Diversity in Reinforcement Learning Through the Occupancy Measure

Arno Feiden
Fraunhofer SCAI
Sankt Augustin, Germany
arno.feiden@scai.fraunhofer.de

Jochen Garcke*
Institut für Numerische Simulation
Universität Bonn
Bonn, Germany
garcke@ins.uni-bonn.de

Abstract

Quality-Diversity algorithms search for a set of diverse, high-performing solutions to optimization problems, including reinforcement learning problems. In the case of reinforcement learning problems, Quality-Diversity algorithms foster diversity by differentiating solutions using behaviour descriptors. We introduce a straightforward, powerful approach to generically characterise behaviour using the so-called occupancy measure. Our approach avoids the manual definition of behaviour descriptors and does not rely on further black-box learning.

We investigate four established benchmark problems inspired by robotics, concerning locomotion and maze navigation. To measure the ability to overcome local optima we consider the number of solved configurations and the maximum average score. The use of the occupancy measure is competitive with problem-specific, custom behaviour descriptors and superior to an established generic behaviour descriptor. Our work contributes to the establishment of MAP-Elites as a versatile, robust, out-of-the-box solver for complex non-convex reinforcement learning scenarios.

CCS Concepts

• **Computing methodologies → Evolutionary robotics**; *Continuous space search.*

Keywords

Quality-Diversity, Unsupervised Machine Learning, Robotics, Reinforcement Learning

ACM Reference Format:
Arno Feiden and Jochen Garcke. 2025. Diversity in Reinforcement Learning Through the Occupancy Measure. In *Genetic and Evolutionary Computation Conference (GECCO '25), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3712256.3726337

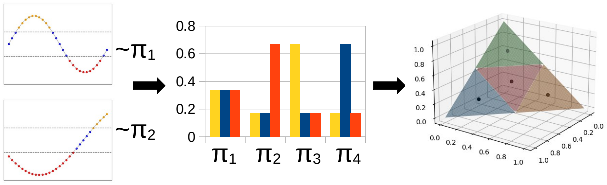

Figure 1: Sketch of the construction of a generic behaviour descriptor by estimating the occupancy measure of a policy. A policy rollout is interpreted as a distribution and added to a niche in a MAP-Elites repertoire.

1 Introduction

There is not one optimal way to optimize complex non-convex problems as posed in reinforcement learning (RL). Quality-Diversity (QD) [5] algorithms address the problem of non-convexity by exploring the solution space with an evolving, diverse population of high-performing solutions. To this end, QD algorithms require a definition of both, the quality of a proposed solution as well as the diversity of a population of solutions. The idea of quality is implicit through the optimization target. Diversity, however, is hard to define conceptually and practically. Carefully constructing the definition of diversity is a prerequisite for using QD algorithms, even if the desired result consists of only one optimal solution. When tackling RL problems with QD, the expression of diversity is determined by behaviour descriptors. They distil the observed behaviour of a potential solution to an expression that fits into the QD framework [1].

In this work, we examine ways to construct behaviour descriptors. Originally they are hand-picked [11] based on a practitioner's understanding of a given problem, but efforts have been made to remove this aspect of supervision through broader setups [20] and unsupervised learning [9]. We introduce a novel strategy to foster diversity in a generic way using the occupancy measure from RL as a behaviour descriptor for a proposed solution. This is done by binning the behaviour space and noting the distribution of visitations of these bins during an episode. This distribution is then treated as a high-dimensional behaviour descriptor and added to an appropriate MAP-Elites repertoire, see fig. 1.

QD algorithms are known to overcome exploration challenges which often stump RL algorithms since gradient-based methods become ensnared in local optima [3]. In the tasks we present, these optima manifest as dead ends or traps. We will measure the success of different strategies to define and encourage diversity by their ability to solve these exploration challenges. Therefore rather than evaluating the performance of an entire population, typical in QD studies [5], we focus on whether the top performer can navigate these challenging exploration tasks, an evaluation typical for RL problems [18].

---

[*] also with Fraunhofer SCAI

GECCO '25, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1465-8/2025/07
https://doi.org/10.1145/3712256.3726337

We specifically focus on the impact of the strategy to define behaviour descriptors within MAP-Elites [11] and its derived methods. We investigate the impact of using different generic behaviour descriptors on the algorithms' ability to navigate problems that require extensive exploration. To this end we use standard benchmark problems inspired by robotics, reflect on their suitability to the task, and gauge the impact of the choice, benchmarking with established ways to construct behaviour descriptors that encourage diversity and a random approach, where no specific diversity is encouraged.

The results indicate that our novel behaviour descriptor based on the occupancy measure is competitive with custom, problem-specific behaviour descriptors in all four instances. It is superior to the established, and next best, generic descriptor in one of the four, while on par for the other three instances. We also argue this strategy is particularly appealing for conceptual reasons: Motivated by theory it uses the full experience of a solution. This facilitates a more thorough theoretical analysis of generalised diversity. The outlined approach requires no prior knowledge about the environment by an end user nor some additional automated learning process. Its simplicity makes it useful as a baseline for further studies and as an out-of-the-box solution for problems where a diverse population of solutions is desirable but not obvious to construct.

2 Background and Related Work

2.1 Quality-Diversity and MAP-Elites

Quality-Diversity (QD) [1, 5] refers to a class of evolutionary algorithms designed to create a population of solutions that are both diverse and high-performing. These algorithms tackle optimization challenges including, but not limited to RL problems. Selecting one or more solutions out of this population may a be method for finding local and global optima, but can also be used to better understand the solution space of a problem ("illumination"), or when building a repertoire of different behaviour itself is the goal, as for damage adaptation [11, 20]. When addressing an underlying optimization problem, the definition of high performance is given by the target of that problem. The definition of diversity, however, needs to be designed by experts specifically to foster the development of a population that can serve as stepping stones [5, 10] towards optimal solutions. Unsupervised methods attempt to eliminate that requirement by using observed behaviours to construct helpful diversity metrics.

MAP-Elites [11] is a prominent QD algorithm that categorizes a given solution with a behaviour descriptor, a function which transforms the observed behaviour of that solution into a usually low-dimensional space, the behaviour space. The image space of that function is partitioned into niches, such that any output of the behaviour descriptor belongs to exactly one niche. The population is defined as up to one solution per such niche. The algorithm is initialized with the creation of some random solutions. These are then iteratively assigned to niches by the behaviour descriptor. If a niche is not already associated with a solution, the solution occupies the niche. If it is already occupied, both the occupying and the new solution compete with regard to the optimization target. The superior one occupies the niche, the inferior is discarded. After random initialization of the population, new solutions are generated as variations to copies of solutions in the population. These new solutions are then again categorized by their behaviour descriptors, and each is assigned to a niche accordingly where they may compete if necessary. The container that holds solutions based on these rules is called the repertoire. This framework spawned several offshoots, mainly by changing the way the new solutions are generated or the way niches are created and solutions associated with them.

The original MAP-Elites strategy [11] constructs a low-dimensional grid in the behaviour space to define niches, and generates new solutions through random mutations. The PGA (Policy Gradient Assisted) MAP-Elites version [12] includes a gradient-based way to create new solutions, analogue to policy gradient methods used in RL. The CVT (Centroidal Voronoi Tessellation) MAP-Elites version [20], samples the behaviour space to construct centroids and defines the niches as a Voronoi tessellation based on these centroids. This permits higher dimensional behaviour descriptors like stacking samples of the trajectory.

2.2 Behaviour Descriptors and Novelty

Behaviour descriptors are critical for the success of diversity-based algorithms, serving as secondary objective functions that encode specific expectations or desired behaviours [5, 10]. The diversity fostered through these descriptors is essential for avoiding early convergence in optimization tasks by helping algorithms overcome local optima, which is important in exploration-hard tasks [3]. However, the effectiveness of a behaviour descriptor hinges on its alignment with the environmental challenges that need to be addressed. If the descriptor does not accurately reflect the obstacles within the environment, the diversity approach may fail to find a viable solution [15]. This is further complicated by the vagueness of this idea of alignment to the task. So, it may be desirable to profit from diversity without having to provide prior knowledge of what kind of diversity is desirable.

A direct approach would add more and more information from the sampled trajectory into the behaviour description. But in a high-dimensional Cartesian grid, the number of niches grows exponentially with the dimension of the behaviour space. CVT MAP-Elites [20] alleviates this problem by defining niches not on a Cartesian grid but by closeness to centroids which are samples in the behaviour space. Especially when sampling many snapshots of the trajectory a prior estimation of expected behaviour is necessary to place these centroids appropriately to construct meaningful niches.

More sophisticated approaches find meaningful low-dimensional representations of the behaviour through means of unsupervised learning and run a MAP-Elites approach on this lower-dimensional space. Here we represent this class of algorithms through AURORA [9] but similar ideas are implemented in TAXONS [14], both of which spawn further developments as in RUDA [8], and STAX [13]. Novelty [10] is a related approach that encapsulates the idea of always looking for solutions different to those encountered prior in the optimization run leading to an open-ended algorithm. In MAP-Elites, unoccupied niches can be interpreted as supporting novelty and occupied niches as novelty with local competition.

3 Method

3.1 Challenges

We investigate the role of behavioural diversity in overcoming local optima. We work with optimization problems that include a clear local optimum difficult to avoid. These problems are sometimes called exploration-hard and usually stump gradient-based methods [3]. An optimizer may have difficulties finding its way out of a local optimum once it gets stuck there as the gradient pulls the solution in a direction which will trap this solution in a local optimum. An extra push to keep exploring after finding a local optimum is required. This extra bit of push can be provided by a QD algorithm.

We specifically look at four established benchmark tasks as implemented in the QDax [2] library. The two-dimensional walker (Walker 2D) is a simple locomotion task that does not seem to feature an obvious local optimum that could trap an optimization algorithm. However, a learner may have to overcome at least two, early local optima: The strategy to fall over gains some reward by rapid forward movement, but fails to collect the reward for being healthy for the whole episode. The strategy to stand still accumulates the reward for being healthy without moving forward. It demonstrates that the trap of local optima is not a constructed issue but pervasive in reinforcement learning problems. The analysis of this environment also examines if and how intensely the effort to develop diversity is detrimental to improving performance after these early optima have been overcome.

Point-Maze is a maze environment in which a particle navigates a square from one point in the lower middle to another in the upper left. Two horizontal walls block the way to the goal. The lower wall has an opening on its left, and the upper wall has an opening on its right. The reward at any timestep is the negative distance to the goal. This makes a pure representation of an exploration-hard challenge.

Ant-Trap and Ant-Maze (fig. 2) are exploration-hard tasks that combine a locomotion task, the quadruped walker, usually called ant, which in itself is challenging, with a navigation task defined by a visible trap or maze-like obstacles. The trap adds a U-shaped obstacle to a unidirectional walking task and so provides a very clear local optimum. Walking straight into it is a local optimum while taking a detour around the trap is a better solution. The maze constrains the movement of the ant within a square where movement is further obstructed by several walls. Success in the environment is defined by moving as close as possible to a goal, where the path to it is obstructed by the walls. Therefore a successful agent needs to take a curved path around the walls of the environment. The local optima in the maze and trap environments are directly visible in fig. 2, as they are given through the obstacles in the environments and can better illustrate the advantage of diversity.

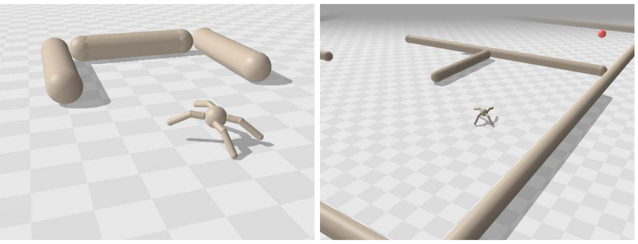

Figure 2: Renderings of the Ant-Trap and Ant-Maze environments. The locomotion task is interrupted by obstacles that need to be navigated.

Evolutionary algorithms without gradient information struggle with the aspect of locomotion tasks [12]. It is difficult for random mutation only to find a solution in a reasonable time frame. So while population-based evolutionary algorithms may be well equipped to deal with local optima the locomotion portion aspect is typically prohibitive for simple evolutionary approaches. The combination with locomotion therefore showcases the power of a merging between evolutionary, specifically quality-diversity approaches with classical reinforcement learning strategies through gradient-based optimization. The gradient-based optimization enables a learning algorithm to find a walking strategy while the population trained with the quality-diversity framework provides enough different solutions such that one of them will find a path that avoids the obstacles in the environment.

These environments pose in different ways the challenge to overcome local optima in reinforcement learning problems. We look at the ability of different ways to define behaviour descriptors to solve the posed problems. A typical way to evaluate the success of a MAP-Elites algorithm is through coverage and QD score [16]. Coverage is the ratio of niches occupied. The QD score is the sum of all performances of all population members offset by a constant such that populated niches always have a positive impact on the score. This is a useful metric if the niches remain fixed and other parts of the algorithm change as it shows the ability of the algorithm to fill niches with high-performing solutions. However, when the definition of niches changes during different runs, QD scores and coverage are not directly comparable any more. Consequently, diverting from these metrics we only look at the best solution found in the archive, which represents the ability of the quality-diversity algorithm to find one appropriate solution to the exploration-hard problem. This is a return to the way reinforcement learning algorithms usually are evaluated.

3.2 The Occupancy Measure as Behaviour Description

The occupancy measure is an important concept in the theory of reinforcement learning. It identifies a policy with the probability of visiting a certain state and choosing a certain action during any episode. As such, the time component of a trajectory-based sampling is lost in the occupancy measure. It has useful properties, for example, a policy can be reconstructed from its occupancy measure, and the mean performance of a policy can be described as the inner product of its occupancy measure and its reward function [19].

The concept of occupancy and its support is particularly important for imitation learning algorithms. The mismatch of the support of the occupancy measure of the learner and the expert is a key problem when designing imitation learning algorithms [17]. Here, enforcing a closeness not only in reaction to states but also in support of occupancy empirically and theoretically improves the performance of algorithms.

Similarly, gradient-based reinforcement learning algorithms estimate improvements in new solutions based on past experiences. This creates a mismatch between the selection of past experiences and the still unknown occupancy of the new solution. As a consequence, the new solution may not improve in performance. This problem has been understood and addressed in trust-region approaches such as PPO [18], which enforces a closeness to past iterations of the solution to guarantee similarity of the support of the occupancy measure, which reduces the mismatch.

In the following, we introduce a practical technique to leverage the occupancy measure of a policy as a behaviour descriptor that supports a MAP-Elites type algorithm to develop a diverse population indiscriminately. We state the relationship between policy and occupancy measure in the following paragraphs as a simplified version of Theorem 2 from [19] and describe an approximation to transfer this relationship to a working algorithm.

Let (S, A, T, α) be a Markov decision process with finite state space S, finite action space A, transition properties T, α the initial state probabilities and a finite time horizon N. For a policy π define the occupancy measure $x : S \times A \longrightarrow \mathbb{R}$ for each state-action pair (s, a) as the expected proportion of visitations of that state-action pair given the policy π, the initial states α, and the transition properties T of the Markov decision process:

$$x^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{N-1} \frac{1}{N} \mathbf{1}_{(s_t,a_t)=(s,a)} | \alpha, \pi, T\right]. \tag{1}$$

Further, $T(s|(a, s'))$ denotes the probability of transitioning to state s from state s' when taking the action a, and α_s is the probability the initial state is s. Then $x : S \times A \longrightarrow \mathbb{R}$ is said to adhere to the *Bellman flow constraints* if for all $s \in S$ it holds:

$$\sum_{a \in A} x(s,a) = \alpha_s + \sum_{(s',a)} x(s',a) T(s|(a,s')), \tag{2}$$

$$x(s,a) \geq 0. \tag{3}$$

With that, we restate a result from [19]:

THEOREM 1. *If x satisfies the Bellman flow constraints then*

$$\pi(a|s) = \frac{x(s,a)}{\sum_{a'} x(s,a')} \tag{4}$$

is a stationary policy and x is the occupancy measure of π. If π is a stationary policy such that x is its occupancy measure, then

$$\pi_{(s,a)} = \frac{x(s,a)}{\sum_{a'} x(s,a')} \tag{5}$$

and x satisfies the Bellman flow constraints.

In plain terms, a policy is fully described by its occupancy measure when disregarding unreachable states where $\sum_a x(s,a) = 0$ and assuming a stationary policy. Since we consider behavioural and therefore observable difference, the first assumption is reasonable. Since many successful approaches to solve RL problems use stationary policies [12, 18], the second assumption is as well. This justifies the use of the occupancy measure as a behaviour descriptor beyond mere practicality, but rather as a complete characterisation of the policy.

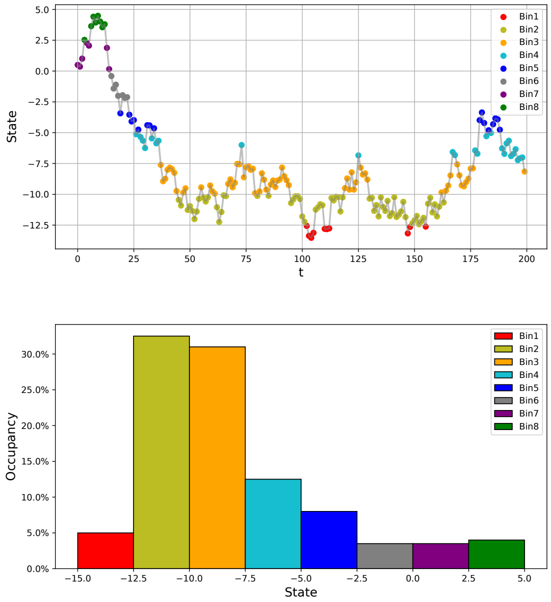

Figure 3: Rolling out a policy generates a time series in the state-action space, here symbolically portrayed as one-dimensional. Bin each encountered value according to a fixed tessellation of the state-action space. The proportion of visitations of the bins then functions as the behaviour descriptor that approximates the occupancy measure.

With this idea, we construct a technical implementation of a behaviour descriptor based on the occupancy measure. To this end, we partition the state-action space of dimension D into bins by constructing a Voronoi-Tessellation with M centroids on $[-1, 1]^D$ in the same way a behaviour space is tessellated in CVT MAP-Elites [20]. The behaviour of a policy is then described by rolling it out once, and counting for each bin how many state-action pairs belong to it. This is then normalized to sum up to 1. This approximates the occupancy measure. The binning process is sketched in fig. 3, where the D-dimensional state-action is represented by one "state" dimension. The approximation of the occupancy measure then serves as the behaviour description of the policy.

To build a MAP-Elites repertoire, we construct a second Voronoi-Tessellation in this behaviour space. By construction this is $X = \{x \in [0, 1]^M \mid \sum_{i=1,\ldots M} x_i = 1\}$. Therefore, we construct the centroids by sampling the Dirichlet distribution with the concentration parameter $\alpha \equiv 1$, which returns a sample of uniform distribution on X. During the MAP-Elites loop, the behaviour description of a new solution is the approximation of the occupancy measure as described before. It is attributed to a niche according to this second Voronoi-Tessellation of X that serves as a tessellation of the space of occupancy measures. We refer to this method as *CVT-ME, Occupancy.*

3.3 Established Solvers and Baseline

We want to investigate how different ways to encode diversity contribute to the solution of exploration-hard problems. As outlined in section 3.1, we expect that locomotion requires gradient descent,

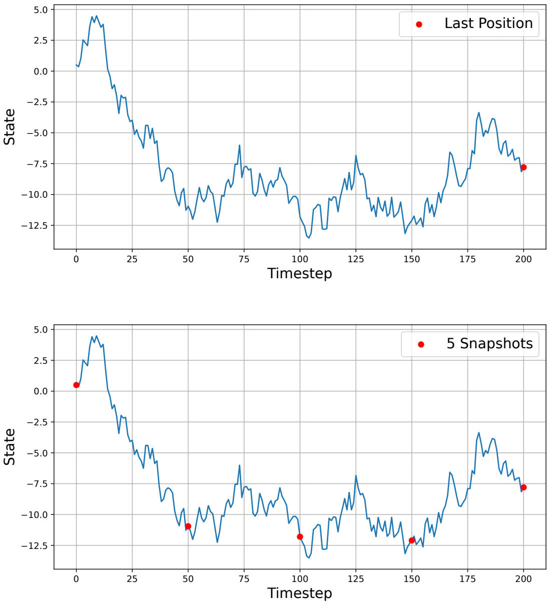

Figure 4: Rolling out a policy generates a time series, here symbolically portrayed as one-dimensional. In a trap- or maze-like environment the custom, problem-specific behaviour descriptor for MAP-Elites is usually the last position, while the established choice for constructing a generic behaviour descriptor in CVT MAP-Elites are snapshots along the time series.

and navigation of mazes and traps requires diversity. Therefore, we mostly limit our investigation to algorithms that combine gradient descent and population-based methods in a way that promises success. We use the standard implementations provided by QDax [2] for the benchmark algorithms and components already implemented for the proposed algorithm. We test our assumption that gradient-based approaches struggle in these environments with a PPO [18] benchmark implementation in Brax [7].

We use different types of MAP-Elites algorithms to solve the problem. In all approaches we incorporate policy-gradient assistance (PGA) [12]. Combining that with a Cartesian grid for behaviour descriptors in the classical approach belongs to the category PGA MAP-Elites, combining it with a centroidal Voronoi tessellation belongs to the category PGA-CVT MAP-Elites, and combining it with an autoencoder belongs to the category PGA-Aurora, see [1] for dedicated analysis of these methods.

We change in every instance of experiment the way behaviour is characterized. The classical way to define a custom problem-specific behaviour descriptor in trap- or maze-like locomotion tasks is through the last physical position of the agent. For the walker, it is the percentage of time either foot touches the ground. We will call this *ME, Custom.* The classical way to characterize behaviour in an unsupervised way through CVT MAP-Elites is to take snapshots of the state encountered during a rollout and regard this time series as the behaviour descriptor. We refer to this as *CVT-ME, Snapshots.* For these behaviour descriptors refer to fig. 4 for a simplified graphical representation. A more sophisticated way to create an unsupervised behaviour descriptor is represented through PGA-Aurora, which uses an autoencoder to find lower dimensional representations of the states encountered during a rollout, which we will refer to as *Aurora.* As much as the different approaches allow, we keep other hyperparameters fixed at reasonable values.

3.4 Random Niching as Lower Baseline

We introduce the idea of random niching as a lower baseline. Here we define a usual MAP-Elites repertoire with a grid in $[0, 1]$, partition it into niches of equal size and assign new solutions to a niche by sampling the uniform distribution leading to random niche assignment independent of the actual behaviour. This then runs in the PGA MAP-Elites framework but encourages behavioural diversity in the minimal way that the population provides, not by defining niches that protect different behaviours. This leads to full coverage very quickly. We expect less diverse solutions in the population, but more diversity in comparison to a regular policy-gradient method, which always works with one solution modified in each loop.

This method still receives the eventual benefits the MAP-Elites setup itself offers even when using behaviour descriptors that do not provide meaningful distinctions of new solutions. We expect any behaviour descriptor to perform at least as well as random niching in an exploration-hard task, but in an environment where diversity does not matter as much, random niching may perform better, as it will spend more resources on higher-performing solutions. We will refer to this method as *ME, Random.*

4 Experimental Validation

4.1 Configuration

For the environments, established algorithms, and shared components of the newly introduced algorithms, we use the reference implementation of the QDax [2] library. The PPO implementation stems from Brax [7].

We manipulate the Ant-Maze environment such that its performance is only defined by the distance from the goal position in its last position. This is different from the default configuration in QDax where in each timestep the reward is defined by secondary rewards of the ant environment and its distance to the goal. We still use that reward information for the policy gradient steps both in the policy-gradient assistance and PPO. This is in line with earlier iterations of this problem [3], and makes it easier to read the conceptual success of the policy from its performance. The number of times steps is set to 1000 in Ant-Trap and 3000 in Ant-Maze.

We implemented the approximation of the occupancy measure through the standard component of Voronoi tessellation implemented in QDax and used TPU-KNN [4] for an approximation of the nearest neighbour calculation. We use tanh to cast the unbounded observation components to $[1, -1]$ for both CVT MAP-Elites algorithms demonstrating the out-of-the-box capabilities, although defining box constraints tailored to the environment have better prospects to achieve good performance. We use the boundaries provided in QDax for each problem for the *ME, Custom.*

The policies are functions from the state to the action space, implemented as fully connected feedforward networks with two hidden layers of size 256 each in Ant-Maze and Ant-Trap, and 64 each in Walker 2D and Point-Maze. We set 1001 centroids for the repertoires, 501 centroids for the binning of the state space for the occupancy measure and the batch size for each generation to 128. For *CVT-ME, Snapshots*, we take 10 snapshots in time in all environments, as higher sampling ratios quickly result in a drop in coverage and eventually in performance. Other hyperparameters are set to default, or, if provided, to recommended values by QDax and Brax.

The runtimes of all methods are on average very similar, ranging from 1.3-1.9 ms per generation for Point-Maze, 50.1-55 ms/gen for Ant-Maze, 20.6-22.3 ms/gen for Ant-Trap, and 4.5-5.3 ms/gen for Walker 2D. All methods achieve reasonable coverage, the lowest for *CVT-ME, Occupancy* in Point-Maze with 13.8%, which still represents a population of 138 individuals.

4.2 Results in single environments

The following plots of performance over generations display the performance of the highest-scoring individual in each population with the mean and the standard error over generations. PPO does not learn in generations but is still represented in the same plot, by aligning the number of observed transitions in the environment in both methods for a fair comparison.

4.2.1 Walker 2D. The Walker 2D (fig. 5) shows a situation where diversity seems superfluous or obstructive in finding a good solution. We would expect and finally see in the results, that all types of MAP-Elites and CVT MAP-Elites perform similarly well. *Aurora*, however, while learning over time, lags behind the more direct methods. *ME, Custom* here uses the ratio of any of the walker's feet touching the ground, instead of its last position. We can consider the environment solved, if the agent learns to walk resulting in at least 300 points of reward and stays healthy over all 1000 timesteps resulting in 1000 points of reward, totalling at least 1300 points of reward. Remarkably, the PPO algorithm fails to solve this problem, because it always gets stuck in the trivial solution and local optimum of falling over. This shows the inherent problems of these algorithms and the potential benefit population-based approach can bring even to problems that are not an obvious fit.

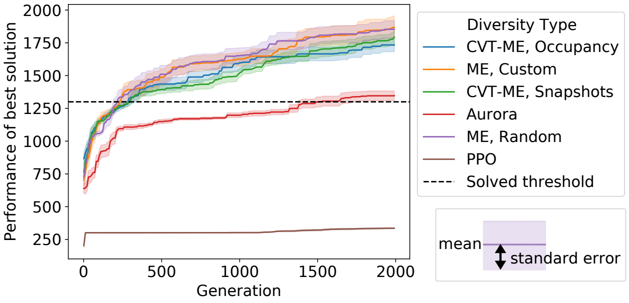

Figure 5: The maximum performance achieved in the Walker 2D environment.

4.2.2 Point-Maze. The Point-Maze environment both presents the clearest case of the exploration problem and also exhibits the clearest results (fig. 6). Without fail, *ME, Custom* and *CVT-ME, Occupancy*, solve the exploration problem in all 10 instances. *ME, Random*, as expected, gives a lower bound, with *Aurora* and *CVT-ME, Snapshots* falling in between. A correct solution navigates the agent into the appropriate area with a reward of no less than −30. As expected, PPO cannot solve this problem and gets stuck either in the first or second wall.

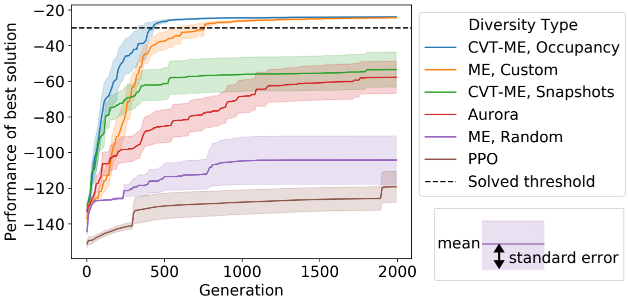

Figure 6: The maximum performance achieved in the Point-Maze environment.

4.2.3 Ant-Trap. The results in the Ant-Trap environment (fig. 7) show how powerful the MAP-Elites approach itself is, even if the behaviour descriptor is not inherently helpful. All configurations with different behaviour descriptors come up with solutions to the problem that avoid the trap, even *ME, Random.* PPO however, will always steer the ant into the wall with such force it stops being healthy and ends the episode.

The way the problem is set up, positive reward signals come from being healthy, with 1000 points from 1000 timesteps at most and moving in the x-direction, where the trap waits. The trap itself is positioned with the far wall at 12 units of distance to the ant's starting point, which leads to 240 points at most when the ant is not moving around the trap. Additionally, negative rewards are incurred from the control signals. So, the problem is considered solved, for any accumulated reward over 1240. With this definition of solving the environment, finding a solution that avoids the trap, any algorithm but PPO confidently solves the problem.

4.2.4 Ant-Maze. In the Ant-Maze environment, the classical MAP-Elites approach of using the last physical position of the agent, is the strongest choice, with two CVT-ME methods coming in a close second, followed by *ME, Random* (fig. 8). *Aurora* again takes a longer time to learn. PPO sometimes learns a way to cheat the system, by jumping over the wall, which in the physics simulation is represented by a tube. Manipulating this environment by placing a second tube on top of the first one prevents jumping. Then, PPO consistently gets stuck, denoted as "PPO no jump" in fig. 8.

The target is located at $(35, 0)$, with walls around 5 units distant. A reward signal larger than -4 is considered a solution because it suggests the agent made it into the critical zone of the maze but also stopped within that zone instead of running into a wall close to the target to stop its movement.

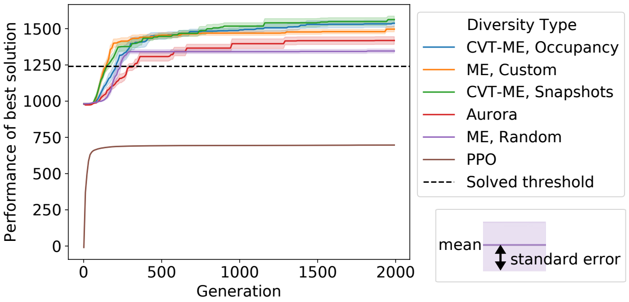

Figure 7: The maximum performance achieved in the Ant-Trap environment.

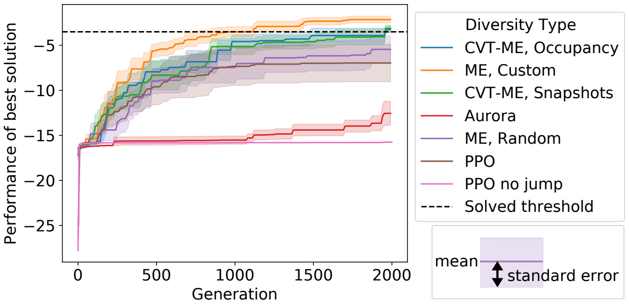

Figure 8: The maximum performance achieved in the Ant-Maze environment.

4.3 Statistical Evaluation

To examine the statistical significance of the results, for each environment, we looked at the maximal scores achieved with the different methods in each of the 10 runs. We use the Mann–Whitney *U* test to determine if the difference in observations is statistically significant. The null hypothesis is the medians of the two underlying distributions are the same. If the null hypothesis is rejected, the performance of the methods can be seen as significantly different, see fig. 9 for a visualisation where green signifies different grades of significance and red indicates no significant difference. Both in the Ant-Maze and in the Walker 2D environment, the diversity methods are not significantly distinguishable, except for *Aurora*, which falls behind. The Point-Maze shows the high performers, *CVT-ME, Occupancy* and *ME, Custom* are not significantly distinguishable from each other, but this group and each of the other three methods seem distinct from each other. The Ant-Trap shows two groups, the high performers represented by *ME, Custom* and both CVT approaches, with *ME, Random* and *Aurora* in the second group. PPO is significantly different (worse) in every environment compared to every method save Ant-Maze, where the "cheating" of jumping over the wall, lumps it with all methods but *Aurora.*

4.4 Evaluation Overall

Evaluating the success of different methods in different environments, we look at the best performance achieved by any solution encountered during the run. For all environments, we also set a threshold for the performance. If the threshold is exceeded, we assume the fundamental underlying problem, which the environment presents, has been solved. These are learning to walk, or navigating the obstacles, or both. We explained the thresholds for each environment in the earlier paragraphs. The obtained results for this measure of success are shown in table 1. This binary distinction shows the difficulty of solving the underlying problem and the ability of a certain diversity mechanism to avoid falling into the local optimum. Looking at Walker 2D and Ant-Trap any MAP-Elites approach seems sufficient. Ant-Maze is more ambiguous, but there some differentiation is visible, with *ME, Custom* solving the problem 9 out of 10 times, and *ME, Random* only 5 out of 10 times. That suggests an appropriate behaviour descriptor improves the chance of success. An unambiguous distinction can be seen in the Point-Maze environment, where *ME, Custom* and *CVT-ME, Occupancy* always solve the problem, whereas the others struggle.

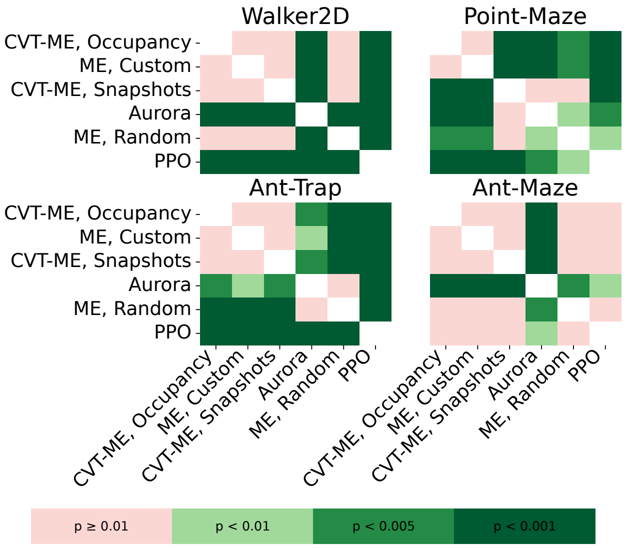

Figure 9: Presentation of statistical significance where darker green shows a more significant rejection of the null hypothesis which implies one method outperforms the other.

	Walker 2D	Point-Maze	Ant-Trap	Ant-Maze
CVT-ME, Occupancy	10	10	10	8
ME, Custom	10	10	10	9
CVT-ME, Snapshots	10	4	10	7
Aurora	6	3	10	0
ME, Random	10	2	10	5
PPO	0	0	0	4

Table 1: Number of trials that solve the underlying problem out of 10.

The min-max-scaled performance in the different environments, depicted in fig. 10, reflects this analysis. Here, the y-value 1 represents the best performance seen in any run, 0 the worst. The large

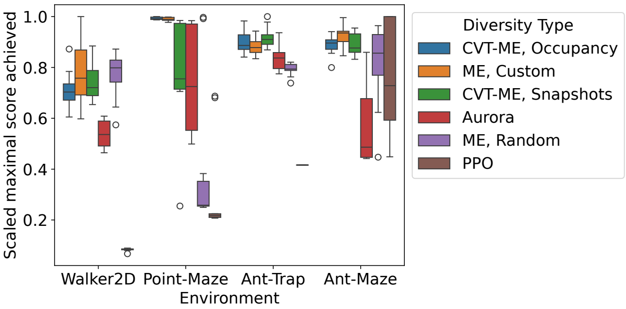

Figure 10: Scaled maximizing score achieved at the end of a run as a *box plot* showing outliers, extreme values, quartiles, and medians.

	Walker 2D	Point-Maze	Ant-Trap	Ant-Maze
CVT-ME, Occupancy	2084.3	**-22.9**	1675.3	-1.7
ME, Custom	**2367.3**	-23.1	1606.3	-0.1
CVT-ME, Snapshots	2111.3	-25.0	**1704.8**	-1.3
Aurora	1499.6	-25.1	1595.2	-4.0
ME, Random	2084.2	-23.3	1395.2	-1.0
PPO	349.3	-66.5	697.9	**-0.0**

Table 2: Maximum performance over all runs.

spread of *Aurora*, and *CVT-ME, Snapshots* in Point-Maze show these methods only sometimes solve the underlying problem. Similarly, in the Ant-Maze environment, the first three that often solve the underlying problem have a smaller spread than the others. The plot shows several instances where some methods seem to perform similarly, as in the Ant-Trap environment. For reference, table 2 shows the top score achieved over all runs by the different methods.

To summarise, our approach with the occupancy measure is just as suited to solve the problem as the custom, problem-specific behaviour descriptors in all four environments. It retains the performance, without the need to hand-craft a behaviour descriptor for each specific problem. In one environment, the Point-Maze, it improves on CVT MAP-Elites with taking snapshots of the trajectory, the classical approach for generic behaviour descriptors. In the other three, both perform similarly well. So overall, it improves on the classical approach for generic behaviour descriptors.

As expected, random niching underperforms everywhere but in the Walker 2D environment, but especially when compared with PPO, is still remarkably effective. Aurora features some successes in exploring the environment, particularly visible in the Point-Maze environment, but would need more computation time, a more focused preselection of the input data as in [1], or hyperparameter tuning.

Overall, the introduced approach, CVT MAP-Elites with occupancy measure, shares the place of top-performer with the classic MAP-Elites approach that uses custom, problem-specific behaviour descriptors. But of course, it has the advantage to skip that step of handcrafting a suitable behaviour descriptor.

5 Discussion

The newly introduced behaviour descriptor using the occupancy measure offers a real alternative to the trajectory-based characterisation for non-specific diversity in reinforcement learning settings. It maintains behavioural diversity well enough to solve the exploration-hard problems presented in QDax. It outperforms taking snapshots in the purer exploration problem presented in the Point-Maze environment. It has the conceptual advantage of using the information of the full trajectory. In the presented exploration problems snapshots along the trajectory always include the information of the last position, which is particularly important, but still, the approach with the occupancy measure slightly outperforms taking snapshots.

The reflection on the given problems also uncovers properties of the benchmark environments. While the Ant-Trap and Ant-Maze environments both demonstrate the value of the quality-diversity algorithms, the evaluation is difficult: In Ant-Trap, the total score will be higher for fast movement, which obfuscates the underlying problem of avoiding the trap, which all quality-diversity algorithms achieve. The Ant-Maze environment provides a challenging problem but should be slightly adjusted to avoid the exploit of jumping over a wall by PPO. The Point-Maze environment provides a clearer picture to differentiate the compared methods. The Walker 2D demonstrates the real value quality-diversity can provide when PPO falls into a local optimum invisible at first glance.

Another interesting aspect not explored here is that the behaviour descriptor using the occupancy measure can meaningfully aggregate the behaviour of several runs, something classic behaviour descriptors like a last physical position cannot: The occupancy measure in theory gets more accurate the more trajectories are rolled out and aggregated. This could prove helpful to evolve solutions for environments with stronger random influences, but also alleviate the problem of (un)lucky individuals [6], where the behaviour or performance of an individual in an uncertain environment is not representative of its average behaviour or performance.

There may be even more efficient ways to encode diversity using the occupancy measure than the presented trick of using two linked centroidal Voronoi tessellations, like using the earth mover's distance when searching for the appropriate niche in the space of occupancy measures. The introduction of a prior to explore a more realistic representation of the state space as is typically done in CVT MAP-Elites may further improve performance, a technique passed over in favour of clarity.

We see the greatest practical advantage of the newly presented approach with the occupancy measure not in its performance, but in the simplicity as an out-of-the-box solution, that uses all behavioural information, does not require a hand-crafted behaviour descriptor and can handle high-dimensional behaviour spaces. As such it is highly suited as a lower-bound baseline for any quality-diversity problem where behaviour descriptors are being compared or constructed.

Finally, the presented approach demonstrates that generalised diversity, that is diversity based on considering the complete observed behaviour, can be feasible and useful in Quality-Diversity.

References

[1] Félix Chalumeau, Raphaël Boige, Bryan Lim, Valentin Macé, Maxime Allard, Arthur Flajolet, Antoine Cully, and Thomas Pierrot. 2023. Neuroevolution is a Competitive Alternative to Reinforcement Learning for Skill Discovery. In *The Eleventh International Conference on Learning Representations, ICLR*. OpenReview.net, Kigali, Rwanda. https://openreview.net/forum?id=6BHlZgyPOZY

[2] Félix Chalumeau, Bryan Lim, Raphael Boige, Maxime Allard, Luca Grillotti, Manon Flageat, Valentin Macé, Guillaume Richard, Arthur Flajolet, Thomas Pierrot, et al. 2024. Qdax: A library for quality-diversity and population-based algorithms with hardware acceleration. *Journal of Machine Learning Research* 25, 108 (2024), 1–16.

[3] Félix Chalumeau, Thomas Pierrot, Valentin Macé, Arthur Flajolet, Karim Beguir, Antoine Cully, and Nicolas Perrin-Gilbert. 2022. Assessing Quality-Diversity Neuro-Evolution Algorithms Performance in Hard Exploration Problems. https://doi.org/10.48550/ARXIV.2211.13742 arXiv:2211.13742

[4] Felix Chern, Blake Hechtman, Andy Davis, Ruiqi Guo, David Majnemer, and Sanjiv Kumar. 2022. TPU-KNN: K Nearest Neighbor Search at Peak FLOP/s. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), New Orleans, LA, USA, 15489 – 15501. http://papers.nips.cc/paper_files/paper/2022/hash/639d992f819c2b40387d4d5170b8ffd7-Abstract-Conference.html

[5] Antoine Cully and Yiannis Demiris. 2018. Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Transactions on Evolutionary Computation* 22, 2 (2018), 245–259. https://doi.org/10.1109/TEVC.2017.2704781

[6] Manon Flageat, Bryan Lim, and Antoine Cully. 2024. Beyond Expected Return: Accounting for Policy Reproducibility When Evaluating Reinforcement Learning Algorithms. *Proceedings of the AAAI Conference on Artificial Intelligence* 38 (03 2024), 12024–12032. https://doi.org/10.1609/aaai.v38i11.29090

[7] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. 2021. *Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation*. http://github.com/google/brax

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

[9] Luca Grillotti and Antoine Cully. 2022. Unsupervised Behavior Discovery With Quality-Diversity Optimization. *IEEE Trans. Evol. Comput.* 26, 6 (2022), 1539–1552. https://doi.org/10.1109/TEVC.2022.3159855

[10] Joel Lehman and Kenneth O. Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comput.* 19, 2 (June 2011), 189–223. https://doi.org/10.1162/EVCO_a_00025

[11] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. arXiv:1504.04909

[12] Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted MAP-Elites. In *GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France). Association for Computing Machinery, New York, NY, USA, 866–875. https://doi.org/10.1145/3449639.3459304

[13] Giuseppe Paolo. 2021. *Learning in Sparse Rewards setting through Quality Diversity algorithms*. Theses. Sorbonne Université. https://theses.hal.science/tel-03707344

[14] Giuseppe Paolo, Alban Laflaquière, Alexandre Coninx, and Stéphane Doncieux. 2020. Unsupervised Learning and Exploration of Reachable Outcome Space. In *2020 IEEE International Conference on Robotics and Automation, ICRA* (Paris, France). IEEE, New York City, US, 2379–2385. https://doi.org/10.1109/ICRA40945.2020.9196819

[15] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Searching for Quality Diversity When Diversity is Unaligned with Quality. In *Parallel Problem Solving from Nature - PPSN XIV* (Edinburgh, UK) *(Lecture Notes in Computer Science, Vol. 9921)*. Springer, Cham, Switzerland, 880–889. https://doi.org/10.1007/978-3-319-45823-6_82

[16] Justin K. Pugh, Lisa B. Soros, Paul A. Szerlip, and Kenneth O. Stanley. 2015. Confronting the Challenge of Quality Diversity. In *GECCO '15: Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, Madrid, Spain, 967–974. https://doi.org/10.1145/2739480.2754664

[17] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*. PMLR, Fort Lauderdale, FL, USA, 627–635. https://proceedings.mlr.press/v15/ross11a.html

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[19] Umar Syed, Michael H. Bowling, and Robert E. Schapire. 2008. Apprenticeship learning using linear programming. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008) (ACM International Conference Proceeding Series, Vol. 307)*. ACM, Helsinki, Finland, 1032–1039. https://doi.org/10.1145/1390156.1390286

[20] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2016. Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm. *IEEE Transactions on Evolutionary Computation* 22 (2016), 623–630.

A Key figures of the experiment averaged over all runs

Environment	Technique	top score	s/gen	QD score	mean score	coverage
Point-Maze	CVT-ME, Occupancy	**-23.8**	1.3	36843	-177.2	13.8
	ME, Custom	-24.3	1.3	269893	-175.0	100.0
	CVT-ME, Snapshots	-53.4	1.3	128692	-172.3	47.2
	Aurora	-57.7	1.9	228017	-217.0	100.0
	ME, Random	-104.2	1.3	**339003**	**-106.1**	100.0
	PPO	-119.2	n/a	n/a	n/a	n/a
Ant-Maze	CVT-ME, Occupancy	-3.2	55.0	12105	-20.72	35.2
	ME, Custom	**-2.1**	54.9	22881	-29.66	90.0
	CVT-ME, Snapshots	-3.1	54.9	8472	-18.86	23.3
	Aurora	-12.6	50.8	31068	-24.02	100.0
	ME, Random	-5.5	50.1	**39467**	**-15.62**	100.0
	PPO	-7.0	n/a	n/a	n/a	n/a
Ant-Trap	CVT-ME, Occupancy	1537.7	20.7	1064546	538.2	29.2
	ME, Custom	1497.1	22.3	3193614	916.9	79.4
	CVT-ME, Snapshots	**1563.1**	20.6	935478	450.4	29.3
	Aurora	1418.4	21.3	3891014	781.4	100.0
	ME, Random	1346.0	20.6	**4276066**	**1168.0**	100.0
	PPO	696.9	n/a	n/a	n/a	n/a
Walker 2D	CVT-ME, Occupancy	1733.5	4.5	361472	849.0	40.2
	ME, Custom	1346.7	4.5	989722	1203.6	79.0
	CVT-ME, Snapshots	1793.6	4.5	906942	1047.3	82.7
	Aurora	**1867.9**	5.3	380674	332.1	100.0
	ME, Random	1853.9	4.5	**1473018**	**1423.4**	100.0
	PPO	335.7	n/a	n/a	n/a	n/a

Table 3: More key figures for the experiments with mean values at the end of the 10 runs. Offsets for QD scores are determined empirically as Point-Maze: -444.81, Ant-Maze: -55.05, Ant-Trap: -3103.80, Walker 2D: -48.15. The dominance of *ME, Random* in QD score, mean score, and coverage shows that these typical metrics are not suitable to compare methods with differing archives.

B Hyperparameters and Configuration

For completeness the full list of hyperparameters used. The entry "Distinct centroids (double CVT)" in table 4 refers to the number of centroids selected to represent the state-action space or the number of bins. The presented environments exhibit a wide spread of both length (200-3000) and dimensionality (4-109) of the rollouts.

Parameter	Value
Initial CVT samples	50,000
Distinct centroids (double CVT)	501
Distinct centroids	1,001
New generation size	128
Proportion of PG updates	0.5
Replay buffer size	1,000,000
Critic learning rate	0.0003
Greedy learning rate	0.0003
Policy learning rate	0.001
Noise clip	0.5
Policy noise	0.2
Discount factor	0.99
Reward scaling factor	1.0
Transitions batch size	256
Soft τ update	0.005
Critic training steps	300
Policy-gradient training steps	100
Policy delay	2

Table 4: PGA-ME parameters

Environment	Episode Length	State dim.	Action dim.
Point-Maze	200	2	2
Ant-Maze	3000	101	8
Ant-Trap	1000	95	8
Walker 2D	1000	17	6

Table 5: Key figures of the environments

Environment	Episode Length
Sampling frequency	10
AURORA dimensions	5
l threshold value	0.2

Table 6: AURORA parameters