Evolving Morphological Robustness for Collective Robotics

***Abstract*—This study evaluates a *Neuro-Evolution* (NE) method for controller evolution in simulated robot teams, where the goal is to evaluate the *morphological robustness* of evolved controllers. *Artificial Neural Network* (ANN) controllers are evolved for a specific sensory configuration (morphology) and then evaluated on a set of different morphologies. The morphological robustness of evolved controllers is evaluated according to team task performance given a collective construction task of increasing complexity. The overall objective was to ascertain an appropriate method for evolving ANN controllers that are readily transferable to robot teams with varied morphologies. Such controller transfer is necessary if task specifications change and different sensory configurations are required, or if robots are damaged and some sensors become disabled. In both cases it is ideal if teams continue to exhibit consistent behavior and a similar task performance. Results indicate that an indirect (developmental) encoding NE method consistently evolves controllers that fully function when transferred to teams with varied morphologies. That is, where comparable or higher task performances were yielded compared to controllers evolved specifically for the varied morphology.**

I. Introduction

An open problem in *collective robotics* [1] and related fields such as *swarm robotics* [2] is ascertaining the most appropriate sensory-motor configurations (morphologies) for individual robots comprising a team that must work collectively given automatically generated controllers [3]. In *evolutionary robotics* [4], a popular method is to co-evolve robot behaviors and morphologies [5], [6], [7], [8]. Within the purview of such *body-brain* co-evolution, indirect encoding (developmental) methods have been effectively demonstrated for various single-robot tasks [9], [10], [11]. However, with few exceptions [12], [13], [14] there has been relatively little work that evaluates developmental methods to evolve behavior and morphology for collective robotics tasks, excluding related research in self-assembling and reconfigurable collective robotics [15], [16]. Specifically, this study focuses on developmental methods to evolve controllers that exhibit consistently robust behaviors such that robot teams effectively adapt to the loss of sensors or new sensory configurations without significant degradation of collective task performance, that is, *morphologically robust behavior*.

This study contributes to an open problem in evolutionary robotics, that is to evolve robot controllers that can continue to appropriately function given unforseen morphological change to the robot such as sensor damage or malfunction [17], [18]. Such robust controller evolution would yield significant advantages in applications where robust autonomous behavior is continually required [19]. In this study, the specific focus is on the evolution of morphologically robust behavior for robotic teams that must accomplish collective behavior tasks.

Given the added computational complexity of co-evolving body-brain couplings for robots that are behaviorally and morphologically heterogenous, this study focuses on evolving collective behaviors for *homogenous* teams with a range of fixed morphologies. Specifically, we apply the HyperNEAT [20] neuro-evolution method to evolve behavior for a range of morphologically homogenous teams that must solve various collective construction tasks. HyperNEAT was selected as it is a developmental encoding NE method with demonstrated benefits that include exploiting task geometry to evolve modular and regular ANN controllers with increased problem-solving capacity [21], [22].

The core contribution of this research is thus to demonstrate the efficacy of a developmental neuro-evolution method (HyperNEAT) for addressing the problem of evolving morphologically robust controllers in the context of collective robotics. To address this objective, this study tested and evaluated three different robot team sensory configurations (morphologies) in company with three collective construction tasks of increasing complexity. The fittest *Artificial Neural Network* (ANN) controller evolved for a given morphology was then transferred and evaluated in each of the other two team morphologies. Such transferred controllers were also evaluated in each of the three tasks. In all cases, evolved controller efficacy was evaluated in terms of collective construction task performance yielded by a robot team.

The *Collective construction* task [23] was selected since it benefits from fully automated robot teams that must exhibit robust behavior to handle changing task constraints, potential robot damage and sensor noise. For example, collective construction of functional structures and habitats in remote or hazardous environments [24]. As in related work [23], the task was for robots to search the environment for *building-block* resources, then move them such that are connected to other blocks. The task is solved if the team connects all blocks forming a structure during its *lifetime*, and this is equated with optimal task performance. However, the task is considered partially solved if only a sub-set of the blocks are connected by the team, though in such instances, team task performance is proportional to the number of blocks connected. In this study, task complexity was equated with the degree of cooperation required to move blocks in order that are connected. Task complexity was regulated via the environment containing three block types, where one, two and three robots were required to move each of the block types, respectively.

Hence, we report a preliminary investigation into developmental methods (HyperNEAT is tested in this case study) for evolving ANN controllers that are robust to morphological change in robotic teams or swarms that must operate in dynamic, noisy and hazardous environments.

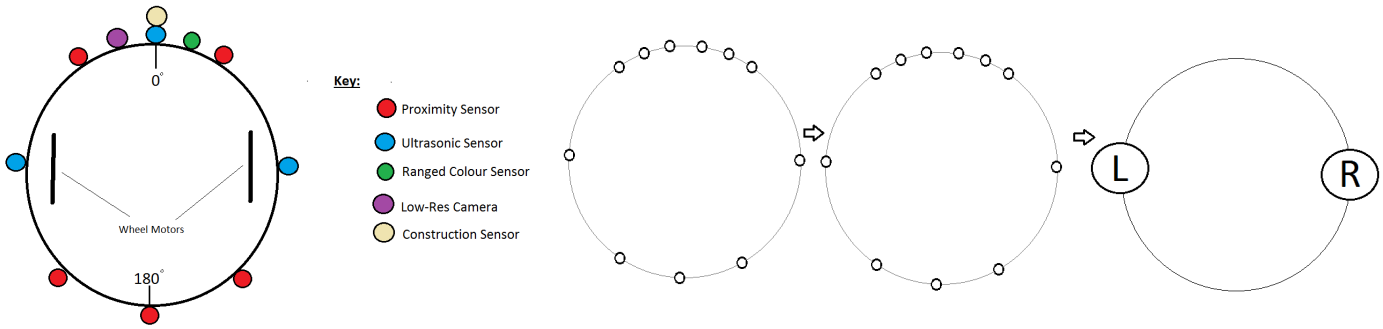

Fig. 1. *Left:* Example robot sensory configuration (*morphology 1*) indicating the relative positions of various sensors on the robot. *Right:* ANN Topology as it relates to robot morphology: Sensory input layer (left), hidden layer (center) and motor output layer (right). Output nodes *L* and *R* determine the speed of the left and right wheels, respectively, and thus the robot's speed and rotation. Arrows indicate the direction the agent is facing.

II. Methods

HyperNEAT [20] is an extension of the NEAT (*Neuro-Evolution of Augmented Topologies*) [25] method, where ANNs are indirectly encoded using a CPPN (*Compositional Pattern Producing Network*) [26]. HyperNEAT was selected as it has a number of benefits demonstrated in previous work [22], [27]. This includes its capability to exploit geometric features such as symmetry, regularity and modularity in the task environment during controller evolution. In the collective construction task, such geometric features include the relative positions of other robots, blocks, the direction robots and blocks are facing and the shape of the environment. Also, the structures to be built are modular (comprised of blocks) and often regular (the same sequence of blocks can be repeated).

In this study, HyperNEAT evolves the connection weights and the connectivity between a sensory input layer, hidden layer and motor output layer. Controller evolution experiments were initialized with a given morphology (table I). However, in one experiment set the sensor configuration of team morphology could be co-evolved via HyperNEAT activating and deactivating sensory input node connections over the course of its evolutionary process. For these co-evolution experiments (section III), *add connection* and *remove connection* mutation operators (table II) from previous work [14] were applied every generation to a sensory input node chosen with uniform random selection. The mutation operator applied depended on whether the chosen input node was connected or not. The construction sensor (table I) was permanently activated for all morphologies and could not be disconnected.

Table I presents a list of all sensor types (left column) and the corresponding morphologies that use the listed sensors (right column). The values in parentheses indicate the number of a given sensor type present in each morphology. The position of *morphology identification* (ID) number in the list corresponds to the number of the given sensor type. For example, morphology 2 has four proximity sensors, one ultrasonic sensor, one colour-ranged sensor, and one low-resolution camera. All morphologies have a construction sensor as this is necessary to complete the collective construction task.

A. Robot Team Controller

Teams were homogenous meaning all robots in a team used the same ANN controller, where each controller had *N* sensory input nodes, determined by a given morphology being evaluated (table I). Each robot's ANN adaptively mapped sensory inputs, via a hidden layer, to two motor outputs, the robot's left and right wheels (figure 1) using HyperNEAT [20].

Figure 1 illustrates an example ANN configuration for *N* = 11. The ANN uses a three dimensional coordinate system for processing *x*, *y*, *z* positions in the CPPN in order to generate weight and bias values and connectivity. The CPPN indirect encoding of HyperNEAT allows evolved controllers to compactly represent and exploit the geometry of the task and the environment, thus boosting controller task performance [20]. Previous work [28], [13] also demonstrated that HyperNEAT exploits the sensory-motor node configuration in ANN controllers, thus sensory input and motor output nodes must be in an appropriate configuration reflecting their position as part of a robot's morphology. Thus, the input layer of the ANN controller is a circle of *N* nodes distributed about the robot's periphery, where the exact geometric configuration corresponds to the team morphology being evaluated (table I). Collectively all sensory nodes approximate a complete 360 degree *Field of View* (FOV).

The intermediate ANN hidden layer reflects the configuration of the input layer, preserving the geometry of the sensory input layer, that is, the direction of each sensor's FOV (figure 1). The ANN was initialized with full connectivity between adjacent layers, however, partial connectivity was evolvable via the CPPN generating a zero weight. During evolution, the CPPN was developed via having nodes and connections added and removed, as well as connection weight values mutated [20]. Table II presents the neuro-evolution parameters used in this study, where the CPPN inputs affecting the weight or bias of a given node were the *x, y, z* position of the connecting nodes, and *delta* was the difference between their positions and the angle between them. Parameter values given in table II were determined experimentally and all other HyperNEAT parameters were set as in previous work [29].

1) Detection Sensors: Each robot can be equipped with a various sensor types, where the exact sensor complement, including the relative position and direction on the robot depends upon the given experiment (section III) and team morphology tested (table I). Each robot had *N* sensors corresponding to the *N* inputs comprising the robot's ANN sensory input layer (figure 1), each with a range of *r* (portion of the environment's length). A robot's sensory FOV was split into *N*

TABLE I. SENSORY CONFIGURATION FOR EACH MORPHOLOGY. NOTE: MORPHOLOGIES 1-4 ARE USED FOR CONTROLLER EVOLUTION AND 1-7 FOR EVALUATION EXPERIMENTS.

Sensor Type	Morphology ID (number of sensors)
Proximity	1, 2, 4, 5, 6, 7 (5, 4, 2 , 2, 5, 5)
Ultrasonic	1, 2, 5, 6, 7 (3, 1, 2, 3, 3)
Colour Ranged	1, 2, 3, 4, 5, 7 (1, 1, 1, 1, 1, 1)
Low-Resolution Camera	1, 2, 3, 4, 5 (1, 1, 1, 1, 1)
Construction	1, 2, 3, 4, 5, 6, 7 (1, 1, 1 ,1, 1, 1, 1)

sensor quadrants, where all sensors were constantly active for the duration of the robot's lifetime. The *nth* sensor returned a value in the normalized range [0.0, 1.0], in the corresponding *nth* sensor quadrant. A value of 0.0 indicated that no blocks were detected and a value of 1.0 indicated that an object was detected at the closest possible distance to the given sensor.

Table II presents the different sensor types used in this study, where the functional properties of each sensor (range and FOV) were abstractions of corresponding physical sensors typically used on the Khepera III robots [30]. In table II, range values are units defined in relation to the environment size (20 x 20) and FOV values are in radians. Each morphology also included a special construction zone detection sensor that activated with a value in the range [0.0, 1.0] whenever a robot came into contact with a block that must be connected with other already connected blocks. 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, where a value of 1.0 indicated the robot (pushing a block) was in contact with the construction zone and a value of 0.0 indicated that the robot (pushing a block) was the maximum possible distance from the closest construction zone. Robots were unable to detect each other, thus all cooperative interactions were *stigmergic* [31] where robots interacted via pushing blocks into the environment's construction zone.

2) Movement Actuators: Two wheel motors control a robot's heading at constant speed. Movement is calculated in terms of real valued vectors (*dx* and *dy*). Wheel motors (*L* and *R* in figure 1) need to be explicitly activated. A robot's heading is determined by normalizing and scaling its motor output values by the maximum distance a robot can traverse in one iteration (table II). That is:

$$dx = d_{max}(o_1 - 0.5)$$

$$dy = d_{max}(o_2 - 0.5)$$

Where, o_1 and o_2 are the 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 equates to straight forward motion and unequal output results in the robot rotating about its own axis. The d_{max} value indicates the maximum distance a robot can move in one simulation iteration (normalized to 1.0, table II).

III. EXPERIMENTS

Experiments tested 15 robots in a bounded two dimensional continuous environment (20 x 20 units) containing a random

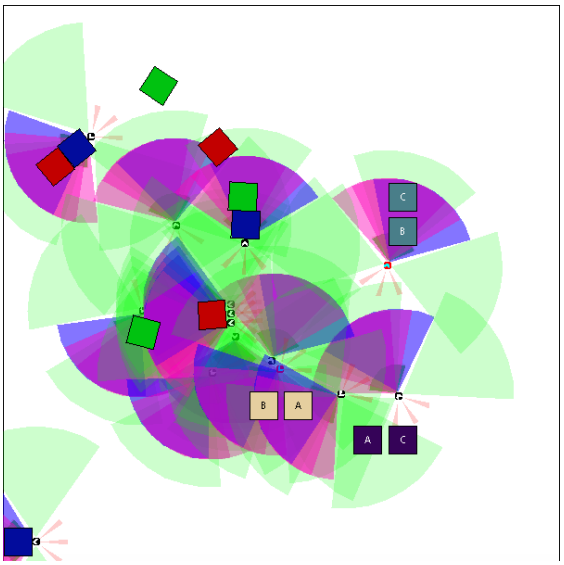

Fig. 2. Example of the simulation environment. Robots search for randomly distributed type A, B, and C blocks (blue, green and red, respectively). Other colored and labeled blocks indicate those already connected in construction zones. Different coloured semi-circles emanating from each robot represent the field of view of currently active different sensor types (table II).

Fig. 3. Task level 3 construction schema: *A*, *B*, and *C* are the block types. The label on each side of each block type indicates what block type can be connected to this side. An *X* label indicates that no block can be connected.

distribution of type *A*, *B* and *C* blocks (table II). Robots and blocks were initialized with random orientations and positions throughout the environment. A *construction schema* dictated the sequence of block types that must be connected together in order that a specific structure be built [32].

Figure 2 presents an example of the team of 15 robots working to solve the collective construction task in the simulation environment containing a distribution of five of each block type (*A*, *B* and *C*), colored blue, green and red, respectively. Other colored blocks in the environment indicate those already connected in construction zones (three illustrated). The purple, blue and green semi-circles emanating from each robot represent the FOV of currently active sensors, where the different colors correspond to different sensor types (table II).

As the purpose this study was to demonstrate the morphological robustness of HyperNEAT evolved controllers for a collective behavior task of increasing complexity, the first two versions of the collective construction task required no cooperation and some degree of cooperation, respectively, though any block could be connected to any other block. Where as, the most complex version of the task required cooperation and block types to be connected according to a construction schema (table III).

A. Collective Construction Task

This task required the robot team to search the environment for building-blocks and cooperatively push them together in order that they connected to form a structure, where connected blocks then formed a *construction zone*. Task complexity was equated with the degree of cooperation (number of robots required) to collectively push blocks and connect them together in the construction zone. In this construction task, there were three block types, *A*, *B* and *C* requiring one, two and three robots to push, respectively. Cooperation occurred when at least two robots simultaneously pushed a type *B* block, or at least three robots pushed a type *C* block.

Table III presents the three levels of task complexity for the collective construction task. Level 1 was the least complex as it did not require any cooperation, given that it in this case there were only type *A* blocks in the environment. Level 2 was of medium complexity as there are equal numbers of type *A*, *B*, and *C* blocks in the environment, where block types *B* and *C* required at least two and three robots to push, respectively. Level 3 was the most complex, as it required the same degree of cooperation as task level 2, though blocks had to be connected according to a construction schema. Figure 3 illustrates this construction schema, where the label on each of the four sides of each block type indicates what other block type can be connected to the given side. The *X* label indicates that no block can be connected to a given side.

The construction zone is formed via two blocks being pushed together, where the construction zone is any structure that has been built in the environment. Once a construction zone is created, all blocks attached to it are fixed in position and cannot be disconnected. The task mandated a maximum of three construction zones and unconnected blocks had to be pushed and connected to one of these construction zones. For task levels 1 and 2, any block could be connected to any other block, meaning that when two blocks were pushed together they automatically connected. For task level 3, blocks had to be pushed together such that they were connected on specific sides according to the construction schema (figure 3).

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 calculated as the highest task performance selected at the end of each run (100 generations) and averaged over 20 runs (table II). The fitness function to direct controller evolution was a weighted sum of the number of times *type A* blocks were pushed by *one robot* and connected (a in equation 1), the number of times *type B* blocks were pushed by *two robots* and connected (b in equation 1), and the number of times *type C* blocks were pushed by *three robots* and connected (c in equation 1).

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

Parameter tuning experiments found that setting the weights (reward values r_a, r_b and r_c) in equation 1 to 0.3, 0.6, 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 pushed and connected in construction zones.

TABLE II. EXPERIMENT, NEURO-EVOLUTION AND SENSOR PARAMETERS

Generations	100	
Sensors per robot	11, 8, 4, 6, random	
Evaluations per genotype	5	
Experiment runs	20	
Environment length, width	20	
Max Distance (Robot movement per iteration)	1.0	
Team size	15	
Team Lifetime (Simulation iterations)	1000	
Lifetimes per generation	5	
Type A blocks (1 robot to push)	15	
Type B blocks (2 robots to push)	15	
Type C blocks (3 robots to push)	15	
Mutation rate	Add neuron	0.25
	Add connection	0.8
	Remove connection	0.002
	Weight	0.1
Population size	150	
Survival rate	0.3	
Crossover proportion	0.5	
Elitism proportion	0.1	
CPPN topology	Feed-forward	
CPPN inputs	Position, delta, angle	
Sensor	**Range**	**FOV**
Proximity Sensor	1.0	0.2
Ultrasonic Sensor	4.0	1.2
Ranged Colour Sensor	3.0	1.5
Low-Res Camera	3.0	1.5
Colour Proximity Sensor	3.0	3.0

TABLE III. TASK COMPLEXITY. NOTE: TASK LEVEL 3 INCLUDES A CONSTRUCTION SCHEMA (FIGURE 2).

Construction Task Complexity	Level 1	Level 2	Level 3
Type A blocks (1 robot to push)	15	5	5
Type B blocks (2 robots to push)	0	5	5
Type C blocks (3 robots to push)	0	5	5
Construction schema	No	No	Yes

B. Experiment Design

Experiments evaluated the *morphological robustness* of HyperNEAT evolved controllers for robot teams that must accomplish collective construction tasks of increasing complexity (section III-A). That is, the change in average task performance of controllers evolved for a given team morphology and task and then transferred to another team morphology given the same task. Thus, teams that achieved an average task performance not significantly different across all *re-evaluation* morphologies were considered morphologically robust.

This study comprised five experiment sets, where the first four experiment sets evolved controllers given team morphologies 1 to 4 (table I) for three levels of increasing collective construction task complexity (table III). The fifth experiment set investigated the *co-evolution* of team morphology and behavior given morphology 5 (table I) as the initial sensory configuration for all robots in the team. This experiment set was included as a task performance benchmark given beneficial results of related work where team behavior and sensory morphology were co-evolved for collective behavior tasks [14].

Each experiment set comprised a team controller evolution and an evaluation stage. For controller evolution, each

TABLE IV. TWO-WAY ANOVA: IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM PERFORMANCE - TASK LEVEL 1.

	Df	SS	MS	F	p
Complexity	2	27.88	13.94	1187.36	1.6×10^{-181}
Morphology	7	3.68	0.53	44.8	3.9×10^{-48}
Complexity:Morphology	14	2.28	0.16	13.86	1.29×10^{-27}
Residuals	456	39.19	0.12		

TABLE V. TWO-WAY ANOVA: IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM PERFORMANCE - TASK LEVEL 2.

	Df	SS	MS	F	p
Complexity	2	17.81	8.92	649.21	3.8×10^{-134}
Morphology	7	0.19	0.03	1.98	0.06
Complexity:Morphology	14	0.15	0.01	0.79	0.68
Residuals	456	6.26	0.01		

experiment applied HyperNEAT to evolve team behavior for 15 robots for 100 generations, where a generation comprised five team *lifetimes* (1000 simulation iterations). Each team lifetime tested different robot starting positions, orientations, and block locations in an environment. The fittest controller evolved for each task level (yielding the highest absolute task performance) was then *re-evaluated* for morphological robustness in all other morphologies (1 to 7 in table I) for the given task.

Also, each re-evaluation run was equivalent to one generation, that is, five team lifetimes comprising 1000 simulation iterations (table II). Average task performance for evolved controllers re-evaluated in the other morphologies was calculated over 20 runs. As per this study's objectives, these morphological re-evaluation runs tested how robust the fittest evolved controllers were to variations in team morphology. That is, testing these controllers on a range of other morphologies emulated loss of sensors due to damage or variable robot morphologies introduced due to changing task constraints.

IV. RESULTS & DISCUSSION

For each experiment, the highest team task performance of controllers evolved during an evolutionary run was recorded, and average maximum fitness values computed over 20 runs for each level of task complexity. This is presented in figures 4, 5 and 6 (left side), where controllers were evolved for morphologies 1-5 (table I). In figures 4, 5 and 6, these results are presented from left to right. That is, average task performance results for morphology 1 are plotted to left-most side and average task performance results for morphology 5 are plotted to right-most side.

Similarly, in each experiment the fittest controller evolved for each task level was re-evaluated in morphologies 1-7 (table I) and an average task performance (taken over all morphologies), was computed over 20 runs for each level of task complexity. These morphological re-evaluation results are presented in figures 4, 5 and 6 (right side), where each of the five plots corresponds to the original morphology from which the fittest controller was taken and then run in all seven morphologies. For example, the left-most plot in figure 4 (right-side) presents the average task performance of the fittest controller evolved in morphology 1 and re-evaluated on all morphologies 1-7 (table I).

To gauge impact of a given team morphology (table I) in company with a given level of task complexity a two-way analysis of variance (ANOVA) [33] test was run on the controller evolution results (figures 4, 5 and 6, left). No statistically significant difference $p < 0.05$ was found between morphologies 1-5, for each given level of task complexity. Morphologies 1-4 were those implementing controller evolution in fixed sensory configurations. Morphology 5 was adaptive as both the controller and sensory configuration were co-evolved. The lack of statistical difference between controllers evolved in fixed versus the adaptive morphology, supports previous results stating that co-evolution of behavior and morphology yield at least comparable task performance benefits in collective behavior tasks of varying complexity [14].

TABLE VI. TWO-WAY ANOVA: IMPACT OF SENSOR CONFIGURATION AND TASK COMPLEXITY ON TEAM PERFORMANCE - TASK LEVEL 3.

	Df	SS	MS	F	p
Complexity	2	24.67	12.34	920.91	7.3×10^{-161}
Morphology	7	0.25	0.04	2.69	0.01
Complexity:Morphology	14	1.49	0.11	7.94	3.41×10^{-15}
Residuals	456	6.11	0.01		

However, to address this study's main objective it was necessary to ascertain the morphological robustness of the fittest controller evolved in each morphology when re-evaluated in all other morphologies. Thus, we also used two-way ANOVA tests to measure the impact of task complexity and re-evaluating the fittest controller evolved for each morphology on all morphologies (in comparison to controller evolution results).

The ANOVA statistical test results indicate a significant difference between average task performance results yielded by controller evolution and morphological re-evaluation experiments for all task complexity levels. That is, average task performance yielded by controllers evolved given morphologies 1-5 (figures 4, 5, 6, left) was significant lower than average task performance yielded by the fittest controllers evolved for morphologies 1-5, but re-evaluated on morphologies 1-7 (figures 4, 5, 6, right). Statistical test results are summarized in tables IV, V and VI for task complexity levels 1, 2 and 3, respectively, and include performance results comparisons between controller evolution and morphological re-evaluation experiments with respect to task complexity and morphology.

These results thus indicate that team controllers evolved by HyperNEAT for a given morphology, have the capacity yo effectively operate when transferred to a range of other morphologies, thus demonstrating the morphological robustness of HyperNEAT evolved controllers. This result was found to hold for all morphologies that controllers were evolved for, and for all levels of task complexity tested (table III).

The efficacy of HyperNEAT for evolving morphologically robust controllers is further supported by the controller evolution experiments that used morphology 5 (section III). In this case, the number of sensors was adaptable meaning that team behavior and morphology were co-evolved. Specifically, these controller evolution experiments began with the sensory configuration of morphology 5 (table I) and enabled and disabled sensor connections to better couple morphology with the evolved controller. Hence, the fittest controller evolved in this case often corresponded to a sensory configuration dissimilar to morphology 5 (initial sensory configuration).

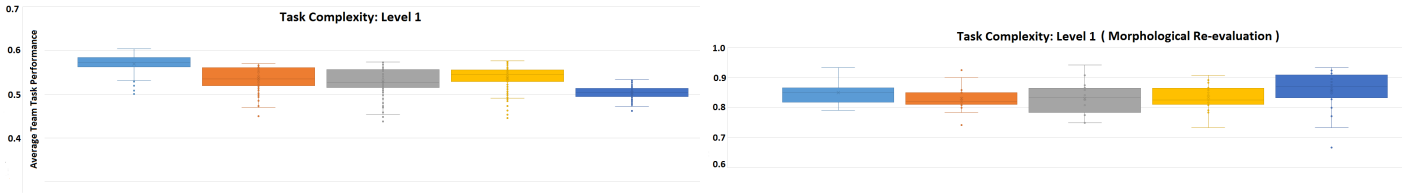

Fig. 4. Average team task performance for controller evolution (*left*) given morphologies 1-5 (depicted from left to right) and average re-evaluation task performance of the fittest evolved controller in all morphologies (*right*), for *task level 1*. Note: for clarity, only portions of the task performance scales are shown.

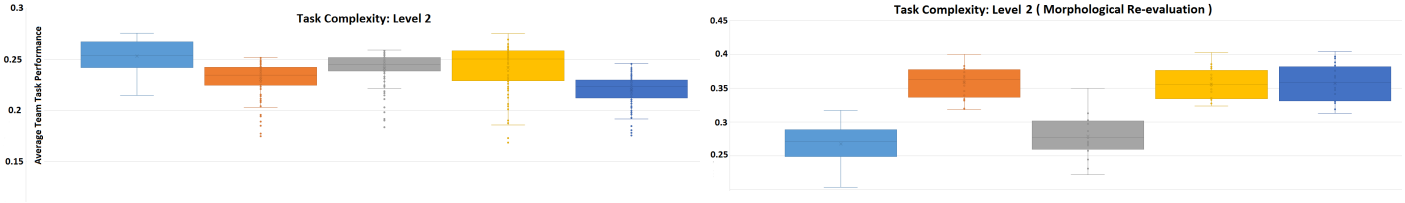

Fig. 5. Average team task performance for controller evolution (*left*) given morphologies 1-5 (depicted from left to right) and average re-evaluation task performance of the fittest evolved controller in all morphologies (*right*), for *task level 2*. Note: for clarity, only portions of the task performance scales are shown.

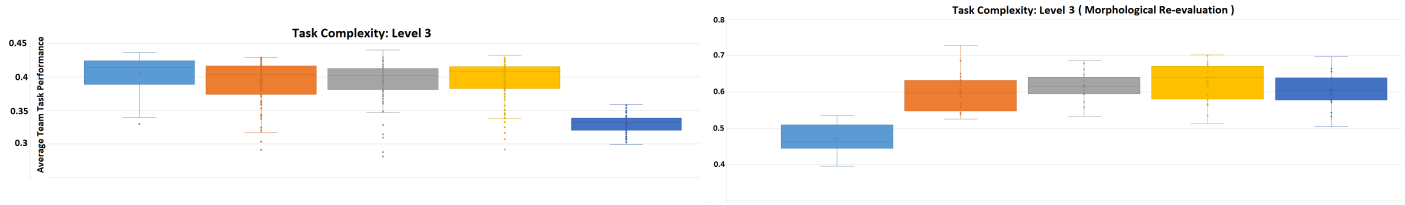

Fig. 6. Average team task performance for controller evolution (*left*) given morphologies 1-5 (depicted from left to right) and average re-evaluation task performance of the fittest evolved controller in all morphologies (*right*), for *task level 3*. Note: for clarity, only portions of the task performance scales are shown.

For the fittest controller evolved for morphology 5 re-evaluated in all other morphologies, the average re-evaluated task performance was statistically higher than average task performance yielded by controller evolution results. This result was observed for all three task complexity levels (figures 4, 5, and 6, right). Thus, evolving controllers for fixed versus adaptive morphologies had no significant impact on the morphological robustness of HyperNEAT evolved controllers (gauged in terms of average team task performance) for all morphologies tested.

Corroborating related work [34], [28], [13], these results contribute further empirical evidence that HyperNEAT yields significant benefits enabling the evolution of behaviors that are effective in other robot morphologies (sensory configurations). In this study this was demonstrated for a range of team morphologies in increasingly complex collective behavior tasks. The benefits include HyperNEAT's capability to exploit modularity and geometric properties such as regularity and symmetry in the task environment [20]. This is particularly relevant in the collective construction task which contains multiple instances of three types of square blocks that must be connected together in a repeated manner. Also, the environment used in the collective construction task was symmetrical (that is, a bounded environment of 20 x 20 units, table II).

Previous work has also demonstrated that the indirect sensory to motor mapping evolved by HyperNEAT is able to encapsulate complex yet effective behaviors with relatively few geometric relationships (such as desired positions and angles between different object types) discovered in the task environment [35]. That is, HyperNEAT evolves CPPNs representing ANN controllers of varying complexity with their own symmetries and regularities which are able to effectively exploit varyingly complex robot (agent) sensory-motor configurations. This in turn, often significantly boosts the efficacy of evolved collective behaviors [22].

Hence, the results of this study demonstrate that HyperNEAT is appropriate for evolving *connectivity patterns* [36] that are broadly applicable as fully functional controllers in a range of team morphologies. This is supported by previous work that similarly demonstrated the robustness of HyperNEAT evolved controllers for various quadruped robot morphologies [34], and for co-evolved behavior and sensor morphology in robot teams given a relatively simple collective behavior task [13].

The efficacy of HyperNEAT for evolving morphologically robust controllers for collective behavior tasks of varying complexity is also supported by related research in *multi-agent policy transfer* [35], [37], [38]. Policy transfer aims to derive methods that facilitate the transfer of behaviors across tasks of increasing complexity or between dissimilar tasks. Such studies have demonstrated that HyperNEAT is an effective method for evolving behaviors in one collective behavior task and then transferring the evolved behavior to a more complex task with relatively little loss in team task performance.

A benefit of HyperNEAT elucidated from such collective behavior transfer research was found to be its capability to evolve connectivity patterns between the sensory-motor layers of agent controllers that are broadly applicable to collective

behavior tasks of varying complexity. That is, HyperNEAT encodes team behaviors that do not rely upon specific sensory-motor mappings in team controllers and thus does not necessitate specific task environment configurations (such as specific numbers of agents or objects) [35], [37], [38]. Similarly, in this study HyperNEAT evolved controllers that were effective for varying levels of collective construction task complexity. Furthermore, for each level of task complexity, a higher average task performance was observed for the fittest controllers re-evaluated in all team morphologies (figures 4, 5, and 6).

However, we hypothesize that the morphological robustness of HyperNEAT evolved controllers demonstrated across all morphologies tested (table I) and all levels of task complexity (table III) was facilitated by the use of morphologically and behaviorally homogenous teams. Specifically, one controller was evolved for all robots in a team and all robots used the same sensory configuration, meaning all robots had the same *collective behavior geometry* [39]. That is, the fittest controllers evolved for a given level of task complexity encoded compact representations of specific relationships between behaviors for any given robot position and orientation relative to objects in the environment.

Furthermore, it has been demonstrated that HyperNEAT is advantageous for evolving solutions to tasks that are regular, modular and symmetrical in nature [20]. The efficacy of HyperNEAT has been corroborated in regular, modular and symmetrical collective behavior tasks such as RoboCup keep-away [37] and collective construction [13], though, to date, has not been demonstrated on irregular, non-modular and asymmetrical collective behavior tasks.

Thus, current research is evaluating the impact of behaviorally and morphological teams on the evolution of morphologically robust behaviors for collective behavior tasks that are non-modular, irregular and without repetition or symmetry.

V. Conclusions

This research presented a study on the efficacy of HyperNEAT for evolving *morphologically robust* behaviors for homogenous robot teams that must solve a collective behavior task of increasing complexity. That is, the task performance effectiveness of behaviors evolved for a given team morphology (robot sensory configuration) that was then transferred to a different team morphology. Controllers that did not yield degraded task performance when transferred to another morphology were considered to be morphologically robust. The objective was to test and evaluate methods for collective behavior controller design that generate behaviors that continue to function without significant loss of performance in robotic teams that experience sensor damage or intentional changes to their sensory systems due to changing task constraints.

Results indicated that the HyperNEAT method was appropriate for generating morphologically robust artificial neural network controllers for a collective construction task of increasing complexity. The collective construction task required robots to cooperatively push blocks such that they connected together to form a structure. Task complexity was regulated by the number of robots required to push blocks and a construction schema mandating that block types be connected in specific ways. These results support the notion that developmental neuro-evolution methods, such as HyperNEAT, are appropriate for controller evolution in collective robotics applications where robot teams must adapt during their lifetime to damage or otherwise must dynamically adapt their sensory configuration to solve new unforseen tasks.

References

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

[2] G. Beni, "From swarm intelligence to swarm robotics," in *Proceedings of the First International Workshop on Swarm Robotics*. Santa Monica, USA: Springer, 2004, pp. 1–9.

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

[4] 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.

[5] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic life forms," *Nature*, vol. 406, no. 1, pp. 974–978, 2000.

[6] H. Lund, "Co-evolving control and morphology with lego robots," in *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, H. Fumio and R. Pfeifer, Eds. Berlin, Germany: Springer, 2003, pp. 59–79.

[7] G. Buason, N. Bergfeldt, and T. Ziemke, "Brains, bodies, and beyond: Competitive co-evolution of robot controllers, morphologies and environments," *Genetic Programming and Evolvable Machines*, vol. 6(1), pp. 25–51, 2005.

[8] J. Auerbach and J. Bongard, "Environmental influence on the evolution of morphological complexity in machines," *PLoS Computational Biology*, vol. 10(1), p. e1003399. doi:10.1371/journal.pcbi.1003399, 2014.

[9] C. Mautner and R. Belew, "Evolving robot morphology and control," *Artificial Life and Robotics*, vol. 4(3), pp. 130–136, 2000.

[10] G. Hornby and J. Pollack, "Creating high-level components with a generative representation for body-brain evolution," *Artificial Life*, vol. 8(3), pp. 1–10, 2002.

[11] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson, "Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Amsterdam, Netherlands: ACM Press, 2013, pp. 167–174.

[12] Y. Asai and T. Arita, "Coevolution of morphology and behavior of robots in a multiagent environment," in *Proceedings of the SICE 30th Intelligent System Symposium*. Tokyo, Japan: The Society of Instrument and Control Engineers, 2003, pp. 61–66.

[13] 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.

[14] J. Hewland and G. Nitschke, "Evolving robust robot team morphologies for collective construction," in *The Benefits of Adaptive Behavior and Morphology for Cooperation in Robot Teams*. Cape Town, South Africa: IEEE, 2015, pp. 1047–1054.

[15] R. O'Grady, A. Christensen, and M. Dorigo, "Swarmorph: Morphogenesis with self-assembling robots," in *Morphogenetic Engineering, Understanding Complex Systems*, R. Doursat, Ed. Berlin, Germany: Springer-Verlag, 2012, pp. 27–60.

[16] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345(6198), pp. 795–799, 2014.

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

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

[19] R. Brooks and A. Flynn, "Fast, cheap and out of control: A robot invasion of the solar system." *Journal of the British Interplanetary Society*, vol. 1, no. 1, p. 478485, 1989.

[20] 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.

[21] P. Verbancsics and K. Stanley, "Constraining connectivity to encourage modularity in hyperneat," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2011, pp. 1483–1490.

[22] 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.

[23] 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.

[24] 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.

[25] K. Stanley, *Efficient Evolution of Neural Networks Through Complexification. Ph. D. Dissertation*. Austin, USA: Department of Computer Sciences, The University of Texas, 2004.

[26] ——, "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.

[27] 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.

[28] ——, "Deriving minimal sensory configurations for evolved cooperative robot teams," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Sendai, Japan: IEEE, 2015, pp. 3065–3071.

[29] 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.

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

[31] R. Beckers, O. Holland, and J. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Proceedings of the International Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, USA: MIT Press, 1994, pp. 181–189.

[32] 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.

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

[34] 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.

[35] P. Verbancsics and K. O. Stanley, "Evolving static representations for task transfer," vol. 11, pp. 1737–1769.

[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] 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.

[38] ——, "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.

[39] D. D'Ambrosio, J. Lehman, S. Risi, and K. Stanley, "Evolving policy geometry for scalable multi-agent learning," in *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*. Richland, USA: ACM Press, 2010, pp. 731–738.