QED: using Quality-Environment-Diversity to evolve resilient robot swarms

David M. Bossens and Danesh Tarapore

***Abstract*—In quality-diversity algorithms, the behavioural diversity metric is a key design choice that determines the quality of the evolved archives. Although behavioural diversity is traditionally obtained by describing the observed resulting behaviour of robot controllers evaluated in a single environment, it is often more easily induced by introducing *environmental diversity*, i.e., by manipulating the environments in which the controllers are evaluated. This paper proposes Quality-Environment-Diversity, an algorithm that repeatedly generates a random environment according to a probability distribution over environmental features (e.g., number of obstacles, arena size and robot sensor and actuator characteristics), evaluates the controller in that environment, and then describes the controller in terms of the features of that environment, the *environment descriptor*. Our study compares Quality-Environment-Diversity to three baseline task-specific and generic behavioural descriptors, in 5 different robot swarm benchmark tasks. For each task, the quality of the evolved archives is assessed by their capability to provide high-performing compensatory behaviours following injection of 250 unique faults to the robots of the swarm. The evolved archives achieve a median 2- to 3-fold reduction in the impact of the faults on the performance of the swarm. A qualitative analysis of evolved archives is done by visualising the relation between diversity of compensatory behaviours, here called *useful* behavioural diversity, and fault recovery metrics. The resulting signatures indicate that, due to the diversity of environments inducing useful behavioural diversity, archives evolved by QED provide robot swarm controllers that are capable of recovering from high-impact faults.**

***Index Terms*—quality-diversity algorithms, evolutionary robotics, behavioural diversity, fault recovery, swarm robotics**

I. Introduction

Swarm robotics [1], [2] studies the emergence of collective behaviour in large-scale teams of robots. The robots in a swarm are simple and have limited sensory, computational and communication capabilities. The tasks they have to accomplish are relatively complex and may not be achievable by individual members of the swarm [3]. Robot swarms have been investigated in loosely-coordinated tasks such as collaborative exploration, monitoring and surveillance [4], [5], as well as tightly-coordinated tasks such as self-assembly [6], coordinated movement [7] and foraging [8], [9].

Despite the robustness conferred to robot swarms by the decentralised inter-robot coordination [10], they remain brittle systems that are rendered inoperable when inadvertent faults are sustained by individual robots of the swarm. Studies on fault tolerance in swarm robotics have revealed that even partial failures to one or a few robots may significantly hamper the capability of the swarm to complete its mission [11], [12]. In developing fault-tolerant robot swarms, many studies have investigated fault-detection algorithms, for individual robots of the swarm to robustly detect faults, both endogenously in themselves [13], and exogenously in neighbouring robots [14], [15], [16], [17]. However, to the best of our knowledge, fault-recovery in swarm robotics, wherein robots of the swarm adapt their behaviour to compensate for the different faults that they may sustain, is an open challenge [18].

Fault recovery has been previously investigated in the context of single-robot systems with multiple actuated degrees of freedom providing redundancy, such as quadruped and hexapod walking robots and multi-joint pick-and-place robot arms [19]. Many of these studies are model-based, and involve updating the model of the robot to restore the accuracy of movements when unexpected faults, such as damages in the actuators, perturb the robot-environment interaction [20], [21], [22]. While such an approach is promising when the model learned is accurate, its extension to swarm robotic systems seems elusive, since: a) even small deviations of the self-model to reality may accumulate rapidly when considering the large number of robots in the swarm; and b) the self-model of one robot may not be able to anticipate changes due to faults sustained by other robots of the swarm. An alternative, model-free approach, explored for a hexapod robot with damaged actuators [23], is to recover from faults by intelligently searching over a diverse archive of walking behaviours. Such behaviours were evolved using *quality-diversity algorithms* [24], [25], special evolutionary algorithms that aim to evolve a behaviourally diverse archive of high-performing solutions. This model-free approach appears promising for fault recovery in a robot swarm.

An important aspect of quality-diversity algorithms is the choice of behavioural descriptor, a feature vector that is used to characterise the behaviours of the evolved solutions based on observable characteristics during the fitness evaluation. Many studies employing quality-diversity algorithms have relied on hand-coded descriptors based on domain-specific insights or dimensions that are of particular interest to the end-user [26], [27], [28], [29], [30]. Generic behavioural descriptors have recently been proposed, including methods that automatically derive the behavioural description from the sensor-actuator trajectories of the robot [31], [32], such as Stochastic Policy Induction for Relating Inter-task Trajectories (SPIRIT) [31], without the need for any domain-specific information. Other approaches such as Systematically Derived Behaviour Characterisations (SDBC) [33] exploit domain-specific information in a systematic manner, deriving the behavioural description from the averaged relations between

Authors are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K. d.m.bossens@soton.ac.uk

different entities, such as robots and objects of interest in the environment. Importantly, for swarm robotic systems, the effect of the choice of behavioural descriptor on the quality of the evolved archive *for fault recovery* remains to be investigated.

A commonality amongst quality-diversity algorithms is that all of the individuals in the archive are evaluated in the same operating environment. However, the bias-variance dilemma [34] implies that controllers trained or evolved in a specific environment will excel in that environment but may fail to generalise to the larger domain of interest. To allow the evolution of robust robot controllers, various studies have explored modifying the fitness evaluation to provide tolerance to faults injected in the robot hardware [35], to bridge the robot simulation-reality gap [36], [37], and produce robot controllers with better generalisation capabilities [38]. Moreover, recent work in active curriculum learning [39], [40] and open-ended co-evolution [41], [42] demonstrates the beneficial impact of enabling controllers to learn on ever-more challenging and diverse environments targeted to the controller's current skill levels. Other studies in evolutionary computation show that recording a variety of solutions associated with the task-objective solved, provides evolution with stepping stones leading to higher behavioural diversity and the ability to solve problems of greater complexity [43], [44]. The above-mentioned findings demonstrate that the environment in which learning takes place is crucial for generalisation, robustness, and complex skills; therefore, some robot behaviours may best, if not *only*, be obtained by learning in a variety of environments.

We investigate the effect of environmental diversity in quality-diversity algorithms on the quality of the evolved archive of solutions, and consequently on the fault-recovery performance of the robot swarm. Quality-diversity algorithms may be improved by evaluating robot swarms in a diversity of environments because the selected behavioural descriptors may omit important information on the dynamics of the swarm (e.g., the summary statistics of the SDBC descriptor [33] fail to capture inter-robot interactions at fine temporal resolutions). Additionally, a given behaviour may be more easily characterised by the type of environment in which it is high-performing (e.g., a high-performing behaviour for patrolling *a cluttered arena with a low robot density swarm*). The contributions of this work are the following:

- a novel framework, called Quality-Environment-Diversity (see Section II), which describes the behaviour of individual solutions based on the randomly generated environment in which they are evaluated – such that an environment descriptor serves as a convenient *implicit* behavioural descriptor and encourages *useful* diversity, variations in behaviour tailored to plausible environmental challenges;
- a comparative study of the impact of the choice of the behavioural descriptor on behavioural diversity (see Section IV-A) and fault recovery (see Section IV-B) in five common swarm robotics tasks;
- a visualisation tool to analyse useful behavioural diversity and performance in fault recovery (see Section IV-B).

II. Quality-environment-diversity algorithm

In quality-diversity algorithms, one evolves an archive of high-performing and behaviourally diverse solutions. In this context, a genotype is a set of parameters that encodes the solution (e.g., a neural network) to a particular problem (e.g., controlling a robot). To obtain behavioural diversity, each solution is evaluated in the same environment, $\mathcal{E}$, which represents all aspects of the application of interest other than those encoded by the genotype. The evaluation in $\mathcal{E}$ results in two outcomes of interest: i) a fitness score, representing the solution's performance; and ii) a behavioural descriptor, a vector of floating-point features of interest to the end-user that may be orthogonal to the fitness (e.g., various walking gaits of a multi-legged robot, the distance between different mobile robots in a swarm, etc.). The behavioural descriptor serves as a practical description of the phenotype, the observable characteristics of the genotype when interacting with the environment, and storing solutions with varying behavioural descriptor in an archive results in behavioural diversity.

This paper proposes quality-environment-diversity algorithms, which evolve an archive of high-performing solutions that is organised based on *environmental diversity* rather than behavioural diversity. To obtain environmental diversity, a probability distribution over environments generates an environment in which a new child solution is to be evaluated. After evolution, the resulting archive then represents high-performing solutions for widely differing regions in an environment space. The environment space is centred around a normal operating environment $\mathcal{E}$, which represents the typical application of interest. We hypothesise that quality-environment-diversity results in a generic and implicit description of *useful* behavioural diversity.

a) Quality-environment-diversity with MAP-Elites: We use quality-environment-diversity to evolve parameters of a neural network controller for robots in a homogeneous swarm, where each robot has a clone of the same controller. In this case, the environment comprises all aspects of the simulation that are extraneous to the controller – this includes the surroundings (e.g., obstacles, friction, weather) but also how the robots are embodied (e.g., sensory range, actuator velocity, number of robots). To evolve the neural network controller, we make use of a direct encoding of a recurrent neural network with mutation operators that can change the topology – by adding or removing neurons or connections – as well as change the interconnection weights between neurons [45]. More details about the evolutionary operators and parameters are provided in Section I.B of Supplemental Materials.

The QED implementation is based on Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [26], a quality-diversity algorithm used in numerous studies in evolutionary computation [46], [47], [30], [48], [29], [26], [25], [28]. MAP-Elites evolves a topologically organised behaviour-performance map $\mathcal{M}$, or *map*. The map's organisation is based on a behavioural descriptor, a D-dimensional vector in which each dimension represents a feature of the robot swarm's behaviour. Our QED algorithm differs from MAP-Elites in that solutions are located in the map $\mathcal{M}$ based on their *environment descriptor*, a set of features of the environment in which the solution has been evaluated.

The QED algorithm is initialised with an empty map $\mathcal{M}$, and first generates a set of random controllers P. An environment generator then randomly generates an environment $\tilde{\mathcal{E}}$, one for each controller $i \in \{1, \ldots, |P|\}$, and evaluates the performance of the controller in that environment. Each of the evaluated solutions is described by the environment descriptor of the environment it was evaluated in. Finally, similar to MAP-Elites, if the performance of the evaluated solution exceeds that of the current solution at that location in the environment-performance map, it is added to the map, replacing the solution at that location. Therefore, solutions are only retained in the environment-performance map if they are the best for the environment type defined by their location in the environment-performance map or if no solution exists at that location. After initialisation, QED improves the solutions in the environment-performance map through: i) the generation of new environments; and ii) random variation and selection of the existing solutions in the map. At each iteration, the algorithm picks a solution from the map at random, following a uniform distribution. A copy of that solution is then randomly mutated (see Table S1 for parameters of mutation operators). The environment generator generates a random environment $\tilde{\mathcal{E}}$, in which the mutated solution is evaluated. If it outperforms the current solution at the location corresponding to $\tilde{\mathcal{E}}$, the mutated solution is inserted in that location. The evolutionary process is repeated until the maximal number of evaluations is expended. An implementation of the QED algorithm is illustrated in Algorithm 1.

Algorithm 1 Quality-Environment-Diversity with MAP-Elites, evolving a D-dimensional environment-performance map $\mathcal{M}$.

```
1:  $\mathcal{M} \leftarrow \emptyset$                          ▷ Empty $D$-dimensional map.
2:  for $i = 1$ to $p$ do                      ▷ Random initial population.
3:      $P[i] \leftarrow$ random-controller()
4:      Perform add-controller($P[i]$)
5:  end for
6:  for $i = 1$ to $I$ do                      ▷ Repeat for $I$ iterations.
7:      $c \sim \mathcal{M}$          ▷ Randomly select controller $c$ from $\mathcal{M}$.
8:      $c' \leftarrow$ mutate($c$)      ▷ Mutate $c$ (details in Table S1).
9:      Perform add-controller($c'$)
10: end for
11: procedure ADD-CONTROLLER(controller $c$)
12:     Randomly select $A_j \in \mathbf{P}_j \; \forall j \in \{1, \ldots, D\}$
13:     Generate environment $\tilde{\mathcal{E}}$ parametrised by $\mathbf{A}$
14:     $\beta \leftarrow$ environment-descriptor($\tilde{\mathcal{E}}$)
15:     if $\mathcal{M}[\beta] = \emptyset$ or $f(\tilde{\mathcal{E}}, c) > f(\tilde{\mathcal{E}}, \mathcal{M}[\beta])$ then
16:         $\mathcal{M}[\beta] = c$        ▷ Add individual $c$ to the map $\mathcal{M}$.
17:     end if
18: end procedure
```

b) Environment generation for quality-environment-diversity: The generation of a diverse set of environments is an essential aspect of the QED algorithm. In our implementation of the QED, the environment for the robot swarm is characterised by the following six attributes $A = \langle A_1, \ldots, A_6 \rangle$: i) maximum linear speed of the robots in the swarm; ii) size of the robot swarm; iii) size of the arena the swarm is operating in; iv) number of obstacles in the arena; and v) maximum range-and-bearing sensor range of the robots in the swarm; and vi) maximum proximity sensor range of the robots in the swarm. The environment attributes are selected to elicit a diverse repertoire of swarm behaviours from perturbations reshaping the pathways of robots' sensory-motor interactions. In our environment generator, the values for each of the attributes $A_j,\ j \in \{1, \ldots, 6\}$ of the generated environments are randomly selected following a uniform distribution from a select set of perturbations around the value of the attribute in a normal operating environment. The selected perturbations on the environment attributes typically vary in range from a quarter of to two-fold the normal value of the attribute (see Table I for details on the range of perturbations).

TABLE I: Environment attributes of the QED algorithm, and their value in normal and perturbed environments. The environment attributes are characteristics of the robots of the swarm, and their operating environment.

Attribute	Description	Value	Perturbations injected
A_1	Robots' maximal linear speed	$10\,\mathrm{cm/s}$	$\mathbf{P}_1 = \{5, 10, 15, 20\}\mathrm{cm/s}$
A_2	Number of robots in the swarm	10	$\mathbf{P}_2 = \{5, 10, 15, 20\}$
A_3	Arena size	$16\,\mathrm{m}^2$	$\mathbf{P}_3 = \{4, 9, 16, 25\}\mathrm{m}^2$
A_4	Number of obstacles	0	$\mathbf{P}_4 = \{0, 2, 4, 6\}$
A_5	Robots' range-and-bearing sensor range	$1\,\mathrm{m}$	$\mathbf{P}_5 = \{25, 50, 100, 200\}\mathrm{cm}$
A_6	Robots' proximity sensor range	$11\,\mathrm{cm}$	$\mathbf{P}_6 = \{5.5, 11, 22, 44\}\mathrm{cm}$

III. Experimental method

A. Robot swarm simulation

To evaluate the fitness and the behavioural descriptor of the solutions, we use a physics-based, discrete-time robot swarm simulator named ARGoS [49], designed to realistically simulate complex swarm behaviours. Exemplary simulations of the normal operating environment and a QED environment are shown in Fig. 1.

In a normal operating environment, the robot swarm simulated is composed of 10 two-wheel differential-drive Thymio mobile robots [50] in an environment of size 4×4 m^2. The Thymio robot is $11\,\mathrm{cm}$ long and $8.5\,\mathrm{cm}$ wide, with a maximum linear speed of $10\,\mathrm{cm/s}$, a maximum angular speed of $127.32\,^\circ/\mathrm{s}$, and a control cycle of $0.20\,\mathrm{s}$. The Thymio robot model is equipped with five frontal and two rear infrared proximity sensors of range $11\,\mathrm{cm}$ for obstacle avoidance and two actuators which control the robot's speed and direction. The model is further augmented with 8 range-and-bearing sensors with range of $1\,\mathrm{m}$ to estimate the distance of the closest neighbouring robot within angular segments of size $45\,^\circ$ (the maximal distance within range is chosen if no robot is in range for that segment). Each robot in the swarm is controlled by a copy of the same neural network controller, which takes as inputs the activations of the 7 proximity sensors, the 8 range-and-bearing sensors, and a bias activation. The outputs of the neural network are the left and right wheel velocities of the

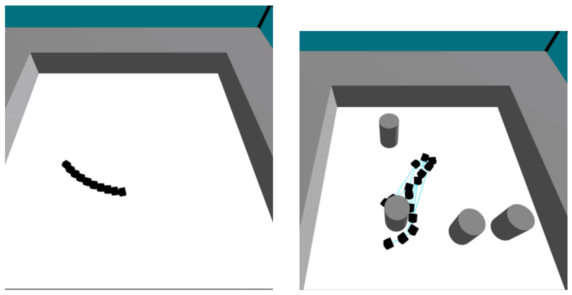

(a) Normal environment (b) Cluttered environment

Fig. 1: Exemplary simulations in (a) the normal operating environment and (b) a cluttered environment based on a $3 \times 3\,\mathrm{m}^2$ arena with 15 robots and 4 obstacles, one of many QED environments.

robot. All sensory activations input to the neural network are scaled in the range $[-1, 1]$, while the output neuron activations in range $[-1, 1]$ are mapped into wheel velocities in the range $[-10, 10]\ \mathrm{cm/s}$.

In QED evolution, the normal operating environment is as described above, however, attributes of the simulation are varied according to Table I. In this case, any obstacles introduced into the environment are static, cylindric objects with height $0.5\,\mathrm{m}$ and a radius of $0.15\,\mathrm{m}$.

To ensure desirable swarm behaviours, a performance evaluation consists of multiple trials, such that robots of the swarm and obstacles in the environment, if any, are positioned randomly at the start of each trial. The performance evaluation comprises 50 trials for evolutionary experiments and 10 trials for analysing the behaviours after evolution, and each fitness trial has a duration of $400\,\mathrm{s}$.

B. Tasks for the robot swarm

Robot swarm controllers are evolved in independent experiments for the aggregation, dispersion, flocking, patrolling, and border-patrolling tasks. Multiple evaluation trials are performed for each of the tasks to evaluate fitness accurately. We outline the robot swarm tasks below (for detailed task specifications and fitness functions, see Section S1.A of Supplemental Materials):

- **Aggregation:** Robots of the swarm are tasked to form a coherent and stable cluster. The fitness function penalises robots for having a large distance to the centre of mass of the swarm, and averages the penalty over robots in the swarm and control cycles in the evaluation trial.
- **Dispersion:** The robots of the swarm are tasked with maximising their total sensing coverage of the arena, relevant for the coverage of large areas in patrolling and monitoring missions [5], [51]. The fitness function is defined as the average distance of all the robots of the swarm to their nearest neighbours, averaged across control cycles in the evaluation trial.
- **Flocking:** The swarm is tasked to move in a tightly-coordinated flocking manner. The fitness functions rewards pairs of robots in the swarm for moving rapidly in the same direction within a range of $50\,\mathrm{cm}$. The fitness function considers the average of this reward over the control cycles in the evaluation trial.
- **Patrolling:** Robots of the swarm are tasked to actively patrol an arena, a behaviour relevant for surveillance type missions. The arena of the swarm is discretised uniformly into a 10-by-10 grid of cells, initially all given the minimal cell value of 0. A cell's value is set to 1 when visited by a robot and decays continually at a rate $0.005\,/\mathrm{s}$ otherwise. The fitness function is the average of the values for all the cells in the arena aggregated across control cycles in the evaluation trial.
- **Border-patrolling:** The task, while similar to patrolling, requires the robots of the swarm to patrol along the walls of the arena. The fitness function is similar to patrolling, except that aggregation only considers the outermost cells.

C. Baseline algorithms

Our study compares QED to three baseline algorithms, variants of MAP-Elites that employ different *behavioural* descriptors, ranging from task-specific to generic: i) a 3-dimensional hand-coded descriptor designed with detailed knowledge of the swarm robotic tasks, in the sense that the shape and dimensions of the arena are assumed to be known and that it may not be suitable when the environment contains dynamic entities other than the swarm; ii) Systematically Derived Behaviour Characterisations (SDBC), a 10-dimensional behavioural descriptor that requires some domain-specific knowledge on the task domain, namely the types of entities of interest, but is otherwise generic [52]; and iii) Stochastic Policy Induction for Relating Inter-task Trajectories (SPIRIT), a completely generic 1024-dimensional behavioural descriptor of the policy of the robot [31]. Importantly, as maps generated with the QED algorithm may contain a maximum of 4096 solutions (six environment attributes and four perturbations per attribute, see Table I), the behaviour-performance maps of the baseline algorithms are discretised to contain the same number of 4096 solutions. To find a suitable discretisation for high-dimensional behavioural descriptors, we apply the Centroidal Voronoi Tesselations (CVT) MAP-Elites algorithm [53] for SPIRIT and SDBC.

Hand-coded behavioural descriptor (HBD): During the evaluation trial, the behavioural descriptor tracks the positions of the robots in a swarm over a uniform grid of cells of size equal to the Thymio robot, to compute the following features: i) the uniformity of the visitation probabilities of different cells in the arena; ii) the average distance of the robots to the centre of the arena; and iii) the total number of cells visited by the robots in the arena at least once during a trial. The resulting behavioural descriptor for the swarm is the average of these features across all evaluation trials.

Systematically Derived Behaviour Characterisations (SDBC): We replicate the 10-dimensional characterisation used in [52], the mean and standard-deviation over the evaluation trial of five features recorded at each control cycle: i) the average linear velocity of the robots in the swarm; ii) the average angular velocity of the robots in the swarm; iii) the average distance between the robots of the swarm and the walls of the arena; iv) the average distance between each robot in the swarm; and v) the average distance between each robot in the swarm and its closest neighbouring robot. The resulting behavioural descriptor for the swarm is the geometric median of these features across all evaluation trials.

Stochastic Policy Induction for Relating Inter-task Trajectories (SPIRIT): The behavioural descriptor counts the frequencies of sensory states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ over all the robots of the swarm and all the evaluation trials during the simulation, to estimate conditional probabilities $p(a|s)$ for all $a \in \mathcal{A}$ and all $s \in \mathcal{S}$. If a state s has been visited N times, where $N > 0$, then $p(a|s)$ is estimated as the frequency of the state-action pair (s, a) divided by N. If a state $s \in \mathcal{S}$ has not been visited, $p(a|s) = 1/|\mathcal{A}|$ for all $a \in \mathcal{A}$. The action space $\mathcal{A}$ comprises the left and right wheel velocities of the robot, each binned into four equal sized intervals in range $\pm 10\,\mathrm{cm/s}$. The sensor space $\mathcal{S}$ comprises 6 sensor groups, the front-left, front-centre, front-right and rear proximity sensors of the robot, as well as the front- and rear-facing range-and-bearing sensors, each binarised such that a sensor group is considered active (set to 1) if any one of its sensors' readings exceeds half of the maximum range of that sensor. Therefore, SPIRIT in this set-up results in $|\mathcal{S}| = 64$ probability distributions of 16 actions each, for a 1024-dimensional descriptor.

D. Metrics for analysis

Performance: The performance of a solution is defined as the fitness averaged across all independent trials in which it is evaluated. To find the best performance for a given environment, QED maps must be re-evaluated[1]. Due to the large number of solutions in QED maps, 50 trials of fitness evaluation would exceed our computational budget. Therefore, the performances of solutions evolved by QED are all re-evaluated for 10 trials each, and for comparable estimates of performance, the solutions evolved by the other algorithms (HBD, SDBC, and SPIRIT) are also re-evaluated for 10 trials. Where indicated, the empirical maximum performance of a swarm robotic task, the maximal performance observed across all replicates and algorithms, is used to normalise performance.

Behavioural diversity: For a behavioural diversity metric comparable across QED, HBD, SDBC and SPIRIT, the solutions of all generated maps are projected into the 1024-dimensional space of SPIRIT, which provides a completely generic behavioural descriptor. The projected maps are further processed into valid behaviour-performance maps by maintaining only the highest performing solution for each of the centroids in the projected SPIRIT space. In this projected space, two probability distributions p_1 and p_2, corresponding to a given pair of behavioural descriptors, are compared by computing the average of the total variation distances between their corresponding conditional probability distributions:

$$d(p_1, p_2) = \frac{1}{2|\mathcal{S}|} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |p_1(a|s) - p_2(a|s)|\,.$$

[1]Note that for QED, the best solution for the normal operating environment may not be at that location in the environment-performance map, thus requiring a re-evaluation of all the solutions in the environment-performance map.

IV. Results

The maps generated by the QED, HBD, SDBC and SPIRIT quality-diversity algorithms are assessed in two phases. In the first phase, the maps are evaluated on their performance in the normal operating environment and on the coverage, i.e., the number of unique solutions in the map (see Section IV-A). In the second phase, we introduce sensor/actuator faults one or more robots of the swarm and assess: i) their impact on the performance of the swarm; ii) how well the swarm is able to recover from them; and iii) as a metric for *useful diversity*, how behaviourally diverse are the recovery solutions (see Section IV-B). The results are aggregated across all swarm robotic tasks; detailed task-specific results are available in Section II-III of the Supplemental Materials.

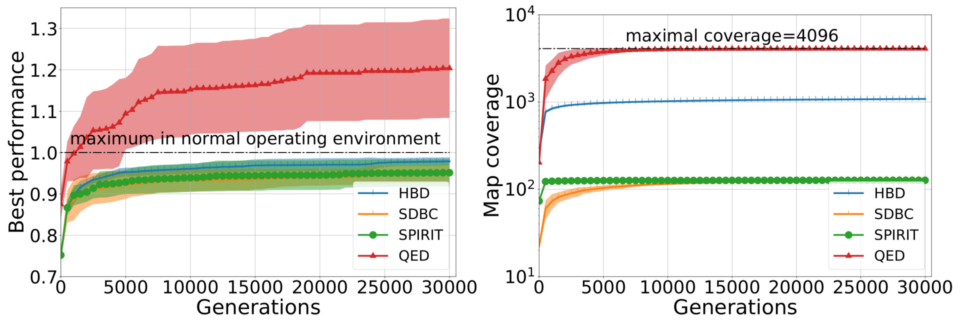
A. Map quality analysis

Evolutionary experiments are repeated in five independent replicates. Comparing QED to baseline algorithms on all 5 tasks yields a total of 4 algorithms $\times$ 5 tasks $\times$ 5 replicates $= 100$ evolutionary runs. The performance awarded to the evaluated robot controller is the average fitness across 50 independent trials. The quality-diversity algorithms are evolved over 30,000 generations to ensure at least a weak convergence in the performance and coverage of the generated maps (see Fig. 2a-b, and Fig. S1 in Supplemental Materials), while respecting the available computation budget for our experiments[2].

Performance: During evolution, the maps generated by the baseline algorithms (HBD, SDBC, SPIRIT) do not vary widely in their performance (see Fig. 2a), and while QED achieved the highest performance, a cautious interpretation is required because solutions in QED's environment-performance maps are evolved in environments with varying levels of difficulty; that is, some QED environments (e.g., those with smaller arenas in the patrolling task) allow a higher performance than can be obtained in the normal operating environment.

After re-evaluation in the normal operating environment (see method in Section III-D), all the algorithms evolve high-performing solutions for aggregation, patrolling and border-patrolling tasks, differing no more than 3% of the empirical maximum performance (see Table S2 in Supplemental Materials). By contrast, in the flocking task, the performance varies widely between 0.25-0.85 for QED, 0.70-0.98 for SDBC, and 0.66-1.0 for SPIRIT, with the exception of HBD which consistently evolved high performing solutions between 0.97-0.99. The performance was less divergent in the dispersion task, with the baseline algorithms and QED achieving a performance in 0.93-1.0, and 0.81-0.87 respectively. A summary of the data across all tasks (see Fig. 3a) suggests that the QED algorithm may sacrifice the performance of its solutions in the normal operating environment for an improved performance on the perturbed environments in which they were evolved; that is, since all algorithms have the same budget of function evaluations, QED allocates on average 1 out of 4096 function

[2]A single replicate required about 300-700h (resp. 900h) of computational time on a 40-cores Intel Xeon Gold 6138 at 2GHz, for baselines (resp. QED).



(a) Performance time-line (b) Coverage time-line

Fig. 2: Evolutionary analysis: Mean±SD of best performance (a) and map coverage (b) evolution over 30,000 generations, aggregated across 25 independently evolved maps (5 independent replicates for each of 5 swarm robotic tasks). Performance is the average fitness across 50 trials, normalised by the empirical maximum for the normal operating environment, the environment in which baseline algorithms evolve solutions – in QED, this is only one of its many environments.

evaluations to the normal operating environment and 4095 out of 4096 function evaluations to perturbed environments.

Behavioural diversity: As an approximate indicator of behavioural diversity in the algorithms' own behaviour spaces, we compute the coverage (see Fig. 2b), the number of cells in an algorithm's own behaviour space that are filled with a solution. The coverage of maps generated by QED, HBD, SDBC and SPIRIT is 4094 ± 2, 1091 ± 19, 130 ± 5, and 127 ± 1, respectively (Mean $\pm$ SD across all swarm robotic tasks and replicates). With a maximal coverage of 4096 for all algorithms, this finding indicates that the behaviour spaces of the baseline algorithms are sparse, i.e., not all cells represent feasible robot swarm behaviours, whereas the behaviour space of QED is dense, with all cells representing feasible robot swarm behaviours. The high QED coverage is explained by its property of controllable coverage; that is, QED's environment generator can be set to reliably generate all environments within its environment space. These findings are not sufficient, however, to demonstrate that QED is behaviourally diverse, since the QED space is based on *environmental diversity* rather than behavioural diversity.

To compare the behavioural diversity of maps generated by the QED, HBD, SDBC and SPIRIT algorithms, we project their solutions into a common generic behaviour space (see Section III-D for details). All the maps generated have a similar averaged pair-wise distance between solutions, ranging between $0.60 - 0.62$, in the projected space. The coverage in the projected behaviour-performance maps differs more widely (see Fig. 3b), with the QED achieving coverage of 49 ± 20 (Mean $\pm$ SD across all swarm robotic tasks and replicates), at least on par with the baseline algorithms – HBD (55 ± 10), SDBC (26 ± 2), and SPIRIT (51 ± 11) – which are based on behavioural descriptors. In sum, as hypothesised, QED's environment descriptor serves as an implicit behavioural descriptor.

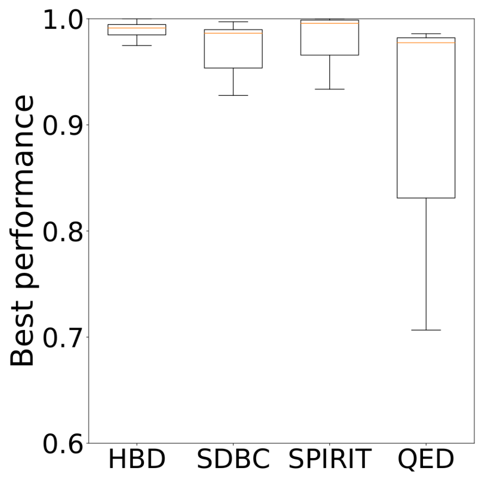
B. Fault recovery analysis

A traditional analysis of behavioural diversity is limited in the sense that behavioural diversity in itself is not a sufficient condition for fault recovery or adaptation; there may be behaviours which contribute to diversity but not to fault recovery. Similarly, a high performance in the normal operating environment does not necessarily generalise towards environments not seen during evolution. Therefore, we analyse behavioural diversity and performance within the context of robot swarm fault recovery.

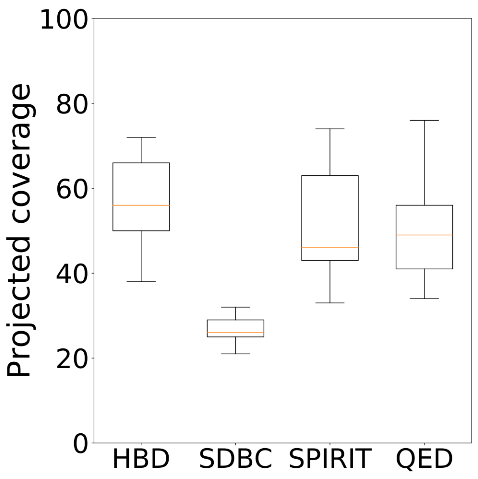
Fault injections: When a robot swarm is operating in its normal operating environment, any fault to a single robot's sensors or actuators effectively changes the operating environment. Because different robots in the same swarm may have different faults, the space of possible environments is greatly expanded. To sample a part of this space, the fault injection scheme determines for each robot in the swarm independently a randomly chosen fault before any trials are started. The fault is chosen from the following 8 fault types previously used in studies on fault-detection in robot swarms [17], [54], which are applied at each control cycle with random variables being sampled anew:

- **PMIN**: the front proximity sensor readings are set to the minimum (0);
- **PMAX**: the front proximity sensor readings are set to the maximum (1);
- **PRAND**: the front proximity sensor readings are generated randomly in the range $[0, 1]$;
- **LW-H**: the speed of the robot's left wheel is halved;
- **RW-H**: the speed of the robot's right wheel is halved;
- **BW-H**: the speed of both wheels is halved;
- **ROFS**: a large offset vector (r, θ) is added to range-and-bearing readings, with $r \sim U(0.75, 1.0)$ and $\theta \sim U(-180°, 180°)$;
- **NONE**: no fault is applied.

Importantly, none of the resulting environments are seen during the evolutionary phase by any of the algorithms and a particular challenge is that different robots experience different faults. In our experiments, a total of 250 unique combined faults are sampled and applied for all 5 swarm robotic tasks. Due to the large number of faulty environments and the large number of solutions in the evolved maps, the number of trials for the fitness evaluation is limited to 10.



(a) Re-evaluated performance



(b) Projected coverage

Fig. 3: Post-evolutionary analysis: all solutions in the maps are re-evaluated in the normal operating environment for 10 independent trials, resulting in 25 data points (5 replicates on 5 tasks). (a) boxplot of the best performance across maps; and (b) coverage of the solutions when projected in the SPIRIT behaviour space[3].

Fault recovery: In our experiments, fault recovery searches the map for the best solution to the faulty environment, i.e., the recovery solution. To assess map quality in the context of fault recovery, we compare the best performance in the faulty and normal operating environments and assess the useful diversity, the behavioural diversity across the recovery solutions.

The analysis of the fault recovery includes a total of three metrics: i) the impact, the proportional change in performance after transferring the best solution of the normal operating environment to the faulty environment; ii) the recovered performance, the performance of the best solution to the faulty environment; and iii) the resilience $\mathscr{R}$, the proportional change in performance comparing the best solution for the faulty environment $\mathcal{E}_{\mathscr{F}}$ to the best solution for the normal operating environment $\mathcal{E}$:

$$\mathscr{R}(\mathcal{M}, \mathcal{E}_{\mathscr{F}}|\mathcal{E}) = \frac{\max_{c \in \mathcal{M}} f(\mathcal{E}_{\mathscr{F}}, c) - \max_{c' \in \mathcal{M}} f(\mathcal{E}, c')}{\max_{c' \in \mathcal{M}} f(\mathcal{E}, c')}, \quad (1)$$

where $\mathcal{M}$ is the map. A summary of these metrics, using the median and inter-quartile range across all fault injections, demonstrates the viability of our approach to fault recovery (see Table II; for task-specific fault-recovery statistics and statistical significance, see Table S3 in Supplemental Materials). All quality-diversity algorithms have a median recovered performance between 86-88% of the empirical maximum performance. Despite the median impact of the fault being 23-25% of the original performance in the normal operating environment, the median resilience score indicates that after fault recovery, the drop in performance is reduced to 8-12%. These results also demonstrate differences between the algorithms: QED has the highest score (Median $\pm$ IQR) for impact of the fault (-0.23 ± 0.26), recovered performance (0.88 ± 0.16), and resilience (-0.08 ± 0.07), whereas SPIRIT has the lowest score for impact of the fault (-0.25 ± 0.29), recovered performance (0.86 ± 0.13), and resilience (-0.12 ± 0.13). Significance scores indicate that the magnitude of QED's positive resilience effect compared to other conditions is medium to large, with few exceptions. Explaining its high IQR, the impact of the fault depends strongly on the type of task, ranging between 0-20% for aggregation, 10-40% for dispersion, 10-50% for patrolling, 0-60% for border-patrolling, and 60-100% for flocking. Interestingly, our data indicate no direct relation between fault type and the impact of the fault.

To predict an algorithm's fault recovery performance in the face of unknown faults, we visualise its resilience as a function of the impact of the fault, giving each algorithm its unique *impact-resilience signature* (see Fig. 4a)[4]. This analysis demonstrates that QED's comparative resilience advantage derives not only from having lower impacts, typically giving only a 10% drop in performance, but also from being more resilient to high-impact faults; although all algorithms have a positive correlation between impact and resilience (0.4-0.6), QED has a smaller slope ($a = 0.17$) compared to other algorithms ($a = 0.19$ for HBD, $a = 0.19$ for SDBC, and $a = .30$ for SPIRIT) when considering a linear regression model of resilience as a function of impact.

TABLE II: Median $\pm$ IQR of the impact of faults, the recovered performance, and the resilience, aggregated across all 1250 injected faults for 5 swarm robotic tasks in 10 independent trials.

	Impact of fault	Recovered performance	Resilience
HBD	-0.25 ± 0.33	0.87 ± 0.11	-0.12 ± 0.11
SDBC	-0.24 ± 0.26	0.87 ± 0.10	-0.11 ± 0.08
SPIRIT	-0.25 ± 0.29	0.86 ± 0.13	-0.12 ± 0.13
QED	-0.23 ± 0.26	0.88 ± 0.16	-0.08 ± 0.07

[3]Upper (resp. lower) edges of the box mark the first (resp. third) quartile of the data, the red line indicates the median, and whiskers extend to the highest (resp. lowest point) within a distance of 1.5 times the IQR from the quartiles.

[4]To better visualise the majority of the data at a fine resolution and allow meaningful estimates of the slope that are not affected by extreme observations, the observations with extreme impacts (i.e., those with impacts larger than 50%) are removed from the analysis of Fig. 4, resulting in around 1000 data points for each algorithm.

To explain its improved resilience to high-impact faults, our hypothesis is that QED overcomes high-impact faults by finding high-performing fault recovery behaviours that are especially different from the *normal behaviour*, i.e., the behavioural descriptor obtained from the best-performing solution to the normal operating environment. To assess this interpretation, the resilience and the behavioural diversity of the fault recovery solutions are visualised in Fig. 4b. Confirming its resilience property, QED has typical values around -0.07 with most of its probability mass in $[-0.10, 0]$ while other algorithms have a resilience centred around -0.10 with most of the probability mass in $[-0.20, 0]$. Importantly, QED exhausts nearly the entire spectrum of distances around the normal behaviour, while other algorithms have a maximal distance of 0.7-0.8. As a signature in contrast to QED, SPIRIT has both low diversity and resilience. Combining the diversity data with the above-mentioned slope data, where resilience degrades slowly for QED but swiftly for SPIRIT as the fault impact grows larger, provides a first supporting argument for our hypothesis.

To further demonstrate that the solutions that are highly differing from the normal behaviour are frequently used for recovery from high impact faults, we visualise behavioural diversity as a function of the impact of the fault. This analysis (see Fig. 4c) confirms that faults with a large, negative impact on performance are more likely to yield solutions with a higher behavioural distance to the normal behaviour. The visualisation is supported by the negative correlations between impact of the fault and behavioural diversity and a large negative slope for all algorithms ($a = -0.66$ for HBD, $a = -1.01$ for SDBC, $a = -0.36$ for SPIRIT, and $a = -0.98$ for QED) when considering a linear regression model of behavioural diversity as a function of fault. Another observation is that, although baseline algorithms (HBD, SDBC, and SPIRIT) have larger fault impacts, some of which exceed a drop of 40% of the original performance, they only rarely find solutions at high behavioural distances ($d > 0.70$) to the normal behaviour, whereas QED does so frequently for faults with an impact greater than 20%. These findings confirm our hypothesis because for high-impact faults, QED more frequently finds solutions differing strongly from the normal behaviour. Large behavioural differences to the highest-performing behaviour in the normal operating environment are expected because QED allows solutions with low performance on the normal operating environment to reproduce and direct evolution.

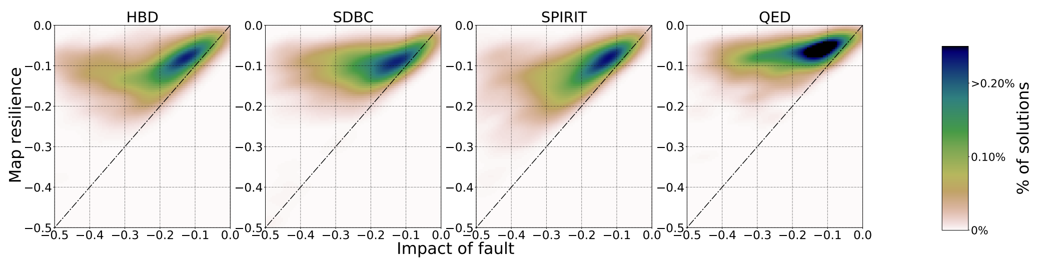
As a visual demonstration, the above-mentioned relation between high-impact faults, resilience, and behavioural diversity around the normal behaviour is supported by video footage of the border-patrolling task (see https://youtu.be/BN6i-NugCGg). In this footage, all algorithms (HBD, SDBC, SPIRIT, and QED) can reduce the impact of faults to the front proximity sensors by using a controller which drives backwards and relies on the rear proximity sensors. Another common observation for all algorithms is that the robots with one faulty wheel make circular movements at a fixed position. However, in the baseline algorithms (HBD, SDBC, and SPIRIT), such robots interfere with the entire swarm, resulting in a traffic congestion, while in QED these robots are isolated and other robots in the swarm stay close to the walls and avoid collision. We believe this is due to the environment in which the solution was evolved: with a larger number of robots each having a higher speed, the robots in the swarm came in frequent contact with each other, and therefore avoiding other robots was important to obtain a high performance. This footage is in line with the above-mentioned findings: despite the high impact of the fault, QED drops a mere 2% in performance, using a recovery solution which has a behavioural distance of 0.8 to its normal behaviour, while other algorithms drop 7-8% and have a distance between 0.6-0.7 to their normal behaviour.

As a final analysis of the results, we investigate which environment attribute values are most important for QED's fault recovery by analysing the environments in which the fault recovery solutions were evolved. A first part of the analysis investigates a frequency-based metric (see Table S4a in Supplemental Materials), which attribute values are most frequently observed in fault recovery solutions, while a second part of the analysis investigates a resilience-based metric (see Table S4b in Supplemental Materials), which attribute values correspond to the highest resilience scores. Analysis of the frequency statistic demonstrates that all environment attribute values are frequently represented in fault recovery solutions. Attribute values corresponding to the normal operating environment are not observed to be the most frequent for any of the attributes, indicating that the attributes indeed are useful to vary and therefore well-chosen. Although for each attribute, its frequencies are close to uniform, there are the following notable effects: i) the larger the range-and-bearing sensor range of the environment, the more fault recovery solutions, with a frequency of 34.2% for the $2\,\mathrm{m}$ setting, as opposed to a frequency of only 15.0% for the $0.5\,\mathrm{m}$ setting; ii) a lowered arena size, particularly the $3 \times 3\,\mathrm{m}^2$ setting, leads to a higher proportion of fault recovery solutions, 29.5%; iii) low maximal linear velocities have a higher frequency, with the minimal value of $5\,\mathrm{cm/s}$ being represented in 30.7% of the fault recovery solutions and the maximal value of $20\,\mathrm{cm/s}$ being represented in 21% of the fault recovery solutions; and iv) the setting of 4 obstacles is represented in 29.4% of the fault recovery solutions. Computing the median resilience across the recovery solutions with perturbed environment attributes, no clear difference is observed, with scores ranging between -0.080 and -0.077. In sum, all environment attributes appear to be important to vary, although fault recovery solutions are more frequently found for cluttered settings where agents must frequently interact with objects and each other.

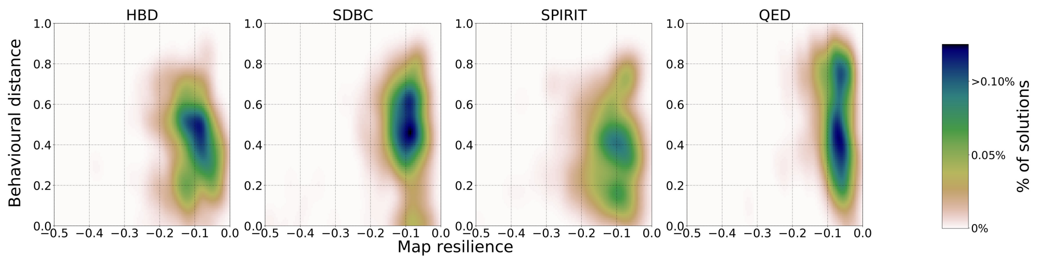
V. Discussion

This paper investigates the use of quality-diversity algorithms in a model-free approach to fault recovery in swarm robotic systems, where different robots in a swarm are affected by different faults. Our novel Quality-Environment-Diversity (QED) framework describes the behaviour of individual solutions implicitly based on the environment in which they are evaluated.

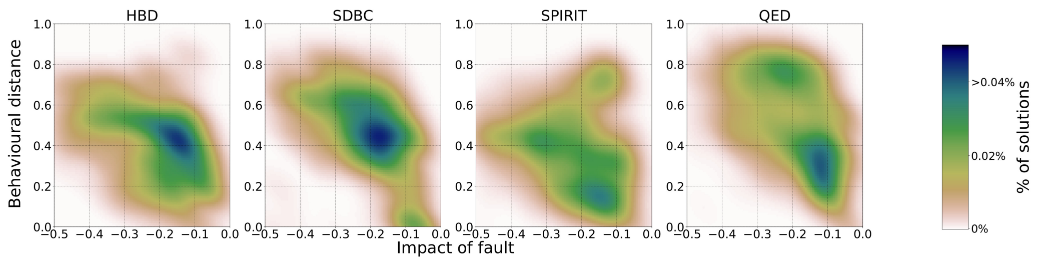
[5]The proportion estimates are based on a Gaussian kernel density estimation with 100-by-100 grid size and bandwidth according to Scott's rule, $n^{-1/(d+4)}$, where n is the number of data points and $d = 2$ is the dimensionality.



(a) Impact-resilience signature



(b) Diversity-resilience signature



(c) Diversity-impact signature

Fig. 4: Visualisation of fault recovery based on impact of the fault, resilience, and useful behavioural diversity of fault recovery solutions. Useful behavioural diversity is based on the behavioural distances of fault recovery solutions to the normal behaviour and is computed in the projected SPIRIT behaviour space. (a) **Impact-resilience signature**, the joint probability of resilience (y-axis) and impact of the fault (x-axis), with the dashed line indicating the identity line – ideally, most of the probability is located in top-right, where faults do not affect the swarm's performance, and resilience is high across the impact spectrum, indicating recovery for high-impact faults; (b) **Diversity-resilience signature**, the joint probability of behavioural distance to the normal behaviour (y-axis) and resilience (x-axis) – an ideal fault recovery profile with useful diversity has high resilience and covers the entire distance spectrum; and (c) **Diversity-impact signature**, the joint probability of behavioural distance to the normal behaviour (y-axis) and impact of the fault (x-axis) – the correlation between impact and behavioural diversity indicates the need for useful diversity to recover from high-impact faults[5].

With QED, the designer can easily formulate dimensions of interest to evolve archives with useful behavioural diversity tailored to a wide variety of plausible environmental challenges. In our study, QED is compared to different baseline-algorithms, including a domain-specific, hand-coded behavioural descriptor, as well as two generic behavioural descriptors, SDBC and SPIRIT, on five different swarm robotics benchmark tasks, with and without injection of 250 unique combinations of faults. Overall, the study demonstrates the use of quality-diversity algorithms as a means for fault recovery in swarm robotics; the quality-diversity algorithms are typically able to reduce the loss of performance from 23-25% immediately after fault injection to 8-12% after fault recovery with exhaustive search through the archive of solutions. Due to the environmental diversity during its evolution, QED confers benefits to generalisation. *On the map-level*, the environmental diversity leads to a significantly higher resilience due to the useful behavioural diversity profile of QED, in which solutions that are especially different from the normal behaviour help to recover from high-impact faults. *On the solution-level*, a minor benefit of QED is observed for the impact of the fault. Since its offspring solutions are genetically similar to their parents but may be evolved in highly differing environments, QED may increasingly select for solutions that generalise across environments.

The results of our investigation provide us with several recommendations. Based on the large resilience advantage and the favourable diversity features of QED, environment diversity is an important component to be considered when evolving a diversity of solutions to a given optimisation problem; this recommendation likely extends beyond the use case of swarm robotics and fault recovery. Analogous to the bias-variance dilemma in supervised learning settings, where generalisation is often improved by presenting data from the target distribution but the fit to the data may be reduced, QED provides an improved map-level generalisation but at the cost of reducing performance in the normal operating environment; that is, map-resilience is improved but a trade-off to consider is that performance in the normal operating environment may in some cases be significantly reduced by using QED. In fact, in the dispersion task, QED's resilience property was not sufficient to overcome its low performance in the normal operating environment. Our study also explores a hand-coded behavioural descriptor, HBD, which exploits the prior knowledge of the arena used in all the tasks. Despite its comparatively high fault recovery performance in aggregation, dispersion, and flocking, HBD has a comparatively low fault recovery performance in patrolling and border-patrolling. This may be explained by the high alignment [24] of the HBD descriptor with the fitness function of the patrolling and border-patrolling tasks: each behaviour is limited to a narrow range of fitness and neighbouring regions of behaviour space have a correlated upper limit to fitness. In patrolling, HBD only gives high-performing solutions that have a large number of unique cells visited. In border-patrolling, a large number of unique cells visited combined with high uniformity of visitations amongst those cells visited cannot lead to high-performing solutions. Therefore, for fault recovery in swarm robotics, we cautiously advise against the use of behavioural descriptors that are 'aligned' with the fitness function, contrasting to results in deceptive maze problems [55], [56]. Further, our study finds that, in the flocking task, fault recovery was not satisfactory for any of the quality-diversity algorithms, with a loss of 60-80% of the original performance levels. Based on this observation, the recommendation is that, when applying our model-free approach to fault recovery in tightly-coordinated tasks involving coordinated movement of robots [7], [8], [9], each robot should use its own unique controller; this would enable fault recovery in case different robots are affected by different faults. For example, when one robot is affected by a fault on its right actuator, and another robot is affected by a fault on its left actuator, these two robots can only move coordinately when they have a different controller. However, using a unique controller for each robot results in at least two challenges: i) there is a linear increase in the number of dimensions in the genotypic space and therefore an exponential increase in the search space; and ii) for descriptors based on the policy of a single robot, such as the SPIRIT descriptor, this also results in an exponential increase in the number of behaviours, unless the resolution of the behaviour-performance map is significantly reduced.

Only limited work has exploited environment diversity to generate behaviourally diverse archives of controllers, with no work so far in swarm robotics or fault recovery; QED provides an alternative to existing approaches and is characterised by applicability, scalability, and user-control. One approach that uses environment diversity within quality-diversity algorithms is the Innovation Engines [43], which automatically extracts features from behaviour but scores the fitness dependent on the behaviour characterisation, and in this sense the task objective and the behavioural description are mutually dependent. While Innovation Engines has been applied to generate images, this method may not be applicable to evolutionary robotics, where the performance of a robot must be assessed empirically in expensive simulations rather than through a neural network classification. Although not explicitly mentioned as a quality-diversity algorithm, the Combinatorial Multi-Objective Evolutionary Algorithm [44] stores a multitude of elite solutions for different task-subtask decompositions and evaluates new offspring on each of the decompositions. In comparison, the QED framework is not limited to combinatorial tasks, and, due to the use of the environment generator, may be able to apply evolution on a wider diversity of environments. Finally, the Paired Open-Ended Trailblazer (POET) [41] generates an ever-increasing diversity of environments within a certain difficulty level, and pairs each environment with a single solution that is optimised on this environment based on local rather than population-based optimisation. With its emphasis on challenging environments, empirical results demonstrate solutions to increasingly challenging single-robot control problems. POET may not be applicable to expensive cost functions due to the need to evaluate all individuals on all environments (to avoid local minima on existing environments and to initialise the best solution to a new environment); in swarm robotics, this may be problematic because high-fidelity simulations are required with many trials, long evaluation times and many robots. In comparison to POET, QED provides an increased user-control since environments are generated by a user-defined probability distribution, rather than by incremental mutations on existing environments. Further, QED may be preferable when the space of environments must be exhausted since POET rejects environments if they lead to either too high or too low fitness scores. In fault recovery, this may imply that POET rejects environments where robots are affected by faults with a strong impact on performance.

A point of discussion is to what extent the fault recovery results in simulation environments can be extrapolated to real-world swarm robotic systems. While the present study assumes that taking the best-performing controller in the archive comes at no cost, a real-world swarm robotics fault recovery system must select a compensatory behaviour in a limited number of evaluations. However, as demonstrated by Cully et al. (2015) in a single-robot study, Bayesian Optimisation represents one approach to search efficiently for the best solution in an evolved archive [23], [57]. This approach may potentially work in robot swarms as well, and in this case, QED may provide an alternative to existing behavioural descriptors to improve fault recovery. As a potential alternative to Bayesian Optimisation, with no restrictions on the map's geometry and no need for expensive trial and error, the topologically organised environment-performance map used in QED with MAP-Elites may be exploited by first detecting the environment and then taking the corresponding controller in the map. Additionally, QED may improve robustness to the simulation-reality gap, because systems that are robust to a transfer from one simulation environment to another may also be robust to a simulation-to-reality transfer [58], and environment diversity has been used previously in the control of a robotic arm to improve the simulation-reality gap [37].

Our fault recovery study also presents a novel method to evaluate the quality of archives developed by quality diversity algorithms. Traditionally, archives evolved by quality-diversity algorithms are evaluated based on behavioural diversity and performance metrics obtained from evolution [26], [25]. However, because the designer cannot predict the types of conditions that may arise in real-world application, this analysis is limited. To predict how well an archive will fare in adverse conditions, where faults strongly impact the performance of the swarm, our study has considered a novel model selection tool, called the impact-resilience signature. The signature provides a unique profile for a quality-diversity algorithm by visualising how strongly performance is expected to degrade in the face of high-impact faults. A similar visual analysis of diversity as a function of impact and resilience illustrates QED's unique profile, which increases resilience by finding behaviours that differ strongly from the normal behaviour.

VI. Conclusion

We investigate fault recovery in swarm robotics, where each robot in a swarm may be affected by different faults. Our approach is to evolve a behaviourally diverse archive of behaviours using quality-diversity algorithms. We formulate a novel quality-diversity algorithm, Quality-Environment-Diversity (QED), which selects an environment at random in which the solution is to be evaluated and then uses an environment descriptor to characterise behaviour implicitly based on this environment. This implicit behavioural characterisation makes it easy to design behaviour spaces tailored to overcome a wide variety of plausible environmental challenges. Our extensive fault recovery study assesses the fault recovery of QED, with its environment descriptor, and traditional quality-diversity algorithms, each with their own behavioural descriptor, on 5 swarm robotic tasks and 250 unique fault conditions per task. Results demonstrate a successful fault recovery for all quality-diversity algorithms and the unique profile of QED, with a high robustness to faults and a high behavioural diversity in its fault recovery solutions. In future work, the QED framework may be implemented with alternative environment generators or other quality-diversity algorithms such as Novelty Search with Local Competition [59], and fault recovery performance may be targeted directly by an adaptive selection of environments during evolution.

VII. Acknowledgements

This work has been supported by the Engineering and Physical Sciences Research Council (EPSRC) under the New Investigator Award grant (EP/R030073/1). The authors thank Mohammad Divband Soorati and Jean-Baptiste Mouret for comments on a draft of this paper. The authors further acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

References

[1] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Intelligence. Natural Computing Series.*, E. Sahin and W. Spears, Eds. Springer, Berlin, Heidelberg, 2005, vol. 3342, pp. 10–20.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[3] L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[4] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," *Robotics: Science and Systems*, vol. 2, pp. 49–56, 2007.

[5] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira, and A. L. Christensen, "Evolution of collective behaviors for a real swarm of aquatic surface robots," *PLoS ONE*, vol. 11, no. 3, pp. 1–25, 2016.

[6] L. Garattoni and M. Birattari, "Autonomous task sequencing in a robot swarm," *Science Robotics*, vol. 3, no. 20, pp. 1–12, 2018.

[7] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, "Flocking algorithm for autonomous flying robots," *Bioinspiration and Biomimetics*, vol. 9, no. 2, 2014.

[8] R. Groß and M. Dorigo, "Towards group transport by swarms of robots," *International Journal of Bio-Inspired Computation*, vol. 1, no. 1-2, pp. 1–13, 2009.

[9] A. F. Llenas, M. S. Talamali, X. Xu, J. A. R. Marshall, and A. Reina, "Quality-Sensitive Foraging by a Robot Swarm Through Virtual Pheromone Trails," in *International Conference on Swarm Intelligence*. Springer, 2018, pp. 135–149.

[10] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems.* New York, NY: Oxford University Press, 1999.

[11] J. D. Bjerknes and A. F. T. Winfield, "On Fault Tolerance and Scalability of Swarm Robotic Systems," in *Distributed Autonomous Robotic Systems, STAR*, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Stoy, Eds. Springer-Verlag Berlin Heidelberg, 2013, pp. 431–443.

[12] A. F. Winfield and J. Nembrini, "Safety in numbers: Fault-tolerance in robot swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[13] A. L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo, "Fault detection in autonomous robots based on fault injection and learning," *Autonomous Robots*, vol. 24, no. 1, pp. 49–67, 2008.

[14] D. Tarapore, P. U. Lima, J. Carneiro, and A. L. Christensen, "To err is robotic, to tolerate immunological: fault detection in multirobot systems," *Bioinspiration & Biomimetics*, vol. 10, no. 1, p. 016014, 2015.

[15] A. G. Millard, "Exogenous fault detection in swarm robotic systems: an approach based on internal models," PhD Thesis, University of York, 2016.

[16] A. L. Christensen, R. O. Grady, and M. Dorigo, "From Fireflies to Fault-Tolerant Swarms of Robots," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 754–766, 2009.

[17] D. Tarapore, A. L. Christensen, and J. Timmis, "Generic , scalable and decentralized fault detection for robot swarms," *PLoS ONE*, vol. 12, no. 8, pp. 1–29, 2017.

[18] G. Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, "The grand challenges of Science Robotics," *Science Robotics*, vol. 3, no. 14, 2018.

[19] K. Chatzilygeroudis and V. Vassiliades, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *arXiv preprint arXiv:1807.02303v4*, pp. 1–21, 2019.

[20] J. Bongard, V. Zykov, and H. Lipson, "Resilient Machines Through Continuous Self-Modeling," *Science*, vol. 314, no. November, 2006.

[21] S. Koos, A. Cully, and J. B. Mouret, "Fast damage recovery in robotics with the T-resilience algorithm," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1700–1723, 2013.

[22] K. Chatzilygeroudis, V. Vassiliades, and J. B. Mouret, "Reset-free Trial-and-Error Learning for Robot Damage Recovery," *Robotics and Autonomous Systems*, vol. 100, pp. 236–250, 2018.

[23] A. Cully, J. Clune, D. Tarapore, and J. B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[24] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality Diversity: A New Frontier for Evolutionary Computation," *Frontiers in Robotics and AI*, vol. 3, no. July, pp. 1–17, 2016.

[25] A. Cully and Y. Demiris, "Quality and Diversity Optimization: A Unifying Modular Framework," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 245–259, 2018.

[26] J.-b. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909v1*, pp. 1–15, 2015.

[27] J. K. Pugh, L. B. Soros, P. A. Szerlip, and K. O. Stanley, "Confronting the Challenge of Quality Diversity," in *GECCO '15: Genetic and Evolutionary Computation Conference*. New York, NY, USA: ACM, 2015.

[28] J. Nordmoen, K. O. Ellefsen, and K. Glette, "Combining MAP-Elites and Incremental Evolution to Generate Gaits for a Mammalian Quadruped Robot," in *21st International Conference, EvoApplications 2018*, K. Sim and P. Kaufmann, Eds. Parma, Italy: Springer International Publishing, 2018, pp. 159–170.

[29] S. Engebråten, O. Yakimenko, J. Moen, and K. Glette, "Towards a Multi-Function Swarm That Adapts to User Preferences," in *International Conference on Robotics and Automation (ICRA 2018)*, 2018.

[30] E. Hart, A. S. Steyven, and B. Paechter, "Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm," in *Proceedings of the 2018 Genetic and Evolutionary Computation Conference (GECCO 2018)*, Kyoto, Japan, 2018, pp. 101–108.

[31] E. Meyerson, J. Lehman, and R. Miikkulainen, "Learning Behavior Characterizations for Novelty Search," in *GECCO '16: Genetic and Evolutionary Computation Conference*, 2016, pp. 149–156.

[32] F. J. Gomez, "Sustaining diversity using behavioral information distance," in *GECCO '09: Genetic and Evolutionary Computation Conference*, Montréal Québec, Canada, 2009, pp. 113–120.

[33] J. Gomes, P. Mariano, and A. Christensen, "Systematic Derivation of Behaviour Characterisations in Evolutionary Robotics," in *ALIFE 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, 2014, pp. 212–219.

[34] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias variance dilemma," *Neural Computation*, vol. 4, pp. 1–58, 1992.

[35] A. Thompson, P. Layzell, and R. S. Zebulum, "Explorations in design space: Unconventional electronics design through artificial evolution," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 3, pp. 167–195, 1999.

[36] S. Koos, J. B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 122–145, 2013.

[37] Ilge Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's Cube with a Robot Hand," *arXiv preprint*, pp. 1–51, 2019.

[38] T. Pinville, S. Koos, J. B. Mouret, and S. Doncieux, "How to promote generalisation in evolutionary robotics: The ProGAb approach," *Genetic and Evolutionary Computation Conference, GECCO'11*, pp. 259–266, 2011.

[39] J. H. Schmidhuber, "PowerPlay: Training an Increasingly General Problem Solver by Continually Searching for the Simplest Still Unsolvable Problem." *Frontiers in psychology*, vol. 4, no. June, p. 313, 2013.

[40] R. K. Srivastava, B. R. Steunebrink, M. Stollenga, and J. Schmidhuber, "Continually adding self-invented problems to the repertoire: First experiments with POWERPLAY," *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL 2012*, 2012.

[41] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, "Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions," *arXiv preprint*, pp. 1–28, 2019. [Online]. Available: http://arxiv.org/abs/1901.01753

[42] J. C. Brant and K. O. Stanley, "Minimal criterion coevolution: A new approach to open-ended search," *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, no. Gecco, pp. 67–74, 2017.

[43] A. Nguyen, J. Yosinski, and J. Clune, "Innovation engines: Automated creativity and improved stochastic optimization via deep learning," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015).*, 2015, pp. 545–572.

[44] J. Huizinga and J. Clune, "Evolving Multimodal Robot Behavior via Many Stepping Stones with the Combinatorial Multi-Objective Evolutionary Algorithm," *arXiv preprint*, pp. 1–21, 2018. [Online]. Available: http://arxiv.org/abs/1807.03392

[45] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[46] V. Vassiliades and J.-b. Mouret, "Discovering the Elite Hypervolume by Leveraging Interspecies Correlation," in *Proceedings of the 2018 Genetic and Evolutionary Computation Conference (GECCO 2018)*, Kyoto, Japan, 2018, pp. 623–630. [Online]. Available: http://arxiv.org/abs/1804.03906

[47] D. Tarapore, J. Clune, A. Cully, and J.-b. Mouret, "How Do Different Encodings Influence the Performance of the MAP-Elites Algorithm ?" in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference (GECCO '16)*, Denver, CO, USA, 2016, pp. 173–180.

[48] N. Justesen, S. Risi, and J.-B. Mouret, "MAP-Elites for Noisy Domains by Adaptive Sampling," in *Proceeding of the 2019 Genetic and Evolutionary Computation Conference (GECCO '19)*, 2019, pp. 121–122.

[49] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[50] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada, "Thymio II, a robot that grows wiser with children," *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, pp. 187–193, 2013.

[51] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. Yue, "Swarm-enabling technology for multi-robot systems," *Frontiers in Robotics and AI*, vol. 4, p. 12, 2017.

[52] J. Gomes and A. L. Christensen, "Task-Agnostic Evolution of Diverse Repertoires of Swarm Behaviours," in *11th conference on Swarm Intelligence*. Springer International Publishing, 2018, pp. 225–238.

[53] V. Vassiliades, K. Chatzilygeroudis, and J. B. Mouret, "Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 623–630, 2018.

[54] D. Tarapore, J. Timmis, and A. L. Christensen, "Fault Detection in a Swarm of Physical Robots Based on Behavioral Outlier Detection," *IEEE Transactions on Robotics*, pp. 1–7, 2019.

[55] S. Kistemaker and S. Whiteson, "Critical factors in the performance of novelty search," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, 2011, pp. 965–972.

[56] J. K. Pugh, L. Soros, and K. O. Stanley, "Searching for Quality Diversity When Diversity is Unaligned with Quality," in *PPSN 2016: Parallel Problem Solving from Nature – PPSN XIV*, 2016, pp. 880–889.

[57] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[58] A. Ligot and M. Birattari, "Simulation-only experiments to mimic the effects of the reality gap in the automatic design of robot swarms," *Swarm Intelligence*, 2019.

[59] J. Lehman and K. O. Stanley, "Evolving a Diversity of Creatures through Novelty Search and Local Competition," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*. Dublin, Ireland: ACM, New York, 2011, p. 1408.