NEUROEVOLUTION IS A COMPETITIVE ALTERNATIVE TO REINFORCEMENT LEARNING FOR SKILL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Reinforcement Learning (RL) has emerged as a powerful paradigm for training neural policies to solve complex control tasks. However, these policies tend to be overfit to the exact specifications of the task and environment they were trained on, and thus do not perform well when conditions deviate slightly or when composed hierarchically to solve even more complex tasks. Recent work has shown that training a mixture of policies, as opposed to a single one, that are driven to explore different regions of the state-action space can address this shortcoming by generating a diverse set of behaviors, referred to as skills, that can be collectively used to great effect in adaptation tasks or for hierarchical planning. This is typically realized by including a diversity term - often derived from information theory - in the objective function optimized by RL. However these approaches often require careful hyperparameter tuning to be effective. In this work, we demonstrate that less widely-used neuroevolution methods, specifically Quality Diversity (QD), are a competitive alternative to information-theory-augmented RL for skill discovery. Through an extensive empirical evaluation comparing eight state-of-the-art methods on the basis of (i) metrics directly evaluating the skills' diversity, (ii) the skills' performance on adaptation tasks, and (iii) the skills' performance when used as primitives for hierarchical planning; QD methods are found to provide equal, and sometimes improved, performance whilst being less sensitive to hyperparameters and more scalable. As no single method is found to provide near-optimal performance across all environments, there is a rich scope for further research which we support by proposing future directions and providing optimized open-source implementations.

1 INTRODUCTION

In the past decade, Reinforcement Learning (RL) has shown great promise at tackling sequential decision making problems in a generic fashion, leading to breakthroughs in many fields such as games (Silver et al., 2017), robotics (Andrychowicz et al., 2020), and control in industrial settings (Degrave et al., 2022). However, neural policies trained with RL algorithms tend to be over-tuned to the exact specifications of the tasks and environments they were trained on. Even minor disturbances to the environment, to the starting state, or to the task definition can incur a significant loss of performance (Kumar et al., 2020; Pinto et al., 2017). A standard approach to improve generalization is to introduce more variations during training (Tobin et al., 2017) but this assumes we can foresee all possibilities, which is not always true in the real world. Even in settings where this is feasible, introducing a wide spectrum of variations will make the problem harder to solve and the resulting policy may not perform as well in the nominal case. Another approach consists in co-training an adversarial agent whose task is to perturb the environments so as to minimize the policy's performance (Pinto et al., 2017). However, adversarial methods are notoriously unstable in Deep Learning (Arjovsky & Bottou, 2017) and can also compromise performance in the nominal scenario.

To improve robustness without explicitly identifying all possible variations, jointly training multiple policies to solve the same task in diverse ways has emerged as a promising line of work in the RL literature (Kumar et al., 2020). To motivate the approach, consider the problem of learning a policy to control the joints of a legged robot with the goal of running as fast as possible. Any damage to the

robot legs might affect an optimal policy's ability to make the robot run fast, if at all. Yet, many of the slightly sub-optimal policies to the original problem (e.g. a policy making the robot hop using only one leg) would perform equally well in this perturbed setting. Two seemingly-opposed main lines of work have been pursued to maximize both performance and diversity in a collection of policies.

RL-rooted approaches Eysenbach et al. (2018); Sharma et al. (2019); Kumar et al. (2020) introduce a randomly-generated latent variable and parametrize the policy to be a function of the state as well as this latent variable. At training time, the latent variable is drawn from a known static distribution and fed as an input alongside the state to the policy, effectively defining a mixture of policies. To encourage diversity among these induced policies, a term derived from information-theory that depends both on the policy parameters and the latent variable is added to the objective function (hereinafter referred to as fitness function). This term is typically formulated as the mutual information between the latent variable and a subset of the policy's trajectory, possibly conditioned on some observations from the past.

Neuroevolution-rooted approaches instead stem from the subfield of Quality Diversity (QD) optimization (Pugh et al., 2016; Cully & Demiris, 2017; Chatzilygeroudis et al., 2021) and combine the tools developed in this space with RL algorithms to get the best of both worlds (Nilsson & Cully, 2021; Pierrot et al., 2022). QD optimization aims at generating and maintaining large and diverse collections of solutions, as opposed to a single optimal solution in Optimization Theory, by imitating the natural evolution of individuals competing for resources in their respective niches. In comparison to traditional Evolutionary Strategies, QD algorithms explicitly use a mapping from solution to a vector space, referred to as behavior descriptor space, to characterize solutions and maintain a data structure, a *repertoire*, filled with high-performing solutions that cover this space as much as possible.

Evolutionary Strategies (possibly hybridized with RL algorithms) have proven to be a competitive alternative to RL algorithms for many common sequential-decision making problems (Pierrot et al., 2022; Salimans et al., 2017). Hence, it is natural to believe that QD algorithms could also be competitive with information-theory-augmented RL approaches to generate diverse populations of high-performing policies in similar settings. Yet, QD approaches remain neglected in skill-discovery studies (Kumar et al., 2020), perhaps because they lack the sample-efficiency of state-of-the-art RL algorithms, sometimes requiring two orders of magnitude more interactions with the environment to solve a task (Pierrot et al., 2022). While this is a significant shortcoming for real-world applications that cannot be accurately described by a computational model, simulators are readily available for many applications. Additionally, when the simulator and the algorithm are implemented using modern vectorized frameworks such as JAX (Bradbury et al., 2018) and BRAX (Freeman et al., 2021), evolutionary approaches are competitive with RL approaches in terms of total training time on an accelerator in spite of the low sample-efficiency of these methods (Lim et al., 2022).

The contributions of this work are the following. **(1.)** We provide extensive experimental evidence that QD methods are competitive with RL approaches for skill discovery in terms of performance given fixed compute and training time budgets and hyperparameter sensitivity. Specifically, using environments taken from the QD and RL literature, we compare eight state-of-the-art skill-discovery methods from the RL and QD world on the basis of (i) metrics directly evaluating the skills' diversity, (ii) the skills' performance on adaptation tasks, and (iii) the skills' performance when used as primitives for hierarchical planning. **(2.)** We open source efficient implementations of all environments and algorithms. Armed with these, running any of the experiments, some of which require hundreds of millions of environments steps, takes only 2 hours on a single affordable accelerator. **(3.)** We provide a detailed analysis of the strengths and weaknesses of all methods, we show that no single method outperforms all others on all environments and we identify future research directions.

2 Preliminaries and Problem Statement

We consider sequential decision making problems formulated as Markov Decision Processes (MDPs) and defined by $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\gamma \in [0, 1]$ is the discount factor, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward signal and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function. Policies are assumed to be implemented by neural networks. We consider two cases. **(1.)** *A mixture of policies is conditioned on a latent variable* $z \in \mathcal{Z}$, in which case it is denoted by $\pi_\theta : \mathcal{S} \times \mathcal{Z} \to \mathcal{D}(\mathcal{A})$ where θ denotes the neural network weights and $\mathcal{D}(\mathcal{A})$ is the space of distributions over the action space. The collection of policies Π is obtained by enumerating all possible values for the latent

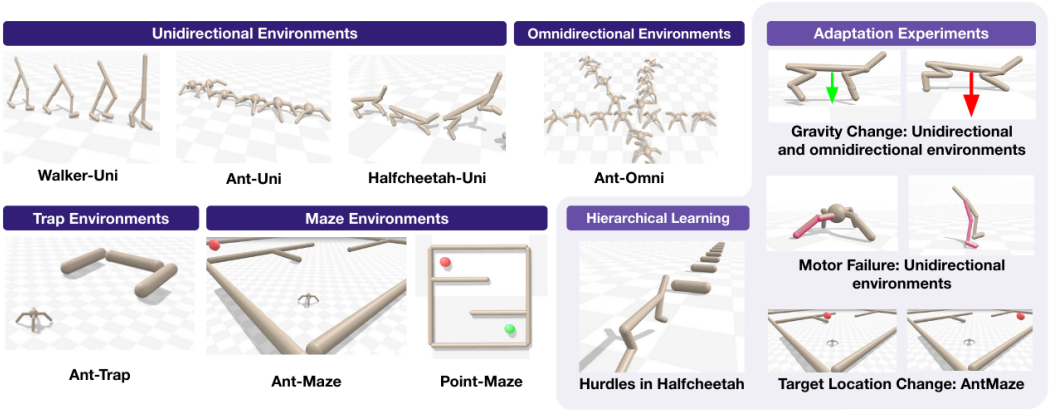

Figure 1: Illustrations of environments used for benchmarking (left) and perturbations applied to them for adaptation and hierarchical learning experiments (right). In UNIDIRECTIONAL environments, policies are trained to make the robots run forward with diverse gaits. In the OMNIDIRECTIONAL environment, the goal is to make a legged robot move on the 2D plane while minimizing the control energy. In ANT-TRAP, a robot learns to run in the forward direction while avoiding a trap. In ANT-MAZE and POINT-MAZE, a robot must reach a target position in a maze. Once trained over these tasks, policies are evaluated on perturbed versions of the tasks and environments.

variable. **(2.)** *Policies share the same neural network architecture but have their own sets of parameters.* In this case, Π corresponds to the collection of N policies $\pi_{\theta_i} : \mathcal{S} \to \mathcal{D}(\mathcal{A}),\ i \in [1, N]$. We denote the trajectory of a single policy π in the environment by $\tau \in \Omega$. The expected sum of rewards of a policy π, referred to as its fitness, is $F(\pi) = \mathbb{E}_\tau \sum_t \gamma^t \mathcal{R}(s_t, a_t)$.

In this work, compared to traditional RL, we consider a second objective: the policies in the collection Π, referred to as *skills*, must not only be high-performing but also diverse as a whole. The notion of diversity can be defined in various ways. In the RL literature, diversity is often measured by looking at the volume spanned by components of the trajectories induced by the policies, such as the sequences of visited states (Parker-Holder et al., 2020). In the QD literature, diversity is measured in a so-called *behavior descriptor space* Φ using a mapping $\phi : \Omega \to \Phi$ that is either hand-engineered or progressively refined in an unsupervised fashion. This approach is more general as it includes as a special case the first one but also allows to leverage expert knowledge on the task at hand, which is often available. We use the notion of behavior descriptor space in order to quantitatively compare the diversity of policies generated by RL and QD methods. Specifically, we define a bounded behavior descriptor space for each MDP, that may differ from the ones used internally by QD methods, which is discretized into a discrete tessellation of cells. Policies from Π are inserted into the tessellation as a function of their behavior descriptors which enables us to use diversity metrics developed in the QD literature. This process is made precise in Section 5.2.

3 DIFFERENT SOLUTIONS TO THE SAME PROBLEM

Information-theory-augmented RL methods. Most methods aiming to maximize diversity in the RL literature, such as DIAYN (Eysenbach et al., 2018) and DADS (Sharma et al., 2019), fall into the class of latent-conditioned policies introduced in Section 2. They strive to maximize the Mutual Information (MI) between a latent variable z, which is drawn randomly from a pre-specified distribution, and some components of the trajectories induced by the conditioned policy. The MI maximization is approximately carried out through the introduction of a discriminator function $z \mapsto q(z|s)$ given a state $s \in \mathcal{S}$ (resp. a dynamics function $z \mapsto q(s_{t+1}|s_t, z)$ given the current state s_t and the next state s_{t+1}) for DIAYN (resp. DADS) trained separately to differentiate the latent variables that lead to visit different states. The latent-conditioned policy is optimized using an off-policy model-free RL algorithm by taking the reward as the discriminator value, referred to as *intrinsic reward*.

DIAYN and DADS are unsupervised algorithms as they do not aim to maximize a reward function associated with a specific task. Recent work in MI-based RL, namely SMERL (Kumar et al., 2020), has extended DIAYN and DADS to supervised settings where both diversity and performance w.r.t. a

reward function should be optimized. They formulate the problem as a Constrained MDP (CMDP) where the objective is to maximize the diversity of a mixture of policies, while also constraining the learned policies to be close to optimality w.r.t. the reward function. By solving these CMDPs, they obtain a collection of policies that are both diverse and high-performing. SMERL can use either DIAYN or DADS internally to generate collections of policies, two variants we refer to as SMERL(DIAYN) and SMERL(DADS). An alternative approach to SMERL simply consists in summing the task reward and the intrinsic reward and using the original DIAYN and DADS algorithms (Kumar et al., 2020; Osa et al., 2021; Gaya et al., 2021). This approach is very similar to Hausman et al. (2018). We refer to these variants as DIAYN+REWARD and DADS+REWARD. For clarity's sake, we only report the performance of the original DIAYN (resp. DADS) algorithm in the Appendix as it turns out to be significantly outperformed by DIAYN+REWARD (resp. DADS+REWARD) in our experiments.

QD methods fall into the second category of policies introduced in Section 2. These methods explicitly evolve and maintain a discrete repertoire of independent policies. QD algorithms traditionally rely on Evolutionary algorithms to incrementally update policies almost independently. Such techniques have the advantage of being easily parallelizable and can be used with non-differentiable policies and controllers (Tang et al., 2020). They have been shown to be viable alternatives to RL algorithms (Salimans et al., 2017).

MAP-ELITES is one of the most widespread QD algorithms. This method assumes that the behavior descriptor space has been discretized into a finite tessellation of cells, which together define a *repertoire* where policies are stored. This algorithm proceeds by iterations. At each iteration, a batch of policies is sampled uniformly from the repertoire and copied. Mutations and crossover operations are applied to them to obtain a new batch of policies which are evaluated in the environment to compute their fitnesses and behavior descriptors. For any of these policies, if the cell in the repertoire corresponding to its behavior descriptor is empty, then it is copied into this cell. Otherwise, the new policy replaces the current incumbent only if its fitness is higher than the current incumbent's fitness.

The exact choice of mutation and crossover operators significantly affects the performance of QD algorithms. Choices that rely exclusively on random perturbations typically require a significant number of iterations. To remedy this problem, several works have combined first-order optimization methods with Evolutionary algorithms to guide the parameter search (Colas et al., 2020; Pierrot et al., 2022). In particular, PGA-MAP-ELITES (Nilsson & Cully, 2021) is a hybrid algorithm that builds on the MAP-ELITES framework to select, mutate, and replace policies in the repertoire but further introduces a policy-gradient mutation operator which is used alongside the genetic operator. As part of every MAP-ELITES iteration, PGA-MAP-ELITES collects the experience of all evaluated policies and stores it into a replay buffer in order to train a pair of critics using the TD3 algorithm (Fujimoto et al., 2018). These critics are used to compute the policy-gradient updates applied to the policies.

While MAP-ELITES and PGA-MAP-ELITES are provided with a pre-specified behavior descriptor space, it is defined in an unsupervised fashion in a recent extension to MAP-ELITES dubbed AURORA (Cully, 2019; Grillotti & Cully, 2022a). This is achieved by training an autoencoder to reconstruct the trajectories collected in the environments and defining the behavior space as the latent space of the autoencoder. In this paper, we introduce a new unsupervised method which extends both AURORA and PGA-MAP-ELITES. This method, referred to as PGA-AURORA, use the same policy-gradient mutation operator as PGA-MAP-ELITES and the same unsupervised technique as AURORA to define the behavior descriptor space.

4 Related Work

Historically, QD algorithms originated from Evolutionary algorithms, which also implement selection, mutation, and evaluation of populations of solutions. Novelty Search (NS) (Lehman & Stanley, 2011a) first introduced the idea of searching for novelty alone. In a sense, NS is similar to DIAYN and DADS as it is an exploration-only algorithm. Novelty Search with Local Competition (NSLC) (Lehman & Stanley, 2011b) and MAP-ELITES (Mouret & Clune, 2015) are two of the most popular QD algorithms. NSLC builds off the NS algorithm and, in contrast to MAP-ELITES, it maintains an unstructured repertoire of solutions that are selected for their performance when compared to similar individuals. Relying on the same diversity-seeking strategies, QD-ES algorithms, which combine QD and Evolutionary Strategies (ES), such as NSR-ES and NSRA-ES, have been successfully applied to challenging continuous control environments (Conti et al., 2018). Aiming for improved sample

efficiency, Colas et al. (2020) developed ME-ES which optimizes both for quality and diversity using MAP-ELITES and two ES populations. Similarly, CMA-ME (Fontaine et al., 2020) uses a covariance matrix adaptation technique to model and update a population distribution to sample individuals from and also maintains a population of varied emitters to generate solutions. More recently, some authors proposed to incorporate a policy-gradient component in the evolutionary framework (Nilsson & Cully, 2021; Pierrot et al., 2022; Tjanaka et al., 2022). Inspired by PGA-MAP-ELITES, the authors of QD-PG (Pierrot et al., 2022) introduced a diversity policy gradient to make diversity information available at the timestep level.

Multiple methods from the RL literature build an intrinsic reward with MI to learn a diverse set of skills. They are similar in spirit to DIAYN and can be adapted, along the same lines as what is done for SMERL, to also use extrinsic rewards. VIC (Gregor et al., 2016) originally introduced MI for skill discovery. Most recent efforts in the skill-discovery RL literature focus on ways to extend DIAYN and DADS to improve the state coverage of the learned skills using several methods (Durugkar et al., 2021; Campos et al., 2020), such as contrastive learning (Laskin et al., 2022) and adding constraints to the skills (Hansen et al., 2021; Kamienny et al., 2021). Nevertheless, Zahavy et al. (2021) introduces an original approach where the intrinsic rewards are defined based on successors features rather than MI to induce diversity. We choose DIAYN, DADS, and SMERL as baselines in our experiments as they are the most widely used RL methods for skill discovery at the time of writing. There also exist methods based on tools other than MI that strive to evolve a diverse population of RL agents, such as Zhou et al. (2022); Parker-Holder et al. (2020); Zhang et al. (2019); Masood & Doshi-Velez (2019), but these studies mostly consider diversity as a means to increase performance w.r.t. the rewards, whereas DIAYN, SMERL, and DADS are evaluated on pure diversity, on adaptation tasks, or in hierarchical learning settings.

5 EXPERIMENTS

In this section, we present the frameworks used to compare the methods listed in Section 3 and analyze the results. First, in Section 5.1, we introduce the environments and tasks upon which the frameworks are built. In Section 5.2, we leverage tools and metrics from the QD literature to directly evaluate the diversity of the solutions found by each method in the process of solving the tasks. In Section 5.3, we investigate how diversity translates into robustness with few-shot adaptation experiments inspired from the RL literature. Finally, in Section 5.4, we evaluate the policies found by each method when used as primitives in a hierarchical learning setting.

5.1 ENVIRONMENTS

In order not to favor any method in the study, we use three different categories of environments and tasks: (i) continuous control locomotion tasks from the RL literature, (ii) exploration tasks with deceptive rewards from the QD literature, and (iii) hybrid tasks mixing both types as a middle ground. All environments are illustrated on Figure 1.

Low-dimensional exploration task with deceptive rewards. Following several works from both the skill-discovery RL and QD communities (Eysenbach et al., 2018; Kamienny et al., 2021; Campos et al., 2020), we consider a simple navigation environment where a point is controlled to navigate on a 2D plane to a specified target position. To turn the task into a challenging exploration problem, we introduce a maze through which the point must navigate to reach the target position. We refer to this task as POINT-MAZE.

High-dimensional locomotion tasks. We consider three locomotion tasks, namely ANT, HALFCHEETAH and WALKER, often used in the RL community to evaluate algorithms, including in RL works focused on diversity-seeking strategies (Nilsson & Cully, 2021; Eysenbach et al., 2018; Fujimoto et al., 2018). In these environments, the problem is to move legged robots by applying torques to their joints through actuators. These tasks are challenging for evolutionary algorithms: they typically require orders of magnitude more environment interactions than RL methods to evolve competitive policies (Colas et al., 2020; Pierrot et al., 2022). We define two subsets of tasks for these environments. The first subset, referred to as UNIDIRECTIONAL locomotion tasks and with the reward signal defined as the x-axis velocity minus the energy consumption, includes ANT-UNI, WALKER-UNI, and HALFCHEETAH-UNI. The second subset, referred to as OMNIDIRECTIONAL locomotion tasks and with the reward signal defined as the opposite of the energy consumption, includes ANT-OMNI.

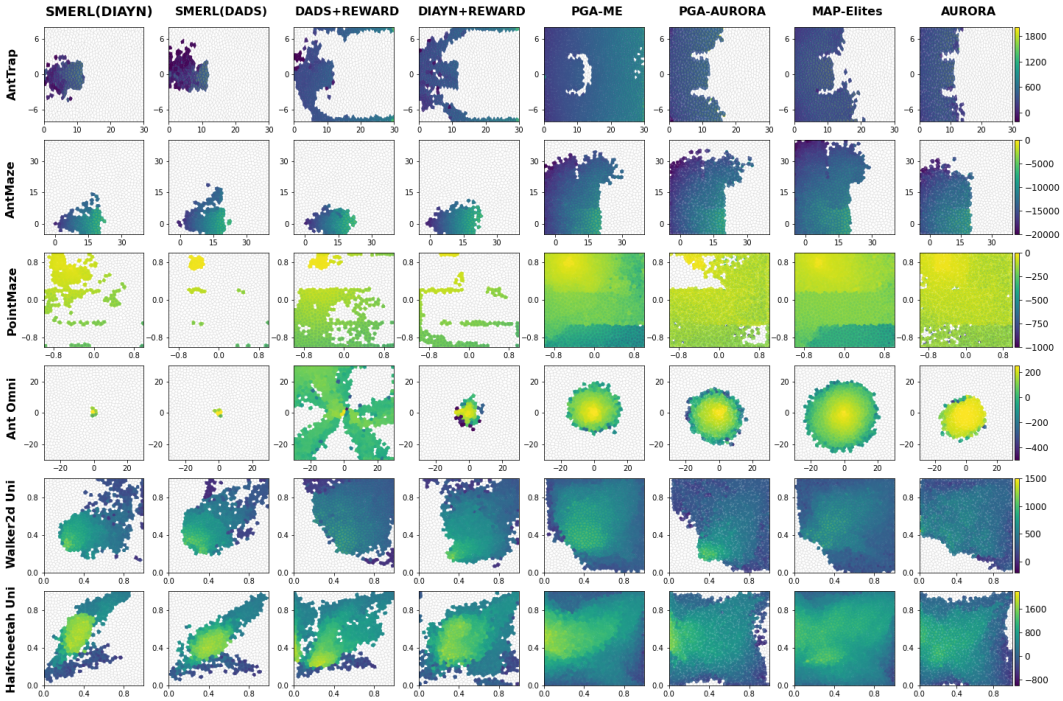

Figure 2: Visualizations of repertoires post training. Each repertoire divides the behavior descriptor space into 1024 cells. Cells are left blank when no policy with the corresponding behavior descriptor has been found and are otherwise colored as a function of the best fitness achieved across policies with a matching behavior descriptor. For ANT-TRAP, ANT-MAZE, POINT-MAZE, and ANT-OMNI, the behavior descriptor is the (x, y) position reached at the end of an episode: these plots highlight explored areas of the 2D plane. For HALFCHEETAH-UNI and WALKER-UNI, the descriptor is the fraction of time the robots' feet were touching the ground during an episode (one dimension per foot).

Mixing locomotion and exploration tasks. Following Pierrot et al. (2022); Parker-Holder et al. (2020); Frans et al. (2018); Shi et al. (2020), we consider two additional hybrid tasks built upon ANT that exhibit both continuous control and exploration challenges, making them particularly difficult even for state-of-the-art evolutionary and RL algorithms. In ANT-TRAP, the ant has to run forward while initially facing a trap consisted of three walls that prevents it from moving forward along the x-axis direction indefinitely. In ANT-MAZE, the ant is initially placed in a maze and the goal is to reach a specified target position.

5.2 EXPERIMENTS DIRECTLY EVALUATING DIVERSITY

On the one hand, metrics quantifying the amount of diversity in a collection of policies are lacking in the MI RL literature. Success in inducing diversity is instead typically assessed indirectly through performance on adaptation or hierarchical learning experiments where either the tasks are modified or new ones are introduced (Eysenbach et al., 2018; Kumar et al., 2020; Sharma et al., 2019). On the other hand, diversity metrics were introduced very early in the QD literature and have been refined over time since QD algorithms are primarily assessed on this basis. Hence, we use metrics developed in the QD literature and extend them to MI RL approaches to be able to compare the methods on the same ground. Noting that almost all QD metrics crucially rely on the definition of a behavior descriptor space (to characterize solutions) and the availability of a repertoire of solutions indexed with respect to this space for all methods, we specify behavior descriptors for all tasks and extend the concept of repertoires to MI RL methods.

Behavior descriptor spaces. For POINT-MAZE, ANT-TRAP, ANT-MAZE, and ANT-OMNI, following several prior QD and RL works (Pierrot et al., 2022; Hansen et al., 2021), the behavior descriptor of a policy is defined as the (x, y) position of the robot's center of gravity at the end of an evaluation episode. For UNIDIRECTIONAL tasks, the i-th component of the behavior descriptor vector is defined as the proportion of time during which the i-th foot of the robot is in contact with the ground in an

Table 1: Maximum fitness and QD score at the end of the training phase (median over 5 seeds). Additional statistics (mean, standard deviation, and interquartile mean) are included in the Appendix.

	ANT-TRAP	ANT-MAZE	POINT-MAZE	ANT-UNI	ANT-OMNI	WALKER-UNI	HALFCHEETAH-UNI
				Maximum fitness			
SMERL(DIAYN)	4.49 e2	-7.88 e3	-4.00 e1	**1.71 e3**	**2.49 e2**	1.19 e3	1.89 e3
SMERL(DADS)	4.48 e2	**-7.62 e3**	-8.10 e1	1.61 e3	**2.49 e2**	1.17e3	1.92 e3
DADS+REWARD	**1.28 e3**	-7.81 e3	-8.10 e1	1.24 e3	2.38 e2	6.30 e2	1.89 e3
DIAYN+REWARD	9.42 e2	-8.13 e3	-4.50 e1	1.34 e3	2.29 e2	1.00 e3	**2.13 e3**
PGA-MAP-ELITES	9.77 e2	-9.32 e3	**-2.30 e1**	1.45 e3	**2.49 e2**	**1.22 e3**	1.75 e3
PGA-AURORA	4.72 e2	-9.27 e3	-6.3 e1	1.44 e3	**2.49 e2**	1.05 e3	1.47 e3
MAP-ELITES	3.81 e2	-1.01 e4	-2.5 e1	4.25 e2	**2.49 e2**	6.87 e2	1.23 e3
AURORA	4.05 e2	-1.10 e4	-2.35 e1	5.05 e2	**2.49 e2**	5.71 e2	1.21 e3
				QD score			
SMERL(DIAYN)	1.02 e5	7.84 e5	38.9 e4	6.20 e4	1.95 e3	3.01 e5	7.92 e5
SMERL(DADS)	9.88 e4	8.22 e5	3.57 e3	0.83 e5	4.15 e3	2.69 e5	7.51 e5
DADS+REWARD	2.42 e5	9.12 e5	0.42 e4	2.88 e5	3.05 e5	3.89 e5	8.31 e5
DIAYN+REWARD	2.27 e5	9.18 e5	4.04 e4	1.29 e5	4.00 e4	4.92 e5	1.68 e6
PGA-MAP-ELITES	**7.89 e5**	2.74 e6	3.92 e5	**9.19 e5**	1.58 e5	**8.23 e5**	2.98 e6
PGA-AURORA	4.40 e5	2.59 e6	2.88 e5	7.33 e5	1.69 e5	5.10 e5	2.39 e6
MAP-ELITES	4.53 e5	**2.98 e6**	4.22 e5	9.08 e5	3.04 e5	6.45 e5	**3.05 e6**
AURORA	3.54 e5	2.18 e6	**4.58 e5**	5.82 e5	**4.00 e5**	4.59 e5	2.45 e6

evaluation episode. This is a simple but effective way to capture the gait of a robot as demonstrated in Cully et al. (2015) where it is used to help robots adapt after having some of their legs damaged.

We stress that, in our experiments, MAP-ELITES and PGA-MAP-ELITES make use of the same behavior descriptors used to compute the QD metrics, which induces a bias in the analysis. This bias does not affect AURORA and PGA-AURORA as these methods build their own behavior descriptors in an unsupervised fashion. Nevertheless, for fairness, we embed the choice of the behavior descriptor space into MI RL methods whenever possible. Hence, in ANT-MAZE, ANT-TRAP, ANT-OMNI and POINT-MAZE, MI RL algorithms are provided with the prior that diversity w.r.t. the (x, y) position of the robot is what needs to be optimized. Implementation details are deferred to the Appendix.

Passive repertoires for RL methods. The MI RL methods considered in this paper do not maintain a repertoire to store diverse solutions. For fairness, and since increasing the size of the discrete latent space to match the size of the QD repertoires makes these methods unstable (Eysenbach et al., 2018), we fill a repertoire on the side during training with all the intermediate policies found by these methods, similarly to what is done internally in QD methods. The repertoires of size 1024 are identically defined across methods using Centroidal Voronoi Tessellations (Vassiliades et al., 2017).

QD Metrics. Armed with behavior descriptor spaces for all environments and repertoires for all methods, we compute three metrics that are often used in the QD community to track the performance of a collection of solutions: (i) the **maximum fitness** of the repertoire, computed as the maximum fitness attained across the solutions stored in the repertoire, (ii) the **coverage** of the repertoire, computed as the number of cells (behavior niches) that have been filled, and (iii) the **QD score**, computed as the sum of fitnesses attained by the solutions stored in the repertoire.

Results and analysis. All methods are trained for exactly two hours on a Quadro RTX 4000 GPU for each environment described in Section 5.1. Each experiment is repeated with 5 different seeds and the final repertoire is evaluated using the aforementioned QD metrics. The maximum fitness and QD score are reported in Table 1. Out of the 5 seeds, the best - in terms of QD score - final repertoires obtained at the end of the training phases are depicted on Figure 2 for all methods and environments.

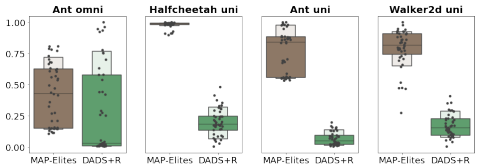

Figure 3: Distribution of QD scores obtained over a range of hyperparameters on UNIDIRECTIONAL tasks and ANT-OMNI.

First, observe that, setting aside ANT-OMNI where DADS+REWARD has managed to largely explore the environment, QD methods outperform the MI RL methods for all environments and tasks if we base the comparison on QD scores and coverage only. This is a particularly desirable property in ANT-MAZE (resp. ANT-TRAP) because reaching a high coverage of the repertoire means that the method has been able to explore the maze (resp. get past the trap) effectively given that the behavior descriptor vector is defined as the final (x, y) position of the robot.

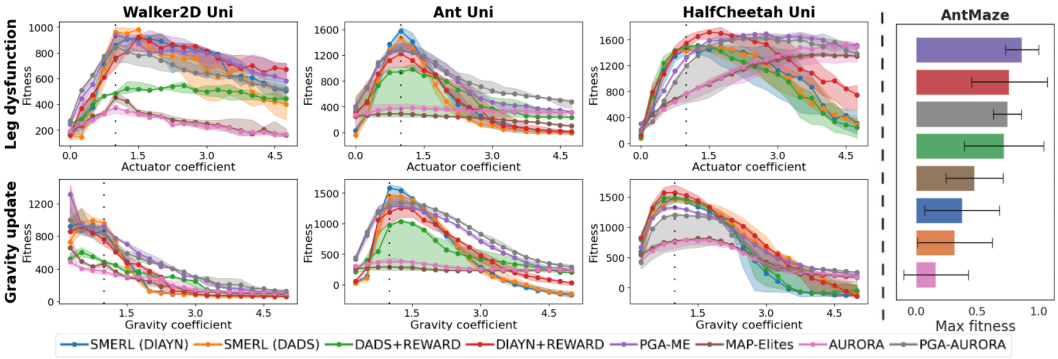

Figure 4: Maximum fitness of the methods when confronted to (left) a perturbed environment and (right) a modified task (median and interquartile range on 5 seeds, additional statistics are in the Appendix). We observe varying resilience levels across methods. In particular, PGA-MAP-ELITES and PGA-AURORA adapt better to extreme environment perturbations in UNIDIRECTIONAL environments.

Second, note that no single method outperforms all others in terms of maximum fitness. PGA-MAP-ELITES ranks first on POINT-MAZE and WALKER-UNI but is always outperformed by an MI RL method in the others. Additional observations are included in the Appendix.

While the excellent performance of DADS+REWARD on ANT-OMNI is an impressive achievement in itself, we run an additional set of experiments to investigate whether this may be the result of a lucky guess on the exact choice of hyperparameters. Specifically, we run the same experiments on ANT-OMNI and UNIDIRECTIONAL environments multiple times for DADS+REWARD and MAP-ELITES, sampling from a grid of reasonable hyperparameters values for both methods (exact details are deferred to the Appendix). The medians and quartiles of the QD-score distributions obtained are shown on Figure 3. We observe that the performance of DADS+REWARD is very sensitive to the exact choice of hyperparameters and that the median performance of MAP-ELITES is always significantly better than the median performance of DADS+REWARD on all tasks.

5.3 FEW-SHOT ADAPTATION EXPERIMENTS

While metrics borrowed from the QD literature provide a first basis for comparison, they suffer from an inherent bias in favor of QD methods as discussed in Section 5.2. Following prior work that analyze the resilience of RL methods to changes in the environment or task definition (Kumar et al., 2020), we design two series of few-shot adaptation experiments that do not favor any particular method a priori. Few-shot here refers to the fact that no re-training is allowed upon modification of the task or the environment, only evaluations of pre-computed policies are allowed. Throughout this section, MAP-ELITES and PGA-MAP-ELITES use the behavior descriptor spaces introduced in Section 5.2.

Environment adaptation experiments. In the first series of experiments, we modify a property of the UNIDIRECTIONAL environments after training, as illustrated on Figure 1. First, we vary the gravity coefficient in a neighborhood grid of size 20 around the default value (i.e. 1). We jointly refer to these experiments as the gravity-update setting. Second, we vary the actuators input-to-torque coefficients for a whole leg of the robot in a neighborhood grid of size 20 around the default value (i.e. 1). We jointly refer to these experiments as the leg-dysfunction setting.

Task adaptation experiments. In the second series of experiments, we modify the task of the ANT-MAZE environment by moving the target position after training. This has no impact on the physics of the environment but directly modifies the reward signal. New target positions are sampled uniformly on the 2D plane delimited by the maze.

For both series of experiments, we train all methods on the nominal versions of the environments for exactly two hours and re-evaluate the final output of the training phase on the modified environment or task. Specifically, for each method, we enumerate all their learned skills, evaluate them one hundred times in the modified environment, and pick the one that performs best in terms of median fitness across evaluations. Each experiment is repeated with 5 different seeds and the maximal fitnesses obtained are reported on Figure 4 for the environment and task adaptation experiments.

Results and analysis. First, note that, on the environment adaptation experiments, PGA-MAP-ELITES and PGA-AURORA are performing on par with MI RL methods in the case of small perturbations, i.e. when the coefficient is perturbed by less than 50%. Methods that do not leverage policy gradients (i.e. MAP-ELITES and AURORA) are outperformed on all tasks in this regime.

Second, QD methods are more resilient than MI RL when confronted with significant perturbations. For instance, in the gravity-update setting, the relative performance loss is limited to 40% for PGA-MAP-ELITES when the gravity coefficient is divided by 4 compared to at least 60% for MI RL methods. As a testament of the resilience of QD methods, note that the best performing method is a QD one for 10 out of the 12 most extreme scenarios. Similar observations hold for ANT-MAZE where PGA-MAP-ELITES and PGA-AURORA perform as well as DIAYN+REWARD and DADS+REWARD.

Third, it is remarkable that PGA-AURORA and AURORA perform almost as well as their QD supervised counterparts in these experiments. This shows that unsupervised behavior descriptor learning is really effective for QD methods. Furthermore, observe that PGA-AURORA adapt with a similar loss in fitness than MI RL methods (recall that all of these methods have access to the exact same priors).

5.4 HIERARCHICAL LEARNING EXPERIMENTS

To further assess the skills discovered by the methods during training on a nominal environment, we introduce a hierarchical learning setting inspired from the RL literature (Eysenbach et al., 2018). In HALFCHEETAH-HURDLES, hurdles are spaced out along the x-axis direction in front of the robot's initial position in HALFCHEETAH, see Figure 1. The robot has to jump over them to move forward, the reward definition remaining unchanged. Just like in Eysenbach et al. (2018), we train a PPO meta-controller (Schulman et al., 2017) to choose a skill, which is to be unrolled in the environment for the next ten consecutive timesteps, among the ones learned by a given skill-discovery method during training on the nominal version of the environment, see Section 5.2. The meta-controller is trained for 200 million policy interactions, corresponding to 20 million actions of the meta-controller.

Results and analysis. The performance of the meta-controller is shown on Figure 5 for various choices of skill-discovery methods used to generate primitives. By visualizing the meta policies in the environment, we observe that PPO is able to learn a meta-controller that consistently manages to jump over multiple consecutive hurdles when it is trained with primitives derived from PGA-MAP-ELITES and DIAYN+REWARD, which achieve similar performance on this experiment. Surprisingly, all other skill-discovery methods fail to provide primitives that can be composed to make the robot jump over hurdles.

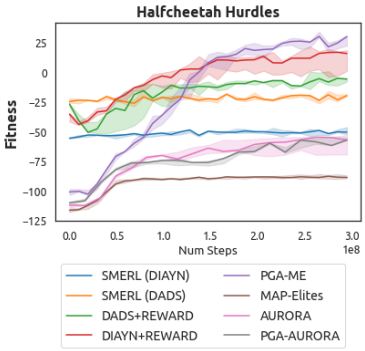

Figure 5: Fitness of the meta-controller as a function of environment steps for all types of skills primitives (median on 5 seeds, more statistics in the Appendix).

6 DISCUSSION AND CONCLUSION

Our experiments show that QD methods are competitive with MI RL methods when the goal is to find a set of diverse and high-performing policies in continuous control and exploration environments. MI RL methods are task-agnostic and data-efficient but struggle to generate large collections of policies. Moreover, they are limited in the set of descriptors priors they can accommodate. QD methods are less data-efficient but can better leverage the vectorization capabilities of modern libraries such as JAX. Moreover, they are significantly less sensitive to the choice of hyperparameters than MI RL methods and their ability to evolve large sets of diverse solutions is a precious tool for hierarchical learning and adaptation applications when the task definition changes or when confronted with extreme environment perturbations. Our benchmarks however show that QD methods do not outperform MI RL methods and vice-versa, which opens the door to promising work directions. We believe that MI RL methods could make use of skill repertoires to improve their memory capacity thereby increasing the pool of available options for downstream applications. Conversely, QD methods leveraging gradients, such as PGA-MAP-ELITES, could benefit from the introduction of intrinsic rewards described in the MI RL literature to guide the search for diversity. Ultimately, we believe that combining the coverage capacity of a repertoire with the data-efficiency of gradient-based techniques is key to develop robust, diverse, and high-performing repertoires of solutions for challenging continuous control.

REFERENCES

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Víctor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giro-i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills, 2020. URL `https://arxiv.org/abs/2002.03647`.

Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pp. 109–135. Springer, 2021.

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

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Advances in neural information processing systems*, pp. 5027–5038, 2018.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pp. 81–89, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361118. doi: 10.1145/3321707.3321804. URL `https://doi.org/10.1145/3321707.3321804`.

Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Gaurav Dixit, Everardo Gonzalez, and Kagan Tumer. Diversifying behaviors for learning in asymmetric multiagent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 350–358, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392372. doi: 10.1145/3512290.3528860. URL `https://doi.org/10.1145/3512290.3528860`.

Ishan Durugkar, Steven Hansen, Stephen Spencer, and Volodymyr Mnih. Wasserstein distance maximizing intrinsic control, 2021. URL `https://arxiv.org/abs/2110.15331`.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pp. 94–102, 2020.

Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. Meta learning shared hierarchies. *Proc. of ICLR*, 2018.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation, 2021. URL https://arxiv.org/abs/2106.13281.

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

Jean-Baptiste Gaya, Laure Soulier, and Ludovic Denoyer. Learning a subspace of policies for online adaptation in reinforcement learning. *arXiv preprint arXiv:2110.05169*, 2021.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, 11 2016. URL https://arxiv.org/abs/1611.07507v1.

Luca Grillotti and Antoine Cully. Unsupervised behaviour discovery with quality-diversity optimisation. *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022a. doi: 10.1109/TEVC.2022.3159855.

Luca Grillotti and Antoine Cully. Relevance-guided unsupervised discovery of abilities with quality-diversity algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 77–85, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450392372. doi: 10.1145/3512290.3528837. URL https://doi.org/10.1145/3512290.3528837.

Steven Hansen, Guillaume Desjardins, Kate Baumli, David Warde-Farley, Nicolas Heess, Simon Osindero, and Volodymyr Mnih. Entropic desired dynamics for intrinsic control. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 11436–11448. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/5f7f02b7e4ade23430f345f954c938c1-Paper.pdf.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rk07ZXZRb.

Pierre-Alexandre Kamienny, Jean Tarbouriech, Sylvain Lamprier, Alessandro Lazaric, and Ludovic Denoyer. Direct then diffuse: Incremental unsupervised skill discovery for state covering and goal reaching, 2021. URL https://arxiv.org/abs/2110.14457.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery, 2022. URL https://arxiv.org/abs/2202.00161.

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

Joel Lehman and Kenneth O Stanley. 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.

Bryan Lim, Maxime Allard, Luca Grillotti, and Cully Antoine. Accelerated quality-diversity for robotics through massive parallelism. *CoRR*, abs/2202.01258, 2022. URL http://arxiv.org/abs/2202.01258.

Muhammad A Masood and Finale Doshi-Velez. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *arXiv preprint arXiv:1906.00088*, 2019.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *ArXiv*, abs/1504.04909, 2015.

Olle Nilsson and Antoine Cully. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, pp. 866–875, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383509. doi: 10.1145/3449639.3459304. URL https://doi.org/10.1145/3449639.3459304.

Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Discovering diverse solutions in deep reinforcement learning by maximizing state-action-based mutual information, 2021. URL https://arxiv.org/abs/2103.07084.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population-based reinforcement learning. In *Neural Information Processing Systems*, 2020.

Thomas Pierrot, Valentin Macé, Félix Chalumeau, Arthur Flajolet, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin-Gilbert. Diversity policy gradient for sample efficient quality-diversity optimization. In Jonathan E. Fieldsend and Markus Wagner (eds.), *GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, USA, July 9 - 13, 2022*, pp. 1075–1083. ACM, 2022. doi: 10.1145/3512290.3528845. URL https://doi.org/10.1145/3512290.3528845.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR, 2017.

Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Longxiang Shi, Shijian Li, Qian Zheng, Min Yao, and Gang Pan. Efficient novelty search through deep reinforcement learning. *IEEE Access*, 8:128809–128818, 2020.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 414–424, 2020.

Bryon Tjanaka, Matthew C. Fontaine, Julian Togelius, and Stefanos Nikolaidis. Approximating gradients for differentiable quality diversity in reinforcement learning, 2022. URL https://arxiv.org/abs/2202.03666.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017. URL https://arxiv.org/abs/1703.06907.

Vassiiis Vassiliades and Jean-Baptiste Mouret. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 149–156, 2018.

Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22(4):623–630, 2017.

Tom Zahavy, Andre Barreto, Daniel J Mankowitz, Shaobo Hou, Brendan O'Donoghue, Iurii Kemaev, and Satinder Baveja Singh. Discovering a set of policies for the worst case reward. *arXiv preprint arXiv:2102.04323*, 2021.

Yunbo Zhang, Wenhao Yu, and Greg Turk. Learning novel policies for tasks. In *International Conference on Machine Learning*, pp. 7483–7492. PMLR, 2019.

Zihan Zhou, Wei Fu, Bingliang Zhang, and Yi Wu. Continuously discovering novel strategies via reward-switching policy optimization. *arXiv preprint arXiv:2204.02246*, 2022.