

Empirical analysis of PGA-MAP-Elites for Neuroevolution in Uncertain Domains

MANON FLAGEAT, FELIX CHALUMEAU, and ANTOINE CULLY, Imperial College London, UK

Quality-Diversity algorithms, among which MAP-Elites, have emerged as powerful alternatives to performance-only optimisation approaches as they enable generating collections of diverse and high-performing solutions to an optimisation problem. However, they are often limited to low-dimensional search spaces and deterministic environments. The recently introduced Policy Gradient Assisted MAP-Elites (PGA-MAP-Elites) algorithm overcomes this limitation by pairing the traditional Genetic operator of MAP-Elites with a gradient-based operator inspired by Deep Reinforcement Learning. This new operator guides mutations toward high-performing solutions using policy-gradients. In this work, we propose an in-depth study of PGA-MAP-Elites. We demonstrate the benefits of policy-gradients on the performance of the algorithm and the reproducibility of the generated solutions when considering uncertain domains. We first prove that PGA-MAP-Elites is highly performant in both deterministic and uncertain high-dimensional environments, decorrelating the two challenges it tackles. Secondly, we show that in addition to outperforming all the considered baselines, the collections of solutions generated by PGA-MAP-Elites are highly reproducible in uncertain environments, approaching the reproducibility of solutions found by Quality-Diversity approaches built specifically for uncertain applications. Finally, we propose an ablation and in-depth analysis of the dynamic of the policy-gradients-based variation. We demonstrate that the policy-gradient variation operator is determinant to guarantee the performance of PGA-MAP-Elites but is only essential during the early stage of the process, where it finds high-performing regions of the search space.

Additional Key Words and Phrases: Quality-Diversity, MAP-Elites, Uncertain domains, Neuroevolution, Reinforcement Learning, Policy Gradients

1 INTRODUCTION

Natural evolution can generate diverse species with distinct characteristics existing within the same ecological niche. This variety of survival strategies adopted by cohabiting species has allowed life to endure for instance multiple mass extinction events. Inspired by this importance of diversity, researchers have proposed Quality-Diversity (QD) optimisation [Chatziligeroudis et al. 2021; Cully and Demiris 2018b; Pugh et al. 2016]. This branch of Evolutionary Computation does not generate one solution to a problem based on performance only, but rather a collection of diverse and high-performing solutions. QD approaches, and in particular, the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm [Mouret and Clune 2015], have shown promising results in a wide range of domains, from content generation for video games [Gravina et al. 2019], and aerodynamic design [Gaier et al. 2017] to robotics [Cully et al. 2015]. They allow the discovery of creative solutions [Gaier et al. 2017; Gravina et al. 2019], but also non-trivial stepping stones toward even better solutions [Cully and Demiris 2018b], and enable fast adaptation to damages or unseen situations [Chatziligeroudis et al. 2021; Cully et al. 2015].

QD algorithms traditionally rely on elitism. This is beneficial in environments where performance scores are deterministically associated with solutions; in other words, environments in which a given solution always exhibits the same behaviour. However, many environments, referred to as uncertain, do not fulfil this assumption: real robots, for example, use imperfect sensors and actuators. In such environments, a given solution might exhibit slightly different behaviours from one evaluation to another with different performance scores. As a consequence, solutions might be lucky and get a higher performance score than they could expect on average. The elitism of QD

Authors' address: Manon Flageat, manon.flageat18@imperial.ac.uk; Felix Chalumeau, felix.chalumeau20@imperial.ac.uk; Antoine Cully, a.cully@imperial.ac.uk, Imperial College London, Department of Computing, Adaptive and Intelligent Robotics Lab, London, UK, SW7 2AZ.

algorithms preserves these lucky solutions, making uncertain applications an important challenge for QD [Flageat and Cully 2020; Justesen et al. 2019]. In particular, some solutions might be more robust than others to the source of uncertainty and thus be more reproducible. For example, if one actuator is an important source of uncertainty, solutions relying on this actuator would be less reproducible than others. Most QD algorithms do not have mechanism to favour reproducible solutions [Flageat and Cully 2020; Justesen et al. 2019]. Besides, QD algorithms are traditionally driven in their exploration of the search space by a Genetic Algorithm (GA). Due to its sample inefficiency, this mechanism makes it difficult -if not impossible- to apply QD to high-dimensional problems [Such et al. 2017], making it another important limitation. This paper focus on domains that are both high-dimensional and uncertain.

On the other side, Deep Reinforcement Learning (DRL) approaches [Arulkumaran et al. 2017] show encouraging results in a wide range of uncertain domains with high-dimensional search-spaces [Lillicrap et al. 2016; Mnih et al. 2015]. Unlike QD, DRL aims to find a single performance-maximising solution. It relies on the powerful directed-search abilities of gradient-based learning, as well as the function-approximation capabilities of Deep Neural Networks (DNNs) to find a high-performing and highly-reproducible solution [Fujimoto et al. 2018; Lillicrap et al. 2016]. Policy Gradient Assisted MAP-Elites (PGA-MAP-Elites) [Nilsson and Cully 2021] has been recently introduced as an alternative algorithm mixing QD and DRL to scale QD algorithms to large neuroevolution problems. PGA-MAP-Elites extends MAP-Elites with a new variation operator based on policy-gradients (PG). Half of the solutions are generated using this PG variation operator and the other half using the usual GA variation operator.

This work proposes an in-depth study of PGA-MAP-Elites and, more generally, the benefits of integrating PG in MAP-Elites. We focus on the same tasks as in the PGA-MAP-Elites paper, referred to as QD-Gym [Nilsson and Cully 2021]. These tasks are all uncertain locomotion tasks with high-dimensional parameter spaces. Our contributions can be summarised as followed:

- A comparison of PGA-MAP-Elites to a larger set of QD and DRL baselines, demonstrating extensively that it maintains the divergent search capabilities of MAP-Elites while finding solutions as performing as those found by compared DRL approach.
- Additional results in deterministic QD-Gym environments, highlighting the benefit of PGA-MAP-Elites applied to high-dimensional search-space independently of the uncertainty.
- A reproducibility-study in uncertain environment, demonstrating that PGA-MAP-Elites does not only generate solutions that are better performing but also more reproducible in uncertain environments, matching and sometimes out-performing the performance of QD algorithms specifically designed for such applications.
- To support these analysis, we also propose a new set of loss metrics quantifying the reproducibility of solutions found by QD algorithms in uncertain domains.
- An ablation of the proportion of PG and GA variation in PGA-MAP-Elites, proving the importance of both type of variations.
- An analysis of the impact of PG and GA variation operators on improvement of the collection showing that PG is only essential during the early stage of the process, where it finds high-performing regions of the search space.

2 BACKGROUND AND RELATED WORK

2.1 Quality-Diversity Optimisation and MAP-Elites algorithm

While standard optimisation searches for a single high-performing solution, QD optimisation allows finding a collection of solutions that are as high-performing and diverse as possible. To this end, QD algorithms distinguish solutions that produce different behaviours. This distinction is

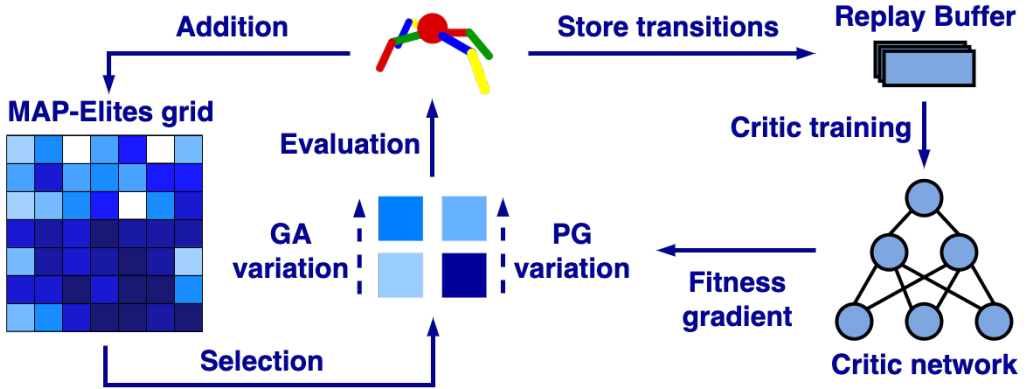


Fig. 1. PGA-MAP-Elites is based on the standard MAP-Elites loops: parent solutions are selected from the archive and mutated to generate offspring that are evaluated in the environment and added back to the archive. However, in PGA-MAP-Elites, the mutation relies on two distinct variation operators: (1) a Policy Gradient (PG) operator that directs variation toward high-performing solutions, and (2) a Genetic Algorithm (GA) operator that maintains divergent search. The PG operator relies on a critic neural network, trained asynchronously to the MAP-Elites loop using experience collected in a replay buffer during evaluations.

based on metrics chosen as part of the task definition to be meaningful for the type of diversity sought. Each metric corresponds to a dimension of interest, denoted as Behavioural Descriptor (BD) [Chatzilygeroudis et al. 2021; Cully and Demiris 2018b; Pugh et al. 2016]. For example, different solutions to a robotic locomotion task might induce different gaits. To characterise each gait, the BD of a solution could be the proportion of time each foot of the robot is in contact with the ground [Colas et al. 2020; Cully et al. 2015; Vassiliades et al. 2017]. QD algorithms attempt to find solutions as distant as possible in terms of BD while being high-performing in their local region of the BD space; in other words, collections of both diverse and high-performing solutions. There are currently two leading QD approaches: Novelty Search with Local Competition [Lehman and Stanley 2011] and MAP-Elites [Mouret and Clune 2015], the focus of this work.

The core idea of MAP-Elites is to discretise the BD-space into a grid. This grid constitutes the final collection of solutions returned at the end of the algorithm. Each of its cells corresponds to a behavioural niche and can store one solution: the elite of this niche. MAP-Elites aims to fill as many niches as possible with the highest-performing solutions possible. The grid is created empty and initially filled with a set of randomly-generated solutions. One iteration of MAP-Elites can be summarised as: (1) randomly selecting parent solutions from the grid, (2) applying variations (such as mutations or crossovers) to generate offspring, and evaluating them, (3) adding to the grid offspring that either populate an empty cell or outperform an existing elite.

2.2 Deep Reinforcement Learning setting and TD3 algorithm

2.2.1 Reinforcement Learning (RL). In the typical RL setting, an agent strives to find the optimal way of acting in an environment. The agent-environment interaction is modelled as a Markov Decision Process (MDP) given as $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$. The agent acts sequentially at discrete time-steps $t \in [0, T]$, performing actions $\mathbf{a}_t \in \mathcal{A}$. These actions induce transitions detailed by \mathcal{P} between states \mathbf{s}_t and $\mathbf{s}_{t+1} \in \mathcal{S}$ of the environment, and each transition generates a reward r_t . In an MDP, the immediate next state \mathbf{s}_{t+1} only depends on the current state \mathbf{s}_t of the environment. Internally, the agent chooses its next action based on the current state \mathbf{s}_t following a policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$. The agent

aims to learn the policy that maximises the expected return $\mathbb{E}_{\pi_\phi} \left[\sum_{t=0}^T \gamma^t r_t \right]$, where $\gamma \in [0, 1]$ is the discount factor which regulates importance of future rewards.

2.2.2 Deep Reinforcement Learning (DRL). DRL solves continuous and large-dimensional RL applications by leveraging the powerful directed-search abilities of gradient-based learning, as well as the function-approximation capabilities of Deep Neural Network (DNNs). Most DRL algorithms rely on approximating with a DNN the action-value function $Q^{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\pi_\phi} \left[\sum_{k=0}^{T-t} \gamma^k r_{t+k+1} \mid \mathbf{s}_t, \mathbf{a}_t \right]$, which encodes the expected return from following the policy π_ϕ after having performed action \mathbf{a}_t in state \mathbf{s}_t . The DNN is trained to approximate Q^{π_ϕ} via the Bellman equation in Eq. 1: it incrementally learns a better approximation of Q^{π_ϕ} by bootstrapping from its current estimates.

$$Q^{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E} \left[Q^{\pi_\phi}(\mathbf{s}_{t+1}, \pi_\phi(\mathbf{s}_{t+1})) \right] \quad (1)$$

In most DRL algorithms, the policy π_ϕ is also modelled by a DNN. The DNN approximating Q^{π_ϕ} is often referred to as the critic while the DNN approximating π_ϕ as the actor. Methods using both an actor network and a critic network are thus known as actor-critic.

2.2.3 Twin Delayed Deep Deterministic policy-gradient algorithm (TD3). This work is based on TD3 [Fujimoto et al. 2018], one of the state-of-the-art actor-critic DRL algorithms. In TD3, the policy π_ϕ is learned by an actor network, while the approximation of Q^{π_ϕ} is done simultaneously by two distinct critics: Q_{θ_1} and Q_{θ_2} . These two critics have the same structure but are trained separately to counteract overestimations; more information are given below. In TD3, Q^{π_ϕ} is learned in an off-policy fashion, meaning using experience collected by any policy, and not only by the current best policy [Lin 1992]. Thus, TD3 stores all encountered transitions $(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})$ in a replay buffer \mathcal{B} to use them to train the critics. Based on batch of N transitions sampled from \mathcal{B} , the loss L defined in Eq. 2 updates the critics towards a target y that approximates Eq. 1.

$$L(\theta_1, \theta_2) = (y - Q_{\theta_1}(\mathbf{s}_t, \mathbf{a}_t))^2 + (y - Q_{\theta_2}(\mathbf{s}_t, \mathbf{a}_t))^2 \quad (2)$$

$$y = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \min_{i=1,2} Q_{\theta'_i}(\mathbf{s}_{t+1}, \pi_{\phi'}(\mathbf{s}_{t+1})) + \epsilon \quad (3)$$

The target y uses the minimum action-value prediction between the two critics Q_{θ_1} and Q_{θ_2} to counteract overestimations done by the critics and avoid instabilities. This is the reason why there is two critic networks. On top, TD3 also uses target networks $Q_{\theta'_1}$, $Q_{\theta'_2}$ and $\pi_{\phi'}$ [Mnih et al. 2013, 2015]. Target networks are copies of the original networks, but updated with a delay τ to amortise variations and provide further stability. Finally, in the expression of the target y , $\epsilon \sim \text{clip}(\sigma_p \mathcal{N}(0, I), -c, c)$ is a noise on the policy's action to promote reproducibility in uncertain environments.

The experiences collected in \mathcal{B} are also used to learn the policy π_ϕ to maximise Q^{π_ϕ} , using an approximation of the deterministic policy-gradient [Silver et al. 2014]. As the policy is deterministic, exploration is achieved by adding Gaussian noise with a variance σ_a to the selected action: $\mathbf{a}_t = \pi_\phi(\mathbf{s}_t) + \sigma_a \mathcal{N}(0, I)$. This exploration noise is applied when interacting with the environment, while the reproducibility noise mentioned earlier is only considered to compute the critics target.

2.3 Alternative variation operator for Quality Diversity

In MAP-Elites, as in standard GAs, variations are fitness-agnostic and aim to explore around existing solutions. However, this lack of directed search can cause slow convergence even in low-dimensional search space [Fontaine et al. 2020a], and this problem is increasingly prominent as the number of dimensions increases [Colas et al. 2020]. Previous works in QD have proposed alternatives to the standard GA mechanism for MAP-Elites. **Covariance Matrix Adaptation MAP-Elites (CMA-MAP-Elites)** [Fontaine et al. 2020a] proposes a directed variation scheme

relying on Covariance Matrix Adaptation Evolution Strategy [Hansen 2016]. This approach allows directing the variations toward regions of the search space that maximise the archive improvements and has proven to outperform MAP-Elites in multiple domains [Cully 2020; Fontaine et al. 2020a]. However, it relies on covariance matrix inversions, which do not scale easily to high-dimensional space. Alternatively, **Policy Manifold Search** [Rakicevic et al. 2021] and **Data-Driven Encoding** [Gaier et al. 2020] proposed mutations scheme relying on learned low-dimensional representation of the solutions space to scale QD to high-dimensional search-space. Another approach, **MAP-Elites with Evolution Strategies (MAP-Elites-ES)** [Colas et al. 2020] replaces GA with Evolution Strategies proposed in Salimans et al. [2017] to extend MAP-Elites to high-dimensional search spaces. This method directs the variation toward an objective using empirical gradients estimated from evaluating a range of perturbations around a current solution. MAP-Elites-ES proposes to use a mixture of fitness and diversity objectives to direct the search. However, it requires a large number of evaluations and typically finds archives containing far fewer behaviours than MAP-Elites given an equal amount of experience in the environment [Colas et al. 2020]. Recently, Fontaine et al. have proposed the **Differentiable-QD** framework [Fontaine and Nikolaidis 2021] for the specific case of QD applied to differentiable domains. This work augments QD search with the information given by the explicit gradient of the function to optimise. However, it is limited to differentiable domains, which is not the case with DRL, thus it is not directly applicable to the domains considered in this work. Differentiable-QD has recently been extended to non-differentiable domain in Tjanaka et al. [2022] but the authors prove that this approach does not work well for the domains considered in this work. Finally, **Quality-Diversity Policy-Gradient (QD-PG)** [Pierrot et al. 2022] has been recently introduced as a way to combine QD and DRL. QD-PG introduces a policy-gradient for diversity, computed using BDs defined at the state level and referred to as state-descriptors. At each generation of QD-PG, half of the solutions are updated using this diversity policy-gradient and the other half using the quality policy-gradient.

2.4 Quality-Diversity applied to uncertain environments

In many environments, the exact same solution might not exhibit the same behaviour when evaluated twice. One solution might thus get different fitness or BD values from one evaluation to another. Such environments have been extensively studied in the Evolutionary Algorithm literature [Jin and Branke 2005], we refer to them as uncertain environments.

2.4.1 Performance estimation in uncertain environments. The lack of reproducibility in uncertain environments directly impacts the performance of QD algorithms. In MAP-Elites, solutions are evaluated once and the result of this unique evaluation is used to determine their cell and their fitness. However, this single evaluation might be "lucky" and display higher fitness or higher novelty than it would display on average if more evaluations were performed. Due to this uncertainty, solutions might be kept in the wrong cell, leading to a loss in the actual coverage of the BD-space. Lucky solutions might also be kept in the place of truly good-performing ones, resulting in a loss in the actual quality of the collection. To handle such uncertain environments, the most widely used approach is to sample multiple times solutions [Cully and Demiris 2018a]. In this approach, that we refer to as **MAP-Elites-sampling**, each solution is evaluated a fixed number of times M before being added to the archive. The fitness and BD of a solution are approximated using the mean fitness and BD of these re-evaluations. However, this approach is costly and impact drastically the algorithm speed-of-convergence. Thus, Justesen et al. [2019] proposed **Adaptive-sampling**: an alternative sampling-based approach that distributes samples more wisely across solutions. This solution mitigates the cost of MAP-Elites-sampling but stays significantly slower in convergence

than the original MAP-Elites. Alternatively, Flageat and Cully [2020] proposed **Deep-grid**, a MAP-Elites variant that tackle uncertain environments without sampling. In Deep-grid, sampling is done implicitly by storing D previously encountered solutions in each cell and considering them as samples of the same solution.

2.4.2 Reproducibility in uncertain environments. Most uncertainty-handling approaches focus on computing a better approximation of the "ground-truth" fitness and BD of solutions. However, in certain tasks, some solutions are more reproducible than others and thus more robust to uncertainty. Said differently, while it is important to approximate the expected value of the fitness and BD distributions of each solution, it is as valuable to minimise the variance of these distributions as this leads to more reproducible solutions. A challenge of QD approaches in such domains is to generate solutions as reproducible as possible, while producing large collections of truly diverse and high-performing solutions. Adaptive-sampling proposes an explicit mechanism to delete from the grid solutions whose BD is evaluated too often outside of their cell. Deep-grid on the other side enforces reproducible solutions via its selection and addition mechanisms. These mechanisms systematically question solutions, even after their addition to the grid, and slowly reject from the grid non-reproducible solutions. We hypothesise that algorithms that use empirical estimation of the gradient for variation should enforce reproducible solutions. Gradient approximations are based on sampling and are thus be less sensitive to luck. It is the case of MAP-Elites-ES, described in Section 2.3, which, on top, also uses 30 re-sampling of the offspring to tackle uncertainty. Finally, we demonstrate in this paper that the use of policy-gradient variation operators also promotes reproducible solutions in QD collections.

2.5 PGA-MAP-Elites

In this work, we study PGA-MAP-Elites, a new QD algorithm proposed by Nilsson and Cully [2021] and illustrated in Fig. 1 and Alg. 1. PGA-MAP-Elites allows to apply the QD approach to high-dimensional search spaces by combining ideas from MAP-Elites and TD3. It follows the usual MAP-Elites loop (selection, variation, evaluation, addition) but uses two independent variation operators: a standard Genetic Algorithm (GA) operator from MAP-Elites, and a Policy Gradient (PG) operator inspired by TD3. The PG operator relies on policy-gradients derived from critics trained in parallel to the main loop. In summary, PGA-MAP-Elites adds three components to standard MAP-Elites: (1) a replay buffer collecting experience from evaluations, (2) two critic networks and their associated policy, trained using the replay buffer, and (3) a PG variation operator.

2.5.1 Replay buffer. At each iteration of the algorithm loop, a fixed number of offspring policies are evaluated in the environment. A replay buffer \mathcal{B} collects every transition (s_t, a_t, r_t, s_{t+1}) of these experiences. \mathcal{B} has a limited capacity, and old transitions are removed on a first-in-first-out basis.

2.5.2 Critics and greedy actor. Like TD3, PGA-MAP-Elites trains two critic networks Q_{θ_1} and Q_{θ_2} to approximate the action-value function, and an associated "greedy" actor π_ϕ . PGA-MAP-Elites keeps π_ϕ separate from the actors in the archive to compute the critic update target in Eq. 3. This way the target is computed using an actor that approximate the optimal action in each state, not conditioned by a bd. The two critics and the greedy actor are trained at each MAP-Elites loop for n_{crit} steps of gradient descent, with the loss given by Eq. 2 and the same sampled noise ϵ that implicitly favours reproducible behaviours. This part of PGA-MAP-Elites is exactly the same as in TD3. It only starts after initialisation to ensure a good initial distribution of experience in the replay buffer. Finally, a copy of the greedy controller is always considered as an offspring for evaluation as it may provide useful behaviours.

Algorithm 1: PGA-MAP-Elites algorithm.

Data: n_{gen} total generations, b batch-size, p_{evo} proportion of GA and PG variations, n_{crit} and n_{act} PG parameter, and σ_1 and σ_2 GA parameters

Result: the final archive \mathcal{A}

```

for  $i = 0 \rightarrow n_{gen}$  do
  if  $i = 0$  then
     $\pi_{\hat{\phi}_1}, \dots, \pi_{\hat{\phi}_b} = \text{random\_solutions}()$ 
  else
    // Train critics and greedy actor
    for  $k = 1 \rightarrow n_{crit}$  do
       $Q_{\theta_1}, Q_{\theta_2}, Q_{\theta'_1}, Q_{\theta'_2} = \text{train\_critics}(\mathcal{B}, Q_{\theta_1}, Q_{\theta_2}, Q_{\theta'_1}, Q_{\theta'_2}, \pi_{\phi'_{greedy}})$ 
       $\pi_{\phi_{greedy}}, \pi_{\phi'_{greedy}} = \text{train\_actor}(\mathcal{B}, Q_{\theta_1}, \pi_{\phi_{greedy}}, \pi_{\phi'_{greedy}})$ 
    // Apply GA variation
    for  $j = 1 \rightarrow \lfloor p_{evo} \cdot b \rfloor$  do
       $\pi_{\phi_{p_1}}, \pi_{\phi_{p_2}} = \text{uniform\_selection}(\mathcal{A})$ 
       $\pi_{\hat{\phi}_j} = \pi_{\phi_{p_1}} + \sigma_1 \mathcal{N}(\mathbf{0}, \mathbf{I}) + \sigma_2 (\pi_{\phi_{p_2}} - \pi_{\phi_{p_1}}) \mathcal{N}(0, 1)$ 
    // Apply PG variation
    for  $j = \lfloor p_{evo} \cdot b \rfloor \rightarrow b - 1$  do
       $\pi_{\phi_p} = \text{uniform\_selection}(\mathcal{A})$ 
      for  $k = 1 \rightarrow n_{act}$  do
         $\pi_{\hat{\phi}_j} = \text{train\_actor}(\pi_{\hat{\phi}_j}, \mathcal{B}, Q_{\theta_1}, \pi_{\phi_p})$ 
      // Return the greedy actor as offspring
       $\hat{\phi}_b = \phi_{greedy}$ 
    // Evaluate offspring and update  $\mathcal{B}$  and  $\mathcal{A}$ 
     $fit, bd, transitions = \text{evaluate}(\pi_{\hat{\phi}_1}, \dots, \pi_{\hat{\phi}_b})$ 
     $\mathcal{B} = \text{add\_to\_replay\_buffer}(\mathcal{B}, transitions)$ 
     $\mathcal{A} = \text{add\_to\_archive}(\mathcal{A}, fit, bd)$ 

```

2.5.3 PG and GA variation operators. At each iteration of the algorithm's main loop, the randomly selected parents are updated by variation operators to generate offspring. The core idea of PGA-MAP-Elites is to introduce a new PG variation operator, based on the critic networks Q_{θ_1} and Q_{θ_2} and the replay buffer \mathcal{B} . This operator updates a solution by applying n_{act} consecutive steps of gradient ascent based on N transitions sampled uniformly from \mathcal{B} . It thus directs variations toward high-performing regions of the search space. To maintain the divergent search methodology of standard MAP-Elites, the PG variation operator is paired with a GA variation operator, here we use the directional variation introduced by [Vassiliades and Mouret \[2018\]](#). This variation is based on two parents sampled uniformly from the archive. It combines a displacement in the parameter space along the line between the two parents and a random perturbation. The PG and GA operators used in PGA-MAP-Elites are independent and used to generate distinct offspring controllers. The proportion of controllers generated by each of them is determined by the parameter p_{evo} . At each iteration, the next batch of offspring controllers to be evaluated is of size b , among which $\lfloor p_{evo} \cdot b \rfloor$ are generated through GA variation and $\lceil (1 - p_{evo}) \cdot b \rceil$ through PG variation. Among the controllers generated through PG, one is always a copy of the greedy controller associated with the critics.

3 EXPERIMENTAL METHODOLOGY

This work proposes an in-depth analysis of PGA-MAP-Elites divided in three main parts:

- Section 4 studies the performance of PGA-MAP-Elites on tasks with high-dimensional search space, and with or without uncertainty on the fitness and BD estimations.
- Section 5 proposes a study of the reproducibility of the solutions found by PGA-MAP-Elites.
- Section 6 displays an ablation of the main parameter of PGA-MAP-Elites: the proportion of GA and PG variations; and a study of the dynamics of these variation operators.

All these results use the same environments, baselines, implementations, and metrics. We introduce them in this section.

3.1 Environments

3.1.1 Tasks. We consider the four QD-Gym tasks defined in Nilsson and Cully [2021], derived from the standard DRL benchmark OpenAI Gym [Brockman et al. 2016] in PyBullet [Coumans and Bai 2019]. In all these tasks, a robot aims to discover all the ways it can walk while maximising a trade-off between speed and energy consumption. The details of these tasks are given in Table 1.

Table 1. QDGym tasks used in this work.

	QDWALKER	QDHALFCHEETAH	QDANT	QDHOPPER
STATE DESCRIPTION	Position and velocity of the centre of gravity and joints			
ACTION DESCRIPTION	Continuous-valued torques in each joint			
FITNESS DEFINITION	Forward progress reward + surviving reward + energy usage penalty			
BD DEFINITION	Proportion each foot of the robot is in contact with the ground			
STATE DIMENSIONS	22	26	28	15
ACTION DIMENSIONS	6	6	8	3
BD DIMENSIONS	2	2	4	1
NR BD NICHES	1024	1024	1296	1000
EPISODE LENGTH	1000 simulation steps			
CONTROLLER	Deep Neural Network with 2 hidden layers of 128 neurons			
NR PARAMETERS	20230	20742	21256	18947

3.1.2 Uncertainty. In the QD-Gym tasks, the initial joint positions are sampled from a Gaussian distribution, making these tasks uncertain. This stochasticity has a limited impact and intuitively one might think that controllers should reach a stationary gait, independent of the initialisation. However, these stationary behaviours can only be reached at a long horizon, and the simulation time is limited to a few seconds. In addition, for some controllers, the initialisation can lead to a fall which simply stops the simulation. Thus the fitness and BD of a given controller are computed based on a transitional gait, and as such, they are uncertain (see Section 2.4). In addition, we hypothesise that, in the QD-Gym tasks, some controllers are more reproducible than others and thus more robust to this uncertainty.

3.2 Baselines

In the next sections, we consider baselines among the list below, also detailed in Table 2. All the baselines we re-implemented use a CVT-shaped MAP-Elites-archive [Vassiliades et al. 2017] to store solutions. For MAP-Elites-ES and QD-PG only we use the original implementation from the authors with standard MAP-Elites grid-based archive. All baselines use the same number of cells so this difference in archive type does not impact the results.

- **MAP-Elites** (Section 2.1) with directional variation [Vassiliades and Mouret 2018].
- **TD3** (Section 2.2) the original TD3 algorithm from [Fujimoto et al. 2018] augmented with a CVT-MAP-Elites archive, used to passively collect behaviours for comparison purposes.
- **CMA-MAP-Elites** (Section 2.3) with 5 improvement emitters [Fontaine et al. 2020b].
- **MAP-Elites-sampling** (Section 2.4) with $M = 10$ samples.
- **Deep-grid** (Section 2.4) with depth $D = 50$ as in the original paper.
- **MAP-Elites-ES** (Section 2.3)
- **QD-PG** (Section 2.3)

Table 2. List of the baselines used in this work and which comparison they figure in. We also outline the type of mutation used for each baseline and the origin of the implementation used to generate the results.

	SEC. 4	SEC. 5	Variation	CODE
PGA-MAP-ELITES	✓	✓	GA & policy-gradient	Original code
MAP-ELITES-ES	✓	✓	natural-gradient	Original code
QD-PG	✓		diversity-gradient & policy-gradient	Original code
CMA-MAP-ELITES	✓		CMA-ES	PyRibs lib
TD3	✓		policy-gradient	Ours
MAP-ELITES	✓	✓	GA	Ours
MAP-ELITES-SAMPLING		✓	GA	Ours
DEEP-GRID		✓	GA	Ours

3.3 Implementation and hyperparameters

Our source code is available at <https://github.com/adaptive-intelligent-robotics/pga-map-elites>. It includes a containerised environment to replicate our experiments. Our code is based on source code from Nilsson and Cully [2021]. Common hyperparameters are identical between PGA-MAP-Elites and MAP-Elites, TD3, MAP-Elites-sampling and Deep-grid, and correspond to the ones used in the original PGA-MAP-Elites paper [Nilsson and Cully 2021] apart from two parameters of the PG variation: the number of actor training steps n_{act} and the actor learning rate lr_{act} . We increased these two parameters values in PGA-MAP-Elites only, as it proves to significantly improve its performance. All hyperparameters for these algorithms are given in Appendix A. CMA-MAP-Elites [Fontaine et al. 2020b] is implemented using the PyRibs library proposed by its authors [Tjanaka et al. 2021] with default parameters. For MAP-Elites-ES [Colas et al. 2020] and QD-PG [Pierrot et al. 2022] we use the implementation provided by the original authors.

3.4 Evaluation Metrics

In the following sections, we consider the following metrics and report p-values based on the Wilcoxon rank-sum test with Bonferroni correction (summarised in Appendix E).

3.4.1 *Main metrics:* In the following, we consider two main metrics:

- **QD-score:** [Pugh et al. 2016] Sum of all fitnesses of the archive, offsetted by the lowest fitness value. It quantifies the diversity and performance of the final collection and allows comparison with QD approaches.
- **Max-fitness:** Fittest solution in the archive, allow comparison with DRL approaches.

For the sake of conciseness, we are not reporting the Coverage metric comparisons within the core of the paper, but we provide them in Appendix D.

3.4.2 *Corrected metrics:* To take into account the effects of uncertainty, we re-evaluate each solution kept by the algorithm a fixed number of times N , and use the average fitness and BD of these N re-evaluations as approximation of the “ground truth” fitness and BD [Flageat and Cully 2020; Justesen et al. 2019]. We place the re-evaluated solutions in a “Corrected archive”, using the same archive addition rules with the “ground truth” fitness and BD. We then compute the QD-score and Max-fitness of this Corrected archive. We refer to them as **Corrected QD-score** and **Corrected Max-fitness**. The N re-evaluations are only used for metric computation: the algorithms cannot access them for optimisation.

3.4.3 *Loss metrics:* We also propose to study the loss induced by the re-evaluations, as a way to estimate the reproducibility of the solutions generated by an algorithm. For each metric, we define the loss as the difference between the original 1-sample-based metric and the Corrected N -samples-based metric, normalised by the original value of the metric reached by the algorithm:

- **QD-score loss:** Normalised loss between QD-score and Corrected QD-score.
- **Max-fitness loss:** Normalised loss between Max-fitness and Corrected Max-fitness.

4 PERFORMANCE OF PGA-MAP-ELITES

We first study the performance of PGA-MAP-Elites on the QD-Gym suite, considering the baselines from Section 3.2 and the new hyper-parameters of PGA-MAP-Elites described in Section 3.3. The results are displayed in Fig. 2. The QD-Gym tasks are both high-dimensional and uncertain, so we also propose to decorrelate these two distinct challenges and demonstrate the performance of PGA-MAP-Elites on deterministic QD-Gym tasks. The uncertainty in the QD-Gym tasks arises from the random initialisation of the robots’ joints positions. Here, we consider a new version of the QD-Gym tasks where all joints are initialised to 0 at every episode. The dynamics in PyBullet being deterministic [Coumans and Bai 2019], this modification leads to a deterministic variant of the QD-Gym environments. Fig. 3 displays the performance across all deterministic tasks. Fig. 4 pictures one final archive per algorithm for the Deterministic QDWalker and Deterministic QDHalfCheetah tasks. Appendix E summarises the p-values of these comparisons.

Due to computational resource limitations, each run is given a maximum budget of 3 days of computation on 32 cores. All algorithms except CMA-MAP-Elites manage to reach 10^6 evaluations within this budget. CMA-MAP-Elites relies on costly covariance matrix inversion having $\approx 4.10^8$ parameters, which explains why it does not reach the maximum number of evaluations.

4.1 QD-score

The QD-score metric in Fig. 2 and 3 shows that PGA-MAP-Elites outperforms all baselines in terms of quality and diversity of the final archive, and reaches the best collection across all tasks ($p < 1e-6$). This observation holds on the archives in Fig. 4. MAP-Elites and CMA-MAP-Elites both find significantly less-performing archives than PGA-MAP-Elites. In the case of CMA-MAP-Elites, this could be due to the early stopping of the algorithm. However, in the case of MAP-Elites, it is likely due to the ineffectiveness of GA variation in large search space [Colas et al. 2020]. On

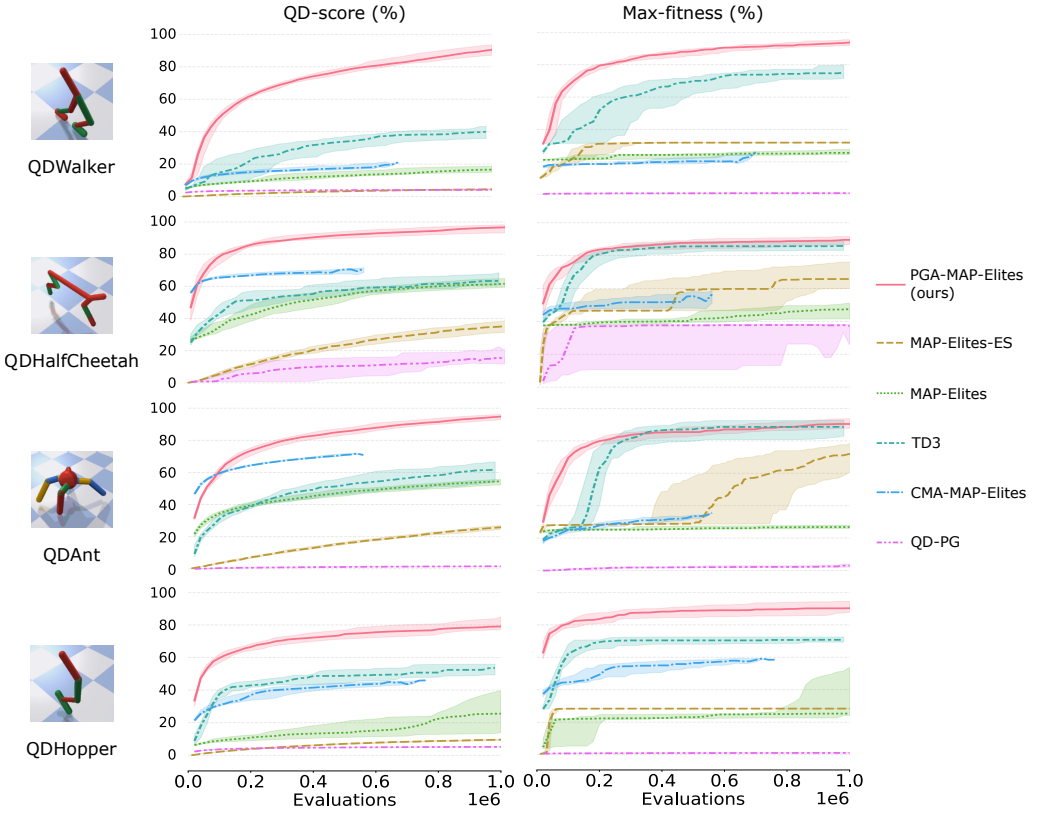


Fig. 2. **QDGym results:** Comparison of the QD-score (left) and Max-fitness (right) of all algorithms on each QDGym task for 10^6 controller evaluations. Each experiment is replicated 20 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

the contrary, the PG variation operator of PGA-MAP-Elites allows directing the variation toward promising regions of the search space. Yet, PGA-MAP-Elites also outperforms TD3, which also relies on PG. But this result is expected as TD3 only focuses on optimising the best solutions and not the intermediate solutions encountered during the optimisation process that populate the archive. PGA-MAP-Elites also performs better than QD-PG and MAP-Elites-ES, two QD approaches designed for high-dimensional search-spaces. MAP-Elites-ES has a higher sample-cost than PGA-MAP-Elites as it relies on sampling to approximate the gradient, and its QD-score does not seem to have fully converged despite the high number of evaluations available. On the contrary, QD-PG seems to have converged, but does not succeed to find diverse and high-performing solutions in the considered tasks. QD-PG assumes that the sum of the novelty of each state provides a meaningful characterisation for the novelty of the entire behaviour. However, there are no guarantees that it is the case for the QD-Gym tasks.

These results are similar in uncertain and deterministic tasks, except for QDAnt: interestingly, all baselines reach a lower QD-score in the deterministic version of this task (Fig. 3) than in its original version (Fig. 2). This task has the larger search space of the QD-Gym domains, and the Ant is the only 4-legged robot in the suite, which might lead to a more complex subspace of promising policies, making it harder to discover and explore. The randomness of the initial positions might

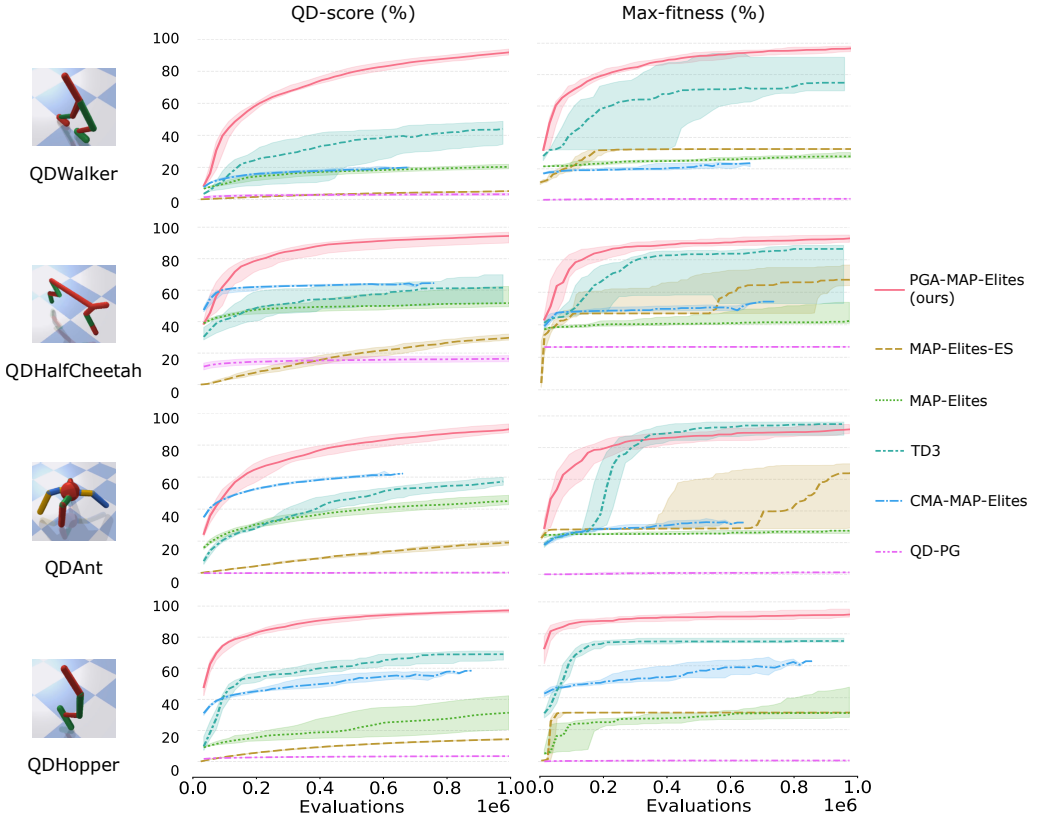


Fig. 3. **Deterministic QDGym results:** Comparison of the QD-score (left) and Max-fitness (right) of all algorithms on each Deterministic task for 10^6 controller evaluations. Each experiment is replicated 20 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

help the algorithms with this exploration, leading to a better QD-Score. Still, PGA-MAP-Elites significantly outperforms all other algorithms in terms of QD-score across all tasks. According to this first metric, PGA-MAP-Elites is the only considered algorithm that finds high-performing solutions throughout the BD-space in all QD-Gym domains.

4.2 Max-fitness

PGA-MAP-Elites finds the best solutions of all QD approaches ($p < 1e-6$). TD3 also finds these high-performing solutions in QDHalfCheetah and QDAnt and in their deterministic versions, but PGA-MAP-Elites outperforms it in QDWalker and QDHopper and their Deterministic versions ($p < 1e-7$). Thus, PGA-MAP-Elites retains the optimisation capability of DRL approaches.

4.3 Conclusion

These results show that PGA-MAP-Elites maintain the capabilities of QD algorithms to generate diverse and high-performing populations while retaining the faculties of DRL approaches to optimise single solutions for performance. As uncertainty might impact some of the baselines [Flageat and Cully 2020], this section also proves that PGA-MAP-Elites outperforms all approaches

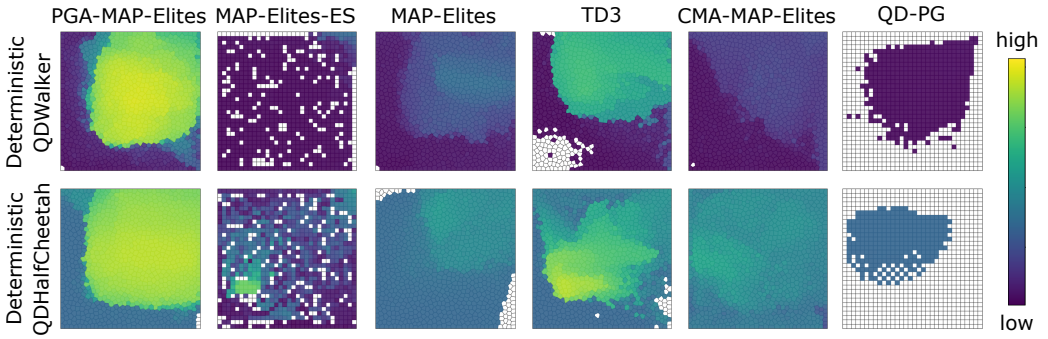


Fig. 4. Final archives in the Deterministic QDWalker (top) and Deterministic QDHalfCheetah (bottom) tasks after 10^6 controller evaluations. The BD in these tasks are 2-dimensional, giving a square grid. Increasing feet contact time from left to right and from bottom to top. The colour of each cell represents the fitness of the controller it contains (the lighter the better), grey colour corresponds to an empty cell.

in Deterministic QD-Gym. Thus, PGA-MAP-Elites allows applying the QD-approach to high-dimensional neuroevolution tasks, independently of their uncertain nature.

5 REPRODUCIBILITY IN UNCERTAIN ENVIRONMENTS

Previous works have shown that QD algorithms applied to uncertain environments tend to favour lucky solutions and struggle to find solutions that are truly diverse and high-performing (see Section 2.4). On the contrary, the critic networks in TD3 encourage reproducible behaviours (see Section 2.2). As PGA-MAP-Elites built on both MAP-Elites and TD3, it should find more reproducible controllers than vanilla QD approaches in uncertain environments. This section analyses the impact of uncertainty on the performance of PGA-MAP-Elites. It compares the reproducibility of the controllers it produces to those of MAP-Elites, MAP-Elites-ES as well as approaches designed specifically for such applications: MAP-Elites-sampling and Deep-grid (see Section 2.4).

Fig. 6 displays the original and Corrected metrics on the QDWalker task, and Fig. 5 the corresponding original and Corrected archives. As detailed in Section 3.4, to compute the Corrected metrics, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The results for the other tasks of the QD-Gym suite can be found in Appendix B. Fig. 7 gives the loss in performance induced by uncertainty across all tasks. The loss corresponds to the difference of each Corrected metric in proportion of the original metric. Lower loss indicates better reproducibility. As already observed in previous uncertainty studies [Flageat and Cully 2020; Justesen et al. 2019], the re-evaluation process results in a loss in performance for all the algorithms.

5.1 Corrected QD-score and QD-score loss

The results in Fig. 6 and Appendix B show that PGA-MAP-Elites outperforms all baselines in Corrected QD-score ($p < 1e-9$). This result is expected as PGA-MAP-Elites was already outperforming all other approaches before the re-evaluation in Sec. 4. However, the QD-score loss in Fig. 7 indicates that PGA-MAP-Elites also manages to find reproducible solutions. In QDWalker, QDHalfCheetah and QDAnt, the QD-score losses of MAP-Elites are consistently higher than those of other algorithms except for Deep-grid ($p < 5e-7$). Only in QDHopper does MAP-Elites observe similar performance-drop. This task seems to produce more reproducible solutions overall. On the contrary, MAP-Elites-sampling has a significantly lower QD-score loss than MAP-Elites on all tasks

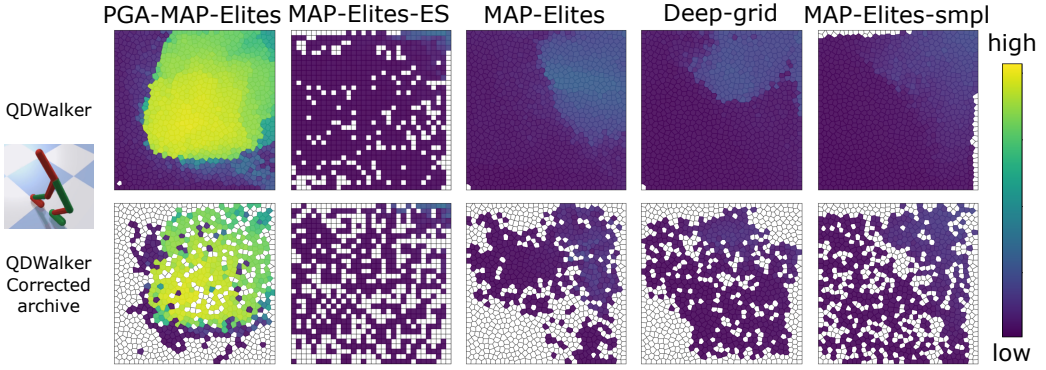


Fig. 5. Final archives (top) and final Corrected archives (bottom) found by all algorithms in the QDWalker task after 10^6 controller evaluations. The BDs in this task are 2-dimensional, giving a square grid. Increasing feet contact time from left to right and from bottom to top. The colour of each cell represents the fitness of the controller it contains (the lighter the better), grey colour corresponds to an empty cell. The Corrected archive is filled with the controllers from the standard MAP-Elites archive with the same archive-addition rules but using the average fitness and BD over 50 replications of this controller.

($p < 1e-7$). Compared to vanilla MAP-Elites, MAP-Elites-sampling gets a better estimate of the true fitness and BD of a solution through sampling, thus it is less likely to keep lucky solutions which reduce the QD-score-drop. The implicit mechanism of Deep-grid fails to find reproducible solutions as efficiently as MAP-Elites-sampling, maintaining QD-score losses close to those of MAP-Elites and significantly higher than other baselines. Deep-grid uses neighbours in the BD-space to approximate the true fitness and BD of a solution. However, QD-Gym tasks have a high-dimensional search space mapped to a low-dimensional BD space. Thus, the relation between genotype and BD space is probably less straightforward than in usual QD-tasks and can be confusing for this BD-neighbourhood-based mechanism. Interestingly, MAP-Elites-ES constantly gets lower QD-score losses than all other baselines ($p < 5e-7$ except for QDHopper: $p < 5e-4$). As mentioned in Sec. 2.4, this algorithm uses a sampling-based approximation of the gradient to direct the variation toward high-performing or diverse solutions. This approximation of the search space structure makes the variation operator less sensitive to luck. In addition, MAP-Elites-ES re-evaluates each solution considered for addition 30 times, where MAP-Elites-sampling only re-evaluates them 10 times. MAP-Elites-ES selects its offspring thoughtfully, unlike MAP-Elites-sampling, it can thus afford to spend more time on re-evaluation. MAP-Elites-sampling with 30 re-evaluations would not even fill the grid, spending too much time on sampling the same unpromising solutions. Thanks to these two mechanisms MAP-Elites-ES finds more reproducible solutions than all other baselines, making it a strong competitor in term of reproducibility of solutions. PGA-MAP-Elites also finds highly reproducible solutions across all tasks, and even manage to find more reproducible solutions than MAP-Elites-sampling in QDHalfCheetah ($p < 5e-5$). Yet, PGA-MAP-Elites does not have any explicit mechanisms to handle uncertainty in its archive-management contrary to Deep-grid, MAP-Elites-ES or MAP-Elites-sampling. It manages to find reproducible solutions while managing the archive the exact same way as MAP-Elites. Thus, the reproducibility of the solutions found by PGA-MAP-Elites is probably inherent to its variation mechanisms. This result corroborates the hypothesis that PG variations bring the optimisation process to reproducible areas of the search space contrary to GA variations.

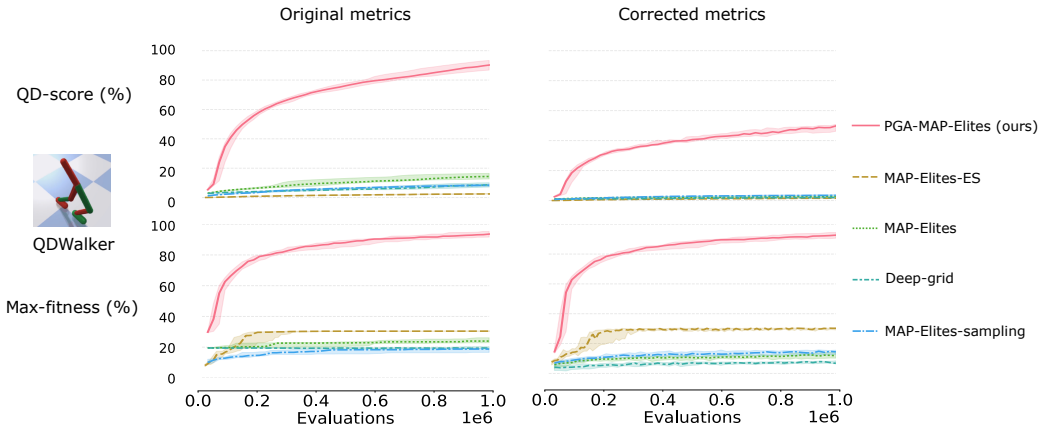


Fig. 6. **Corrected results:** Comparison of the original (left) and corrected (right) QD-score (top) and Max-fitness (bottom) of all algorithms on QDWalker. To compute the Corrected metrics, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. Each experiment is replicated 20 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

5.2 Corrected Max-fitness and Max-fitness loss

PGA-MAP-Elites also outperforms all baselines according to the Corrected Max-fitness ($p < 1e-9$) and also find significantly more reproducible solutions according to the Max-fitness loss, achieving less than 20% loss across all tasks and even less than 10% loss in QDWalker and 5% loss in QDHalfCheetah and QDAnt. MAP-Elites-ES achieves similar score on QDWalker and QDAnt but higher loss on QDHalfCheetah (still less than 20%) and lower loss on QDHopper, while MAP-Elites-sampling achieves similar scores on QDAnt and QDHopper, but significantly higher loss in QDWalker and QDHalfCheetah. In comparison, MAP-Elites reaches significantly higher Max-fitness loss than PGA-MAP-Elites across all tasks ($p < 1e-7$ except QDHopper: $p < 2e-2$), supporting previous results. Deep-grid obtains really poor scores on this metric, getting even bigger loss values than MAP-Elites, especially in QDWalker. As highlighted before, Deep-grid has not been designed for such high-dimensional search spaces. Also, the original Deep-grid work has already shown that the main advantage of the Deep-grid algorithm lies in the BD-reproducibility of its solutions, and less in their fitness-performance or fitness-reproducibility. Deep-grid has a really high variance in results compared to other approaches, causing their boxplots to appear flattened on Fig. 7. Thus, we added the same graph without Deep-grid in Appendix B, to allow to visualise the full range of variation of the other baselines. These results illustrate again the high reproducibility of the solutions found by PGA-MAP-Elites, that has no explicit uncertainty-handling mechanisms.

5.3 Conclusion

PGA-MAP-Elites is the only algorithm considered in this study, apart from vanilla MAP-Elites, that does not have any explicit mechanism to handle uncertainty. In comparison, MAP-Elites-sampling and MAP-Elites-ES both uses sampling to get a better approximate of the true BD and fitness of solutions, while Deep-grid uses neighbouring individuals for the same purpose. In this sense, PGA-MAP-Elites is closer to vanilla MAP-Elites, and despite this similarity it proves to generate solutions that are significantly more reproducible, on top of being diverse and high-performing. Thus, the reproducibility of the solutions found by PGA-MAP-Elites is inherent to its variation

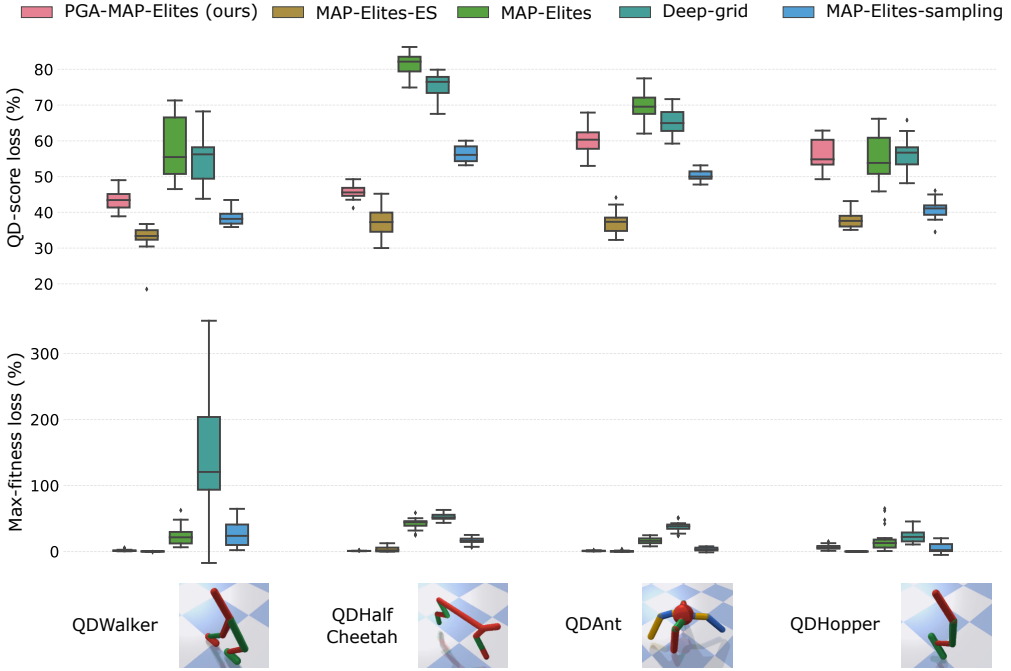


Fig. 7. **Loss comparison:** Comparison of the QD-score loss (top) and Max-fitness loss (bottom) due to the lack of reproducibility of the solutions. To compute these losses, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The loss corresponds to the difference of each metric in proportion of the original metric. Lower loss indicates better reproducibility. The boxes represent the distribution of the loss for a given algorithm across 20 runs.

mechanism, based on policy-gradients. This allows PGA-MAP-Elites to maintain the diversity and the quality of its final archive, making it a strong QD baseline for uncertain applications.

6 ABLATION OF PGA-MAP-ELITES

The core principle of PGA-MAP-Elites is to introduce a new PG variation operator for MAP-Elites. At each generation, PGA-MAP-Elites generates b offsprings from parents selected from the archive. Among these offsprings, $\lfloor p_{evo} \cdot b \rfloor$ are generated using GA, and $\lceil (1 - p_{evo}) \cdot b \rceil$ using PG. The last of the PG-generated solutions is a copy of the greedy actor of the critics. In this section, we propose an ablation study to better understand the impact of these variation operators.

6.1 Impact of the proportion of PG and GA variation

We first study the impact of the proportion of PG and GA variations: p_{evo} .

6.1.1 Baselines: In the above experiments, the implementation of PGA-MAP-Elites uses $p_{evo} = 0.5$ and $b = 100$. In other words, 50 offsprings are generated through GA and 50 through PG, among which 1 is a copy of the greedy actor. In this section, we compare multiple versions of PGA-MAP-Elites with $b = 100$ but different values of the p_{evo} parameter:

- $p_{evo} = 1$ - corresponds to the standard MAP-Elites algorithm with only GA variations.
- $p_{evo} = 0.75$ - corresponds to 75% of GA and 25% of PG.
- $p_{evo} = 0.5$ - is the standard PGA-MAP-Elites algorithm.

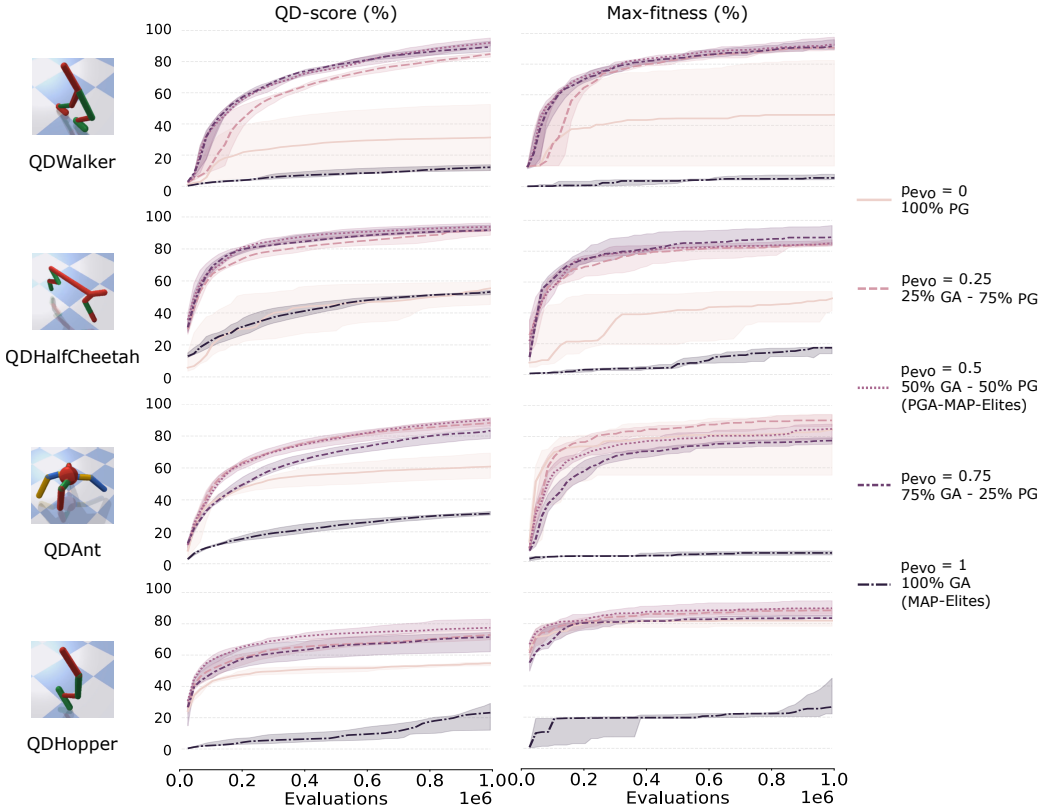


Fig. 8. **Ablation p_{evo} - Main results:** Comparison of the QD-score (left) and Max-fitness (right) of PGA-MAP-Elites with different mutation proportions on the QDGym tasks. Each experiment is replicated 10 times, the solid line gives the median over replications and the shaded area to the first and third quartiles.

- $p_{evo} = 0.25$ - corresponds to 25% of GA and 75% of PG.
- $p_{evo} = 0$ - corresponds to full PG variations. Importantly, it does not correspond to TD3. First, PGA-MAP-Elites with $p_{evo} = 0$ selects the parents from the archive, while TD3 does not perform any selection and always modifies the greedy actor. Second, in PGA-MAP-Elites with $p_{evo} = 0$, the training of the greedy and the variations are applied only every generation. In other word, actors are modified only between episodes, while in TD3 the greedy actor is trained within episode.

6.1.2 *Main results:* Fig. 8 summarises the results of this study, the corresponding archives are given in Appendix C. All runs are replicated 20 times. PGA-MAP-Elites with $p_{evo} = 1$, which corresponds to MAP-Elites, under-performs PGA-MAP-Elites with $p_{evo} = [0.25, 0.5, 0.75]$: its QD-score is order-of-magnitude lower ($p < 5e-4$) as well as its Max-fitness ($p < 5e-4$). This first result illustrates the importance of policy-gradients in PGA-MAP-Elites. On the opposite side of the spectrum, PGA-MAP-Elites with $p_{evo} = 0$, corresponding to PG-only, also reaches significantly lower QD-score ($p < 5e-4$) than $p_{evo} = [0.25, 0.5, 0.75]$. This variant manages to find the same max-fitness solutions as the other baselines in QDAnt and QDHopper, but not in QDWalker ($p < 5e-3$) and QDHalfCheetah ($p < 5e-4$). This second result highlights the importance of the divergent fitness-agnostic search of the GA variation operator.

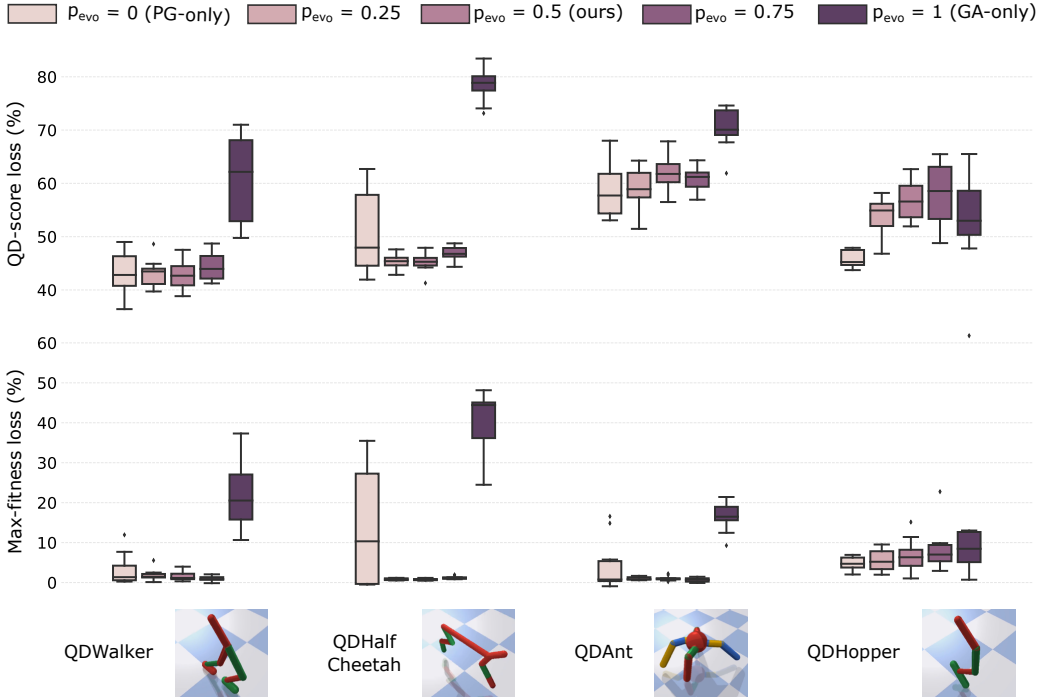


Fig. 9. **Ablation p_{evo} - Loss comparison:** Comparison of the QD-score loss (top) and Max-fitness loss (bottom) due to the lack of reproducibility of the solutions, for PGA-MAP-Elites with different variation operator proportions. To compute these losses, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The loss corresponds to the difference of each metric in proportion of the original metric. Lower loss indicates better reproducibility. The boxes represent the distribution of the loss for a given algorithm across 10 runs.

The last three variants: $p_{evo} = [0.25, 0.5, 0.75]$ perform equally across tasks and metrics. The ideal value of p_{evo} seems to vary with the task. $p_{evo} = 0.25$, which corresponds to PGA-MAP-Elites with predominant PG, converges a bit slower than the two others and even reaches a lower value of QD-score in the QDWalker task ($p < 5e-2$). However, it performs similarly to the others in the QDAnt task. On the contrary, $p_{evo} = 0.75$, PGA-MAP-Elites with predominant GA, performs similarly to the others in QDWalker and QDHalfCheetah but reaches a lower QD-score in QDAnt ($p < 1e-2$). These results illustrate the importance of balancing GA and PG variations in PGA-MAP-Elites and indicate that the algorithm is quite robust to the choice of this hyperparameter as long as it does not take extreme values. Overall, $p_{evo} = 0.5$, which corresponds to the value chosen for PGA-MAP-Elites in the previous sections, shows the best trade-off.

6.1.3 Reproducibility results: Fig. 9 gives the reproducibility analysis for this ablation. The corresponding Corrected metrics and Corrected archives are given in Appendix C. These results draw similar conclusions to previous results of this section but also to those obtained in Sec. 5. Overall, all algorithms are impacted by the re-evaluation mechanisms but PGA-MAP-Elites with $p_{evo} = [0.25, 0.5, 0.75]$ still outperform the extreme values both in term of Corrected metrics and Loss metrics. Again, $p_{evo} = [0.25, 0.5, 0.75]$ are equivalent and $p_{evo} = 0.5$, the value chosen earlier in the paper, shows the best trade-off. Fig. 9 shows that PGA-MAP-Elites with $p_{evo} = 1$, which corresponds to MAP-Elites, is more impacted than the others by the re-evaluation process. As

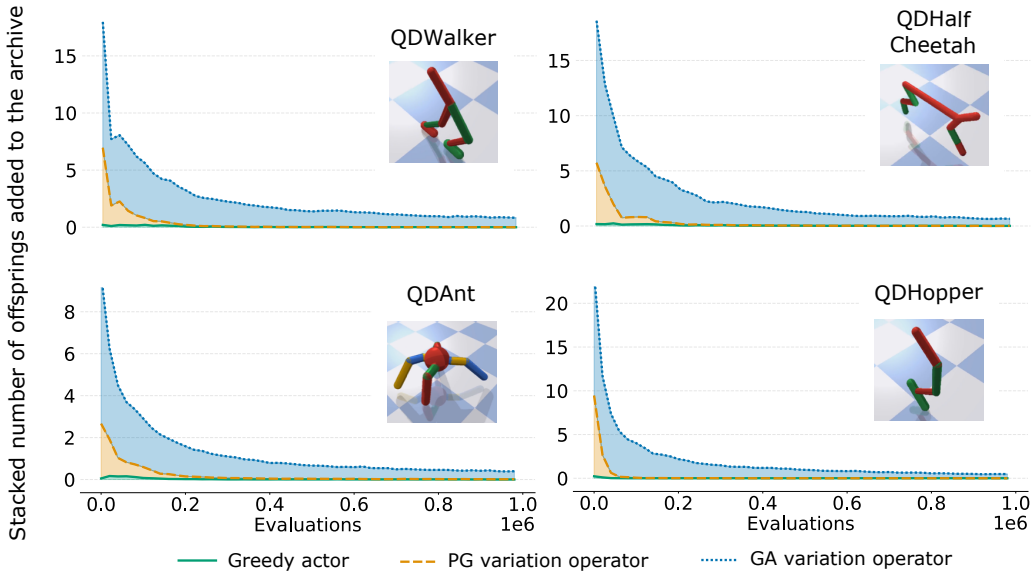


Fig. 10. **Variation operators contribution:** Stacked contribution of each variation operator of PGA-MAP-Elites: GA, PG and Greedy. The number of offspring generated by each operator is fixed: 50 for GA, 49 for PG and 1 for Greedy. This graph gives the number of these offspring added to the archive for each operator. Offspring are added to the archive when they either discover a new cell or improve the existing elite of their cell. The results are averaged over 20 replications of PGA-MAP-Elites, and smoothed on 2000 generations ($2 * 10^4$ evaluations) to increase the readability. They are stacked over operators to facilitate comparison.

all other algorithms rely on critics approximations, this observation corroborates the hypothesis that the robustness mechanism enforced by the PG variation via the TD3 training allows to better handle uncertainty. The Coverage and QD-score loss of PGA-MAP-Elites with $p_{evo} = 0$, which corresponds to PG variations only, are equivalent or even lower in average to those of other baselines, supporting the hypothesis of the importance of critic-based variation for the reproducibility of solutions. However, it displays a greater variance across runs, which is coherent with the high variance observed in the results in Fig. 8. PGA-MAP-Elites with $p_{evo} = 0$ might lack exploration compared to the other algorithms which all use GA variation, which enhance the difference in performance between lucky and unfortunate runs.

6.2 Variation operators contribution

This section studies the contribution to archive improvement of each variation operator.

6.2.1 Metrics. We consider three operators: **GA variation**, **PG variation** and **Greedy actor**. The first two correspond to the two variation operators of PGA-MAP-Elites. **Greedy actor** refers to a copy of the actor associated with the critics. It is considered among the offspring of each generation. We propose to visualise how each of these operators improves the archive across time by counting the number of solutions added to the archive it generates. We represent the median of this quantity across 20 runs against the number of evaluations.

6.2.2 Results. Fig. 10 summarises this analysis. While we observed in previous sections that the PG variation operator is necessary to get good results, here we can see that it is only relevant during the early stage of the process. On the contrary, the GA operator contributes to archive improvement

throughout the process. This result suggests that the PG operator is used to find the region of the search space containing high-performing solutions: the hyper-volume of Elites, as introduced by Vassiliades and Mouret [2018]. The GA operator is then used to exploit this hyper-volume.

Across all tasks, the total number of offspring added to the archive per generation reaches its maximum at the start of the process. These values support previous analysis showing that interesting solutions are harder to find for QDAnt and easier to find in QDHopper. All tasks show a first phase during which the three operators improve the archive. During this phase, the **PG variation** and the **GA variation** have an equivalent impact. The former guide the optimisation process toward promising regions of the search space, and the latter explore these regions. Throughout this first phase, the **Greedy actor**, also guided by the critics toward promising regions, is often added to the archive. This phase is of equivalent length for QDWalker and QDHalfCheetah, but longer for QDAnt, and significantly shorter for QDHopper. During the rest of the process, the impact of all variations decreases. The **PG variation** is the most impacted and adds almost no new solution to the archive, whereas the **GA variation** continues to discover new diverse solutions in the hyper-volume of elites found thanks to the critics. During this second phase, the **Greedy actor** keep on being added to the archive episodically but less frequently. Thus, despite being essential to finding promising solutions in the QD-Gym domains, the PG variation operator is only crucial for the first phase of the process during which it guides the optimisation process toward the hyper-volume of elites, that is then extensively explored by the GA operator. These results corroborate the previous ones, showing the importance of both PG and GA variation operators in PGA-MAP-Elites.

7 CONCLUSIONS AND DISCUSSION

This work proposes an in-depth analysis of PGA-MAP-Elites, a new variant of MAP-Elites that allows to successfully apply the QD approach to uncertain environments with high-dimensional search space. PGA-MAP-Elites extends MAP-Elites with a new DRL-inspired variation operator that leverages the estimation of critic networks and pairs it with the standard GA variation operator for exploration. We first prove that PGA-MAP-Elites significantly outperforms existing methods on four robotic locomotion tasks, demonstrating its ability in uncertain tasks with high-dimensional search-space. We also show that these results hold in deterministic variants of the same tasks. We thus secondly propose an analysis of the reproducibility of the solution generated by PGA-MAP-Elites in such uncertain domains. We demonstrate that these solutions are highly reproducible, approaching the solutions generated by QD baselines designed specifically for uncertain applications. Thirdly, we study in detail the impact of the proportion of PG and GA variations on PGA-MAP-Elites. We demonstrate the importance of the two operators to preserve both the performance and the reproducibility of the solutions. We also show that the performance of PGA-MAP-Elites are robust to the proportion of PG and GA, as long as the two operators are present. Finally, we analyse the contribution of each variation operator to archive improvement. The results indicate that the PG operator is crucial at the beginning of the optimisation process to discover the hyper-volume of elites containing the performing individuals. The GA variation operator remains essential throughout the process to guarantee sufficient exploration of the search space. Overall, these results highlight the importance of policy-gradients in PGA-MAP-Elites and prove their potential to improve MAP-Elites and allow to apply QD to evolve diverse high-quality DNN controllers.

These analyses point toward possible directions for future work. The ablation study indicates that the PG variation is key during the first phase of the optimisation process, where it contributes to improving the collection of solutions. During this phase, it seems to guide the process toward promising regions of the search space that are then explored by the GA. Thus, an interesting direction of research would be to propose an adaptive mechanism for the proportion of GA and PG as variation operators. Also, this work proposes an analysis of PGA-MAP-Elites applied to

robotic locomotion tasks. It would be interesting for future work to study its performance in other domains, possibly uncertain and controlled using high-dimensional DNN controllers, for example manipulation tasks.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/V006673/1 project REcoVER. We also want to thank the members of the Adaptive and Intelligent Robotics Lab at Imperial College London for their useful comments.

REFERENCES

- Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.
- Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *CoRR* abs/1606.01540 (2016). arXiv:1606.01540 <http://arxiv.org/abs/1606.01540>
- Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. 2021. Quality-Diversity Optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*. Springer, 109–135.
- Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. 2020. Scaling MAP-Elites to Deep Neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (Cancún, Mexico) (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 67–75. <https://doi.org/10.1145/3377930.3390217> Implementation: <https://github.com/uber-research/Map-Elites-Evolutionary>.
- Erwin Coumans and Yunfei Bai. 2016–2019. PyBullet, a Python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>. Implementation: https://github.com/bulletphysics/bullet3/blob/master/examples/pybullet/gym/pybullet_envs/gym_locomotion_envs.py.
- Antoine Cully. 2020. Multi-Emitter MAP-Elites: Improving quality, diversity and convergence speed with heterogeneous sets of emitters. arXiv:2007.05352 [cs.NE]
- Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* (2015), 503–507. <https://doi.org/10.1038/nature14422> arXiv:1407.3501
- Antoine Cully and Yiannis Demiris. 2018a. Hierarchical Behavioral Repertoires with Unsupervised Descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference (Kyoto, Japan) (GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 69–76. <https://doi.org/10.1145/3205455.3205571>
- Antoine Cully and Yiannis Demiris. 2018b. Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Transactions on Evolutionary Computation* (2018). <https://doi.org/10.1109/TEVC.2017.2704781> arXiv:1708.09251
- Manon Flageat and Antoine Cully. 2020. Fast and stable MAP-Elites in noisy domains using deep grids. *The 2020 Conference on Artificial Life* (2020). https://doi.org/10.1162/isal_a_00316
- Matthew C Fontaine and Stefanos Nikolaidis. 2021. Differentiable Quality Diversity. *arXiv e-prints* (2021), arXiv–2106.
- Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. 2020a. Covariance Matrix Adaptation for the Rapid Illumination of Behavior Space. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (Cancún, Mexico) (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 94–102. <https://doi.org/10.1145/3377930.3390232>
- Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. 2020b. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*. 94–102.
- Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, StockholmSmässan, Stockholm Sweden, 1587–1596. <http://proceedings.mlr.press/v80/fujimoto18a.html> Implementation: <https://github.com/sfujim/TD3>.
- Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2017. Aerodynamic design exploration through surrogate-assisted illumination. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 3330.
- Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2020. Discovering representations for black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 103–111.
- Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. 2019. Procedural Content Generation through Quality Diversity. In *2019 IEEE Conference on Games (CoG)*. 1–8. <https://doi.org/10.1109/CIG.2019.8848053>
- Nikolaus Hansen. 2016. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772* (2016).

- Yaochu Jin and Jürgen Branke. 2005. Evolutionary optimization in uncertain environments—a survey. *IEEE Transactions on evolutionary computation* 9, 3 (2005), 303–317.
- Niels Justesen, Sebastian Risi, and Jean-Baptiste Mouret. 2019. Map-elites for noisy domains by adaptive sampling. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 121–122.
- Joel Lehman and Kenneth O. Stanley. 2011. Evolving a diversity of creatures through novelty search and local competition. In *Genetic and Evolutionary Computation Conference, GECCO'11*. 211–218. <https://doi.org/10.1145/2001576.2001606>
- Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR. arXiv:1509.02971
- Long-Ji Lin. 1992. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. *Mach. Learn.* 8, 3–4 (May 1992), 293–321. <https://doi.org/10.1007/BF00992699>
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. (dec 2013). arXiv:1312.5602 <http://arxiv.org/abs/1312.5602>
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (feb 2015), 529–533. <https://doi.org/10.1038/nature14236>
- Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. (apr 2015). arXiv:1504.04909 <http://arxiv.org/abs/1504.04909>
- Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 866–875.
- Thomas Pierrot, Valentin Macé, Félix Chalumeau, Arthur Flajolet, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin. 2022. Diversity Policy Gradient for Sample Efficient Quality-Diversity Optimization. (2022).
- Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers Robotics AI* 3, JUL (jul 2016). <https://doi.org/10.3389/frobt.2016.00040>
- Nemanja Rakicevic, Antoine Cully, and Petar Kormushev. 2021. Policy manifold search: Exploring the manifold hypothesis for diversity-based neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 901–909.
- Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. (mar 2017). arXiv:1703.03864 <http://arxiv.org/abs/1703.03864>
- David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *31st International Conference on Machine Learning, ICML 2014*.
- Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567* (2017).
- Bryon Tjanaka, Matthew C Fontaine, Julian Togelius, and Stefanos Nikolaidis. 2022. Approximating gradients for differentiable quality diversity in reinforcement learning. *arXiv preprint arXiv:2202.03666* (2022).
- Bryon Tjanaka, Matthew C. Fontaine, Yulun Zhang, Sam Sommerer, Nathan Dennler, and Stefanos Nikolaidis. 2021. pyribs: A bare-bones Python library for quality diversity optimization. <https://github.com/icaros-usc/pyribs>.
- Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2017. Using Centroidal Voronoi Tessellations to Scale Up the Multi-dimensional Archive of Phenotypic Elites Algorithm. *IEEE Transactions on Evolutionary Computation* (2017), 9. <https://doi.org/10.1109/TEVC.2017.2735550> Implementation: https://github.com/resibots/pymap_elites.
- Vassilis Vassiliades and Jean-Baptiste Mouret. 2018. Discovering the Elite Hypervolume by Leveraging Interspecies Correlation. *Proceedings of the Genetic and Evolutionary Computation Conference* (2018).

A ALGORITHMS HYPERPARAMETERS

PGA-MAP-Elites, MAP-Elites, TD3, MAP-Elites-sampling and Deep-grid are all implemented in a common repository detailed in Sec. 3.3. The corresponding hyperparameters are summarised below ($\#\mathcal{A}$ refers to the number of actions in a task). MAP-Elites-ES and QD-PG are not included in this table as we used the implementation and hyperparameters of the original authors.

PARAMETER	MAP-ELITES VARIANTS	PGA	TD3
NR. OF RANDOM INIT. (I)	500 EPISODES	500 EPISODES	2500 TIMESTEPS
EVALUATION BATCH SIZE (b)	100	100	1
ACTOR NETWORKS	[128, 128, $\#\mathcal{A}$]	[128, 128, $\#\mathcal{A}$]	[128, 128, $\#\mathcal{A}$]
CRITIC NETWORKS	/	[256, 256, 1]	[256, 256, 1]
TRAINING BATCH SIZE (N)	/	256	256
CRITIC TRAINING STEPS (n_{crit})	/	300	/
ACTOR TRAINING STEPS (n_{act})	/	50	/
CRITIC LEARNING RATES (lr_{crit})	/	3×10^{-4}	3×10^{-4}
ACTOR LEARNING RATE (lr_{act})	/	0.005	3×10^{-4}
REPLAY BUFFER MAX. SIZE	/	10^6	10^6
DISCOUNT FACTOR (γ)	/	0.99	0.99
EXPLORATION NOISE (σ_a)	/	0.2	0.2
TARGET UPDATE FREQ. (d)	/	2	2
TARGET UPDATE RATE (τ)	/	0.005	0.005
SMOOTHING NOISE VAR. (σ_p)	/	0.2	0.2
SMOOTHING NOISE CLIP (c)	/	0.5	0.5
GA VARIATION PARAM. 1 (σ_1)	0.005	0.005	/
GA VARIATION PARAM. 2 (σ_2)	0.05	0.05	/

B REPRODUCIBILITY IN UNCERTAIN ENVIRONMENT - COMPLEMENTARY

This section complements the results displayed in Fig. 6 and Fig. 5, and analysed in Sec. 5. Fig. 11 shows the original and corrected metrics for all algorithms across remaining tasks. Fig. 13 shows the corresponding archives for the QDHalfCheetah tasks. We also display the Max-fitness loss for all baselines except Deep-grid in Fig. 12

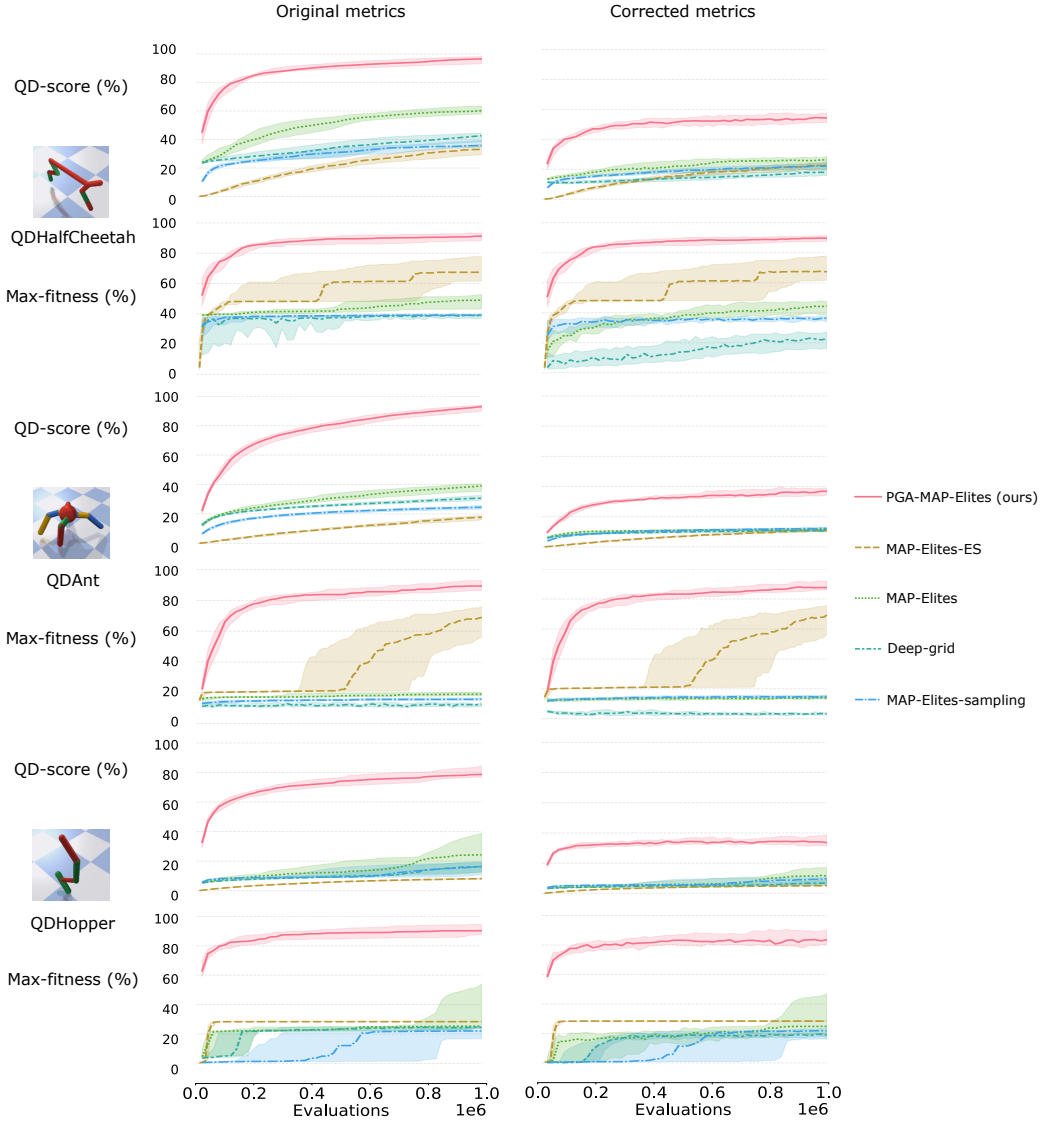


Fig. 11. **Corrected results:** Comparison of the original (left) and corrected (right) QD-score (top) and Max-fitness (bottom) of all algorithms on QDWalker. To compute the Corrected metrics, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. Each experiment is replicated 20 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

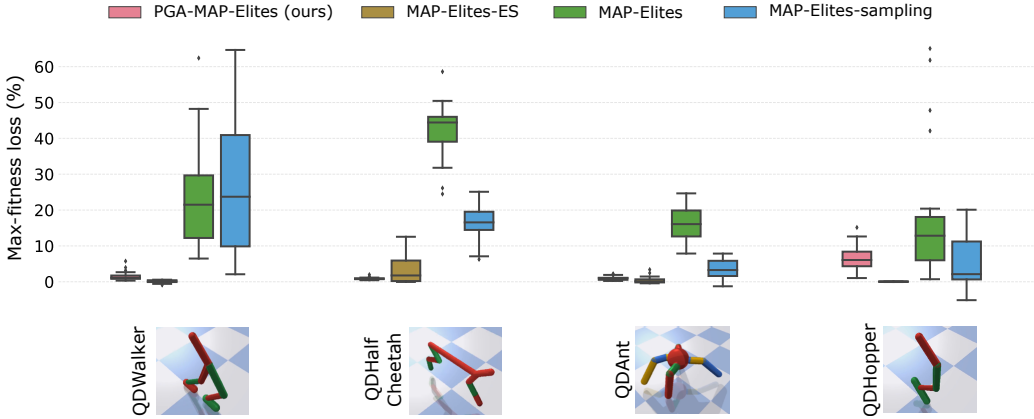


Fig. 12. **Loss comparison:** Comparison of the Max-fitness loss due to the lack of reproducibility of the solutions. To compute these losses, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The loss corresponds to the difference of each metric in proportion of the original metric. Lower loss indicates better reproducibility. The boxes represent the distribution of the loss for a given algorithm across 20 runs.

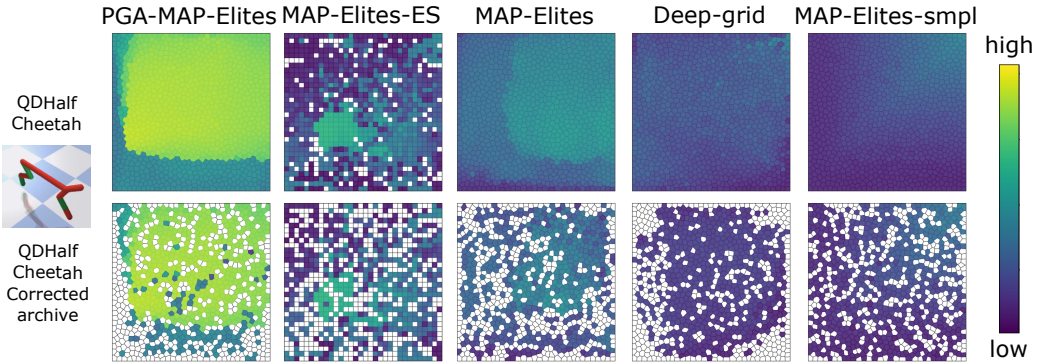


Fig. 13. **Ablation p_{evo} - Corrected archives:** Final archives (top) and final Corrected archives (bottom) found by all algorithms in the QDHalfCheetah tasks after 10^6 controller evaluations. The BDs in these tasks are 2-dimensional, giving a square grid. Increasing feet contact time from left to right and from bottom to top. The colour of each cell represents the fitness of the controller it contains (the lighter the better), grey colour corresponds to an empty cell. The Corrected archive is filled with the controllers from the standard archive with the same archive-addition rules but using the average fitness and BD over 50 replications.

C ABLATION-STUDY - COMPLEMENTARY

C.1 Archives

Fig. 14 display the archives and corrected archives obtained for one run of each baseline considered in the ablation study in Sec. 6. These archives support the results from Sec. 6 and show that $p_{evo} = 0.25$, $p_{evo} = 0.5$ and $p_{evo} = 0.75$ outperform the two extreme baselines $p_{evo} = 0$, only PG, and $p_{evo} = 1$, only GA. This highlights again the complementary of PG and GA variation in PGA-MAP-Elites. However, the archives given in Fig. 14 only represent one run and should be

compared gingerly, especially for PGA-MAP-Elites with $p_{evo} = 0$ which has a really high variance across runs.

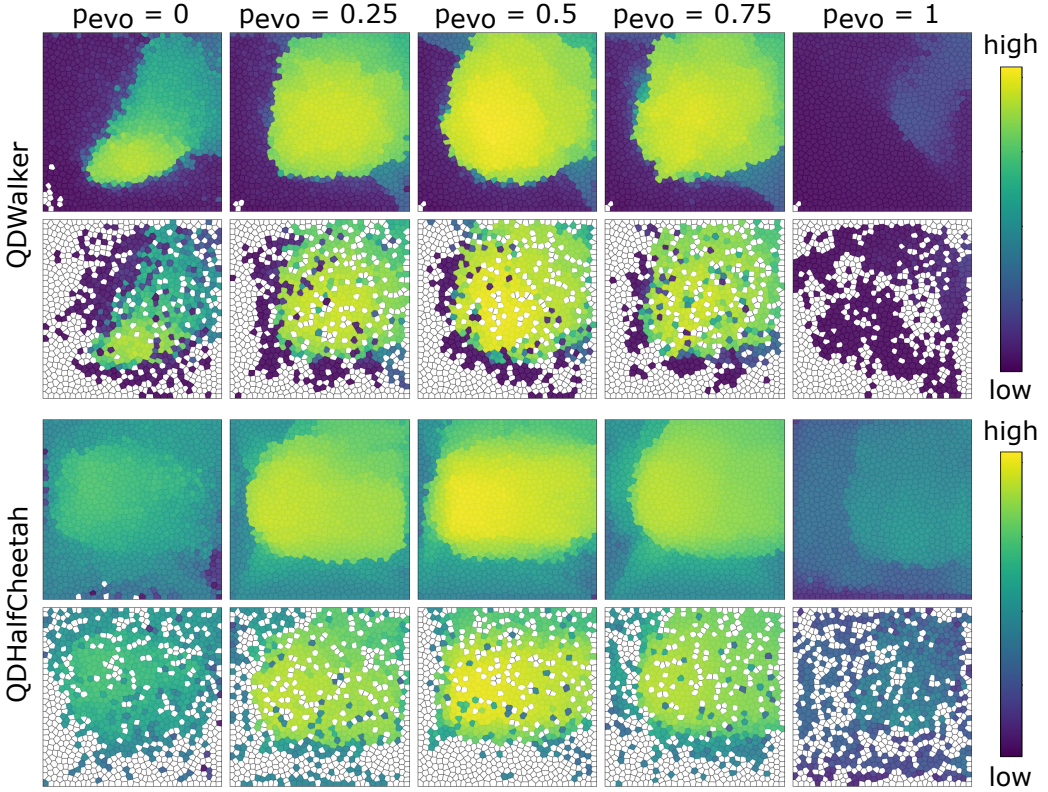


Fig. 14. **Ablation p_{evo} - Corrected archives:** Final archives (top line) and final corrected archives (bottom line) found by PGA-MAP-Elites with different mutation proportion in the QDWalker (top) and QDHalfCheetah (bottom) tasks after 10^6 controller evaluations. The BD in these tasks are 2-dimensional, giving a square grid. Increasing feet contact time from left to right and from bottom to top. The colour of each cell represents the fitness of the controller it contains (the lighter the better), grey colour corresponds to an empty cell. The Corrected fitness and BD of each controller is computed as the average over 50 replications of this controller. The Corrected archive is filled with the same archive rules but using these corrected scores.

C.2 Reproducibility in uncertain environment

We also give the Corrected metrics complementing the ablation study of Sec. 6 in Fig. 15.

D COVERAGE COMPARISON

D.1 Performance of PGA-MAP-Elites

We display in Fig. 16 the Coverage metrics corresponding to Section 4 in both uncertain and deterministic environment. This additional metric shows that PGA-MAP-Elites also retains the divergent search capability of QD-algorithms. It achieves coverages similar to MAP-Elites and CMA-MAP-Elites for the QDWalker and QDHalfCheetah tasks and even outperforms MAP-Elites in QDHopper ($p < 1e-2$) and QDant ($p < 5e-8$). In this last task, CMA-MAP-Elites manages to

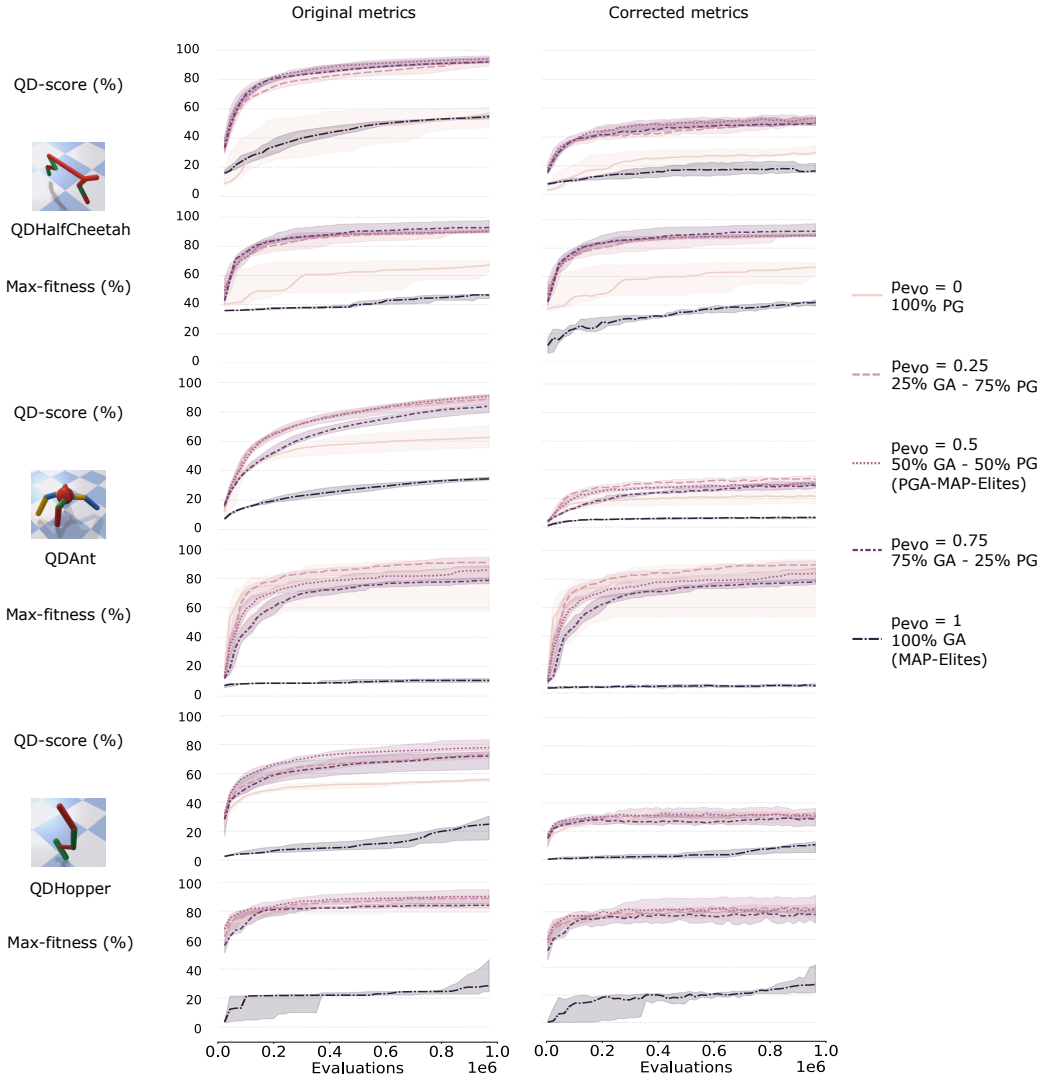


Fig. 15. **Ablation p_{evo} - Corrected results:** Comparison of the original (left) and Corrected (right) QD-score (top) and Max-fitness (bottom) of PGA-MAP-Elites with different mutation proportion on each QDGym task. To compute the Corrected metrics, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. Each experiment is replicated 10 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

reach a better coverage than all the other baselines within only $2 * 10^5$ evaluations ($p < 1e-6$) but longer run-time. As mentioned before, QDAnt likely has a more complex subspace of promising policies than other QD-Gym tasks, making it harder to discover and explore, which might explain the good performance of CMA-MAP-Elites in this domain. PGA-MAP-Elites also outperforms TD3 on the Coverage metric ($p < 5e-5$). Interestingly, TD3, which is not a QD algorithm, manages to get relatively good coverage of the BD space, in particular in QDWalker and QDHalfCheetah. However, it does not optimise these local solutions and only focuses on the best-performing one,

as highlighted by its poor QD-score. PGA-MAP-Elites also performs better than QD-PG and MAP-Elites-ES, for causes similar to those of the QD-Score. These results are similar in both versions of the environments. Only QD-PG obtains significantly lower coverage when applied to Deterministic QD-Gym tasks, probably due to the random initialisation of the robot encouraging exploration.

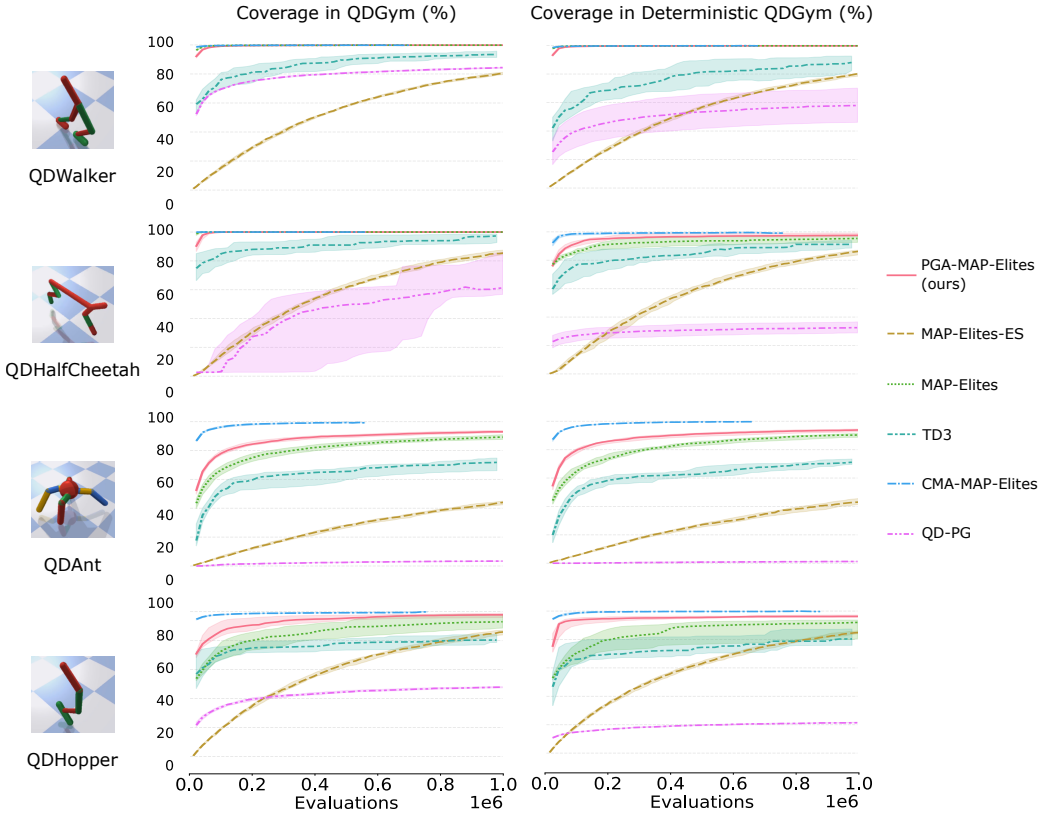


Fig. 16. **QD-Gym and Deterministic QD-Gym results:** Coverage of all algorithms on each QD-Gym (left) and Deterministic QD-Gym (right) task for 10^6 controller evaluations. Each experiment is replicated 20 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

D.2 Reproducibility in uncertain environments

Fig. 17 gives the Coverage loss corresponding to Section 5. According to the metrics in Appendix B, MAP-Elites-sampling and MAP-Elites-ES have a higher diversity in their final Corrected archive than all other algorithm in QDWalker ($p < 1e-5$) and QDHopper ($p < 1e-7$), and MAP-Elites-sampling still does in QDHalfCheetah ($p < 5e-4$). However, in QDAnt they are performing equivalently to PGA-MAP-Elites and Deep-grid. These results, as well as the archives displayed in Fig. 13, indicate that MAP-Elites-sampling and MAP-Elites-ES are the only algorithms that manage to discover reproducible solutions all over the BD-space, but their sample inefficiency makes them less efficient in QDAnt. The Coverage loss results in Fig. 17 are similar to those of QD-score loss, and the comparison across algorithms stay alike: MAP-Elites-ES is the algorithm that finds the more reproducible solutions across all tasks ($p < 1e-7$ except in QDHalfCheetah: $p < 5e-5$), and MAP-Elites-sampling stays second in this category ($p < 5e-7$ except in QDHalfCheetah:

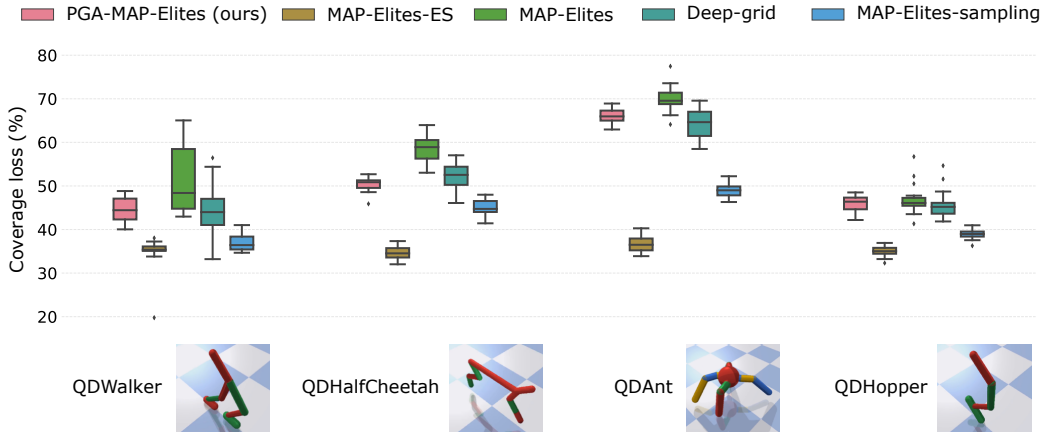


Fig. 17. **Loss comparison:** Comparison of the Coverage loss. To compute these losses, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The loss corresponds to the difference of each metric in proportion of the original metric. Lower loss indicates better reproducibility. The boxes represent the distribution of the loss for a given algorithm across 20 runs.

$p < 5e-4$). Their sampling mechanism increases their chance to store solutions in more accurate cells than MAP-Elites or PGA-MAP-Elites which only rely on one evaluation to estimate the BD of solutions. In the case of MAP-Elites-ES, the sampling-based estimate of the gradient probably also guide variation toward reproducible solutions. Despite also being designed for uncertain applications, Deep-grid obtains higher Coverage-loss than these two baselines ($p < 5e-7$ except in QDHalfCheetah: $p < 5e-4$). As highlighted before, its neighbours-based mechanism is not built for such high-dimensional search spaces. Among all baselines, MAP-Elites observes the higher performance drop ($p < 5e-5$ except in QDHalfCheetah: $p < 6e-3$), except on QDHopper where it performs similarly to PGA-MAP-Elites and Deep-grid.

The coverage-losses of PGA-MAP-Elites are close to those of the three uncertainty-handling QD algorithms, staying significantly lower than those of MAP-Elites ($p < 5e-5$ and in QDHalfCheetah $p < 3e-3$) except on QDHopper. These results lead to the same conclusions as for the QD-score loss metric. However, interestingly, the solutions found by PGA-MAP-Elites are slightly less reproducible in terms of Coverage than they were in terms of QD-score: PGA-MAP-Elites obtains slightly higher Coverage loss than QD-score loss in QDHalfCheetah and QDAnt. On the contrary, MAP-Elites solutions are more reproducible in terms of Coverage than QD-score (even if they stay less reproducible than those of all other baselines). As mentioned earlier, MAP-Elites and PGA-MAP-Elites use the same archive-management and differ only in variation operators. Thus, this difference could be related to critic-based variations: as the critic is only aware of fitness values and not BD, it may encourage reproducibility in terms of fitness but not directly in terms of BD. However, as reproducibility in terms of fitness and BD are likely correlated, we still observe a better Coverage-loss for PGA-MAP-Elites.

D.3 Ablation of PGA-MAP-Elites

We display in Fig. 18 the Coverage metrics from Section 6 and in Fig. 19 the corresponding Coverage loss. These results lead to the same conclusion as the QD-Score and Max-Fitness analysis in Section 6: the performance of PGA-MAP-Elites are robust to the proportion of PG and GA, as long as the two operators are present.

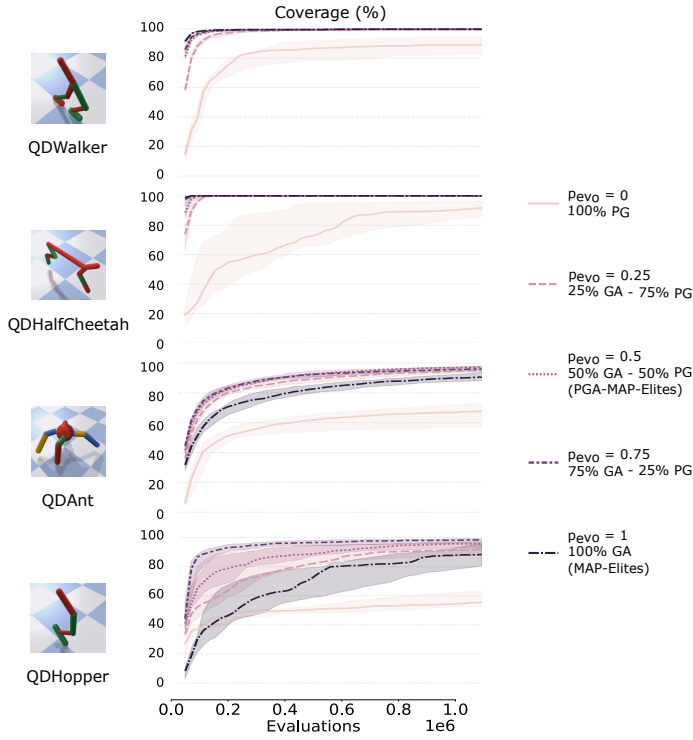


Fig. 18. **Ablation p_{evo} - Coverage:** Coverage of PGA-MAP-Elites with different mutation proportion on each QDGym task. Each experiment is replicated 10 times, the solid line corresponds to the median over replications and the shaded area to the first and third quartiles.

$p_{evo} = 0$ (PG-only)
 $p_{evo} = 0.25$
 $p_{evo} = 0.5$ (ours)
 $p_{evo} = 0.75$
 $p_{evo} = 1$ (GA-only)

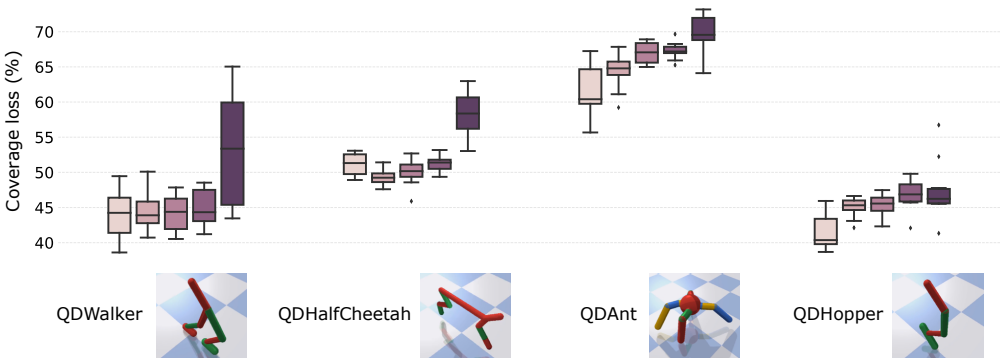


Fig. 19. **Ablation p_{evo} - Loss comparison:** Comparison of the Coverage loss for PGA-MAP-Elites with different variation operator proportions. To compute these losses, all solutions of the archive are replaced in a Corrected archive according to their average BD and fitness over 50 replications. The loss corresponds to the difference of each metric in proportion of the original metric. Lower loss indicates better reproducibility. The boxes represent the distribution of the loss for a given algorithm across 10 runs.

E WILCOXON RANK-SUM TEST P-VALUES

In Table 3, we give all the p-values based on Wilcoxon rank-sum test associated with all comparisons of PGA-MAP-Elites with other baselines in each result Section.

Table 3. **Summary of p-values:** we give the p-value based on Wilcoxon rank-sum test associated with all comparisons of PGA-MAP-Elites with other baselines in each result Section. ME stands for MAP-Elites.

Section 4		ME-ES	ME	TD3	QD-PG
QDWalker	QD-Score	1.38208e-14	1.38208e-14	1.38208e-14	3.55295e-10
	Max-Fitness	1.38208e-14	1.38208e-14	1.38208e-14	3.55295e-10
QDHalfCheetah	QD-Score	3.01981e-14	1.38208e-14	1.38208e-14	1.86346e-07
	Max-Fitness	5.58359e-14	1.38208e-14	0.0153138	1.86346e-07
QDAnt	QD-Score	1.38208e-14	1.38208e-14	3.55295e-10	1.38208e-14
	Max-Fitness	3.47414e-13	1.38208e-14	0.0903488	3.55295e-10
QDHopper	QD-Score	3.01981e-14	1.38208e-14	1.38208e-14	3.55295e-10
	Max-Fitness	3.01981e-14	1.38208e-14	1.38208e-14	3.55295e-10
Deterministic QDWalker	QD-Score	1.38208e-14	1.38208e-14	3.55295e-10	3.01981e-14
	Max-Fitness	1.38208e-14	1.38208e-14	3.42566e-08	3.55295e-10
Deterministic QDHalfCheetah	QD-Score	1.38208e-14	1.38208e-14	3.01981e-14	3.55295e-10
	Max-Fitness	1.38208e-14	1.38208e-14	1.71916e-08	3.55295e-10
Deterministic QDAnt	QD-Score	1.38208e-14	3.01981e-14	1.38208e-14	3.55295e-10
	Max-Fitness	1.38208e-14	3.01981e-14	0.0646717	3.55295e-10
Deterministic QDHopper	QD-Score	1.38208e-14	1.38208e-14	1.38208e-14	3.55295e-10
	Max-Fitness	1.38208e-14	1.38208e-14	1.38208e-14	3.55295e-10

Section 5		ME-ES	ME	Deep-grid	ME-sampling
QDWalker	QD-Score	1.38208e-14	1.38208e-14	3.55295e-10	3.55295e-10
	Max-Fitness	1.38208e-14	1.38208e-14	3.55295e-10	3.55295e-10
QDHalfCheetah	QD-Score	3.01981e-14	1.38208e-14	3.55295e-10	3.55295e-10
	Max-Fitness	5.58359e-14	1.38208e-14	3.55295e-10	3.55295e-10
QDAnt	QD-Score	1.38208e-14	1.38208e-14	3.55295e-10	3.55295e-10
	Max-Fitness	3.47414e-13	1.38208e-14	3.55295e-10	3.55295e-10
QDHopper	QD-Score	3.01981e-14	1.38208e-14	3.55295e-10	3.55295e-10
	Max-Fitness	3.01981e-14	1.38208e-14	3.55295e-10	3.55295e-10

Section 6		$p_{evo} = 0$ (Full PG)	$p_{evo} = 0.25$	$p_{evo} = 0.75$	$p_{evo} = 1$ (Full GA)
QDWalker	QD-Score	0.000379285	0.000880743	0.173617	0.000157052
	Max-Fitness	0.000529757	0.0126111	0.364346	0.000157052
QDHalfCheetah	QD-Score	0.000157052	0.0864107	0.705457	0.000238563
	Max-Fitness	0.000157052	0.567628	0.198765	0.000238563
QDAnt	QD-Score	0.000157052	0.596701	0.00515896	0.000157052
	Max-Fitness	0.762369	0.0696424	0.0126111	0.000157052
QDHopper	QD-Score	0.000157052	0.0342937	0.0101652	0.000157052
	Max-Fitness	0.00149887	0.449692	0.04125	0.000157052