# LEARNING TO WALK AUTONOMOUSLY VIA RESET-FREE QUALITY-DIVERSITY

**Bryan Lim*, Alexander Reichenbach* & Antoine Cully**
Imperial College London
London, UK
{bryan.lim16,a.cully}@imperial.ac.uk

## ABSTRACT

Quality-Diversity (QD) algorithms can discover large and complex behavioural repertoires consisting of both diverse and high-performing skills. However, the generation of behavioural repertoires has mainly been limited to simulation environments instead of real-world learning. This is because existing QD algorithms need large numbers of evaluations as well as episodic resets, which require manual human supervision and interventions. This paper proposes Reset-Free Quality-Diversity (RF-QD) as a step towards autonomous learning for robotics in open-ended environments. We build on Dynamics-Aware QD (DA-QD) and introduce a behaviour selection policy that leverages the diversity of the imagined repertoire and environmental information to intelligently select behaviours that can act as automatic resets. We demonstrate this through a task of learning to walk within defined training zones with obstacles. Our experiments show that we can learn repertoires of legged locomotion controllers autonomously without manual resets and with high sample efficiency in spite of harsh safety constraints. Finally, using an ablation of different target objectives, we show that it is important for RF-QD to have diverse types solutions available for the behaviour selection policy over solutions optimised with a specific objective. Videos and code available at https://sites.google.com/view/rf-qd.

## 1 INTRODUCTION

Despite the recent popularity of Quality-Diversity (QD) algorithms (Pugh et al., 2016; Cully & Demiris, 2017), these algorithms have been limited to domains in which evaluations can be performed in simulation. This is because QD algorithms need to perform evaluations in the order of millions and where the outcomes are not safety critical or dangerous. Examples of these application domains include robotics (Cully & Mouret, 2013; Cully et al., 2015; Chatzilygeroudis et al., 2018), video games (Gravina et al., 2019; Fontaine et al., 2020a) and aerodynamics (Gaier et al., 2018). In the field of robotics, physics simulators (Coumans & Bai, 2016–2020; Lee et al., 2018a; Todorov et al., 2012) are commonly used and QD algorithms depend heavily on these to obtain abundant amounts of data and evaluations to learn behavioural repertoires of robots. However, building fast and accurate physics simulators to model the complex dynamics of robots and the wide variety of potential environments is difficult. Furthermore, even with extensive modelling of different scenarios, there is still the difficult problem of sim-to-real transfer (Zhao et al., 2020; Akkaya et al., 2019; Lee et al., 2020). To realize the potential of QD algorithms for robotics and to have the real-world impact we want them to have, we need algorithms which can effectively learn repertoires of skills autonomously and adapt directly in the real world.

Another key reason we might want use QD algorithms directly in the real-world is in the pursuit of more open-ended learning algorithms (Stanley et al., 2017; Stanley, 2019). QD algorithms have already shown evidence of possessing characteristics of open-ended search (Stanley et al., 2017; Wang et al., 2019; Clune, 2019). We propose a new perspective and that a promising path to open-ended learning algorithms could be to build on existing QD algorithms to continuously interact directly in the real world. The real world itself offers constantly evolving environments and agents with complex diversity, open-ended challenges and almost endless possibilities. Given the right innovations, QD has

the potential to continuously generate new skills by leveraging the already present open-endedness of the real world.

There are some challenges in this paradigm, such as continuous representation of the diversity of the real-world, autonomously learning behavioural descriptors (Cully, 2019; Paolo et al., 2020), and more. In this work, as a step towards using QD algorithms in the real-world, we address two practical issues that arise when attempting to learn behavioural repertoires in the real-world in an autonomous manner: *resets* and *safety*. We then aim to maximise the sample efficiency of learning behavioural repertoires while considering these constraints. We specifically focus on reset-free learning of robotic locomotion skills to highlight these issues and our approach to solving them.

An often overlooked requirement of QD algorithms when used for Reinforcement Learning (RL) is the episodic setting they function in. This requires the environment to be set to a fixed initial state at the start of every episodic trial as the behavioural descriptor is measured as a function of the trajectory of states from this initial state to the final state in the episode. In real-world settings, this would correspond to humans manually resetting the robot and environment after every episode. This is an impractical and expensive solution considering the number of evaluations that conventional QD algorithms require are in the order of millions. With the considerable amount of human supervision and instrumentation required for resets, this defeats the purpose of QD algorithms to autonomously learn complex skills.

Another key challenge of learning in the real-world is safety. Actions taken must not be dangerous to the robot and the environment. For learning locomotion skills, this corresponds to avoiding collisions with objects in the environment during learning. Achieving this while performing QD would require both the capability to predict the outcome of the execution of a new behaviour as well as an understanding of its implications with respect to its safety.

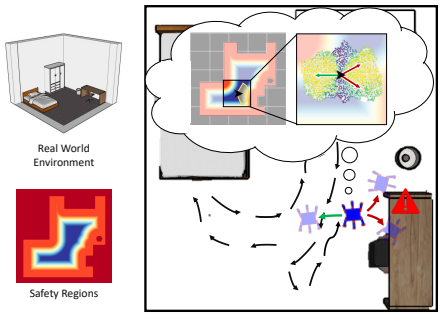


Figure 1: Safe reset-free movement via diversity of behavioural space in a real-world environment.

Finally, we want to learn skills efficiently with efficiency being measured by the number of evaluations taken. In the real-world, this also directly corresponds to the learning time needed. The goal is to intelligently select behaviours that would help improve the behavioural repertoire while minimizing unnecessary non-informative trials.

We introduce Reset-Free Quality Diversity (RF-QD) as a framework for the real-world execution of QD algorithms (see Figure 1). In a nutshell, RF-QD is a Dynamics-Aware Quality-Diversity (DA-QD) algorithm combined with a Behaviour Selection Policy to select only safe and valuable behaviours for evaluation in the (potentially dangerous) real-world. We demonstrate an algorithm which autonomously acquires a diverse repertoire of locomotion skills on a hexapod robot in safety-constrained environments.

## 2 Related Work

### 2.1 Quality-Diversity and Behavioural Repertoire Learning in Robotics

Quality-Diversity (QD) optimization is a class of algorithms that aims to generate a collection of both diverse and high-performing solutions (Pugh et al., 2016; Cully & Demiris, 2017). In the context of robotics, each solution can for instance, be a parametric policy which determines sequences of actions to execute (i.e. motor commands), resulting in a behaviour. A behaviour can then be represented by a numerical vector referred to as the *Behavioural Descriptor (BD)*. The BD is a low-dimensional representation of the trajectory of states the policy visited and is usually defined manually depending on the tasks. However, the BD can also be learned in an unsupervised manner (Cully, 2019; Paolo et al., 2020; Grillotti & Cully, 2021). The choice of the BD is important as it determines the novelty of a solution and will be used to drive the exploration over the BD space (Lehman & Stanley, 2011a).

Conceptually, QD extends the single novelty-seeking objective introduced in the Novelty Search algorithm (Lehman & Stanley, 2011a) with another measure of quality. MAP-Elites (Mouret & Clune, 2015) and Novelty Search with Local Competition (Lehman & Stanley, 2011b) are two commonly

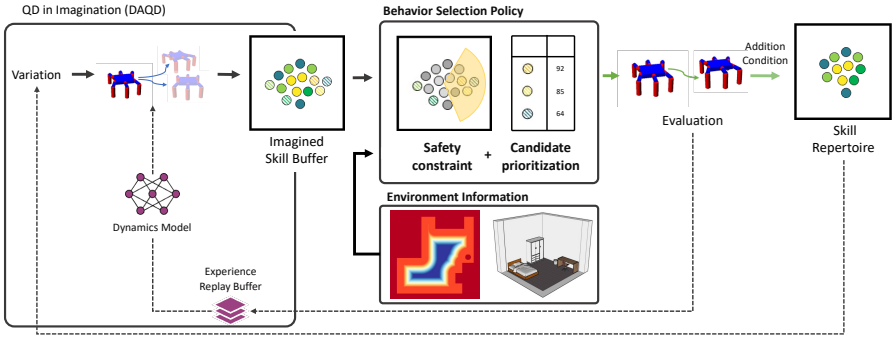


Figure 2: RF-QD performs QD in imagination (as in DA-QD) and uses a more intelligent behaviour selection policy to keep the robot in the safe regions of its environment while maximising the value gained by every real-world evaluation. See Pseudocode 1 in Appendix A.1.3 for more detail.

used and well-known QD algorithms. Cully and Demiris (Cully & Demiris, 2017) suggested that most QD algorithms can be represented using a common framework consisting of two key components; the *archive* and *selector*. Variants of QD algorithms develop around these components, all building on the QD loop of selection, variation, evaluation and (tentative) addition to the archive. The archive is used to store the highest performing solutions for each niche. Instead of a uniform grid used to discretize the BD space, methods like CVT-MAP Elites (Vassiliades et al., 2017) or Sliding-Boundary MAP-Elites (Fontaine et al., 2019) modify this to make the archive more flexible. More recently, there has also been a body of work focused on using more complex selectors and efficient optimizers such as evolutionary strategies (Colas et al., 2020; Fontaine et al., 2020b) and policy gradients (Nilsson & Cully, 2021; Pierrot et al., 2021).

In the field of robotics, it was shown that the final archive of solutions discovered by QD algorithms can be used as a behavioural repertoire to perform downstream tasks (Cully & Mouret, 2013). For example, each solution is a controller which is represented by a set of parametric functions which controls the robot's joints. Coupled with Bayesian optimization, the controllers generated in the archive can be used to quickly adapt to unforeseen mechanical damage, help with sim-to-real transfer and solve long-horizon tasks using planning (Cully et al., 2015; Chatzilygeroudis et al., 2018).

### 2.2 Model-based Quality-Diversity

One of the main bottlenecks of Quality-Diversity (QD) algorithms is the sample efficiency. QD algorithms typically require on the order of millions of evaluations and rely on parallel computation of these evaluations. This single factor alone usually make them unsuitable to be used directly in the real-world. A line of work that attempts to address this problem, now referred to as *model-based quality-diversity* methods, is through the use of models. Surrogate-Assisted Illumination (SAIL) (Gaier et al., 2018) first introduced the use of surrogate models for QD algorithms. SAIL integrates surrogate models, in the form of Gaussian Process (GP) models, to approximate the objective function and reduce the number of evaluations for the computationally expensive application of aerodynamic design. Another algorithm called M-QD (Keller et al., 2020) later follows up on this idea and used neural network models that map the parameter space to the behaviour and fitness space as a surrogate model. They demonstrate this on robotic pushing and placing tasks.

Dynamics-Aware Quality-Diversity (DA-QD) (Lim et al., 2021) is another approach which instead uses learnt dynamics models as a surrogate model. DA-QD introduced the concept of the imagined repertoire which allows QD to be performed fully in imagination using the learnt dynamics models. The dynamics models are trained incrementally and online as data is collected through evaluations. DA-QD showed a significant ($\approx$20 time) increase in sample-efficiency. Our work uses the DA-QD framework to heavily reduce the number of evaluations needed making it feasible to be considered for a real-world application. With RF-QD, we extend DA-QD to make better use of the imagined repertoire to select behaviours more intelligently and in a sequential manner. While DA-QD only uses variation operators for optimization in imagination, we also further study the effect of optimizing for different objectives in imagination.

### 2.3 Reset-free Learning

Reset-free Learning has mainly been studied in gradient-based Deep Reinforcement Learning (DRL) where the episodic setting is also usually a prerequisite of the Markov Decision Process (MDP)

formulation of the problem. One approach taken to enable real-world RL is to automate resets using other manually scripted robots to reset objects and the environment to the initial state distribution required (Nagabandi et al., 2020). While this works well for resetting manipulation tasks in which the workspace is relatively limited, this approach is difficult to apply for learning locomotion behaviours.

Most similar to our work and another promising method is to use a multi-task RL approach (Ha et al., 2020; Gupta et al., 2021). The key idea behind this approach is to use a scheduler and the different tasks present in the multi-task setup as resets for each other. Ha et al. (2020) showed this in the context of learning simple locomotion policies while Gupta et al. (2021) demonstrated this approach on more extensive multi-task setting to learn dexterous manipulation policies. Both these works explicitly learn policies for tasks in a pre-defined distribution of tasks. Each policy is optimized individually using an off-the-shelf deep RL algorithm and separate instances of networks and replay buffers. Our work instead concurrently learns a repertoire of diverse policies using QD algorithms and leverage the diversity of the behavioural repertoire as resets. In Multi-task MAP-Elites (Mouret & Maguire, 2020), it is also showed that the behaviour space can also be viewed and formulated as a task-space where each cell is a task.

In the context of QD algorithms and behavioural repertoire learning, the Reset-Free Trial and Error (RTE) (Chatzilygeroudis et al., 2018) algorithm has also aimed to address the reset problem. RTE demonstrated this for adaptation using the behavioural repertoire as a prior for Gaussian Process models. This is a different setting from the work we present in this paper as the behavioural repertoire generation process itself in RTE is performed fully in simulation using resets. The reset-free in RTE refers to the reset-free adaptation when performing sim-to-real transfer or reset-free adaptation to mechanical damage. In our work, we learn the behavioural repertoire itself in a reset-free manner.

## 3 Background: Dynamics-Aware QD

We build on the DA-QD framework proposed by Lim et al. (2021). We briefly summarize DA-QD here and refer the reader to the full paper for further details. DA-QD is a model-based QD algorithm which extends the conventional QD framework (Pugh et al., 2016; Cully & Demiris, 2017) discussed in section 2.1 with three key components: a *dynamics model*, an *imagined repertoire* and a *selector* from the imagined repertoire.

The learnt dynamics model is a forward dynamics model $\widetilde{p}_{\vec{\theta}}(\vec{s}_{t+1}|\vec{s}_t, \vec{a}_t)$ and is represented by a neural network parameterized by $\theta$. To capture both aleatoric and epistemic uncertainties, an ensemble of probabilistic models are used. Here, the disagreement between predictions of all models in the ensemble captures the epistemic uncertainty, i.e. it indicates the uncertainty of the prediction due to a lack of samples. The overall model disagreement $\mu_d$ can be calculated as the expected difference between any two models in the ensemble $f_\phi$ for one state-action pair, averaged over all time step predictions in one trajectory (i.e. one evaluated behaviour) of length T (Kidambi et al., 2020):

$$\begin{aligned} \text{disag}(s,a) &= \mathop{\mathbb{E}}_{i \neq j} \| f_{\phi_i}(s,a) - f_{\phi_j}(s,a) \|_2 \\ \mu_d &= \tfrac{1}{T} \textstyle\sum_{t=0}^{T} \text{disag}(s_t, a_t) \end{aligned} \tag{1}$$

State transition data is collected and stored in a replay buffer $\mathcal{B}$ as evaluations of robot behaviour are performed in the environment. The model is trained in a self-supervised manner to maximise log-likelihood of the transitions sampled from replay buffer and is optimized via back-propagation.

The dynamics model $\widetilde{p}_{\vec{\theta}}$ can be called recursively to evaluate policies in what is referred to as an imagined roll-out. The expected fitness and BD can be obtained from this imagined roll-out as both these quantities measured are a function of the state trajectory. DA-QD introduced the concept of an imagined repertoire $\widetilde{\mathcal{A}}$ to organise and maintain solutions that have been evaluated in imagination using the dynamics model. The imagined repertoire $\widetilde{\mathcal{A}}$ uses the the same addition conditions as the repertoire $\mathcal{A}$. The imagined repertoire $\widetilde{\mathcal{A}}$ only allows solutions that have been evaluated in imagination that are expected to be novel or better performing than existing solutions to be considered for evaluation. This is where the sample-efficiency of this method is derived from. Additionally, this allows QD to be performed fully in imagination. This means that the selection and mutation of the QD algorithm can be continuously performed from the imagined archive for any desired number of imagined generations without any samples or evaluations on the real system.

Finally, with the introduction of the imagined repertoire $\widetilde{\mathcal{A}}$, this necessitates selection of solutions from $\widetilde{\mathcal{A}}$ to be evaluated. As the original DA-QD algorithm does not consider the reset-free sequential

evaluation setting, the authors select all the solutions that have been added to the imagined archive to be evaluated in parallel. Our work extends DA-QD and proposes a more intelligent method to select and manage the solutions in the imagined archive given the reset-free setting and safety constraints from the environment. Additionally, DA-QD also does not explicitly use the resulting model-disagreement. In this paper, the model-disagreement is used both as a heuristic to select behaviours to execute more intelligently (Sec. 4.1) and as objective in imagination (Appendix A.2.2).

# 4 METHODS

We present Reset-Free Quality-Diversity (RF-QD) as a method to enable the application of QD's behavioral repertoire learning in non-episodic real-world environments (see Algorithm 1). We treat the robot as an actor in its environment that performs a constant search for new and improved behaviours and storing these in the archive. For this, we extend the classical QD loop by two steps. Firstly, we build on the pre-evaluation of any new behaviour "in imagination" by a dynamics model (DA-QD). Secondly, we introduce a behaviour selection policy, that modulates the robot's search for novel and high-performing behaviours as to comply with the safety constraints given by the environment (see Figure 2). In the following section, we first elaborate on the core of our method: the *behaviour selection policy*. Then, we detail its main components: the *safety evaluation*, *safety constraints*, the *prioritisation metrics* and the *recovery policy*.

## 4.1 RESET-FREE BEHAVIOUR SELECTION

To be able to stay safe while acting in its environment, we introduce a behaviour selection policy to modulate the robot's actions in the real world. This "behaviour selection" ensures that every new behaviour will only be performed if it is deemed safe for the robot. At every step, new behaviours are selected from a candidate buffer $\mathcal{C}$. The candidate buffer $\mathcal{C}$ is regularly filled with new policies from the imagined repertoire $\widetilde{\mathcal{A}}$ that are not already present in the repertoire $\mathcal{A}$. $\widetilde{\mathcal{A}}$ is a component introduced in DA-QD that maintains solutions that were evaluated in imagination using the dynamics model $\widetilde{p}_{\widetilde{\theta}}$. Based on the robot's current state in the environment, our policy then selects a subset of candidate behaviours $\mathcal{C}_{safe}$ that have a low risk of violating the safety constraints given by the environment. Out of these, the candidate behaviour with the highest prioritization score will be evaluated in the real world. In the following sections, the core components are described in detail.

## 4.2 SAFETY EVALUATION

In this paper, we assume knowledge of the environment layout, represented by 'safety regions' (see Figure 1), that indicate the region of dangerous states $\Omega$. In practice, this information could as well be obtained using Simultaneous Localisation and Mapping (SLAM) methods with an on-board camera.

Derived from the robot's state $s$, we define the exploration parameter $\epsilon(s)$, which indicates the relative degree of safety in the current state. It is calculated as the smallest distance between $s$ and $\Omega$ and normalised by the maximum encountered distance value (see Equation 2). While inside the safe region (i.e. $s \notin \Omega \rightarrow \epsilon(s) > 0$), the robot must choose any potential solution to be evaluated in the real world that is predicted to keep $\epsilon(s) > 0$, i.e. does not enter any unsafe state. To lower the risk of damage to the robot, an offset $\beta$ can be added in the computation of $\epsilon_s$ as an increased threshold for the minimum distance towards the border of the region of unsafe states within the state space.

$$\epsilon(s) = \frac{dist(s, \Omega) - \beta}{\max\limits_{s_i} dist(s_i, \Omega) - \beta} \tag{2}$$

From the dynamics model, we can obtain the predicted next state $s'$ after the execution of a candidate behaviour and compute $\epsilon(s')$. $s'$ corresponds to the state $s_{t+T}$ after $T$ timesteps, where $T$ is the length of one behaviour. Generally, we seek the robot to stay as close as possible to the safest point/s in the environment, i.e. maintain maximal distance to the region of dangerous states ($\epsilon(s) \approx 1$).

## 4.3 SAFETY CONSTRAINTS

For every behaviour selection performed by our policy, we first employ a safety constraint to determine the safe subset $\mathcal{C}_{safe}$ of all available candidate behaviours with respect to the robot's current state. We can use different constraints depending on our knowledge of the environment and the intended risk aversion of our exploration. In the experiment section below, we evaluate the following constraints, all of which are based on the predicted robot state $s'$ after the execution of each imagined behaviour (given the current robot state $s$):

- As a *minimal constraint* we consider only candidate behaviours with $\epsilon(s') > 0$ to ensure we never execute a behaviour that was already expected to be unsafe.
- Alternatively, a *contextual constraint* carries weight only if the current robot state is near the border of the region of unsafe states ($\epsilon(s) \approx 0$), but enables free exploration if it is far away from potential danger ($\epsilon(s) \approx 1$):

$$\epsilon(s') > \epsilon(s) \cdot (1 - \epsilon(s)) \tag{3}$$

- If we have access to the gradient of the epsilon function, the direction of maximal improvement of safety with respect to the next state can be computed as $\nabla_s \epsilon(s)$. The *gradient-minimal constraint* considers only solutions moving in the general direction of the gradient. Based on the dot product of the unit vectors of the gradient of the epsilon function ($\nabla_s \epsilon(s)$) and the projected movement in state space ($s' - s$), we formulate a lower bound for deviation from the direction of the gradient as:

$$\frac{s' - s}{||s' - s||} \cdot \frac{\nabla_s \epsilon(s)}{||\nabla_s \epsilon(s)||} \geq 0 \tag{4}$$

  Geometrically, this is equal to a maximum deviation of 90° in 2D space (Figure 7 - green semicircle).
- Again, we can modify this into a more strict *gradient-contextual constraint* by using the value of epsilon at the current state of the robot to modulate the constraint. This way, the constraint is more relaxed towards the centre of the region of safe states but only accepts small deviations from the direction of the safety gradient close to the border of the region of unsafe states:

$$\frac{s' - s}{||s' - s||} \cdot \frac{\nabla_s \epsilon(s)}{||\nabla_s \epsilon(s)||} \geq \epsilon(s) \cdot (1 - \epsilon(s)) \tag{5}$$

  Geometrically, this is equal to a deviation from the gradient proportional to $\epsilon(s)$ (see yellow region in Figure 7 (Appendix 4.3 for $\epsilon(s) = 0.5$).
- Finally, safety can also be enforced not by a hard constraint, but as a component of the prioritization measures. This can be useful as a supplement to gradient-free constraints in complex environments.

### 4.4 Prioritization Metrics

After the safe subset of candidate behaviours $\mathcal{C}_{safe}$ has been selected based on the safety constraint, the remaining candidates are ranked according to a prioritization measure as the second step in behaviour selection. This gives priority to the evaluation of candidate behaviours which have the highest value for the overall QD algorithm performance, as real-world samples are expensive. Finally, the candidate with the highest prioritization score is selected. The composition of prioritization measures can be adapted depending on the task at hand. We can either use a single prioritization measure or a (weighed) sum of multiple values.

Firstly, the robot's **safety** can be considered again as a prioritization measure through the dynamic exploration parameter $\epsilon(s')$ as outlined above. Generally, this approach will be used in combination with another metric to enable the behaviour policy to tolerate a possible safety violation in favor of a higher score. Another key measure to score a candidate behaviour is the **dynamics model disagreement** (Equation 1). Further details on the use of this metric can be found in Appendix A.1.2 We also consider the classical metrics used to quantify behaviours in QD, especially the **novelty** of a candidate behaviour as the distance to the k nearest solutions already in the archive $(\nu_1, ..., \nu_k)$ (Lehman & Stanley, 2011a).

### 4.5 Recovery Policy

As a final safeguard to keep the robot in the safe region of the environment, we introduce a recovery policy to return the robot to safety if it ever violates any of the environment's safety constraints. These constraints can be derived from the environment in various ways, e.g. as a minimum distance to obstacles represented by 'safety regions' as in this work. Should the robot ever leave the safe region, the discovery of new behaviours will be halted. A greedy behaviour selection policy will be employed over the archive of behaviours that were already evaluated instead of the imagined buffer of candidate behaviours. Here, we pick the single behaviour that is projected to effect the greatest improvement in safety.


Figure 3: Example trajectories of DA-QD, Vanilla-QD and RF-QD in flat environment with safe region (green) and dangerous region (red).

Table 1: Safety metrics for all variants, averaged over 10 runs (mean ± std).

| Variant | Resets | Steps outside safety | Recovery steps |
|---|---|---|---|
| Vanilla-QD | 54.0 ± 4.2 | 908.0 ± 74.1 | n/a |
| DA-QD | 114.0 ± 17.8 | 1039.5 ± 51.0 | n/a |
| RF-QD | 0.0 ± 0.0 | 1.0 ± 2.8 | 3.5 ± 9.9 |

## 5 EXPERIMENTS

We evaluate our method with an 18 DoF hexapod robot on an adapted version of the omni-directional locomotion task Cully & Mouret (2013). In this task, the robot learns behaviours to walk in every direction from an initial position. For the controllers, we evolve parameters of a sinusoidal control signal that is sent to each motor. This sinusoidal signal acts as a prior towards periodic movement. As we focus on a reset-free setting, all evaluations of new behaviours have to be done sequentially and cannot be parallelised. All simulations are performed in RobotDART building on the Dynamics Animation and Robotics Tookit (DART) simulator (Lee et al., 2018b). To simulate a practical number of trials that would be performed in the real-world, the number of evaluations performed in any single run of the algorithm are limited to 10,000.

### 5.1 BASELINE COMPARISON

Firstly, we evaluate the general capability of the RF-QD method. For this, we compare against "vanilla" QD and DA-QD (Lim et al., 2021) as baselines. As in DA-QD, we use the unstructured archive (Cully & Demiris, 2017) and Isodd Vassiliades & Mouret (2018) variation operator. implementation in all our experiments and baselines. We use a simple flat environment with a circular region of safety with radius $r = 2.0m$. Figure 3 shows example trajectories of the baselines compared to RF-QD. The baselines' random selection of behaviours causes the robot to trail off deeply into the dangerous region, while RF-QD performs its exploration almost entirely within the safe region. The depicted RF-QD run leaves the safe region once, but then deploys the recovery policy (blue line) to return to safety.

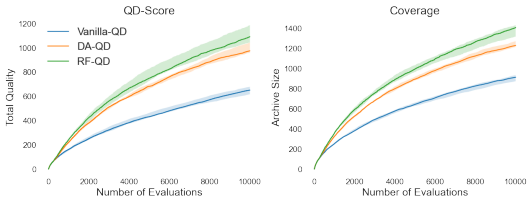


Figure 4: QD-Score and coverage of RF-QD and baselines on the circular safe area environment. The graphs represent the median as a coloured bold line, while the shaded area extends to the first and the third quartiles over 10 runs.

As the baseline methods are not made for a reset-free environment, for all further comparisons we perform manual resets to the starting position if the robot leaves the safe region by more than 50 cm. This is similar to what is done when performing QD on a real-world robot today. For the baseline comparisons, RF-QD was run with a gradient-contextual safety constraint and encouraging maximal novelty through the prioritization strategy. This configuration has proven powerful in our evaluation of different constraints and prioritization measures. Table 1 quantifies the safety of the three algorithms averaged over 10 replications of each. We can see, that RF-QD achieves almost perfect safety - never once requiring a safety reset as described above and only rarely taking a single step outside the safe region.

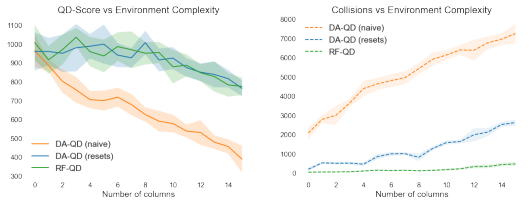


Figure 5: Increasingly complex environments: QD-Scores vs number of obstacles. The bold line is the mean while the shaded area extends to the standard deviations over 10 runs for each environment.

Additionally, RF-QD slightly outperforms its direct baseline DA-QD in terms of both QD-score and coverage as shown in Figure 4. While the distance to vanilla QD is due to DA-QD's increased sample efficiency, RF-QD's behaviour selection policy does not sacrifice performance for safety, but even improves performance by its candidate prioritization strategy (i.e. novelty in this case).

### 5.2 ROBUSTNESS TO ENVIRONMENT COMPLEXITY

To evaluate RF-QD's performance in increasingly complex environments, we exchange the previous circular environment for a closed 4x4m room with a number of column-shaped obstacles. Figure 6 shows examples of such environments including RF-QD's trajectories in them (top row). We can

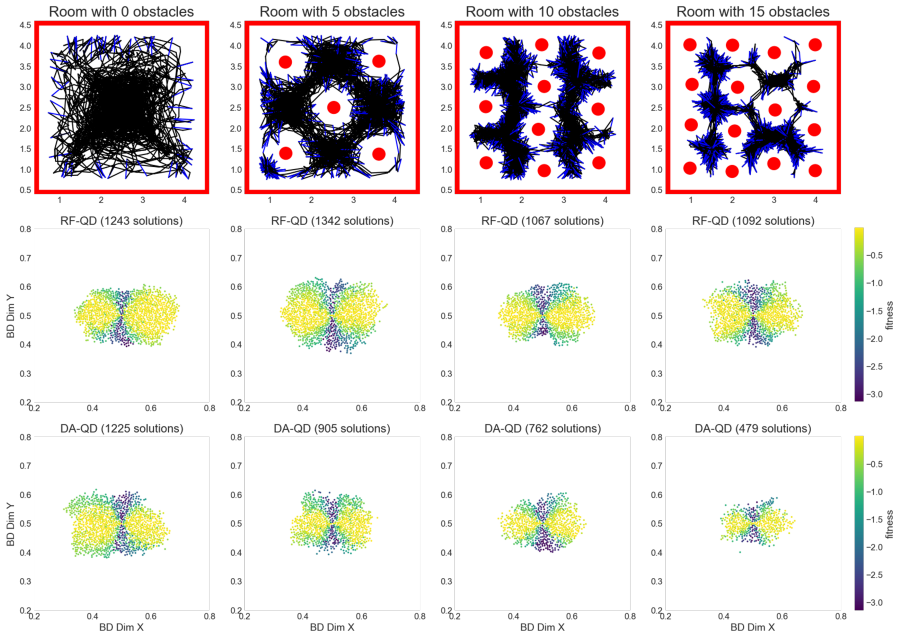


Figure 6: Complex environments with 0, 5, 10 and 15 obstacles. Top: Sample trajectories of hexapod moving with RF-QD. Middle: Example archives by RF-QD. Bottom: Example archives by DA-QD.

observe that the robot acting under RF-QD keeps its distance from the obstacles, while building archives of behaviours (middle row) that are radically less affected by the environment complexity than those created by DA-QD (bottom row).

In these complex environments, we employed RF-QD with a safety-focused configuration. This uses a minimal (hard) safety constraint combined with two equally weighed prioritization measures to select behaviours that maximise safety (through $\epsilon$) and have low model disagreement. As a benchmark for QD performance, we again add a version of DA-QD that uses safety resets, now triggered on any collision with an obstacle. We also keep a 'naive' version of DA-QD, that is not reset upon collision (same as RF-QD). These algorithms were compared in rooms with 0 to 15 obstacles (see Figure 5). While in an empty room, all algorithms perform similarly well, the naive DA-QD variant quickly drops in performance with a growing number of obstacles through a large number of collisions (which render the corresponding evaluations invalid). At the same time, RF-QD manages to fully keep up with the upper baseline of DA-QD (using safety resets). While a more performance-focused prioritization strategy (i.e. novelty as in Section 5.1) for RF-QD might have increased QD-scores slightly (as in Section 5.1), this would have sacrificed the safety of the robot in more challenging environments. Finally, additional experiments performed to study the effect of different objectives in imagination on RF-QD can be found in Appendix A.2.2.

## 6 Discussion

In this paper, we have presented RF-QD, a method to learn behavioural repertoires autonomously without resets in realistic environments. We demonstrate how an intelligent behaviour selection policy can be used with QD in imagination to learn safely and efficiently. We first test RF-QD to learn while remaining within a designated area and show that the behaviour selection policy is necessary to prevent the need for resets and to stay within the safe training area. We then show how RF-QD can also operate in more complex environments with many obstacles and minimal room for error. Our results also show that we can acquire full repertoires despite increasing environment complexity while the performance of DA-QD and Vanilla QD baselines deteriorate with the increase in complexity. Lastly, we conduct an ablation to investigate the effect of the type of solutions present in the candidate buffer on the performance of RF-QD. We demonstrate that using targeted optimization objectives when performing QD in imagination can bias the distribution of solutions presented to the behaviour selection policy. Our results show that it is important to keep the diverse types of solutions in the candidate buffer over just specialised solutions biased towards a single metric. For future work, we hope to show RF-QD learning directly on a real world system, with no dependence on simulators.

### ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/V006673/1 project REcoVER. We would like to thank the members of the Adaptive and Intelligent Robotics Lab for their very valuable comments.

## REFERENCES

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

Konstantinos Chatzilygeroudis, Vassilis Vassiliades, and Jean-Baptiste Mouret. Reset-free trial-and-error learning for robot damage recovery. *Robotics and Autonomous Systems*, 100:236–250, 2018.

Jeff Clune. Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*, 2019.

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

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2020.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 81–89, 2019.

Antoine Cully. Multi-emitter map-elites: Improving quality, diversity and convergence speed with heterogeneous sets of emitters. *arXiv preprint arXiv:2007.05352*, 2020.

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

Antoine Cully and Jean-Baptiste Mouret. Behavioral repertoire learning in robotics. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp. 175–182, 2013.

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

Matthew C Fontaine, Scott Lee, Lisa B Soros, Fernando de Mesentier Silva, Julian Togelius, and Amy K Hoover. Mapping hearthstone deck spaces through map-elites with sliding boundaries. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pp. 161–169, 2019.

Matthew C Fontaine, Ruilin Liu, Ahmed Khalifa, Jignesh Modi, Julian Togelius, Amy K Hoover, and Stefanos Nikolaidis. Illuminating mario scenes in the latent space of a generative adversarial network. *arXiv preprint arXiv:2007.05674*, 2020a.

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

Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation*, 26(3):381–410, 2018.

Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*, pp. 1–8. IEEE, 2019.

Luca Grillotti and Antoine Cully. Unsupervised behaviour discovery with quality-diversity optimisation. *arXiv preprint arXiv:2106.05648*, 2021.

Abhishek Gupta, Justin Yu, Tony Z Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. *arXiv preprint arXiv:2104.11203*, 2021.

Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort. *arXiv preprint arXiv:2002.08550*, 2020.

Leon Keller, Daniel Tanneberg, Svenja Stark, and Jan Peters. Model-based quality-diversity search for efficient robot learning. *arXiv preprint arXiv:2008.04589*, 2020.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL : Model-Based Offline Reinforcement Learning, 5 2020. URL http://arxiv.org/abs/2005.05951.

Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, Feb 2018a. doi: 10.21105/joss.00500. URL https://doi.org/10.21105/joss.00500.

Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic Animation and Robotics Toolkit. *The Journal of Open Source Software*, 3(22), 2 2018b. ISSN 2475-9066. doi: 10.21105/joss.00500.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

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

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

Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. Dynamics-aware quality-diversity for efficient learning of skill repertoires. *arXiv preprint arXiv:2109.08522*, 2021.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Jean-Baptiste Mouret and Glenn Maguire. Quality diversity for multi-task optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 121–129, 2020.

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

Olle Nilsson and Antoine Cully. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 866–875, 2021.

Giuseppe Paolo, Alban Laflaquiere, Alexandre Coninx, and Stephane Doncieux. Unsupervised learning and exploration of reachable outcome space. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2379–2385. IEEE, 2020.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

Thomas Pierrot, Valentin Macé, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin. Diversity policy gradient for sample efficient quality-diversity optimization. 2021.

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

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Kenneth O Stanley. Why open-endedness matters. *Artificial life*, 25(3):232–235, 2019.

Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself*, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

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

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

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Poet: open-ended coevolution of environments and their optimized solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 142–151, 2019.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

# A Appendix

## A.1 Method Details

This section of the appendix provides more detailed description of of parts of the methods presented in this paper to provide more clarity.

### A.1.1 Safety Constraints

This figure provides a more visual explanation of the safety constraints described and referenced in Section 4.3

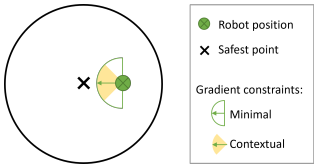


Figure 7: Sketch of the gradient-based safety constraints in a simple circular 2D-environment.

### A.1.2 Dynamics Model Disagreement

The dynamics model used in DA-QD consists of an ensemble of models to capture the epistemic uncertainty via disagreement between the predictions of the models (see Section 3 and Equation 1). The epistemic uncertainty can also be interpreted and formalised as an information theoretic measure of the expected information gain (Pathak et al., 2019; Sekar et al., 2020). Maximising the model-disagreement has been used as a self-supervised intrinsic reward for exploration in Deep RL literature (Pathak et al., 2019; Sekar et al., 2020). The key idea behind this measure is to prioritise policies that are most informative based on our current knowledge which is represented via the ensemble of dynamics models (i.e. epistemic uncertainty). Selecting policies with high-model disagreement would mean visiting states that have been less explored than others. As we incrementally train the dynamics model on incoming data, policies that visit states that have been seen will no longer have a large model disagreement which will allow this measure to continuously be used to explore. Depending on the state of the robot in the environment, we can prioritize high

and low model disagreement behaviours. Conversely, policies with low disagreement should be prioritized in safety-critical situations. Solutions with low expected model disagreement are likely to resemble the expected outcome and indicates the model's confidence.

### A.1.3 PSEUDOCODE

**Algorithm 1** Reset-free Quality-Diversity (RF-QD)

**Input:** archive $\mathcal{A} = \emptyset$, candidate buffer $\mathcal{C} = \emptyset$,
dynamics model $\widetilde{p}_{\vec{\theta}}$, experience replay buffer $\mathcal{B}$

$\mathcal{A}, \mathcal{B} \leftarrow$ random_archive_initialization()
$\widetilde{p}_{\vec{\theta}} \leftarrow$ train_model($\mathcal{B}$)

**for** max_iterations **do**
  **if** $\mathcal{C} == \emptyset$ **then** ▷ Fill the candidate buffer
    $\widetilde{\mathcal{A}} \leftarrow$ generate_candidates($\mathcal{A}, \widetilde{p}_{\vec{\theta}}$) ▷ Using imagination
    $\mathcal{C} \leftarrow \widetilde{\mathcal{A}} \backslash \mathcal{A}$
  $s \leftarrow$ get_robot_state() ▷ Get state to evaluate safety
  **if** $s$ is safe **then** ▷ Apply candidate selection policy
    $\mathcal{C}_{safe} \leftarrow$ apply_safety_constraint($\mathcal{C}$)
    x $\leftarrow$ argmax(prioritize_candidates($\mathcal{C}_{safe}$))
    $b_x, f_x \leftarrow$ execute_behaviour(x)
    $\mathcal{A} \leftarrow$ add_to_archive(x, $b_x$, $f_x$)
  **else** ▷ Apply recovery policy to return to safety
    x $\leftarrow$ recovery_policy($\mathcal{A}$)
    execute_behaviour(x)
  $\mathcal{B} \leftarrow$ add_to_replay_buffer()
  $\widetilde{p}_{\vec{\theta}} \leftarrow$ train_model($\mathcal{B}$)

## A.2 SUPPLEMENTARY RESULTS

This section provides extended results and experiments performed on DA-QD. In Appendix A.2.1, we ablate the different prioritization metrics and safety constraints. In Appendix A.2.2, we study the effect of different objectives in imagination and its effect on the behaviour selection policy used in RF-QD.

### A.2.1 COMPARISON OF POLICY CONFIGURATIONS

Additionally, we evaluated the various configurations of the Behaviour Selection Policy as introduced in Section 4.1. Figure 8 shows an overview over the different combinations of safety constraints and prioritization measures. Here, the policy configurations are evaluated by performance (represented by their final coverage) and safety (represented by the number of recovery steps), both from runs of 10,000 steps over 10 replications. In short, Figure 8 shows strong separation between the relatively unsafe minimal and contextual constraints (both gradient-free) and all remaining constraints. The strongest performance is exhibited by variants combining the novelty or disagreement maximising prioritization measures with a gradient contextual constraint. Out of the naive gradient-free constraints, which must be used if there is no single 'safest' direction of movement (as e.g. in more complex environments such as the one following in Section 5.2), only the soft constraints achieves comparable safety scores and performances as the gradient-based configurations. Which exact configuration should be chosen will however always depend on the exact task at hand.

### A.2.2 EFFECT OF OBJECTIVES IN IMAGINATION

We also study the effect of the type of solutions available in the candidate buffer that the behaviour selection policy chooses from. To study this, we investigate the influence of different optimisation objectives for the generation of the candidate buffer during the QD in imagination. When using Iso-DD (Vassiliades & Mouret, 2018), the solutions are relatively generic and objective-agnostic, i.e., not optimised to fulfil a specific objective. Alternatively, we can use different types of emitters (introduced by CMA-ME (Fontaine et al., 2020b)) to produce solutions that maximise a specific

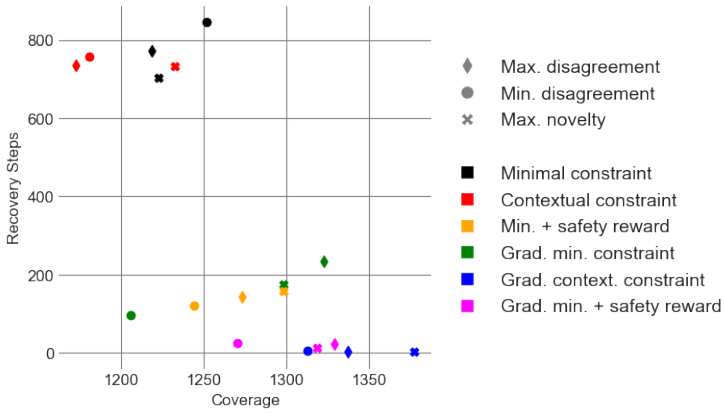


Figure 8: Comparison of different Behaviour Selection Policy configurations on both performance (coverage) and safety (recovery steps) on the circular safe area environment.

objective. We perform experiments using three different optimization objectives: maximising model disagreement, minimising model disagreement, and a random direction objective as a surrogate objective for novelty. We compare this to the standard Iso-dd variations used in all our experiments as a baseline. We perform an ablation of these three different objectives with their corresponding prioritization measures used in the behaviour selection policy. We report results across 10 replications.

First, we evaluate the effect of more targeted objectives by analysing the model disagreement associated with the individuals selected by the behaviour selection policy (Figure 9). The key take-away from Figure 9 (top) is that the optimisation objectives used when running QD in imagination can strongly influence the behaviours that are finally selected. We can see that regardless of the prioritization metric used by the behaviour selection policy, the same overall trends are always observed: The minimising disagreement optimization objective (yellow) always results in low disagreement individuals being selected by the behaviour selection policy regardless of the prioritization metrics. The same observation applied to the maximising disagreement objective (green). This observation corresponds to our initial hypothesis where targeted optimization objectives can skew the distribution of solutions generated towards the target objective. This results in a higher probability for the solutions with the desired metric being selected.

Given that biased/specialised sets of solutions can be generated in the candidate buffer using more targeted objectives, we evaluate the effect of the composition of this candidate buffer on the performance of RF-QD. Figure 9 (middle and bottom) show that the objective-agnostic Iso-DD operator outperforms all the targeted optimization objectives both in terms of coverage and safety (number of resets) across all prioritization measures used by the behaviour selection policy. This is an interesting result as one could expect the variants with aligned prioritization measures and optimization objectives to perform better. We hypothesize that the buffer of candidate solutions being generated by targeted objectives become too specialised while the objective-agnostic Iso-DD can generate a diverse buffer of solutions to choose from. This is not such a surprising observation as Multi-Emitter MAP-Elites (Cully, 2020) had previously also shown that when using simultaneously multiple emitter types, the random emitter (based on Iso-dd) remains the most fruitful through the entire process compared to other objective-driven emitters.

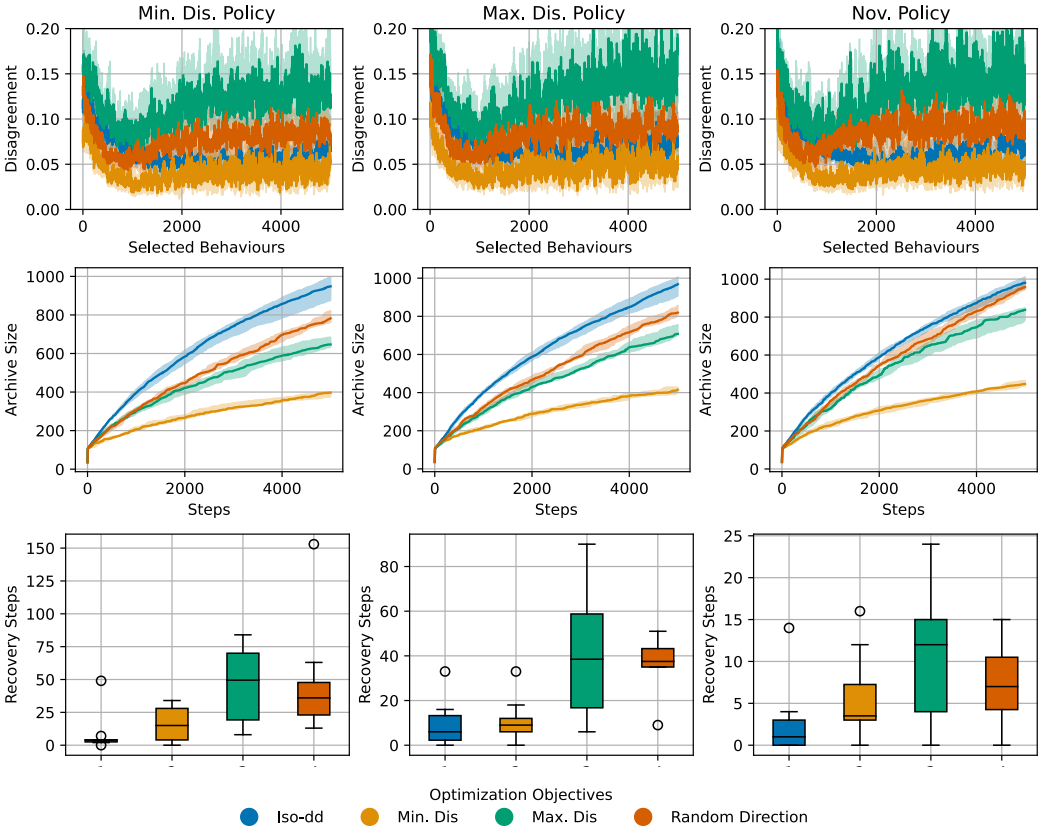


Figure 9: Study of different optimization objectives and prioritization metric configurations. Each panel considers a different prioritisation metric. Top: Disagreement of selected behaviours by RF-QD. The bold lines and shaded areas represent the median and interquartile range over 10 replications respectively. Middle: Progression of the archive size over the number of selected behaviours for each optimization objective. Bottom: Distribution of the total number of recovery steps for each optimization objective.