Evolving Morphologically Robust Robot Controllers

***Abstract*—This study evaluates objective versus non-objective-based evolutionary search methods for behavior evolution in robot teams. The goal is to evaluate the *morphological robustness* of evolved controllers, where controllers are evolved for specific robot sensory-motor configurations (morphologies) but must continue to function as these morphologies degrade. Robots use artificial neural network controllers where behavior evolution is directed by developmental neuro-evolution. Guiding evolutionary controller design we use objective (*fitness function*) versus non-objective (*novelty*) search. The former optimizes for behavioral fitness and the latter for behavioral novelty. These methods are evaluated across varying robot morphologies and increasing task complexity. Results indicate that novelty search yields no benefits over objective search, in terms of evolving *morphologically robust* controllers. That is, both novelty and objective search evolve team controllers that are morphologically robust given varying robot morphologies and increasing task complexity. Results thus suggest behavioral diversity methods such as novelty search mat not be suitable for generating robot behaviors that can continue functioning given changing robot morphologies, for example, due to damaged or disabled sensors and actuators.**

I. Introduction

Autonomous robots are increasingly being applied to remote and hazardous environments [1], [2], where in such environments, damage to sensory-actuator systems (*morphologies* [3]) cannot be easily repaired if damaged. An unsolved problem in the controller design for such autonomous robots is having controllers continue to effectively function given unexpected changes, such as damage, to robot morphology.

Currently, robotic systems recover from damage via self-diagnosis and selection from pre-designed contingency plans in order to continue functioning [4], [5], [6]. However, robots using such self-diagnosis and recovery systems are problematic as such systems are expensive, require sophisticated monitoring sensors and are difficult to design since system designers must have *a priori* knowledge of all necessary contingency plans [7].

Addressing this, recent work in *Evolutionary Robotics* (ER) [8] elucidated the efficacy of population based stochastic *trial and error* methods for *online* damage recovery in autonomous robots operating in physical environments [7]. This was demonstrated as being akin to self-adaptation and injury recovery of animals observed in nature.

This study further contributes to this research area, focusing on *evolutionary controller design* [9] within the broader context of *collective* [10] and *swarm* [2] robotics. That is, evolutionary controller design for robot groups that must continue to accomplish tasks given that damage is sustained to the morphologies of some or all of the robots in the group.

Recent work in evolutionary controller design for ER systems indicated that maintaining *phenotypic* (behavioral) diversity in populations of potential solutions improves the quality (task performance) of evolved robot behaviors [11], [12], [13], [7]. For example, replacing objective search (fitness functions) with the search for behavioral diversity in controller evolution [14], [15], [16] has boosted evolved behavior quality across a range of simulated [17], [11], [13] and physical [7], [12] tasks.

Related research also indicated that *non-objective based* search methods such as *novelty search* [17], applied to direct evolutionary search [11], out-perform objective-based search in various robotic control tasks defined by complex, high dimensional and deceptive fitness landscapes [7], [12], [13]. While the benefits of non-objective based evolutionary search has been demonstrated for increasing evolved behavioral quality in various tasks [11], the *morphological robustness* of controllers evolved with such search methods remains unknown. This is especially the case for controller evolution in groups of robots that must accomplish collective behavior tasks [23]. Morphological robustness refers to the capacity for evolved controllers to continue to effectively function, despite degradation to robot morphology, for example, loss or damage of sensors on some or all robots.

In this study, the evolutionary controller design method is *Hyper-Neuro-Evolution for Augmenting Topologies* (HyperNEAT) [18], the task is *collective construction* [19], and objective versus *Novelty Search* (NS) [17] is applied to direct collective behavior evolution in robot groups. HyperNEAT was selected as it is a neuro-evolution method that has been effectively applied for controller evolution to solve various collective behavior tasks [20], [21], and has demonstrated benefits such as exploiting task geometry and regularity that boosts task performance.

Collective construction was selected as it is a task that benefits from autonomous robot groups that must exhibit robust collective behavior behavior in dynamic, noisy environments [22]. Also, the collective construction task includes the notion of morphological damage to robots that may impede group task accomplishment. The collective construction task requires robots to search the environment for *building-blocks* that are collectively transported and connected to other blocks in *construction zones* [23]. Collective construction task complexity was equated to the number of robots (degree of cooperation) needed to connect blocks together.

A. Research Objectives

This study's research objectives were formulated given results of the following previous related work. First, that non-objective based controller evolution has been effectively demonstrated in collective robotics [24], [25], [16]. Second, that NS has been demonstrated as suitable for evolving controllers (behaviors) that effectively operate across a range of robot morphologies [26], [27], [28]. The research objectives of this study are thus as follows:

1) Demonstrate the efficacy of NS (compared to objective-based search) for evolving *morphologically robust* controllers over increasingly task complexity.

2) Demonstrate the efficacy of NS evolved behaviors versus those evolved by objective-based search in terms of average task performance (behavior quality) over increasingly task complexity.

To test these objectives, experiments evaluated various robot morphologies in an increasing complex collective construction task. That is, morphological robustness of evolved controllers was evaluated in terms of a controller's task performance when coupled with alternate robot morphologies.

This study's contribution was thus to elucidate the impact of specific objective versus non-objective search methods on the evolution of *morphologically robust* controllers in collective robotic systems. To date, the morphological robustness of such controller evolution approaches has not been comparatively evaluated, especially in the context of collective robotics. Specifically, collective robotic systems that must effectively adapt to unforseen morphological change, such as the loss or damage of sensors on one or more robots without significant task performance degradation [6], [7].

II. Methods

HyperNEAT [18] was applied to evolve robot team (collective) behaviors, where teams were behaviorally and morphologically homogenous teams meaning all robots in a given team used the same controller and sensory configuration. HyperNEAT extends NEAT (*Neuro-Evolution of Augmented Topologies*) [29], where *Artificial Neural Network* (ANN) controllers were indirectly encoded using a CPPN (*Compositional Pattern Producing Network*) [30]. HyperNEAT was selected in this study for team controller evolution since it has numerous benefits demonstrated in previous multi-agent (robot) research [21], [31]. This includes HyperNEAT's capacity to evolve controllers that compactly encode and exploit geometric features such as symmetry, regularity and modularity in robot morphologies and the task environment, which in turn results in increased task performance.

HyperNEAT evolved a CPPN that encoded connection weight values between each robot's sensory input layer, hidden layer and motor output layer. Connections from pairs of nodes in the substrate network were sampled and the coordinates passed as inputs to the CPPN, which then output the synaptic weight of each sampled connection (Figure 1). Connection weights between the substrate input, hidden and output layers were encoded by the evolved CPPN.

A. Team Controller

Each robot's ANN controller comprised *N* sensory input and hidden nodes, connected hidden layer to two motor outputs (controlling the robot's left and right wheels, Figure 1, right). Nodes were arranged as a substrate (Figure 1, left) where the number of input and hidden nodes was determined by a given robot morphology (Figure 2). Each substrate node was placed at specific (x, y) locations in the substrate's two-dimensional geometric space (x, y axes in the range: [-1.0, 1.0]). Sensor nodes of the substrate approximated up to a 360° degree sensory *Field of View* (FOV), where the FOV was dependent upon the morphology used. For example, morphology 3 (figure 2, right) used one proximity and one ranged color sensor and one low-resolution camera (table I). Hence, the sensory FOV for morphology 3 was 3.0 radians (proximity color sensor) plus 1.5 radians (ranged color sensor) plus 1.5 radians (camera), which approximated a 340° FOV about the sensors on the robot's periphery (figure 2, right).

Figure 1 presents the team ANN for $N = 11$ (morphology 1, Figure 2, Table I), the associated substrate and an example CPPN. The intermediate ANN hidden layer reflects the input layer geometry and thus the direction of each sensor's FOV (Figure 1). The ANN was initialized with random weights in the range [-1.0, 1.0], with full connectivity between adjacent layers, however, partial connectivity was evolvable via the CPPN generating a zero weight.

Connection weights were evolved via querying the CPPN for the weight of any connection between two points (x_1, y_1) and (x_2, y_2) by inputting (x_1, y_1, x_2, y_2) into the CPPN, which subsequently output the associated weight. A CPPN was evolved via having nodes and connections added and removed, as well as connection weight values mutated (Table II) during HyperNEAT evolution. The CPPN evolved connectivity patterns across ANN geometry via querying all potential connections for their weights. This connectivity pattern was a function of task and ANN geometry, which enabled HyperNEAT to exploit task structure (regularity, repetition and symmetry) and robot morphology.

For example, there was symmetry and regularity in robot morphology in terms of sensor positions, and repetition of sensors about each robot's periphery (Figure 1), as well as regularity and repetition in the collective construction task (section III), in terms of repeating blocks comprising regular structures in construction zones. Table II presents the simulation, experiment and HyperNEAT parameters used in this study, where *delta* was the angle between (x_1, y_1, x_2, y_2) positions of nodes in the substrate. Parameter values were determined experimentally, where HyperNEAT parameters not listed in Table II, were set as in previous work [32].

B. Sensors

Each robot was equipped with various sensor types, where the exact sensor complement, including relative position and direction depended on the morphology being evaluated (Figure 2, Table I). Table II presents the different sensor types, where the functional properties of each sensor (range and FOV) were abstractions of physical sensors used on Khepera III robots [33]. Range values are units defined in relation to the environment size (20x20) and all sensor FOV values are in radians.

Each robot had *N* sensors corresponding to *N* ANN sensory inputs (Figure 1), each with a range of *r* (portion of the environment's size). A robot's sensory FOV was split into *N* sensor quadrants, where all sensors were constantly active for the duration of the robot's lifetime. The *nth* sensor returned a value in the range [0.0, 1.0] in the corresponding *nth* sensor quadrant. A 0.0 value indicated that no objects were detected and a 1.0 value indicated that an object was detected at the

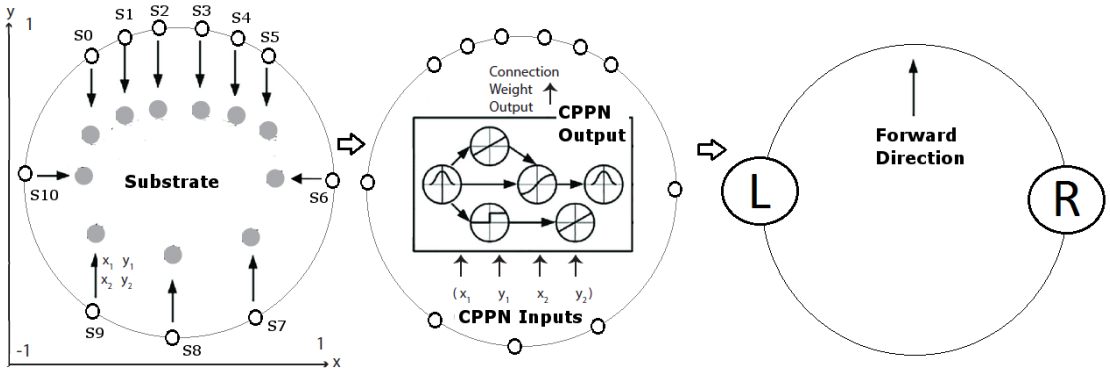

Fig. 1. *Left:* ANN Topology and robot morphology 1: 11 Sensory inputs [S0, S10]. Sensory inputs connect to a hidden layer (11 corresponding grey dots depicted). *Center:* Connection weight values between two nodes (x_1, y_1, x_2, y_2) are evolved by querying the CPPN with x, y values in the range [-1.0, 1.0]. The hidden layer is fully connected to all inputs and outputs (connectivity not depicted). *Right:* Motor outputs *L* and *R* determine the speed of the left and right wheels, respectively, and thus a robot's speed and direction.

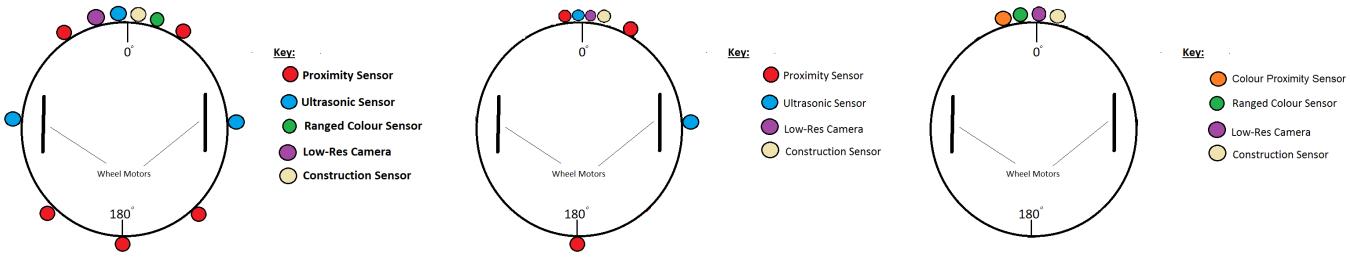

Fig. 2. Robot morphologies 1, 2, 3, from left to right. Table I details the sensory configuration (number and type of sensors) for each morphology.

TABLE I. SENSORY CONFIGURATION (NUMBER AND TYPE OF SENSOR) FOR EACH ROBOT MORPHOLOGY.

Morphology ID	Proximity Sensors	Ultrasonic Sensors	Color Ranged Sensors	Low-Resolution Camera	Construction Zone Sensors
1	5	3	1	1	1
2	3	2	1	1	1
3	1	0	1	1	1

closest possible distance to the given sensor. Thus, the functional capacity for a robot to operate in its environment and accomplish its task was dependent the number and composition of sensors used for its given morphology. Figure 2 depicts the three robot morphologies and Table I specifies the sensory configuration of each morphology.

Each morphology also included a special construction zone detection sensor that was constantly activated with a value in the range: $[0.0, 1.0]$, to enable collective construction. The construction zone sensor calculated the squared Euclidean norm, bounded by a minimum observation distance, as an inversely proportional distance between *this* robot and the closest construction zone. A 1.0 value indicated a robot (pushing a block) was in contact with a construction zone and a 0.0 value indicated the robot was the maximum possible distance from the closest construction zone (section III-A).

Robots were unable to detect each other, rather robots interacted via cooperatively pushing blocks into a construction zone. Once at least two blocks had been connected together this formed a construction zone (section III-A), that was then visible to each robot's construction zone sensor.

C. Actuators

Two wheel motors controlled each robot's heading at a constant speed. Movement was calculated in terms of real valued vectors (*dx* and *dy*), where varying sensory inputs and sensor configurations (morphologies) resulted in various Braitenberg dynamics [34]. Wheel motors (*L* and *R* in Figure 1) were explicitly activated by the ANN, where a robot's heading was determined by normalizing and scaling its motor output values by the maximum distance it could traverse in one simulation iteration (Table II). Specifically:

$$\begin{aligned} dx &= d_{max}(o_1 - 0.5) \\ dy &= d_{max}(o_2 - 0.5) \end{aligned}$$

Where, o_1 and o_2 were motor output values, corresponding to the left and right wheels, respectively, producing an output in the range: [-1.0, 1.0]. These output values indicate how fast each respective wheel must turn. Equal output equated to straight forward motion and unequal output resulted in the robot rotating about its own axis. The d_{max} value indicates the maximum distance a robot can move in one simulation iteration (Table II).

D. Objective-based Search (Fitness Function)

Equation 1 presents the fitness function (objective-based search) used to direct team controller evolution by HyperNEAT. This fitness function used a weighted sum accounting for the number of *type A* blocks *pushed* and *connected* (to any other block type) by *one robot* (a in Equation 1) and the number of *type B* blocks *pushed* and *connected* (to any other block type) by *three robots* (b in Equation 1).

$$f = r_a a + r_b b \tag{1}$$

Parameter tuning experiments found that setting weights (reward values r_a and r_b in Equation 1) to 0.3 and 1.0, respectively, resulted in functional controller evolution. Fitness was normalized to the range $[0.0, 1.0]$ using the maximum possible fitness yielded from all blocks being connected.

E. Novelty Search

In *novelty search* [17], the search for novel robot behaviors replaced the fitness function. In this study, behavioral novelty was characterized with three task specific *behavioral characterization* components, calculated at end of a team's lifetime and over all experiment runs (Table II):

- Γ: Average squared Euclidian distance between each robot's position.
- Θ: Average squared Euclidian distance between positions of joined (construction zone) blocks.
- Ξ: Average sum of differences in cooperation used (to connect each block in construction zones).

These behavior component values were normalized to the range: [0.0, 1.0] and used in behavioral distance calculations (Equation 2). This behavioral characterization was selected given previous work [16], and team behavior observations indicating the value of measuring robot, connected block positions and cooperation required, where these components played an important role in the evolution of novel collective construction behaviors.

Behavioral distance was computed using Equation 2, where, x_i and y_{ij} were (normalized) behavioral characterization vectors of two genotypes. The novelty of genotype *x*, with respect to all other genotypes in the archive (Table II), was then quantified by Equation 3.

$$\delta_i(x, y) = \|x_i - y_{ij}\| \tag{2}$$

Where, δ_x was the behavioral distance between genotypes x and y (Equation 2), and x_j was the jth behavior component (Γ, Θ or Ξ) of genotype x, and y_{ij} was the jth behavior component of the ith nearest neighbor of genotype x.

$$nov_x = \frac{1}{3k} \sum_{i=1}^{k} \sum_{j=1}^{3} \delta_x(x_j, y_{ij}) \tag{3}$$

Equation 3 replaces the fitness function (section II-D), where the nov_x was derived as the mean behavioral distance of a genotype with its k nearest neighbors. The parameter k represented the number of nearest neighbors, where $k = 15$ was selected as this setting has been widely used in related work [16]. The novelty of newly generated genotypes was then calculated with respect to previously novel behaviors stored in the novelty archive, where archived behaviors were ranked by diversity. In this study the maximum archive size was 100, where a maximum of 10 novel behaviors were added to the archive each generation (Table II).

TABLE III. COLLECTIVE CONSTRUCTION TASK COMPLEXITY: NUMBER OF ROBOTS NEEDED TO PUSH GIVEN BLOCK TYPES.

Construction Task Complexity	Level 1	Level 2
Type A blocks (1 robot to push)	15	0
Type B blocks (3 robots to push)	0	15

III. EXPERIMENTS

Experiments[1] tested and evaluated 15 robots in a bounded two dimensional continuous simulation environment (20 x 20 units) with randomly distributed type *A* and *B* blocks (Table II). Robots and blocks were initialized with random orientations and positions throughout the environment. The experimental objective was to evaluate the *morphological robustness* of HyperNEAT evolved controllers for robot teams given collective construction tasks (section III-A). We measured the average task performance of controllers evolved for three team morphologies (Table I) and two levels of task complexity (Table III). Each experiment comprised a team *controller evolution* and *re-evaluation* stage, where the latter was the morphological robustness test.

In the controller evolution stage, each experiment applied HyperNEAT, where evolutionary search (to evolve team behavior) was directed by either objective-based (section II-D) or novelty search (section II-E), running for 100 generations. Each generation comprised three team *lifetimes* (1000 simulation iterations), where each team lifetime tested different robot starting positions, orientations, and building-block locations in the simulation environment.

Teams that achieved an average task performance that was not significantly lower across all *re-evaluated* morphologies were considered to be *morphologically robust*. Specifically, the fittest controller evolved for a given morphology and level of task complexity was re-evaluated in the other morphologies for the same level of task complexity. For example, the fittest controller evolved for morphology 1 was re-evaluated in morphologies 2 and 3 and the average task performance calculated across all re-evaluation runs.

Re-evaluation runs were *non-evolutionary*, meaning controllers were not further evolved, and each re-evaluation was equivalent to one team lifetime (Table II). There were 20 re-evaluation runs for each morphology (Figure 2), in order to account for random variations in robot and block starting positions and orientations, where an average task performance was computed for all re-evaluated morphologies.

[1] Simulator screen-shots and source code for all experiments is online: https://github.com/not-my-name/SSCI18_Appendix

TABLE II. EXPERIMENT, HYPERNEAT, NOVELTY SEARCH AND ROBOT SENSOR PARAMETERS

Generations / Team lifetimes per generation	100 / 5	
Sensors per robot	Figure 2, Table I	
Evaluations per genotype	3	
Experiment runs (evolution & re-evaluation)	20	
Environment length, width	20.0 x 20.0	
Max robot movement per iteration	1.0	
Team size / Team Lifetime (Simulation iterations)	15 / 1000	
Type A / B blocks	15 / 15	
Task performance / complexity	Blocks connected in construction zones / Table III	
Mutation rates	Add neuron	0.25
	Add connection	0.008
	Remove connection	0.002
	Weight	0.1
Population size / Survival rate	150 / 0.3	
Crossover / Elitism proportion	0.5 / 0.1	
Connection weight range	[-1.0, 1.0]	
CPPN topology	Feed-forward	
CPPN inputs	Position, delta, angle	
Novelty Search (NS) nearest neighbor k	15	
NS Maximum archive size / Behaviors added	100 / 10 (per generation)	
NS Compatibility / Behavioral threshold	3 / 0.03	
Robot Sensor	Range	Field of View (FOV)
Proximity Sensor	1.0	0.2 Radians
Ultrasonic Sensor	4.0	1.2 Radians
Ranged Colour Sensor	3.0	1.5 Radians
Low-Res Camera	3.0	1.5 Radians
Colour Proximity Sensor	3.0	3.0 Radians

A. Collective Construction Task

This study's research objective was to evaluate the morphological robustness of team controllers evolved with non-objective versus objective-based evolutionary search coupled with HyperNEAT. Collective construction tasks required robots to search the environment for building blocks and cooperatively push the blocks together into a structure. Task complexity was gauged according to the level of cooperation required to optimally solve the task, that is, connect all the blocks in construction zones (Table III).

For task complexity levels 1 and 2, there were 15 type *A* and *B* blocks, respectively. In the case of level 1, a single robot could push each block, but in the case of level 2, three robots were required to cooperatively push and connect blocks (Table III). In this collective construction task it is assumed that type B blocks were significantly larger than type A blocks, and thus required three robots to cooperatively push and move. A construction zone was formed via at least two blocks being pushed together thus forming a structure. Once a construction zone was created, all blocks attached to it were fixed in position and could not be disconnected. The task used a maximum of three construction zones and unconnected blocks had to be pushed and connected to one of these construction zones.

Team task performance was calculated as the number of blocks connected in construction zones during a team's lifetime (Equation 1), where average task performance was the highest performance at the end of each run (100 generations), averaged over 20 runs (Table II). HyperNEAT was applied to evolve team controllers given evolutionary search was directed by either objective (section II-D) or novelty search (section II-E), where such search methods directed behavior evolution to optimize this task performance metric.

IV. RESULTS AND DISCUSSION

To address the research objectives (section I-A), we compared the average task performance results of team behavior evolution directed by *objective* (section II-D) versus *novelty search* (NS, section II-E). For each experiment, task performance was the number of blocks connected in construction zones over a team's lifetime (section III-A), where the maximum task performance was taken at each run's end and an average task performance calculated over 20 runs (Table II).

Figures 3 and 4, present the *average task performance* (normalized to the range: [0.0, 1.0]) of team controller evolution (directed by objective and NS) and morphological robustness experiments given increasing collective construction task complexity (Table III). For example, *Morphology 1* in the left-hand plot of Figure 3 presents the average task performance of controller evolution with objective-based search in morphology 1. The labels *Morphology 2* and *3* in this left-hand plot (of Figure 3) indicate average task performance results of re-

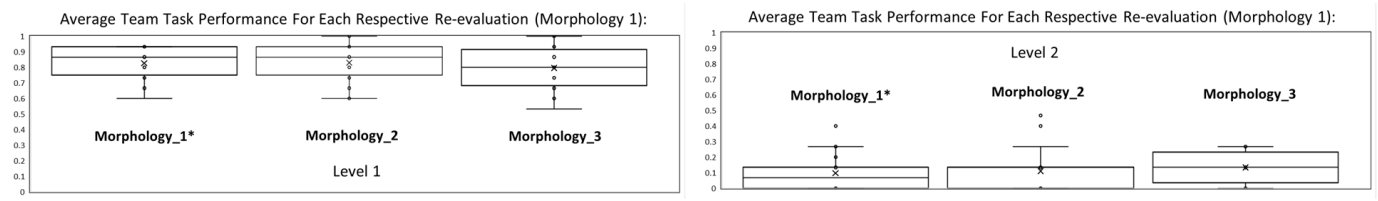

Fig. 3. Morphological Robustness of *Objective Search*: Average task performance of fittest team controller evolved for morphology 1 (left-most box-plot in each Figure) and re-evaluated in morphologies 2 (center plot), 3 (right-most plot). Left and right-hand figures: Task complexity levels 1 and 2 (respectively).

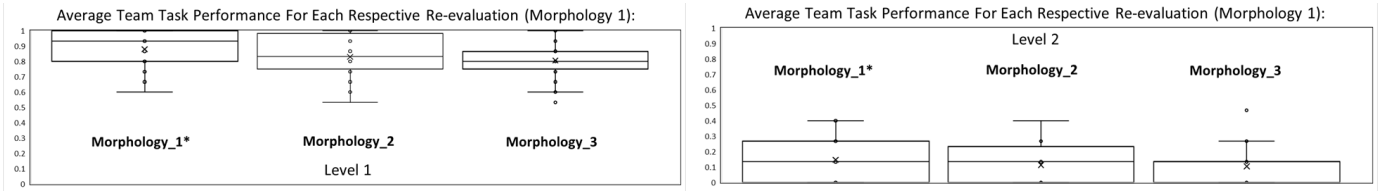

Fig. 4. Morphological Robustness of *Novelty Search*: Average task performance of fittest team controller evolved for morphology 1 (left-most box-plot in each figure) and re-evaluated in morphologies 2 (center plot), 3 (right-most plot). Left and right-hand figures: Task complexity levels 1 and 2 (respectively).

TABLE IV. P-VALUE STATISTICAL RESULTS COMPARING AVERAGE TASK PERFORMANCE OF NOVELTY-SEARCH (TOP TWO ROWS) AND OBJECTIVE-SEARCH (BOTTOM TWO ROWS) EVOLVED TEAM BEHAVIORS (FOR A GIVEN MORPHOLOGY) VERSUS TASK PERFORMANCE WHEN THE FITTEST CONTROLLER WAS RE-EVALUATED IN OTHER MORPHOLOGIES. BOLD: STATISTICALLY SIGNIFICANT DIFFERENCE.

Novelty search **Task Complexity: Level 1**	Re-evaluated in Morphology 1	Re-evaluated in Morphology 2	Re-evaluated in Morphology 3
Morphology 1 (Fittest controller selected)	–	0.12	$<$ **0.05**
Morphology 2 (Fittest controller selected)	0.50	–	0.50
Morphology 3 (Fittest controller selected)	0.18	0.17	–
Novelty search **Task Complexity: Level 2**	Re-evaluated in Morphology 1	Re-evaluated in Morphology 2	Re-evaluated in Morphology 3
Morphology 1 (Fittest controller selected)	–	0.31	0.50
Morphology 2 (Fittest controller selected)	0.50	–	0.50
Morphology 3 (Fittest controller selected)	0.17	0.40	–
Objective search **Task Complexity: Level 1**	Re-evaluated in Morphology 1	Re-evaluated in Morphology 2	Re-evaluated in Morphology 3
Morphology 1 (Fittest controller selected)	–	0.46	0.22
Morphology 2 (Fittest controller selected)	0.50	–	0.5
Morphology 3 (Fittest controller selected)	0.42	0.29	–
Objective search **Task Complexity: Level 2**	Re-evaluated in Morphology 1	Re-evaluated in Morphology 2	Re-evaluated in Morphology 3
Morphology 1 (Fittest controller selected)	–	0.35	$<$ **0.05**
Morphology 2 (Fittest controller selected)	0.35	–	0.38
Morphology 3 (Fittest controller selected)	0.32	0.43	–

evaluating the fittest controller evolved for morphology 1 in morphologies 2 and 3 (Figure 2).

Note that we only presents graphed results for controllers evolved by objective-based (Figure 3) versus NS directed HyperNEAT (Figure 4) in *morphology* 1 and re-evaluated in *morphologies 2* and *3*. Task performance results for objective and NS controller evolution in morphologies 2 and 3 and re-evaluation in other morphologies (1 and 3, and 1 and 2, respectively) were similar to those presented in Figures 3 and 4 and are thus not presented here. However all graphed results are available online[2].

In order to ascertain if the fittest objective (Figure 3) versus NS (Figure 4) evolved controllers were *morphologically robust* we applied statistical tests between average task performance results for controller evolution (evaluation) in each morphology (for example, morphology 1) and average task performances for re-evaluation in other morphologies (for example, 2 and 3). Evolved controllers were considered *morphologically robust* if there was no significant statistical difference between average task performance results yielded for controller evolution and re-evaluation experiments (section III).

Specifically, statistical *t-tests* ($p < 0.05$) [35] were applied in pair-wise comparisons between average task performance results for objective versus NS in each morphology and average re-evaluation task performance results (of the fittest objective and NS evolved controller) in each other morphology.

Statistical tests indicated that, for both objective and NS, and increasing task complexity, there was *no significant statistical difference* in average task performance between controllers evolved in any morphology and re-evaluated in any other morphology.

[2]https://github.com/not-my-name/SSCI2018_Appendix

However, Table IV presents two exceptions to this result. First, NS applied in task complexity 1 to evolve controllers in morphology 1. In this case, the fittest NS evolved controller re-evaluated in morphology 3 yielded a significantly lower average task performance (top, Table IV). Second, objective search in task complexity 2 to evolve controllers in morphology 1. The fittest controller evolved in morphology 1 and re-evaluated in morphology 3 had a significantly lower average task performance (bottom, Table IV).

In both of these cases, one may observe that morphology 1 uses 11 sensors where as morphology 3 uses only four sensors (Figure 2). This indicates that controllers evolved for the high sensor complement (and thus functionality) of morphology 1, are not readily transferable to a simpler sensory configuration (and simpler functionality). However, this was not the case for NS applied in task level 2 or objective-based search applied in task complexity level 1 (Table IV). That is, NS evolved controllers (given morphology 1) were found to be morphologically robust for task complexity level 2, where as controllers evolved with objective search were not. This suggests that NS may be suitable for evolving morphologically robust controllers for some specific types of tasks. However, this is a topic of ongoing research.

The key results of this study were thus two-fold. First, results indicated that for all morphologies and task complexity levels (with two previously discussed exceptions), both objective and NS were comparably effective at evolving morphologically robust controllers (Table IV). Specifically, for any given morphology, there was no significant difference in average task performance between the fittest NS evolved team controllers and re-evaluation of these controllers in the other morphologies. The same result was observed for the fittest controllers evolved by objective-based search.

This first result addresses this study's first objective (Section I-A) and indicates that while NS yields advantages over objective search in terms of evolving high quality behaviors, it yields no benefits over objective-based search for evolving morphologically robust controllers in the collective construction task for the given robot morphologies. Thus, for the given task and morphologies, objective-based search directing HyperNEAT controller evolution was sufficient to evolve morphological robust controllers and NS evolved controllers added no benefits in this respect (objective 1, Section I-A).

The lack of any difference between objective and NS in these morphological robustness experiments is theorized to be a result of the indirect (developmental) controller encoding of HyperNEAT [18]. Similar results have been previously demonstrated and explained by HyperNEAT's capacity to compactly encode complex behaviors with evolved CPPN *connectivity patterns* [36]. This in turn facilitated the transfer of evolved controllers (and behaviors) across different tasks [20] and robot morphologies of varying complexity [26], [28].

This result is also hypothesized to be consequent of the behaviorally and morphologically homogenous teams used in the collective construction task. That is, each robot in the team used the same controller and morphology, meaning that HyperNEAT was readily able to evolve team behaviors with a *multi-agent policy geometry* [21] that was transferable across teams with different morphologies. As in previous work [20], [21], such multi-agent (collective behavior) policy geometries are represented by evolved CPPNs. Previous related work found that such CPPNs encode geometric relationships between robot and block starting positions and orientations and sensory-motor activation values. Such sensory-motor activations were correlated with cooperative robot behaviors that suitably connected blocks, and thus accomplished given collective construction tasks [28], [37].

Second, collective construction behaviors evolved by NS, for all morphologies (Figure 2) and task complexity levels (Table III), significantly out-performed (pair-wise t-tests, $p < 0.05$) team behaviors evolved with objective-based search. This result is supported by previous work [25], [24], [16], similarly demonstrating that NS evolved collective behaviors out-perform those evolved by objective-based search in collective behavior tasks of varying complexity. This was found to be a result of the explorative capacity of NS [38], [39] which enables a broad search of the behavior space and thus discovery of high quality (task performance) solutions that could not be discovered by objective-based search given the same evolutionary parameter constraints. These results thus address the second objective of this study (Section I-A).

As this was a preliminary study investigating the morphological robustness of controllers adapted with varying evolutionary search methods, an extensive comparison between objective and novelty search was beyond this study's purview. However, in order to elucidate the impact of HyperNEAT's developmental encoding versus the evolutionary search method on morphological robustness, current work is investigating coupling varying evolutionary search methods (including novelty and objective search) with other (for example, direct encoding) controller evolution methods.

V. Conclusions

This study evaluated the efficacy of *objective* (fitness function) versus *non-objective* (NS: *Novelty Search*) based evolution of *morphologically robust* controllers in robot groups. Morphological robustness was the capacity for evolved controllers to effectively operate in alternate morphologies (sensory configurations of the robot group). Controllers were evolved for collective construction tasks of varying complexity, where the fittest controllers were evaluated on the same task but in differing morphologies.

The main research objective was to evaluate the efficacy of NS versus objective-based evolutionary search, coupled with HyperNEAT, for evolving controllers that were robust to morphological change. That is, controllers unaffected by morphological change such as sensor loss, damage or changes to sensory systems mandated by changing task constraints.

Results indicated that NS yielded no benefits over objective-based search for evolving morphologically robust controllers across increasingly complex tasks. That is, both NS and objective-based search were found to achieve the same degree of morphological robustness across all tasks, as neither objective or NS evolved controllers resulted in significantly lower task performances when re-evaluated in other morphologies.

Results did however support the efficacy of NS for evolving high quality group (collective) behaviors [11]. This secondary

result contributed further empirical evidence to highlighting the benefits of non-objective-based search for group controller evolution given increasingly complex collective behavior tasks [24], [25], [16], [40].

REFERENCES

[1] R. Brooks and A. Flynn, "Fast, cheap and out of control: Robot invasion of the solar system." *Journal of the British Interplanetary Society*, vol. 1, no. 1, pp. 478–485, 1989.

[2] G. Beni, "From swarm intelligence to swarm robotics," in *Proceedings of the First International Workshop on Swarm Robotics*. Santa Monica, USA: Springer, 2004.

[3] R. Pfeifer and J. Bongard, *How the body shapes the way we think*. Cambridge, USA: MIT Press, 2006.

[4] W. Fenton, T. McGinnity, and L. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31(3), pp. 269–281, 2001.

[5] G. G. V. Verma, R. Simmons and S. Thrun, "Real-time fault diagnosis," *IEEE Robotics and Automation Magazine*, vol. 11(2), pp. 56–66, 2004.

[6] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314(5802), pp. 1118–1121, 2006.

[7] A. Cully, J. Clune, D. Tarapore, and J. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521(1), pp. 503–507, 2015.

[8] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. Eiben, "Evolutionary robotics: what, why, and where to," *Frontiers in Robotics and AI — Evolutionary Robotics*, vol. 2(4), pp. 1–18, 2015.

[9] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.

[10] R. Kube and H. Zhang, "Collective robotics: from social insects to robots," *Adaptive Behaviour*, vol. 2, no. 2, pp. 189–218, 1994.

[11] J. Mouret and S. Doncieux, "Encouraging behavioral diversity in evolutionary robotics: An empirical study," *Evolutionary Computation*, vol. 20(1), pp. 91–133, 2012.

[12] A. Cully and J. Mouret, "Evolving a behavioral repertoire for a walking robot," *Evolutionary Computation*, vol. 24(1), pp. 1–33, 2016.

[13] J. Gomes, P. Mariano, and A. Christensen, "Novelty-driven cooperative coevolution," *Evolutionary Computation*, vol. 25(2), pp. 275–307, 2016.

[14] H. Moriguchi and S. Honiden, "Sustaining behavioral diversity in neat," in *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation*. Portland, USA: ACM, 2010, pp. 611–618.

[15] J. Lehman, K. Stanley, and R. Miikkulainen, "Effective diversity maintenance in deceptive domains," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2013, pp. 215–222.

[16] J. Gomes, P. Mariano, and A. Christensen, "Devising effective novelty search algorithms: A comprehensive empirical study," in *Proceedings of the Genetic Evolutionary Computation Conference*. Madrid, Spain: ACM, 2015, pp. 943–950.

[17] J. Lehman and K. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary computation*, vol. 19(2), pp. 189–223, 2011.

[18] K. Stanley, D'Ambrosio, and J. Gauci, "Hypercube-based indirect encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 1, pp. 185–212, 2009.

[19] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343(6172), pp. 754–758, 2014.

[20] P. Verbancsics and K. Stanley, "Evolving static representations for task transfer," *Journal of Machine Learning Research*, vol. 11(1), pp. 1737–1769, 2010.

[21] D. D'Ambrosio and K. Stanley, "Scalable multiagent learning through indirect encoding of policy geometry," *Evolutionary Intelligence Journal*, vol. 6, no. 1, pp. 1–26, 2013.

[22] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *The International Journal of Robotics Research*, vol. 27(3-4), pp. 463–479, 2008.

[23] G. Nitschke, M. Schut, and A. Eiben, "Evolving behavioral specialization in robot teams to solve a collective construction task," *Swarm and Evolutionary Computation*, vol. 2, no. 1, pp. 25–38, 2012.

[24] J. Gomes, P. Urbano, and A. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intelligence*, vol. 7, pp. 115–144, 2013.

[25] J. Gomes and A. Christensen, "Generic behaviour similarity measures for evolutionary swarm robotics," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. Amsterdam, Netherlands: ACM, 2013, pp. 199–206.

[26] S. Risi and K. Stanley, "Confronting the challenge of learning a flexible neural controller for a diversity of morphologies," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Amsterdam, The Netherlands: ACM, 2013, pp. 255–261.

[27] F. Corucci, M. Calisti, H. Hauser, and C. Laschi, "Novelty-based evolutionary design of morphing underwater robots," in *Proceedings of the Genetic Evolutionary Computation Conference*. Madrid, Spain: ACM, 2015, pp. 145–152.

[28] J. Watson and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Cape Town, South Africa: IEEE Press, 2015, pp. 1039–1046.

[29] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[30] K. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines: Special Issue on Developmental Systems*, vol. 8, no. 2, pp. 131–162, 2007.

[31] J. Watson and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Cape Town, South Africa: IEEE, 2015, pp. 1039–1046.

[32] D. D'Ambrosio and K. Stanley, "Generative encoding for multiagent learning," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Atlanta, USA: ACM Press, 2008, pp. 819–826.

[33] F. Lambercy and J. Tharin, *Khepera III User Manual: Version 3.5*. Lausanne, Switzerland: K-Team Corporation, 2013.

[34] V. Braitenberg, *Vehicles: Experiments in synthetic psychology*. Cambridge, USA: MIT Press, 1984.

[35] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.

[36] J. Gauci and K. Stanley, "Autonomous evolution of topographic regularities in artificial neural networks," *Neural Computation journal*, vol. 22, no. 7, pp. 1860–1898, 2010.

[37] R. Putter and G. Nitschke, "Evolving morphological robustness for collective robotics," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Honolulu, USA: IEEE Press, 2017, pp. 1104–1111.

[38] S. Didi and G. Nitschke, "Hybridizing novelty search for transfer learning," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*. Athens, Greece: IEEE Press, 2016, pp. 10–18.

[39] ——, "Multi-agent behavior-based policy transfer," in *Proceedings of the European Conference on the Applications of Evolutionary Computation*. Porto, Portugal: Springer, 2016, pp. 181–197.

[40] G. Nitschke and S. Didi, "Evolutionary policy transfer and search methods for boosting behavior quality: Robocup keep-away case study," *Frontiers in Robotics and AI — Evolutionary Robotics*, vol. 4, no. 62, p. doi: 10.3389/frobt.2017.00062, 2017.