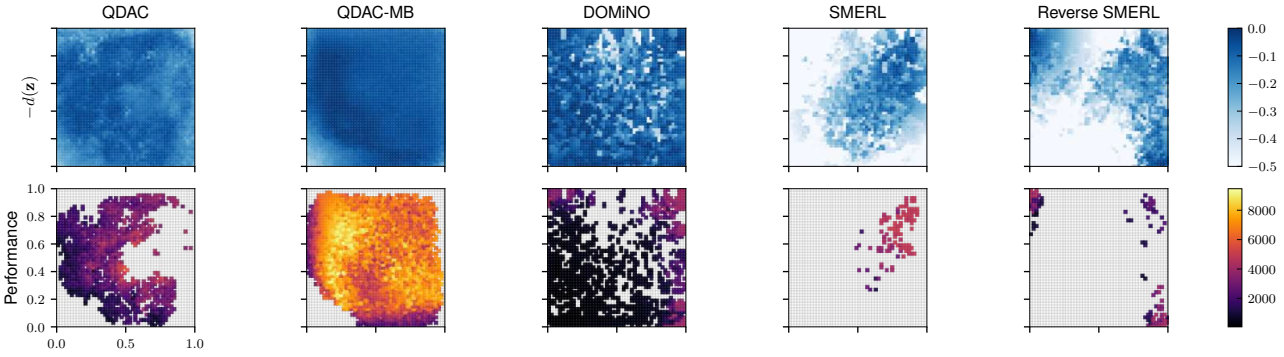


Figure A.26. **Humanoid Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^2$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

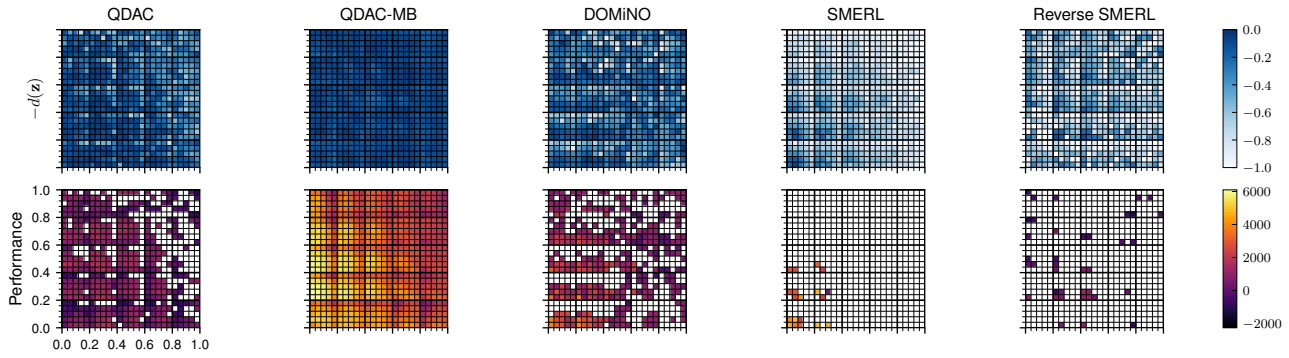


Figure A.27. **Ant Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^4$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

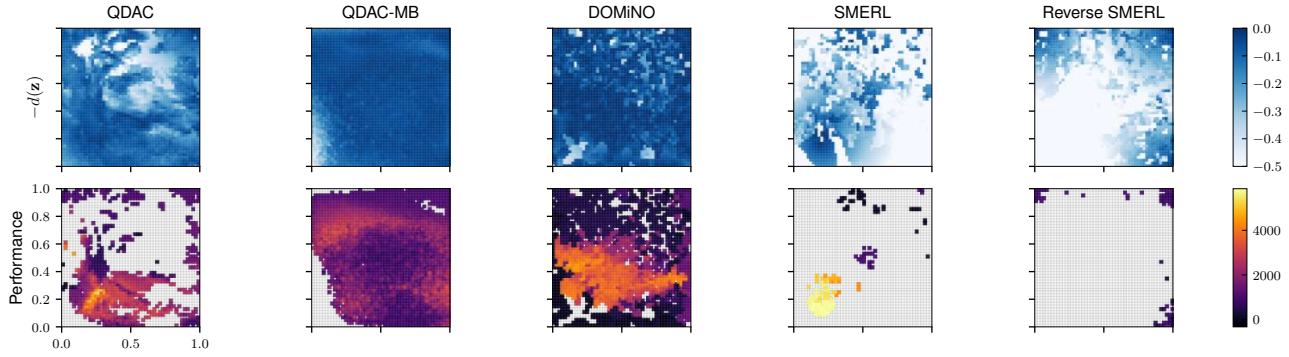


Figure A.28. **Walker Feet Contact** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space of feet contacts $\mathcal{Z} = [0, 1]^2$. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^\intercal$, where z_i is the proportion of time that leg i touches the ground over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

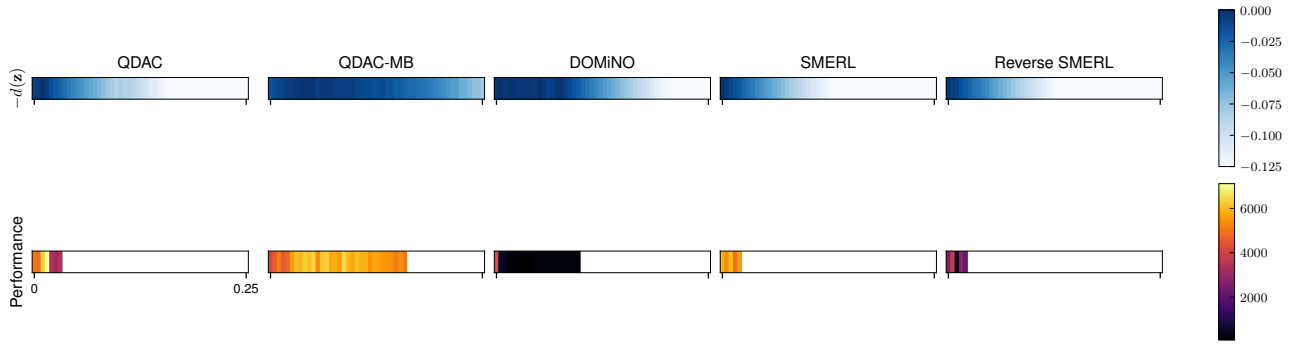


Figure A.29. **Humanoid Jump** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space of jumping skills $\mathcal{Z} = [0, 0.25]$. This space is discretized into cells, with each cell representing a distinct skill; in this task, the skills refer to the average of the lowest foot heights over an entire episode. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

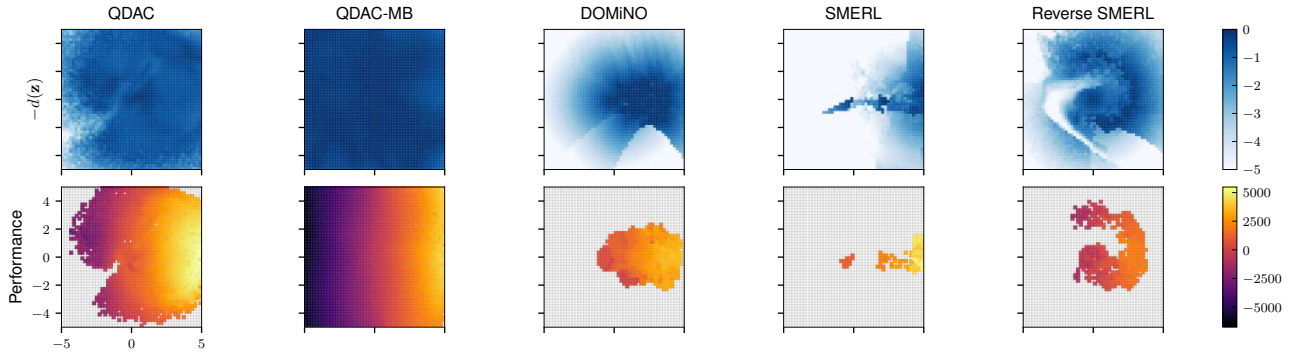


Figure A.30. **Ant Velocity** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space $\mathcal{Z} = [-5 \text{ m/s}, 5 \text{ m/s}]^2$, of target velocities. This space is discretized into cells, with each cell representing a distinct skill $\mathbf{z} = \begin{bmatrix} v_x & v_y \end{bmatrix}^\intercal$. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

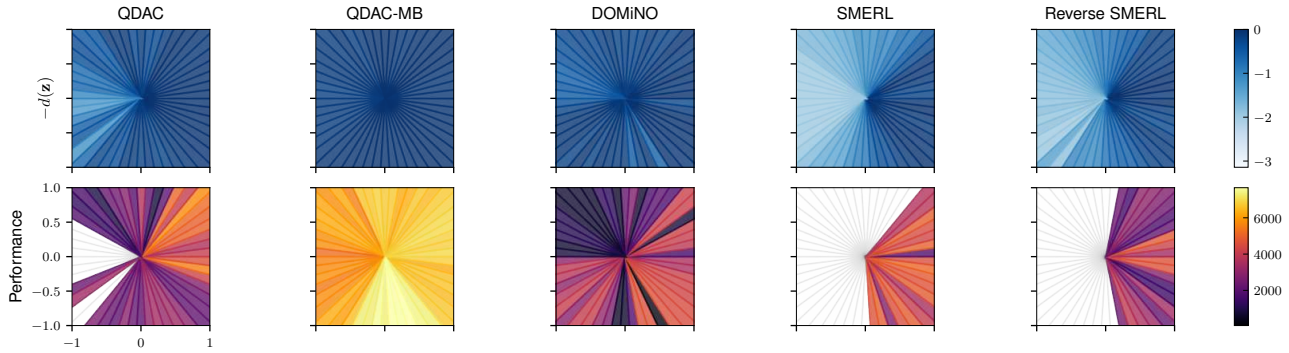


Figure A.31. **Humanoid Angle** Heatmaps of **(top)** negative distance to skill, **(bottom)** performance defined in Section 5.3. We present here the results from DOMiNO, SMERL and Reverse SMERL using a method analogous to that in (Chalumeau et al., 2022). The heatmap represents the skill space of body angles $\mathcal{Z} =]-\pi, \pi]$. This space is discretized into cells, with each cell representing a distinct skill; in this task, the skills refer to the angle of the humanoid body about the z-axis. In the bottom row, empty cells show which skills are not successfully executed (i.e. $d(\mathbf{z}) > d_{\text{eval}}$), while colorized cells indicate the performance score obtained for the corresponding skill.

B. Theoretical Results

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:*

$$\|\mathbb{E}_{\pi_{\mathbf{z}}}[\boldsymbol{\phi}(s,a)] - \mathbf{z}\|_2 \leq \mathbb{E}_{\pi_{\mathbf{z}}}[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2] \tag{6}$$

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

$$\boldsymbol{\psi}(s,a,\mathbf{z}) = \boldsymbol{\phi}(s,a) + \gamma \mathbb{E}_{s' \sim p(.|s,a)}[\boldsymbol{\psi}(s',\mathbf{z})] \tag{7}$$

For all skills $\mathbf{z} \in \mathcal{Z}$ and for all sequences of T states $(s_0, a_0, \ldots, s_{T-1})$ sampled from $\pi_{\mathbf{z}}$, we have:

$$\begin{aligned}
&\left\| \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}(s_t,a_t) - \mathbf{z} \right\|_2 \\
&= \left\| \frac{1}{T} \sum_{t=0}^{T-1} \left(\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \gamma \mathbb{E}_{s_{t+1} \sim p(.|s_t,a_t)}[\boldsymbol{\psi}(s_{t+1},\mathbf{z})] - \mathbf{z} \right) \right\|_2 && \text{(Equation 7)} \\
&= \left\| \frac{1}{T} \sum_{t=0}^{T-1} ((1-\gamma)\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \mathbf{z}) + \frac{\gamma}{T} \sum_{t=0}^{T-1} \left(\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \mathbb{E}_{s_{t+1} \sim p(.|s_t,a_t)}[\boldsymbol{\psi}(s_{t+1},\mathbf{z})] \right) \right\|_2 \\
&\leq \left\| \frac{1}{T} \sum_{t=0}^{T-1} (1-\gamma)\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \mathbf{z} \right\|_2 + \gamma \left\| \frac{1}{T} \sum_{t=0}^{T-1} \left(\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \mathbb{E}_{s_{t+1} \sim p(.|s_t,a_t)}[\boldsymbol{\psi}(s_{t+1},\mathbf{z})] \right) \right\|_2 \\
& && \text{(triangular inequality)}
\end{aligned}$$

We denote $\rho(s) = \lim_{t \to \infty} P(s_t = s | s_0, \pi_{\mathbf{z}})$ the stationary distribution of states under $\pi_{\mathbf{z}}$, which we assume exists and is independent of s_0. Consequently, by taking the right term to the limit as $T \to \infty$:

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

Furthermore, by taking the left term to the limit as $T \to \infty$:

$$\begin{aligned}
\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} (1-\gamma)\boldsymbol{\psi}(s_t,a_t,\mathbf{z}) - \mathbf{z} &= (1-\gamma)\mathbb{E}_{\substack{s \sim \rho \\ a \sim \pi(.|s,\mathbf{z})}}[\boldsymbol{\psi}(s,a,\mathbf{z})] - \mathbf{z} \\
&= (1-\gamma)\mathbb{E}_{s \sim \rho}[\boldsymbol{\psi}(s,\mathbf{z})] - \mathbf{z}
\end{aligned}$$

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

$$\begin{aligned}
&\|\mathbb{E}_{\pi_{\mathbf{z}}}[\boldsymbol{\phi}(s,a)] - \mathbf{z}\|_2 \leq \|(1-\gamma)\mathbb{E}_{\pi_{\mathbf{z}}}[\boldsymbol{\psi}(s,\mathbf{z})] - \mathbf{z}\|_2 + \|\mathbf{0}\|_2 \\
&\boxed{\|\mathbb{E}_{\pi_{\mathbf{z}}}[\boldsymbol{\phi}(s,a)] - \mathbf{z}\|_2 \leq \mathbb{E}_{\pi_{\mathbf{z}}}[\|(1-\gamma)\boldsymbol{\psi}(s,\mathbf{z}) - \mathbf{z}\|_2]} && \text{(Jensen's inequality)}
\end{aligned}$$

□

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 $\pi_{\mathbf{z}}$. If we write $\epsilon := \sup_t \mathbb{E}_{\pi_{\mathbf{z}}}\left[\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}(s_t, a_t, \mathbf{z}) - \boldsymbol{\psi}(s_{t+1}, a_{t+1}, \mathbf{z})\|_2\right]$, then:*

$$\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2\right] \leq \sup_t \mathbb{E}_{\pi_{\mathbf{z}}}\left[\|(1-\gamma)\boldsymbol{\psi}(s_t, a_t, \mathbf{z}) - \mathbf{z}\|_2\right] + \epsilon \tag{8}$$

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:

$$\boldsymbol{\psi}_t := \boldsymbol{\psi}\left(s_t, a_t, \mathbf{z}\right)$$

We define β as follows:

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

Then we have, for all t,

$$\mathbb{E}_{\pi_{\mathbf{z}}}\left[\|(1-\gamma)\boldsymbol{\psi}_t - \mathbf{z}\|_2\right] \leq \beta \tag{10}$$

$$\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}_t - \boldsymbol{\psi}_{t+1}\right\|_2\right] \leq \epsilon \tag{11}$$

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

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

$$\text{or also: } \boldsymbol{\phi}_t = \boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t, a_t\right] \tag{13}$$

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:

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

Thus,

$$\begin{aligned}
\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2\right] \leq\ & \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left((1-\gamma)\boldsymbol{\psi}_t - \mathbf{z}\right)\right\|_2\right] \\
& + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t, a_t\right]\right)\right\|_2\right]
\end{aligned} \tag{14}$$

[1] *almost sure* refers to the almost sure convergence from probability theory where rollouts are sampled from $\pi_{\mathbf{z}}$.

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:

$$\begin{aligned}
\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\right\|_2\right] &\leq \mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\|_2\right] && \text{(triangular inequality)}\\
&\leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\|((1-\gamma)\boldsymbol{\psi}_t-\mathbf{z})\|_2\right]\\
&\leq \frac{1}{T}\sum_{t=0}^{T-1}\beta && \text{(from Equation 10)}\\
&\leq \beta && (15)
\end{aligned}$$

Also, we can prove that the second term of the right-hand side in Equation 14 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:

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

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

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

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

$$
\begin{aligned}
&\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\left(\gamma\boldsymbol{\psi}_t - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t, a_t\right]\right)\right\|_2\right] \\
&\leq \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-2}\left(\gamma\boldsymbol{\psi}_{t+1} - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t, a_t\right]\right)\right\|_2\right] && \text{(triangular inequality)} \\
&\leq \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-2}\left\|\gamma\boldsymbol{\psi}_{t+1} - \gamma\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_{t+1}\middle| s_t, a_t\right]\right\|_2\right] \\
&\leq \eta_T + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\frac{1}{T}\sum_{t=0}^{T-2}\left\|\gamma\boldsymbol{\psi}_{t+1} + \boldsymbol{\phi}_t - \boldsymbol{\psi}_t\right\|_2\right] && \text{(from Equation 12)} \\
&\leq \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\gamma\boldsymbol{\psi}_{t+1} + \boldsymbol{\phi}_t - \boldsymbol{\psi}_t\right\|_2\right] \\
&\leq \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\phi}_t + \gamma\boldsymbol{\psi}_{t+1} - \boldsymbol{\psi}_t\right\|_2\right] \\
&\leq \eta_T + \frac{1}{T}\sum_{t=0}^{T-2}\epsilon \\
&\leq \eta_T + \frac{T-1}{T}\epsilon
\end{aligned} \tag{16}
$$

After combining the two previously derived Equations 15 and 16, we get:

$$
\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\phi}_t - \mathbf{z}\right\|_2\right] \leq \beta + \eta_T + \frac{T-1}{T}\epsilon \tag{17}
$$

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

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

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

$$
\begin{aligned}
\eta_T &\leq \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\mathbb{E}_{\pi_{\mathbf{z}}}\left[\boldsymbol{\psi}_T\middle| s_{T-1}, a_{T-1}\right]\right\|_2\right]\right) \\
&\leq \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_T\right\|_2\middle| s_{T-1}, a_{T-1}\right]\right]\right) && \text{(Jensen's inequality)} \\
&\leq \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \mathbb{E}_{\pi_{\mathbf{z}}}\left[\left\|\boldsymbol{\psi}_T\right\|_2\right]\right) && \text{(law of total expectation)} \\
&\leq \frac{\gamma}{T}\left(\frac{\rho}{1-\gamma} + \frac{\rho}{1-\gamma}\right) \\
&\leq \frac{1}{T}\left(\frac{2\rho\gamma}{1-\gamma}\right)
\end{aligned} \tag{18}
$$

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 14 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, $\mathbb{E}_{\pi_\mathbf{z}} [\rho + \|\mathbf{z}\|_2] < \infty$, 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:

$$\lim_{T\to\infty} \mathbb{E}_{\pi_\mathbf{z}} [X_T] = \mathbb{E}_{\pi_\mathbf{z}} \left[\lim_{T\to\infty} X_T \right] \tag{19}$$

Finally, by taking the Equation 14 to the limit as $T \to \infty$, we get:

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

□

Algorithm 2 Detailed training procedure of QDAC

input Parameters θ_π, θ_V, $\theta_{\boldsymbol{\psi}}$, θ_λ ▷ Initial parameters for the actor, critics and Lagrange multiplier
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay buffer
repeat
  $\mathbf{z} \sim \mathcal{U}(\mathcal{Z})$ ▷ Sample skill uniformly from skill space
  for T steps **do**
    ▷ Environment steps
    $a_t \sim \pi(a_t|s_t, \mathbf{z})$ ▷ Sample action from policy
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t, \mathbf{z})$ ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), \boldsymbol{\phi}(s_t, a_t), s_{t+1}, \mathbf{z})\}$ ▷ Store transition in the replay buffer
    ▷ Training steps
    $\theta_\lambda \leftarrow \theta_\lambda - \alpha_\lambda \nabla J_\lambda(\theta_\lambda)$ ▷ Update Lagrange multiplier with Eq. 5
    $\theta_{Q,i} \leftarrow \theta_{Q,i} - \alpha_Q \nabla J_Q(\theta_{Q,i})$ for $i \in \{1, 2\}$ ▷ Policy evaluation for the Q-networks (Haarnoja et al., 2019)
    $\theta_{\boldsymbol{\psi}} \leftarrow \theta_{\boldsymbol{\psi}} - \alpha_{\boldsymbol{\psi}} \nabla J_{\boldsymbol{\psi}}(\theta_{\boldsymbol{\psi}})$ ▷ Policy evaluation for successor features with Eq. 2
    $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \nabla J_\pi(\theta_\pi)$ ▷ Policy improvement with Eq. 21
    $\beta \leftarrow \beta - \alpha_\beta \nabla J_\beta(\beta)$ ▷ Adjust temperature as in (Haarnoja et al., 2019)
    ▷ Update target networks
    $\theta'_{Q,i} \leftarrow \tau\theta_{Q,i} + (1-\tau)\theta'_{Q,i}$ for $i \in \{1, 2\}$
    $\theta'_{\boldsymbol{\psi}} \leftarrow \tau\theta_{\boldsymbol{\psi}} + (1-\tau)\theta'_{\boldsymbol{\psi}}$
  end for
until convergence

C. Additional Training Details

C.1. Expanded Information on QDAC

The policy parameters θ_π are optimized to maximize the objective function from Equation 4. To that end, we use the Soft Actor-Critic (SAC) algorithm with adjusted temperature β (Haarnoja et al., 2019). Then the objective from Equation 4 needs to be slightly modified to be optimized by SAC:

$$J_\pi(\theta_\pi) = (1 - \lambda(s, \mathbf{z}))\, Q(s, a, \mathbf{z}) - \lambda(s, \mathbf{z}) \|(1-\gamma)\boldsymbol{\psi}(s, a, \mathbf{z}) - \mathbf{z}\|_2 + \underbrace{\beta \log \pi(a|s, \mathbf{z})}_{\substack{\text{Entropy regularization} \\ \text{term used in SAC}}} \tag{20}$$

Using the same notations as (Haarnoja et al., 2019), each action a returned by the policy π can be seen as the function of the state s, the skill $\mathbf{z}$, and a random noise ϵ: $a = f_{\theta_\pi}(s, \mathbf{z}, \epsilon)$. Then the complete form of the actor's objective function is as follows:

$$J_\pi(\theta_\pi) = (1 - \lambda(s, \mathbf{z}))\, Q(s, f_{\theta_\pi}(s, \mathbf{z}, \epsilon), \mathbf{z}) - \lambda(s, \mathbf{z}) \|(1-\gamma)\boldsymbol{\psi}(s, f_{\theta_\pi}(s, \mathbf{z}, \epsilon), \mathbf{z}) - \mathbf{z}\|_2 + \beta \log \pi(f_{\theta_\pi}(s, \mathbf{z}, \epsilon)|s, \mathbf{z}) \tag{21}$$

In our setup, the policy π outputs a vector μ and a vector of standard deviations $\begin{pmatrix}\sigma_1 & \cdots & \sigma_n\end{pmatrix}$. The action a is computed as follows: $a = \mu + \begin{pmatrix}\sigma_1\epsilon_1 & \cdots & \sigma_n\epsilon_n\end{pmatrix}$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The Q-network is trained using the exact same procedure as in (Haarnoja et al., 2019), with a clipped double-Q trick (Fujimoto et al., 2018). The successor features network $\boldsymbol{\psi}$ is trained to minimize the Bellman error (see Eq. 7). The targets of the double Q-network and of the successor features network are updated at each iteration using soft target updates, in order to stabilize training (Lillicrap et al., 2015).

Algorithm 2 provides a detailed description of the training procedure of QDAC.

C.2. Expanded Information on QDAC-MB

We provide here additional details on world models, and on our implementation of QDAC's model-based variant.

Algorithm 3 QDAC-MB

input Parameters θ_π, θ_V, $\theta_{\boldsymbol{\psi}}$, θ_λ, $\theta_{\mathcal{W}}$ ▷ Initial parameters for the actor, critics, Lagrange multiplier and world model
  $\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay buffer
  repeat
    $\mathbf{z} \sim \mathcal{U}(\mathcal{Z})$ ▷ Sample skill uniformly from skill space
    for T steps **do**
      ▷ Environment steps
      $a_t \sim \pi(a_t|\widetilde{s}_t, \mathbf{z})$ ▷ Sample action from policy
      $s_{t+1} \sim p(s_{t+1}|s_t, a_t, \mathbf{z})$ ▷ Sample transition from the environment
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), \boldsymbol{\phi}(s_t, a_t), s_{t+1})\}$ ▷ Store transition in the replay buffer
      $\theta_{\mathcal{W}} \leftarrow \theta_{\mathcal{W}} - \alpha_{\mathcal{W}} \nabla J_{\mathcal{W}}(\theta_{\mathcal{W}})$ ▷ Update world model
      ▷ Training steps from a rollout in imagination with skills $\widetilde{\mathbf{z}} \sim \mathcal{U}(\mathcal{Z})$
      $\theta_\lambda \leftarrow \theta_\lambda - \alpha_\lambda \nabla J_\lambda(\theta_\lambda)$ ▷ Update Lagrange multiplier with Eq. 5
      $\theta_V \leftarrow \theta_V - \alpha_V \nabla J_V(\theta_V)$ ▷ Policy evaluation for value function with Eq. 1
      $\theta_{\boldsymbol{\psi}} \leftarrow \theta_{\boldsymbol{\psi}} - \alpha_{\boldsymbol{\psi}} \nabla J_{\boldsymbol{\psi}}(\theta_{\boldsymbol{\psi}})$ ▷ Policy evaluation for successor features with Eq. 2
      $\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \nabla J_\pi(\theta_\pi)$ ▷ Policy improvement with Eq. 23
    end for
  until convergence

C.2.1. 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: $J_\pi(\theta_\pi) = \mathbb{E}_{\substack{a \sim \pi(.|\widetilde{s}) \\ \widetilde{s}' \sim \widehat{p}(.|\widetilde{s}, a)}} \left[\sum_{t=1}^{H} V_\lambda(\widetilde{s}_t) \right]$.

C.2.2. QDAC-MB

QDAC-MB's pseudocode is provided in Algorithm 3. 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 C.32.

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 $\boldsymbol{\psi}_\lambda$ respectively. The successor features target is defined recursively

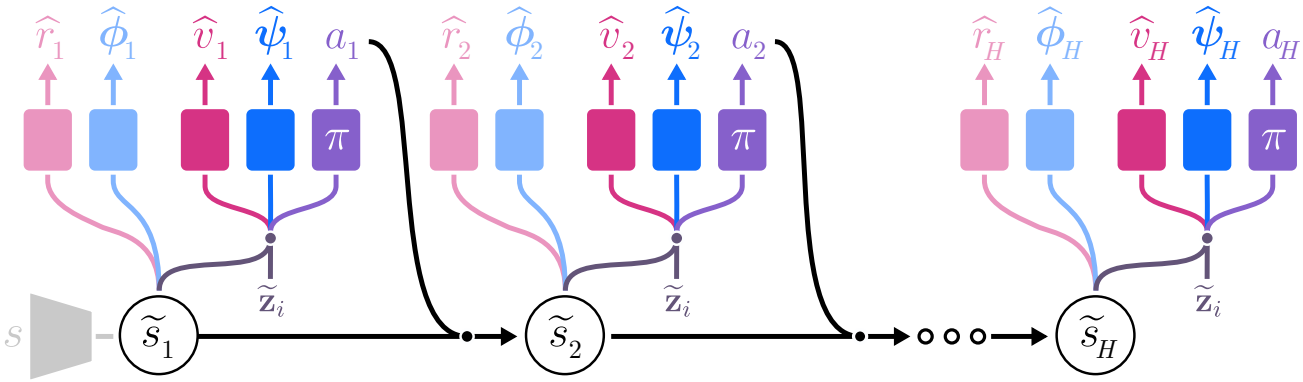


Figure C.32. Imagination rollout performed within the world model $\mathcal{W}$. Each individual rollout i generates on-policy transitions following skill $\widetilde{\mathbf{z}}_i$, starting from a state $\widetilde{s}_1$ for a fixed number of steps H. The world model predicts $\widehat{r}_i$ and $\widehat{\phi}_i$ that enable to compute $\widehat{V}_i$ and $\widehat{\psi}_i$ respectively.

as follows:

$$\begin{aligned} &\psi_\lambda(\widetilde{s}_t, \widetilde{\mathbf{z}}) = \widehat{\phi}_t + \gamma \widehat{c}_t \left((1-\lambda)\, \widehat{\psi}(\widetilde{s}_{t+1}, \widetilde{\mathbf{z}}) + \lambda \psi_\lambda(\widetilde{s}_{t+1}, \widetilde{\mathbf{z}}) \right) \\ &\text{and} \quad \psi_\lambda(\widetilde{s}_H, \widetilde{\mathbf{z}}) = \widehat{\psi}(\widetilde{s}_H, \widetilde{\mathbf{z}}) \end{aligned} \tag{22}$$

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 ψ, and the actor by backpropagating through the dynamics of the model. The actor maximizes the following objective, with an entropy regularization for exploration, where $\mathrm{sg}\,(\cdot)$ represents the *stop gradient* function.

$$J_\pi\left(\theta_\pi\right) = \mathbb{E}_{\substack{\widetilde{s}_{1:H} \sim \mathcal{W}, \pi \\ \widetilde{\mathbf{z}} \sim \mathcal{U}(\mathcal{Z})}} \left[\sum_{t=1}^{H} \left(1 - \mathrm{sg}\,(\lambda)\right) \underbrace{V_\lambda(\widetilde{s}_t, \widetilde{\mathbf{z}})}_{\text{Performance}} - \mathrm{sg}\,(\lambda) \underbrace{\left\| (1-\gamma) \psi_\lambda(\widetilde{s}_t, \widetilde{\mathbf{z}}) - \widetilde{\mathbf{z}} \right\|_2}_{\text{Distance to desired skill } \widetilde{\mathbf{z}}} \right] \tag{23}$$

D. Tasks and Metrics Details

D.1. Tasks

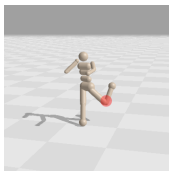
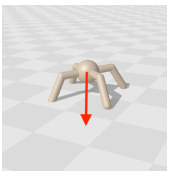
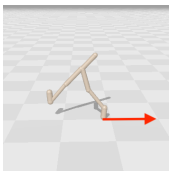
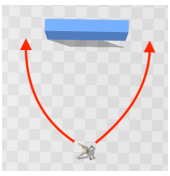
Table D.1. Tasks

	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}	0.1	0.3	0.1	0.025	1.0	0.6

D.2. Few-Shot Adaptation and Hierarchical Learning 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 D.2. Adaptation tasks

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

D.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 6, 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 features to jump over hurdles varying in height from 0 to 50 cm.

On *Humanoid - Motor Failure*, we use the feet contact features 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 6) ranging from 0.0 (no damage) to 1.0 (maximal damage).

On *Ant - Gravity*, we use the feet contact features 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 features 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).

D.2.2. HIERARCHICAL LEARNING

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

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

D.3. Evaluation Metrics Details

In this section, we illustrate how to compute and read the distance and performance profiles in Figure 4. In the Quality-Diversity community, there is a consensus that the best evaluation metric is the "distance/performance profile" (Flageat et al., 2022; Grillotti et al., 2023; Grillotti & Cully, 2022a; Batra et al., 2023). This metric 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. For a given distance d, the distance profile shows the proportion of skills with distance to skill lower than d. For a given performance p, the performance profile shows the proportion of skills with a performance higher than p. The bigger the area under the curve, the better the algorithm is. The profiles have similarities with the cumulative distribution functions in probability.

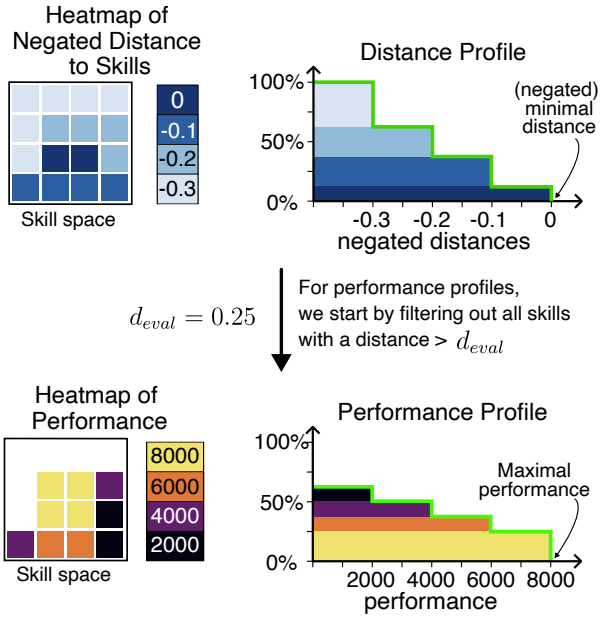


Figure D.33. **Left** Heatmaps of negated distance to skills and performance. **Right** Distance and performance profiles A skill $\mathbf{z}$ is considered successfully executed if $d(\mathbf{z}) < d_{\text{eval}}$, otherwise it is considered failed. All the skills that are not successfully executed by the policy are filtered out before computing the performance heatmap and profile. 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).

E. Baselines details

Table E.3. Comparison of the main features for the different algorithms

Algorithm	Objective Function	Model-based	Lagrange Multiplier λ
QDAC	$(1-\lambda)\sum\gamma^t r_t - \lambda\left\|(1-\gamma)\sum\gamma^t\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✗	✓
QDAC-MB	$(1-\lambda)\sum\gamma^t r_t - \lambda\left\|(1-\gamma)\sum\gamma^t\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	✓
DCG-ME †	$\sum\gamma^t\exp(-\frac{\\|\mathbf{z}-\mathbf{z}'\\|_2}{l})r_t$	✗	✗
QD-PG †	$\sum\gamma^t r_t$ or $\sum\gamma^t\tilde{r}_t$	✗	✗
DOMiNO	$(1-\lambda)\sum\gamma^t r_t + \lambda\sum\gamma^t\tilde{r}_t$	✗	✓
SMERL ‡	$\sum\gamma^t\left(r_t + \lambda\mathbb{1}(R \geq R^* - \epsilon)\tilde{r}_t\right)$	✗	✗
Reverse SMERL ‡	$\sum\gamma^t\left(\mathbb{1}(R < R^* - \epsilon)r_t + \lambda\tilde{r}_t\right)$	✗	✗
No-SF	$(1-\lambda)\sum\gamma^t r_t - \lambda\sum\gamma^t\left\|\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	✓
Fixed-λ	$(1-\lambda)\sum\gamma^t r_t - \lambda\left\|(1-\gamma)\sum\gamma^t\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	✗
UVFA	$(1-\lambda)\sum\gamma^t r_t - \lambda\sum\gamma^t\left\|\boldsymbol{\phi}_t - \mathbf{z}\right\|_2$	✓	✗

† DCG-ME and QD-PG learn diverse skills with mechanisms that are not visible in their objective function.
‡ see Section E.4 for a detailed explanation of the reward used in SMERL and Reverse SMERL.

E.1. DCG-ME

DCG-ME (Faldor et al., 2023a) is a QD algorithm based on MAP-Elites (Mouret & Clune, 2015), that combines evolutionary methods with reinforcement learning to improve sample efficiency. DCG-ME addresses these challenges through two key innovations: First, it enhances the Policy Gradient variation operator with a descriptor-conditioned critic. This allows for a more nuanced exploration of the solution space by guiding the search towards uncharted territories of high diversity and performance. Second, by utilizing actor-critic training paradigms, DCG-ME learns a descriptor-conditioned policy that encapsulates the collective knowledge of the population into a singular, versatile policy capable of exhibiting a wide range of behaviors without incurring additional computational costs.

E.2. QD-PG

QD-PG (Pierrot et al., 2022) is a QD algorithm based on MAP-Elites (Mouret & Clune, 2015) that integrates Policy Gradient methods with QD approaches, aiming to generate a varied collection of neural policies that perform well within continuous control environments. The core innovation of QD-PG lies in its Diversity Policy Gradient (DPG), a mechanism designed to enhance diversity among policies in a sample-efficient manner. This is achieved by leveraging information available at each time step to nudge policies toward greater diversity. The policies in the MAP-Elites grid are subjected to two gradient-based mutation operators, specifically tailored to augment both their quality (performance) and diversity. This dual-focus approach not only addresses the exploration-exploitation dilemma inherent in learning but also enhances robustness by producing multiple effective solutions for the given problem.

The diversity policy gradient is based on the maximization of an intrinsic reward defined as: $\tilde{r}_t = \sum_{j=1}^{J}\left\|\boldsymbol{\phi}(s_t) - \boldsymbol{\phi}(s_j)\right\|_2$, where the J states s_j are coming from an archive of past encountered states.

E.3. DOMiNO

DOMiNO (Zahavy et al., 2022) considers a set of policies $(\pi_i)_{i\in[1,N]}$, and intends to maximize simultaneously their quality while also maximizing the diversity of those that are near-optimal. Thus each policy π^i maximizes the following objective:

$$(1-\lambda)\sum\gamma^t r_t^i + \lambda\sum\gamma^t\tilde{r}_t^i$$

where r_t^i is the task reward (also called *extrinsic reward*) and $\tilde{r}_t^i$ is the *diversity reward*. Also, λ^i refers to the Lagrange multiplier of policy π^i; it is used to balance between (1) the maximization of the extrinsic reward when the policy is not near-optimal, and (2) the maximization of diversity when the policy is near-optimal. The definition of *near-optimality* is given by the first policy π^1.

The first policy π^1 only maximizes the expected sum of extrinsic rewards without considering any diversity. Hence, $\tilde{r}^1 = 0$

and its Lagrange multiplier λ^1 is always equal to 0. Its average extrinsic reward $\widetilde{V}_e^1$ is estimated empirically and used to define when the other policies are near-optimal. The other policies $(\pi^i)_{i\geq 2}$ are considered near-optimal when their average extrinsic reward $\widetilde{V}_e^i$ is higher than $\alpha\widetilde{V}_e^1$ (considering $\widetilde{V}_e^1$ is positive) where α is a constant between 0 and 1. If a policy is not near-optimal, its Lagrange coefficient λ^i decreases to focus on maximizing the task reward; if it is near-optimal, the coefficient increases to give more importance to the diversity reward $\tilde{r}^i$. For policies $(\pi^i)_{i\geq 2}$, the diversity reward balances between repulsion and attraction of the average features $\widetilde{\boldsymbol{\psi}}^i$ experienced by the policies:

$$\tilde{r}_t^i = \left(1 - \left(\frac{l_i}{l_0}\right)^3\right) \phi_t^i \cdot (\widetilde{\boldsymbol{\psi}}^i - \widetilde{\boldsymbol{\psi}}^{j^*})$$

where $j^* = \arg\min_j \left\|\widetilde{\boldsymbol{\psi}}_i - \widetilde{\boldsymbol{\psi}}_j\right\|_2$ and l_0 is a constant.

E.4. SMERL and Reverse 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 SMERL with continuous DIAYN with 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). Finally, we use DIAYN + prior (Eysenbach et al., 2018; Chalumeau et al., 2022) to guide SMERL and Reverse SMERL towards relevant skills as explained in DIAYN's original 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 DIAYN corresponds 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 $\mathbf{z}_{\text{DIAYN}} \sim q(\mathbf{z}_{\text{DIAYN}}|\boldsymbol{\phi}(s))$.

E.5. No-SF

We can show that the constraint in QDAC's objective function is easier to satisfy than No-SF'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:

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

At each timestep t, No-SF tries to satisfy $\boldsymbol{\phi}_t = \mathbf{z}$, whereas QDAC approximately tries to satisfy $\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\boldsymbol{\phi}_t = \mathbf{z}$, which is less restrictive.

F. Hyperparameters

We provide here all the hyperparameters used for QDAC and all baselines. QDAC-MB 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.

The implementation of QDAC-MB is based on the implementation of DreamerV3. Its successor features network is also implemented as a distributional critic. In our implementation, the Lagrange multiplier network λ is only conditioned on the skill $\mathbf{z}$, as we noticed no difference in performance.

The hyperparameters for SMERL, Reverse SMERL and DCG-ME are 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 QDAC. We noticed a statistically significant decrease in performance for Reverse SMERL and DCG-ME, and no statistically significant change in performance for SMERL. Each algorithm is run until convergence for 10^7 environment steps.

The hyperparameters for DOMiNO are based on the ones suggested by Zahavy et al. (2022) and fine-tuned for our tasks.

Table F.4. QDAC hyperparameters

Parameter	Value
Actor network	[512, 512, $\|\mathcal{A}\|$]
Value function network	[512, 512, 1]
Successor feature network	[512, 512, $\|\mathcal{Z}\|$]
Lagrange multiplier network	[512, 512, 1]
Real environment exploration batch size	256
Total timesteps	1×10^7
Optimizer	Adam
Learning rate	3×10^{-4}
Replay buffer size	2×10^6
Discount factor γ	0.99
Target smoothing coefficient τ	0.005

Table F.5. QDAC-MB hyperparameters

Parameter	Value
Actor network	[512, 512, $\|\mathcal{A}\|$]
Value function 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 F.6. DCG-ME hyperparameters

Parameter	Value
Number of centroids	1024
Evaluation batch size b	256
Policy networks	[128, 128, $\|\mathcal{A}\|$]
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, $\|\mathcal{A}\|$]
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 F.7. QD-PG 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	86
Number of Quality-PG variations	85
Number of Diversity-PG variations	85
GA variation param. 1 σ_1	0.005
GA variation param. 2 σ_2	0.05
Critic network — task reward	[256, 256, 1]
Critic network — diversity reward	[256, 256, 1]
TD3 batch size N	100
Critic training steps — task reward	300
Critic training steps — diversity reward	300
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
Archive acceptance threshold	0.1
Archive maximal size	10^4
K-nearest neighbors	3

Table F.8. DOMiNO hyperparameters

Parameter	Value
Actor network	[256, 256, $\lvert\mathcal{A}\rvert$]
Critic network — task reward	[256, 256, 1]
Critic network — diversity reward	[256, 256, 1]
Online batch size	60
Batch size	600
Optimizer	Adam
Learning rate	1×10^{-4}
Replay buffer size	1.5×10^6
Discount factor γ	0.99
Target smoothing coefficient τ	0.005
Number of policies	10
Optimality ratio α	0.9
$\widetilde{v}_{\pi^i}^{\text{avg}}$ decay factor $\alpha_{\widetilde{v}_d^{\text{avg}}}$	0.9
$\widetilde{\boldsymbol{\psi}}_{\pi^i}^{\text{avg}}$ decay factor $\alpha_{\widetilde{\boldsymbol{\psi}}_d^{\text{avg}}}$	0.99
Lagrange learning rate	1×10^{-3}
Lagrange optimizer	Adam

Table F.9. 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 F.10. Reverse 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 F.11. No-SF 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 F.12. Fixed-λ 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
Lambda λ	0.5

Table F.13. 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 λ	0.66