Quality-Diversity Optimisation on a Physical Robot Through Dynamics-Aware and Reset-Free Learning

Simón C. Smith*†
s.smith-bize@imperial.ac.uk

Bryan Lim†
bryan.lim16@imperial.ac.uk

Hannah Janmohamed†
hannah.janmohamed21@imperial.ac.uk

Antoine Cully†
a.cully@imperial.ac.uk

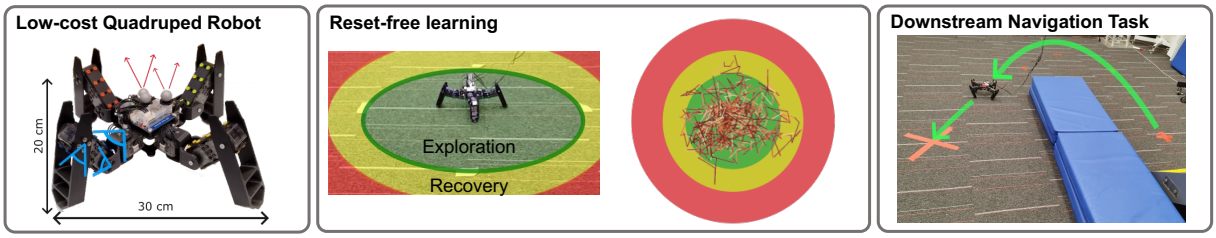

Figure 1: The Qutee robot. A 12 DoF (blue angles) quadruped. Global coordinates given by a motion capture system (red arrows). Middle: the trace of 2 hours of reset-free training within the exploration (green) and recovery (yellow) zones. Right: the navigation task used to test the learned solutions. All training was done on the physical robot, without any simulators.

ABSTRACT

Learning algorithms, like Quality-Diversity (QD), can be used to acquire repertoires of diverse robotics skills. This learning is commonly done via computer simulation due to the large number of evaluations required. However, training in a virtual environment generates a gap between simulation and reality. Here, we build upon the Reset-Free QD (RF-QD) algorithm to learn controllers directly on a physical robot. This method uses a dynamics model, learned from interactions between the robot and the environment, to predict the robot's behaviour and improve sample efficiency. A behaviour selection policy filters out uninteresting or unsafe policies predicted by the model. RF-QD also includes a recovery policy that returns the robot to a safe zone when it has walked outside of it, allowing continuous learning. We demonstrate that our method enables a physical quadruped robot to learn a repertoire of behaviours in two hours without human supervision. We successfully test the solution repertoire using a maze navigation task. Finally, we compare our approach to the MAP-Elites algorithm. We show that dynamics awareness and a recovery policy are required for training on a physical robot for optimal archive generation. Video available at https://youtu.be/BgGNvIsRh7Q

* School of Computing, Engineering and The Built Environment, Edinburgh Napier University, UK.
† Department of Computing, Imperial College London, UK.
This work was partially supported by the Engineering and Physical Sciences Research Council [grant number EP/V006673/1] .

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590625

CCS CONCEPTS

• **Computer systems organization** → **Evolutionary robotics**; • **Theory of computation** → **Evolutionary algorithms**.

KEYWORDS

Quality-Diversity, Robotics, Real-time optimisation

ACM Reference Format:
Simón C. Smith, Bryan Lim, Hannah Janmohamed, and Antoine Cully. 2023. Quality-Diversity Optimisation on a Physical Robot Through Dynamics-Aware and Reset-Free Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590625

1 INTRODUCTION

Quality-Diversity (QD) algorithms are techniques for optimisation that, among other, have been used for learning robotics controllers deployed in the real world. These algorithms find extensive collections of both high-performing and diverse solutions, which is advantageous for downstream applications [2, 4]. However, to the best of our knowledge, all existing QD algorithms applied in robotics use computer simulations to evaluate controller candidates. While this approach affords evaluation of thousands of solutions, it also comes with various disadvantages. For example, simulators require accurate modelling of the physical properties and dynamics of the environment. Though a great variety of advanced simulators have been developed, it is still difficult to model the real world with high fidelity. Furthermore, rigid-body simulation often assumes a controlled environment and a robot with accurate sensors and actuators. Meanwhile, in the real world, sensors and actuators are often prone to noise and damage. Some methods, such as Kalman filters and high-frequency closed-loop control, attempt to mitigate the effects of stochastic environments. However, the controllers

learned in simulation are not always robust when transferred directly to physical robots. Consequently, QD approaches in robotics usually encompass further adaptation mechanisms to bridge the gap between simulation and reality [2, 4, 7] which require additional training and fine-tuning.

In this work, we use the Reset-free Quality-Diversity (RF-QD) algorithm [8] to learn repertoires of controllers directly on a physical quadruped robot without human intervention. RF-QD learns a dynamics model to predict the behaviour of controllers. Based on these predictions, RF-QD applies a behaviour selection policy to prioritise controllers according to predefined safety and exploration heuristics to be executed in the environment. Accordingly, controllers that are predicted to be unsafe or uninteresting are filtered out, leading to better sample efficiency, reduced training time and decreased risk of damage. Furthermore, given environmental information, RF-QD ensures that training always takes place in a safe zone so that continuous learning can take place, without the need of human supervision or manual resets.

Our results show that RF-QD can successfully generate an archive of behaviours without simulation in two hours. We compare RF-QD with baselines to show that the recovery policy and behaviour prioritisation are both essential components for achieving high-performing and diverse repertoires. Finally, we test the generated repertoires on a navigation task.

2 RELATED WORK

Quality-Diversity in Robotics. Quality-Diversity (QD) form a subset of evolutionary algorithms that generate many diverse and high-performing solutions. To achieve this, QD algorithms characterise solutions by their fitness, an objective function, and by a vector known as the *behaviour descriptor* (BD) [5]. The BD is used to quantify the difference in the behaviour of solutions.

In general, QD algorithms are initialised with a set of random solutionsstored in an archive. At each iteration, solutions are selected from the archive and mutated to generate offspring. The offspring are evaluated and considered for addition to the archive, based on their fitness and BD. The precise addition mechanism differs according to the QD algorithm. For example, MAP-Elites [9] employs a grid-based archive, achieved by tessellating the BD space into several fixed-size bins. A solution is added to the grid if the corresponding cell is empty or its fitness is higher than an existing solution in the same cell, discarding the previous solution. Alternatively, other QD algorithms use an unstructured archive [6, 7] in which a solution is added if the distance to its k-nearest neighbours in the descriptor space exceeds a predefined threshold.

QD algorithms are used in robotics to learn a collection of controllers. The diversity of skills learned is advantageous for use in downstream applications such as damage recovery or planning for long horizon tasks [2, 3, 7]. Existing methods first learn skills in simulation, and only after learning they use them on the physical robot. In our work, we apply the QD algorithm directly on the physical robot, with no simulator and prior experience.

End-to-end Learning on Physical Robots. Data-driven and learning based methods are able to learn very complex skills [1]. However, they require a large number of samples or trials, making them infeasible for direct implementation on physical robots. Learning directly on physical robots rely on model-based RL for locomotion [12, 13] and manipulation [10, 12, 14]. Our work demonstrates learning directly on a physical robot is also possible for evolutionary-based algorithms such as QD algorithms.

3 METHODS

Learning a Dynamics Model. Following DA-QD [7], a forward model $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ that predicts the next state $\boldsymbol{s}_{t+1}$ given the current state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$, is learnt based on data collected from interaction in the environment. The model is implemented as a deep neural network where the delta of the next state ($\Delta \boldsymbol{s}_{t+1} = \boldsymbol{s}_{t+1} - \boldsymbol{s}_t$) is learned. Similar to DA-QD and RF-QD, we learn an ensemble of probabilistic models to minimise both aleatoric and epistemic uncertainty. The disagreement between the models in the ensemble is inferred using their distributions, allowing an estimation of the epistemic uncertainty, which can be used to prioritise new controllers. The model is trained via self-supervised learning using gradient descent to maximise the log-likelihood of transitions sampled from the replay buffer. The dynamics model is used to perform rollouts of controllers, called recursively for the defined length of the execution of a controller.

Performing QD in imagination. RF-QD uses the dynamics model to predict a robot's trajectory, fitness and behavioural descriptor for a given controller. Solutions can be evaluated without requiring any real-world interaction. Thus, the QD loop of selection, mutation, evaluation and archive addition is performed using imagined rollouts and maintained in an separated archive. Solutions from this imagined archive are selected to be executed in the real world. These physical executions are added to the main archive and added to the reply buffer for further training of the dynamics model. Performing evaluations in imagination allows better data efficiency as solutions that are not promising will be sieved out and not executed.

Behaviour Selection Policy. BSP ensures that the robot remains in a safe state while learning new skills and interacting with the environment. The BSP determines the solutions from the imagined archive to be executed provided that the user defines a safety signal. In locomotion, BSP comprise exploration zones (a set of safe states) and recovery zones (a set of unsafe states) Ω. The relative safety of the robot in state $\boldsymbol{s}$ is measured by an exploration parameter $\epsilon(\boldsymbol{s})$, calculated as the distance between $\boldsymbol{s}$ and the nearest unsafe state $\boldsymbol{\omega} \in \Omega$ normalised by the maximum distance between any previous state and $\boldsymbol{\omega}$:

$$\epsilon(\boldsymbol{s}) = \frac{dist(\boldsymbol{s}, \boldsymbol{\omega}) - \beta}{\max_{\boldsymbol{s}_i} dist(\boldsymbol{s}_i, \boldsymbol{\omega}) - \beta}. \quad (1)$$

Here, β ensures a buffer space when the robot returns to the exploration zone.

New controllers $\boldsymbol{s}'$ are safe if they are expected to keep the robot in the exploration zone, i.e. $\epsilon(\boldsymbol{s}') > 0$. Using this safety metric, RF-QD filters the solutions in the imagined archive to obtain a subset of solutions expected to be safe. RF-QD prioritise solutions in this safe set by safety, novelty or disagreement between models. In our experiments, we use novelty prioritisation to bias exploration in the BD space.

Recovery Policy. A recovery policy is used for safety if the robot leaves the exploration zone. While in the recovery zone, the policy selects behaviours in the archive that returns the robot to

the exploration zone. In the case of leaving the recovery and the exploration zone the learning is stopped.

Controllers Update. We extend RF-QD to include an update strategy for controllers used during recovery. The executions of these controllers are used to update their fitness and BD. Given a controller with fitness f and behavioural descriptor $\mathbf{b}$, the new values are given by:

$$f = (1-\alpha)f + \alpha f', \ \mathbf{b} = (1-\alpha)\mathbf{b} + \alpha \mathbf{b}',$$

where f' and $\mathbf{b}'$ are the fitness and behaviour descriptor values observed in the new execution, with $\alpha \in [0, 1]$. This updated controller is reevaluated for addition or removal from the archive.

Physical Implementation. We use the Qutee robot (Fig.1) which was designed, 3D printed and assembled by researchers at the Adaptive and Intelligent Robotics Lab at Imperial College. One cable is attached to the robot for power and another for data transmission to a computer.

The action space of the robot corresponds to the angle commands sent to each of these DoF. We restrict the range of movement of each DoF to avoid collisions between the legs and the body of the robot. We restrict the range of the hip actuators in the horizontal plane to reduce the chances of turning upside down.

Exterioception is provided by a motion capture system. The system returns the position, velocity, angle and angular velocity of the centre of mass of the robot. We transform the world coordinates to positions relative to the robot's initial location. These coordinates are used as part of the robot's state space, along with the position and velocities of each joint.

We define four main zones for exploration and recovery as visualised in Fig. 1. The robot is initially placed inside the green exploration zone. In this zone, the robot executes new controllers. Based on their fitness and BD, controllers are evaluated to be added to the archive following the usual QD rules. When executing a controller, if the robot exits the exploration zone, it enters the yellow recovery zone. From this zone, the recovery policy is activated. The policy takes into account the β (Eq. 1) buffer before returning to the exploratory mode. The training is stopped if the robot leaves the recovery zone (red in Fig. 1).

4 EXPERIMENTS AND RESULTS

4.1 Experiments and Ablations

Training. We use an omnidirectional walking task to generate an archive of controllers. In this task, the robot learns walking policies to move in the horizontal plane [4]. The BD is the final position of the robot with respect to the initial position of the behaviour. The fitness is the negative error between the horizontal rotational angle of the robot in the final position and the required angle to arrive following a predetermined arc from the initial position. The genotype is a vector of size 24. This vector represents the parameters of an open-loop sinusoidal controller. The movement of each joint is parameterised by the amplitude, the phase and the duty cycle. We use the same parameters for the foot and knee of each leg.

We assume that the robot has a 2-D map of the exploration and recovery zone, represented as concentric circles of radii 0.5m and 0.75m respectively, with $\beta = 0.3$. The execution of a controller last 5s. We set $\alpha = 0.8$. To generate offspring, we use the ISO+LineDD mutator [11]. We use an unstructured archive to store the solutions and define novelty and gradient-contextual constraints for the behaviour selection policy. For the dynamics model probabilistic ensemble, we use 4 neural networks with two hidden layers of 500 neurons each. We store the action-state pairs in a replay buffer and train the dynamics model after 10 evaluations.

Each experiment runs for 2 hours or until the robot leaves the recovery zone. RF-QD includes an initialisation phase where 10 random controllers are generated, executed and added to the archive. Among all the experiments we ran, this initialisation failed only once due to the robot moving out of the recovery zone.

Comparison to other algorithms. We compare RF-QD to two variants. *RF-QD without dynamics awareness*, we remove the dynamics model. Thus, no training or behaviour selection policy is possible. Removing the training phase allows more evaluations during the 2 hours of the experiment. *RF-QD without recovery function*, the recovery function is removed. This version still has the dynamics model so the behaviour selection policy still bias the robot to remain in the exploration zone. We compare the algorithms to MAP-Elites, which is equivalent to RF-QD without a dynamics model and no recovery policy. All the meta-parameters remain the same between the four algorithms. We trained each version 4 times and tested the solutions on navigation task 5 times each.

Planning on a maze. For testing, we use a maze task in which the robot must navigate to a goal while avoiding obstacles (Fig. 1). The location of the goal and obstacles are known to the robot. We use RTE (Reset-free Trial-and-Error) algorithm [3] with an A* planning algorithm to select the best solution from the archive. We count a test as successful if the robot reaches the goal within a range of 5cm, and unsuccessful if the robot executes more than 100 actions without reaching the goal. We define an action as the execution of a single controller for 5 seconds.

4.2 Results

Archive Generation. Fig. 2 (top) shows the best archive for each algorithm. Algorithms that include recovery, RF-QD and RF-QD no DA, outperform those without it. The coverage and maximum fitness values achieved by RF-QD and RF-QD no DA are equivalent and the best among the algorithms. When no recovery function is available, the robot may leave the recovery zone, resulting in early termination and lower number of evaluations, Fig. 2 (bottom). On average, the non-recovery algorithms run 100 evaluations before leaving the recovering zone. By contrast, the two algorithms with recovery run 600 and 1400 evaluations on average. RF-QD no Recovery performs slightly better than MAP-Elites thanks to the behaviour selection mechanism. This mechanism allows the algorithm to perform slightly more evaluations than MAP-Elites, but was still insufficient to avoid the robot to leave the recovery zone.

RF-QD is more consistent on QD score while using fewer evaluations (Fig.2). In the same time frame, RF-QD no DA executed more evaluations as it does not require time for training the model and predicting behaviours. However, in physical training, requiring fewer evaluations is a desired outcome. Also, asynchronous execution of controllers and training of the model can always be implemented to reduce this difference. Moreover, fewer evaluations are desirable as it reduces the probability of damaging the robot.

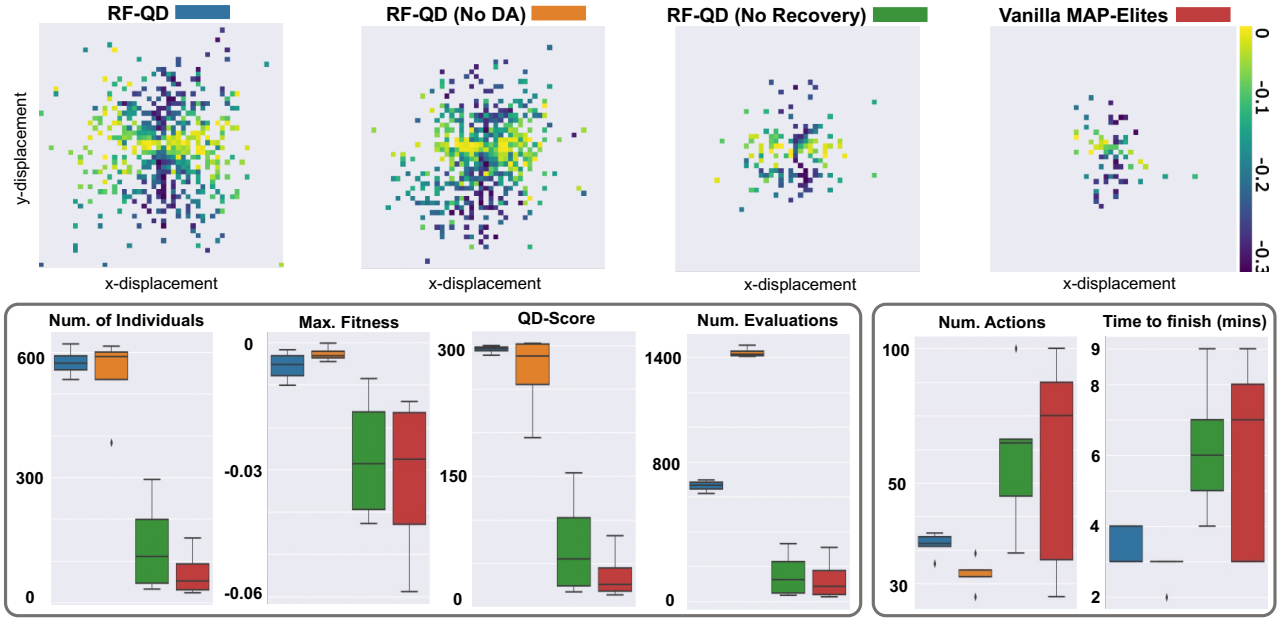

Figure 2: Top: without dynamics awareness (RF-QD no DA, orange), the archive tends to have more solutions around the origin compared to RF-QD (blue). For RF-QD no Recovery (green) and MAP-Elites (red), few individuals are found before the robot is outside the training zone. Bottom: Archive generation and navigation tasks have the best results for RF-QD and RF-QD no DA.

Maze Navigation Task. Fig. 2 also shows the results of each algorithm in the navigation task. RF-QD and RF-QD no DA have the best results with fewer actions and shorter arrival times. These results are related to the coverage, maximum fitness and QD score of the archives after training. There is a slight increase in performance for the no DA variation. This effect is related to the selection policy that prioritises individuals predicted to be safer and novel.

The results for the no-recovery versions show the relative worst performance. The lack of diversity and relatively low fitness results in the robot not having a proper set of actions to reach the goal. The robot was observed to get stuck in corners and hit obstacles.

5 CONCLUSIONS

We trained a physical robot to find diverse walking solutions without using any physical simulations. After 2 hours of training, the robot could generate enough diverse solutions to navigate a maze. Using a recovery function is necessary to keep the robot in the training regime. The data efficiency of our approach is improved by using the dynamics model to predict and select safe and novel controllers before testing them in the robot. To the best of our knowledge, this is the first implementation of a QD algorithm directly on a physical robot without using simulations. A video showing the training and navigation tasks is available at https://youtu.be/BgGNvIsRh7Q

REFERENCES

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113* (2019).

[2] Maxime Allard, Simón C. Smith, Konstantinos Chatzilygeroudis, and Antoine Cully. 2022. Hierarchical Quality-Diversity for Online Damage Recovery. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 58–67. https://doi.org/10.1145/3512290.3528751

[3] Konstantinos I. Chatzilygeroudis, Vassilis Vassiliades, and Jean-Baptiste Mouret. 2016. Reset-free Trial-and-Error Learning for Data-Efficient Robot Damage Recovery. *CoRR* abs/1610.04213 (2016). arXiv:1610.04213 http://arxiv.org/abs/1610.04213

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

[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.

[6] Luca Grillotti and Antoine Cully. 2022. Unsupervised Behavior Discovery With Quality-Diversity Optimization. *IEEE Transactions on Evolutionary Computation* 26, 6 (2022), 1539–1552.

[7] Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. 2022. Dynamics-aware quality-diversity for efficient learning of skill repertoires. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 5360–5366.

[8] Bryan Lim, Alexander Reichenbach, and Antoine Cully. 2022. Learning to Walk Autonomously via Reset-Free Quality-Diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 86–94. https://doi.org/10.1145/3512290.3528715

[9] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. https://doi.org/10.48550/ARXIV.1504.04909

[10] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. 2020. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*. PMLR, 1101–1112.

[11] Vassiiis Vassiliades and Jean-Baptiste Mouret. 2018. Discovering the Elite Hypervolume by Leveraging Interspecies Correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Kyoto, Japan) *(GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 149–156. https://doi.org/10.1145/3205455.3205602

[12] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. 2022. DayDreamer: World Models for Physical Robot Learning. https://doi.org/10.48550/ARXIV.2206.14176

[13] Yuxiang Yang, Ken Caluwaerts, Atil Iscen, Tingnan Zhang, Jie Tan, and Vikas Sindhwani. 2020. Data efficient reinforcement learning for legged robots. In *Conference on Robot Learning*. PMLR, 1–10.

[14] Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. 2019. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*. PMLR, 7444–7453.