Evaluation of frameworks that combine evolution and learning to design robots in complex morphological spaces

Wei Li[‖,†], Edgar Buchanan[†], Léni K. Le Goff[‡], Emma Hart[‡], Matthew F. Hale[*], Bingsheng Wei[‖], Matteo De Carlo[§], Mike Angus[†], Robert Woolley[†], Zhongxue Gan[‖], Alan F. Winfield[*], Jon Timmis[¶], Agoston E. Eiben[§], and Andy M. Tyrrell[†]

[†]School of Physics, Engineering and Technology, University of York, UK

[*]Bristol Robotics Laboratory, University of the West of England, Bristol, UK

[‡]School of Computing, Edinburgh Napier University, UK

[§]Department of Computer Science, Vrije Universiteit Amsterdam, NL

[¶]School of Computer Science, University of Sunderland, UK

[‖]Academy for Engineering and Technology, Fudan University, China

Email: wei.li@york.ac.uk

***Abstract*—Jointly optimising both the body and brain of a robot is known to be a challenging task, especially when attempting to evolve designs in simulation that will subsequently be built in the real world. To address this, it is increasingly common to combine evolution with a learning algorithm that can either improve the inherited controllers of new offspring to fine tune them to the new body design or learn them from scratch. In this paper an approach is proposed in which a robot is specified indirectly by two compositional pattern producing networks (CPPN) encoded in a single genome, one which encodes the brain and the other the body. The body part of the genome is evolved using an evolutionary algorithm (EA), with an individual learning algorithm (also an EA) applied to the inherited controller to improve it. The goal of this paper is to determine how to utilise the results of learning process most effectively to improve task performance of the robot. Specifically, three variants are investigated: (1) evolution of the body+controller only; (2) a learning algorithm is applied to the inherited controller with the learned fitness assigned to the genome; (3) learning is applied and the genome is updated with the learned controller, as well as being assigned the learned fitness. Experiments are performed in three different scenarios chosen to favour different bodies and locomotion patterns. It is shown that better performance can be obtained using learning but only if the learned controller is inherited by the offspring. Our code is available on https://doi.org/10.6084/m9.figshare.24105450.v1.**

***Index Terms*—Morphological Evolution, Evolution and Learning, Embodied Intelligence**

I. Introduction

Starting with the pioneering work of Sims [1] in 1994, the field of evolutionary robotics has sought to use evolutionary algorithms to co-design the body and brain of robots. The current state-of-the-art has realised robots that can be built following evolution from a variety of novel substrates that include soft materials [2] and living cells [3]. The majority of research in this area focuses on modular systems, i.e., evolving designs that are constructed from a fixed set of component parts [4]–[6], which restricts the space of possible designs. A larger design space can potentially contain a more optimal body-plan to achieve better performance. Furthermore, most of these approaches evolve robots that lack sensors: as a result they operate via open-loop control mechanisms in which control is not directly influenced by any feedback from the environment.

In an effort to advance the field, an evolutionary framework that permits both evolution in a rich morphological space and delivers closed-loop controller has been proposed [7]–[9]. Specifically, the framework jointly evolves the body and brain of robots that have free-form skeletons (i.e. chassis), a diverse array of sensors and a range of actuators (wheels and legs). The skeletons can be 3D-printed and then the robot is autonomously constructed with pre-fabricated components such as a CPU (Raspberry Pi) in addition to the range of sensors and actuators previously mentioned. However, evolution in such a complex morphological space is very challenging. The body-plan of offspring robots produced by combining parents can be very different to either parent. As a result an inherited controller is unlikely to be a good match for the new body. For example, the number of sensors on the child robot might be different to both parents, which is especially problematic for neural network controllers which have a fixed number of inputs/outputs. Even changes in the placement of sensors on the body can result in vastly different control. As a result, a *learning* mechanism is usually required to fine-tune the controller [10].

The integration between evolution and learning conceptualized by the 'Triangle of Life', depicted in Figure 1, is a nested optimization system with two loops: the outer loop is an evolutionary algorithm that optimizes the bodies and the brains together, while the inner loop is a learning algorithm that improves the controllers of 'newborn' robots before they get evaluated to determine their fitness. Note that the framework facilitates any kind of learning algorithm — this itself can be evolutionary (e.g. [5], [9], [11]) but there are other potential candidates, e.g., reinforcement learning [12] or Bayesian optimisation [13]. However, using any framework that interweaves evolution and learning raises questions regarding how the two systems interact. Specifically, it introduces choices with respect to how the fitness obtained as a result of learning influences the selection process and whether the inherited genome is updated following learning to reflect the new controller.

This paper seeks to answer these questions. The experiments are grounded in the context of evolving body and control in the rich morphological space defined in previous work [7], [8]. Morphology and controller are *each* encoded by a compositional pattern producing network (CPPN) [14] on a single genome. This indirect method of generating both bodies and controllers is already common in the literature. In terms of controllers, it has the important characteristic of being able to construct a neural controller that matches the newly-generated body in terms of the number of inputs and outputs needed. Two separate CPPNs are used to generate (1) the morphology and (2) the weights in the neural controller. Each CPPN is evolved using neuro-evolution of augmenting topologies (NEAT) [15]. A learner is used which is also an evolutionary algorithm: for each robot (individual) in the outer population, it creates a population of CPPNs representing controllers *and containing the inherited CPPN*. NEAT is again used to evolve this learning population to improve the performance of the controller. Theoretically, any controller that can provide effective control to the evolving body can be used. Hence, there are other potential feasible controllers and optimising methods other than CPPN + NEAT. However, these experiments are restricted to this setup given it is commonly used in the literature and the goal of the paper is to explore the effectiveness of adding a learning system, not to compare different learning methods.

In all experiments, the best fitness obtained after learning is assigned to each robot in the outer population. Three versions of evolution are investigated. In the first, the CPPN defining the controller on the inherited genome is *not* updated following learning, however the learned fitness is used to guide selection. Hence, one might observe a Baldwin effect post-evolution [16]. The second scheme is Lamarckian-like: the CPPN that produces the best fitness following learning overwrites the inherited CPPN on the genome, and the genome is assigned the learned fitness. The third scheme is simply an EA without learning: body and controller are co-evolved without extra learning applied to the controller. These three schemes are compared with respect to performance of the robots evolved, the diversity of morphologies obtained, and speed of convergence.

The main contributions of this paper are as follows: (1) A specific implementation of the Triangle of Life model, which is capable of dealing with complex morphologies, and in which the learning loop is implemented by an evolutionary algorithm. It is referred to in the paper as a dual loop evolution structure (DLES). (2) A comparison of evolution

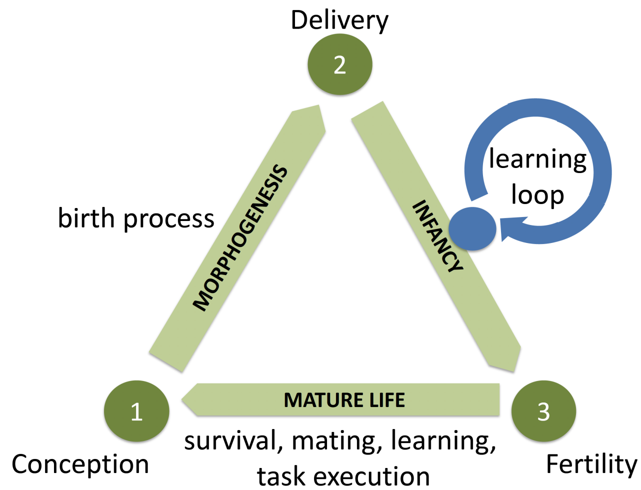

Fig. 1. The nested optimization system for robot evolution with an evolutionary and a learning loop, captured by the Triangle of Life model [17]. The evolutionary loop is formed by the green triangle, and the learning loop is shown by the blue circle.

and learning with controller inheritance, evolution and learning without controller inheritance, evolution only approaches. (3) A rigorous experimental study that seeks to understand the influence of the task and environment on the results obtained by DLES.

The rest of the paper is organized as follows: Section II overviews work on evolution of robot morphology and controller. Section III describes the Dual Loop Evolution Structure (DLES) proposed in this paper. Section IV describes the detailed experimental setup, including tasks, scenarios, evolution setting, etc. Section V analyses and discusses experimental results. Finally, Section VI brings together all the results and concludes the paper.

II. Related Work

In this section, previous studies that examine the joint evolution of robot morphology and control are reviewed, with particular attention paid to those that include intertwining evolution and learning.

As noted in the introduction, previous work is typically concentrated in a limited morphological search space. The first work in this area was pioneered by Sims [1]. A hierarchical graph-based encoding was used to represent ‘creatures’ that were evolved from a set of rigid parts of different dimensions and contained a variety of joint-types that provide different degrees of freedom. The evolutionary process used a hierarchical graph structure to specify the robot, where each individual part had embedded neurons for control. Veenstra *et. al.* [18] also evolved blue-prints that specify both the body and controller of a modular robot, i.e., one that is built from a library of ‘modules’ that can connect together at multiple sites on each module, comparing tree-based and grammar based representations. Brodbeck et. al. [4] evolved robots composed of a set of cubic active and passive modules. Each gene contains information about the module type to be used (active or passive), construction parameters and finally two parameters that specify the motor control of the module (the phase and amplitude of a sinusoidal controller). A CPPN [14] representation is used to evolve robot designs that are then built using living cells [19] while a Gaussian mixture representation is used to evolve robots built using soft materials [20]. In both of the latter cases, each material type had an associated parameter defining the rate of contraction/expansion hence there was no need to encode control separately.

With the exception of the work by Sims [1], the approaches described evolve robots without sensors and therefore have open-loop controllers. Furthermore, they tend to evolve modular robots, composed of a fixed set of component parts. Evolving in more complex morphological spaces, especially where sensors are included, tends to require augmenting evolution with a learning algorithm. Ruud *et al* [21] evolve controllers for a fixed morphology robot, but combine an EA with a local search learning algorithm to evolve control system parameters for a four-legged robot. The local search algorithm is run on every evolved controller. They compare two schemes, one in which the learned controller is inherited (dubbed Larmarkian) and one in which the learned fitness guides selection but without inheritance, finding the Larmarckian scheme to be most effective. Miras *et al* [11] evolve modular robots and their controllers simultaneously. They use the evolution strategy CMA-ES [22] to improve controllers, finding that the controller learning process not only boosts fitness of evolved robots, but also leads to evolution of larger robots (compared to robots that do not learn). Gupta *et al* [12] combine deep reinforcement learning (RL) with an evolutionary algorithm: the RL algorithm is applied to each evolved body-plan to learn a controller from scratch. They study the relationship between environmental complexity, morphological intelligence and the learnability of control, demonstrating existence of a Baldwin effect. However, this is applied within a relatively small design space.

In our previous work, initial studies were undertaken into 'evolution + learning' approaches in the rich morphological space described in the introduction. In Le Goff *et. al.* [23], a hierarchical optimisation framework is proposed in which an outer loop evolves a body-plan and an inner loop applies a learning algorithm to evolve a controller from scratch. In [13], two learning algorithms were compared: a modified evolution strategy named NIPES and Bayesian Optimisation. In [23], a weaker learner (based on Latin Hyper-Cube sampling) was also compared. In [9] an attempt to improve the learner that bootstrapped the learning algorithm from a previously found solution was suggested, rather than start from scratch, leading to improved results. However, this work has not previously made any attempt to design or evaluate methods in which the controller was encoded on the genome and therefore can be inherited by future offspring. Jelisavcic *et. al.* [24] studied evolutionary robot system with both Lamarckian and Darwinian type methods. Fully modular robots are used for the mophological design space.

In summary, the literature demonstrates that although there have been some attempts to combine evolution and learning in the joint optimisation of robot body and control there still exists many weaknesses. For example: (1) most previous work takes place in modular morphological spaces with open-loop control due to a lack of sensors; (2) when attempting to deal with complex morphology, it is typical to refrain from *evolving* the controller and instead apply a learner from scratch. This choice is often made due to the difficulty of evolving neural controllers in which the inputs and outputs match the evolving body-plan. (3) There have not been any studies in a complex morphological space permitting closed-loop control where both body-plan and control can be inherited and that attempt to understand how the results of the learning process should influence evolution. This paper directly addresses this gap.

III. Methods

A. Body-Plan Encoding and Decoding

A body-plan representation defined in [8] is used throughout this paper. The body-plans are encoded indirectly by a CPPN which defines a robot in a 3D voxel-based matrix. Each voxel can contain either skeleton material (which can be 3D-printed in reality) or pre-designed components [8] (organs). Each CPPN has four inputs and six outputs. The three inputs represent the 3D coordinates X, Y, Z of a cell in the 3D matrix, with the fourth input representing the distance from the cell to the centre of the matrix. The first output defines the presence or absence of skeleton in that cell. The following four outputs represent each component type (a robot can have a maximum of 16 components of the same type), i.e., wheel, sensor, joint and caster. The last output defines the orientation of the component. The skeleton is freely evolved and the evolution decides when and where to use the pre-designed components. This results in a very large search space. In order to ensure that robots can ultimately be manufactured via 3D printing and automated assembly, a repair process ensures the design is feasible (e.g. does not contain overhangs that cannot be printed). The algorithm used in this paper to evolve the CPPN is the widely used method NEAT (neuro-evolution of augmenting topology) [15], which evolves both the topology and the weight of the CPPN.

The decoding takes place in four steps: 1) The skeleton is first generated. 2) The skeleton is modified to meet the manufacturability restrictions. 3) The CPPN is queried again with coordinates on the surface of the skeleton to determine where components are attached: the output with the highest value defines the component type to be placed on the surface of the skeleton. 4) Colliding components are removed. This method is described in detail in [8]. The components (organs) are shown in Figure 2.

The decoding used in this paper has the additional feature of generating multi-segmented robots, i.e., 'legs' are composed of multi-segmented joints. The position of each skeleton voxel is queried in CPPN (Figure 3.1). If the component generated is a joint (Figure 3.2) then a cuboid skeleton is generated at the other end of the joint (Figure 3.3). The position of each face

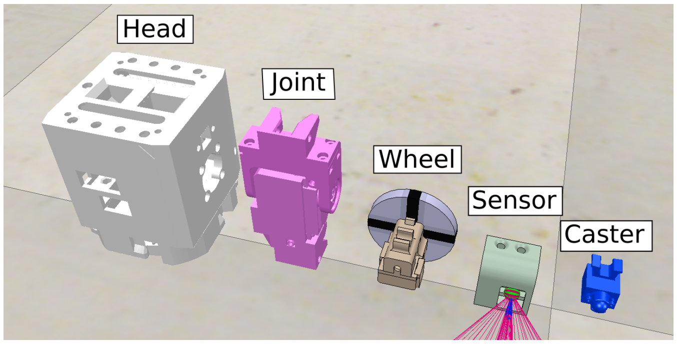

Fig. 2. Robot components (organs) for body-plan generation: The *head* contains a small computer that runs the main controller. Wheels, joints and castors provide locomotion ability. The sensor provides perception ability by identifying the existence of walls and in these experiments a beacon. Joints can be chained to form 'legs' [8].

of cuboid is queried to the same CPPN and components are generated (Figure 3.4). The work of Hale et al. [25] describes how the physical multi-segmented robot is assembled in the robot fabricator.

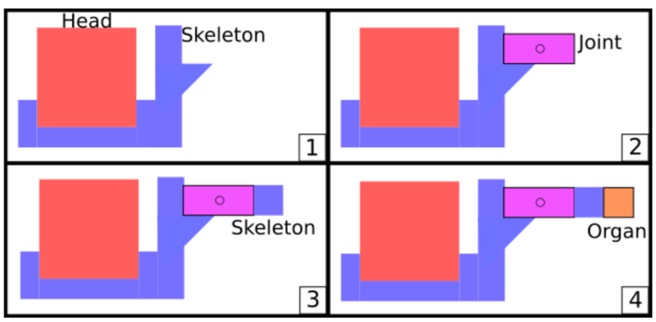

Fig. 3. Generation of multi-segmented robots. (1) The main skeleton is generated first. (2) A joint is placed on the surface of one of the voxels. (3) A cuboid skeleton with 4 cm side is generated at the other end of the joint. (4) The CPPN is queried to generate components at each side of the cuboid.

The ultimate motivation of this work is to evolve AND building physical robots, therefore each component in the body-plan has to meet pre-defined *manufacturability* criteria, first introduced in the work of Buchanan et al. [8]. For example, there should be no collisions between components; components should have the correct orientation; the position of a component can be accessed by a robot arm with a gripper when being manufactured. If a component fails any of the manufacturability tests then the component is removed from the final body-plan phenotype.

The physical head component has eight electrical connections for components, therefore limiting the number of components that can be connected to head skeleton at any time to eight. The joints offer the option to electrically daisy chain one more active component. In total, a body-plan can have up to 16 active components. The size of the skeleton connected to the head component can be as big as 23 cm x 23 cm x 23 cm.

B. Controller Encoding and Decoding

The controller is encoded by a separate CPPN [26] which defines the weights of an artificial neural network (ANN) controller as shown in Figure 4. The number of inputs and outputs of the network is determined by the new body of the robot, i.e., the number of sensors (inputs) and actuators (outputs).

As shown in Figure 4, the ANN controller consists of three parts, namely input layer, hidden layer and output layer. The input layer feeds sensor information into the ANN. The architecture of the hidden layers is fixed following initial empirical experimentation to determine appropriate values. There are two hidden layers, and 10 nodes in each layer with signed sigmoid activation functions. The output layers provides control to actuators. For each architecture, a substrate is defined consisting of the 2D coordinates of each node. CPPN HyperNEAT [27] is then used to evolve the weights between each pair of nodes.

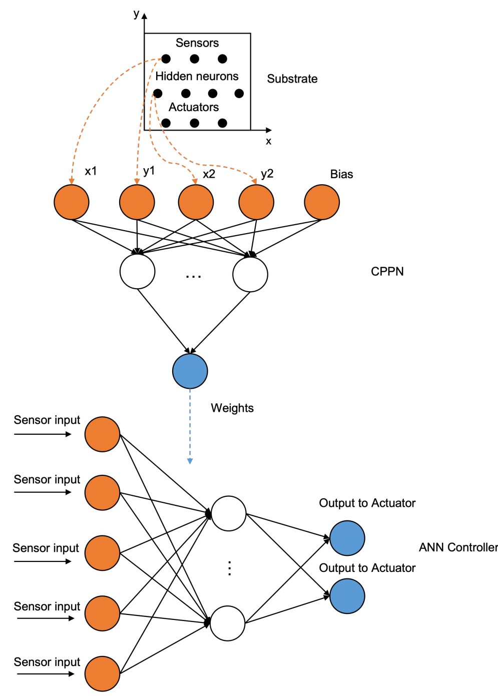

Fig. 4. Controller network: The number of connection between pairs of neurons is not restricted in order to maximize the diversity of the controller. Note that this figure is only an illustration of a possible network as each network has an architecture that maps to the number of sensors and actuators in the morphology.

C. Dual Loop Evolution Structure (DLES)

The proposed 'evolution+learning' framework which uses a dual loop evolution structure (DLES) uses an evolutionary algorithm that adds a nested learning loop for adapting an inherited controller to a new morphology. As mentioned in Sections III-A and III-B, an indirect encoding method is used

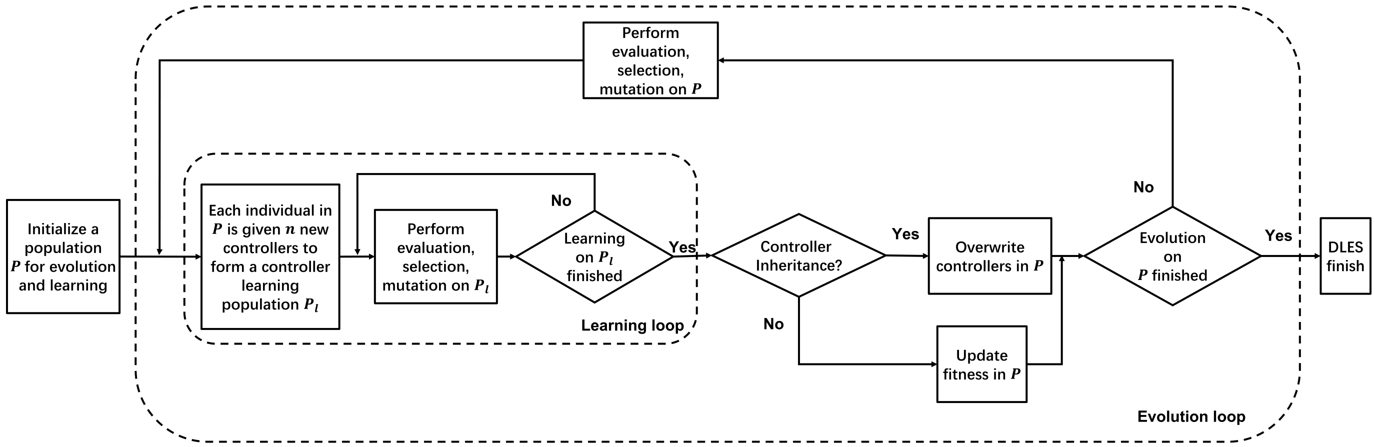

Fig. 5. Dual loop evolution structure (DLES): The outer evolution loop follows a joint evolution on morphology and controller routine, while the inner learning loop evolves controllers only. Details are given in Sections III-C1 and III-C2.

for both morphology and controller, providing the ability to encode various structures of morphology and controller. As noted above, new controllers reproduced from mutation may be a poor match for a new body. The DLES method aims to address this problem by applying a learning algorithm to the new controller to improve its performance via individual learning. The learned controller (represented by a CPPN) can overwrite the inherited controller in the offspring population (evolution and learning with controller inheritance). Alternatively, the learned fitness can be used to guide selection without updating the controller specified on the genome (evolution and learning without controller inheritance). An overview of DLES is illustrated in Figure 5. It includes two loops: an outer evolution loop and an inner learning loop. Pseudo code of DLES can be found in Algorithm 1.

1) Outer Evolution Loop: The outer evolutionary process in DLES evolves a population of individuals where each individual consists of a genome describing both the morphology and controller of a robot. Evolutionary operators (selection and reproduction) are applied on the individuals. An objective function evaluates the performance of an individual on a chosen task.

2) Inner Learning Loop: The learning loop optimises the controller to adapt to its morphology in order to accomplish a specific task. A learner is used which is also an evolutionary algorithm, following previous work [9]. A new set of CPPNs representing controllers are initialised for learning, containing the controllers from the population for evolution. HyperNEAT is used to optimise the controllers, where each controller is paired with the single morphology k from the population for evolution. At the end of this process, the task based fitness is assigned to each of the controllers. In the controller inherited case, the controller is over written by the best controller in the learning population. In the controller not inherited case, the learning stage influences selection by favouring individuals with morphologies that are more conducive to learning.

IV. Experiments

A. Experimental Protocol

A number of experiments were conducted to answer the following research questions:

1. To what extent does the inclusion of a learning loop that uses an intelligent learner improve performance when considering a range of tasks/environments while jointly evolving morphology and control?
2. When using an intelligent learning algorithm to make controllers adapt to morpholgies, to what extent are the results influenced by the inheritance of controllers?
3. To what extent is the proposed DLES approach capable of producing a diverse set of body-plans that adapt to a specific environment and/or task?

In order to answer question **1.**, experiments are conducted using the learning mechanism described in the previous section, compared against a simple baseline which only evolves the individual (no controller learning loop added). Question **2.** is addressed by comparing the two evolution and learning approaches (with and without controller inheritance) discussed above. Finally, by conducting experiments in three different environments aiming to understand whether the environment itself influences the morphological characteristics of the robots that evolve, and to what extent diverse robots are produced.

B. Tasks and Evaluation Scheme

1) Arenas and Tasks: DLES is applied in three arenas, the escape room, amphitheatre and escape amphitheatre shown in Figure 6. Each arena has different features in terms of the number of obstacles present, and the amphitheatre and escape amphitheatre also contain ‘steps’ that the robot must navigate. In each arena the goal is for a robot spawned at a starting position located in the middle of the arena (S) to reach a target located in the top right (T). The size of the arena is 2 m by 2 m. A beacon sensor placed at the top right corner of the arena marks the target position (T). The fitness function

Algorithm 1: Pseudo code of DLES.

Initialize evolution population P.
// Evolution of outer loop starts.
for $i \leftarrow evolution\ generation$ **do**
    // Learning of inner loop starts.
    for $j \leftarrow individuals\ in\ evolution\ population$ **do**
        Initialize a controller population for learning, including the controllers from P^j, with the total size of n.
        Replicate P^j for n times such that each P^j's controller is overwritten by a controller from the controller population to form the population for learning P_l.
        for $k \leftarrow learning\ generation$ **do**
            Perform evaluation, selection and mutation on the controller learning population P_l
        end
        // Learning finishes
        if *Evolution with learning without controller inheritance* **then**
            Update fitness scores for individual P^j by the best score achieved by P_l in learning
        end
        if *Evolution with learning with controller inheritance* **then**
            Update fitness scores for P^j by the best score achieved by P_l in learning
            Overwrite controller for P^j by the controller of the best individual P_l, if better performance is achieved.
        end
    end
    Perform evaluation, selection and mutation on P.
end
// Evolution finishes.

indicates distance from target after an evaluation time of 30 seconds. The simulation stops if a robot reaches the target position or the 30 seconds limit is reached. The final position of the individual is used to evaluate its performance.

The three arenas offer three different challenges to the individuals:

- Escape room: The starting position in this arena is surrounded by four walls with gaps at the corners. Only one gap enables sight of the beacon sensor located at target position via a sensor. Robots evolved in this arena need to have the ability to escape from the surrounding walls and find the target position.
- Amphitheatre: Different from the plain 2D locomotion in escape room, the amphitheatre has the challenge of 3D locomotion. Although there is no obstacle blocking the beacon sensor at target position, the challenge lies in finding the path to the target by overcoming the steps.
- Escape amphitheatre: The escape amphitheatre is a combination of the escape room and amphitheatre. Not only does an individual need to find a path out of the surrounding walls which have narrower gaps than the ones in escape room, but also the robot needs to have the ability to undertake 3D locomotion.

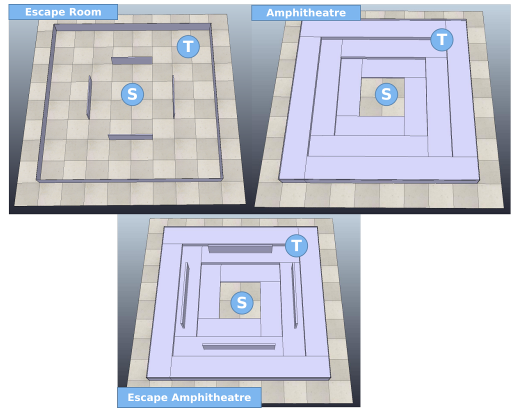

Fig. 6. Experimental arenas: The three arenas all have the same starting (S point) and target (T point) positions. Starting position is located at (0,0) for all three environments, and target positions are located at (0.75, 0.75).

2) Evaluation Scheme: The performance of an individual is evaluated by a fitness function that calculates normalized Euclidean distance between the final position of an individual and the target position in each arena. The fitness function used is shown in Equation 1.

$$fitness = \begin{cases} 1 - \dfrac{\|p_{target} - p_{final}\|}{distance_{max}} & , \dfrac{\|p_{target} - p_{final}\|}{distance_{max}} < 1 \\ 0 & , \dfrac{\|p_{target} - p_{final}\|}{distance_{max}} > 1 \end{cases} \quad (1)$$

Where p_{target} and p_{final} are the position of target and the final position of an individual respectively. $fitness$ should always be non-negative. $\frac{\|p_{target}-p_{final}\|}{distance_{max}} < 1$ means that an individual is doing effective locomotion, i.e., moving towards the target. $\frac{\|p_{target}-p_{final}\|}{distance_{max}} > 1$ implies that an individual is moving in the opposite direction of the target. In this case, fitness is set to 0. $distance_{max}$ is the distance between the start point and target point, $distance_{max} = \sqrt{(0.75-0)^2 + (0.75-0)^2} = 1.06$.

A metric is also defined to quantify morphological diversity within a population, to understand the extent to which DLES falls into local optima. This is motivated by previous research which has shown that morpho-evolution algorithms tend to quickly stagnate to a morphology for which it is easy to learn sub-optimal control, hindering innovation [28]. A morphological descriptor is defined as [wheel: number of wheels, sensor: number of senors, joint: number of joints, caster:number of casters]. It is represented by an encoding that assigns a code for each component combination. Each component can occur at most 16 times. Hence, a body-plan is encoded by 4 digits, representing the number of each component that the body-plan has ([number of wheels, number of sensors, number of joints, number of casters]), ranged by [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G]. For instance, a body-plan which has 1 wheel, 2 sensors, 5 joints and 10 casters can be encoded by 014A. Then, the diversity of a population can be described by a score of D:

$$D = \frac{N_d}{P} \tag{2}$$

where N_d is the number of different body-plans in the population. P is the total number of all possible body-plans, in this case: $P = 17^4 = 83521$. In previous work [8], a number of different diversity metrics were evaluated to find the metric described to provide an appropriate categorisation between robots: a more fine-grained metric that took account of placements of sensors etc., would result in a very large space of potential designs with little overlap. Furthermore, the investigations showed that small changes in placement do not have a significant impact on performance.

C. Experimental Settings

Two setups are considered. The first answers the three reserach questions posed above while the second is an ablation study to obtain more insight into parameter settings.

There are four parameters that define the computational budget for evolution, namely the size of the population in the outer evolution loop, the number of generations in the outer evolution loop, the size of the learning population in the inner loop and the number of learning generations in the inner loop. The same parameters are used for each of the escape room, amphitheatre and escape amphitheatre experiments, and are detailed in Table I specifying the detailed setup. This setup was selected after empirical investigations (see Section V-C) that suggested that a relatively small budgets of 10 generations was sufficient for convergence[1]. This concurs with other work in the field e.g. [24] which use a similar number of generations. It is also important to note that it is preferable to minimise the number of generations as much as possible when working in robotics particularly if the ultimate goal is to evolve in hardware due to the significant computational cost of such experiments. For the ablation study, five sets of parameter settings listed in Table II were considered, and used to investigate the weight of each parameter's effect on DLES.

TABLE I
EXPERIMENTAL SETUP OF DLES

Evolution population	50
Evolution generation	10
Learning population	25
Learning generation	10
Total individual evaluated	125500

The total evaluation number is calculated by the addition of evaluations of evolution and learning: total_evaluation = total_learning_evaluation + total_evolution_evaluation = Evolution population * Evolution generation * Learning population * Learning generation + Evolution population * Evolution generation = 50 * 10 * 25 * 10 + 50 * 10 = 125500.

[1]This contrasts with work in combinatorial optimisation in which much larger budgets are normally used

V. Results and Discussion

For each scenario, experiments are conducted over 20 replicates in order to provide meaningful statistical data. Fitness and diversity are measured in each experiment.

A. Evolution and Learning

The baseline EA experiment applies evolution to the population of morphologies without learning. The controller not inherited version of DLES applies learning then assigns the learned fitness to the individual while the controller inherited scheme overwrites the genome of each offspring with the learned controller. In this section, the three schemes are evaluated on the three environments, namely escape room, amphitheatre and escape amphitheatre. Results are shown in Figure 7 and Figure 8.

The first column of Figure 7 plots the fitness associated with the individuals of the outer loop over each generation for each experimental scheme. Any individual with fitness around 0.9 or higher is considered to be a successful individual (close enough to the target). There are two main observations: (1) evolution + learning (with inheritance) outperforms the other methods, and the effect becomes more apparent as the complexity of the task increases; (2) using learning without inheritance does not improve performance when compared to the baseline of evolution only. The latter point contrasts to some previous work, e.g. [12] which clearly demonstrates a strong Baldwin effect, i.e. finding that selecting for controllers that are more capable of learning improves performance. Suggesting that the framework used in [12] evolves robots in a simpler morphological design-space, consisting only of articulated 3D rigid parts connected via motor actuated hinge joints. In contrast, this framework permits free-form skeletons and a variety of actuators (wheels and/or joints) and sensor types.

Figure 8 compares the improvement per generation of the performance of the evolution+learning (with inheritance) method to each of the other two methods, where improvement is calculated as the fitness score of former approach minus the compared approach. This clearly demonstrates that in the most complex arena (escape amphitheatre) the magnitude of the improvement increases over generations while in the most simple case, the magnitude of the improvement gained is smaller and stays roughly constant. It seems clear that the evolutionary process is boosted by inheriting the learned controller in complex domains, rather than just selecting for controllers that have the capacity to learn. The magnitude of improvement justifies the additional cost associated with learning, for example approximately doubling the best fitness obtained compared to the no-learning method.

The middle column of Figure 7 shows the progress of the inner learning loop, in which there are 10 learning evolutionary

TABLE II
EXPERIMENTAL SETTINGS FOR PARAMETER STUDY

	Outer loop (evolution) population size	Outer loop generations	Learning population size	Learning loop generations
Setup 1	50	10	25	10
Setup 2	100	10	25	10
Setup 3	50	20	25	10
Setup 4	50	10	50	10
Setup 5	50	10	25	20

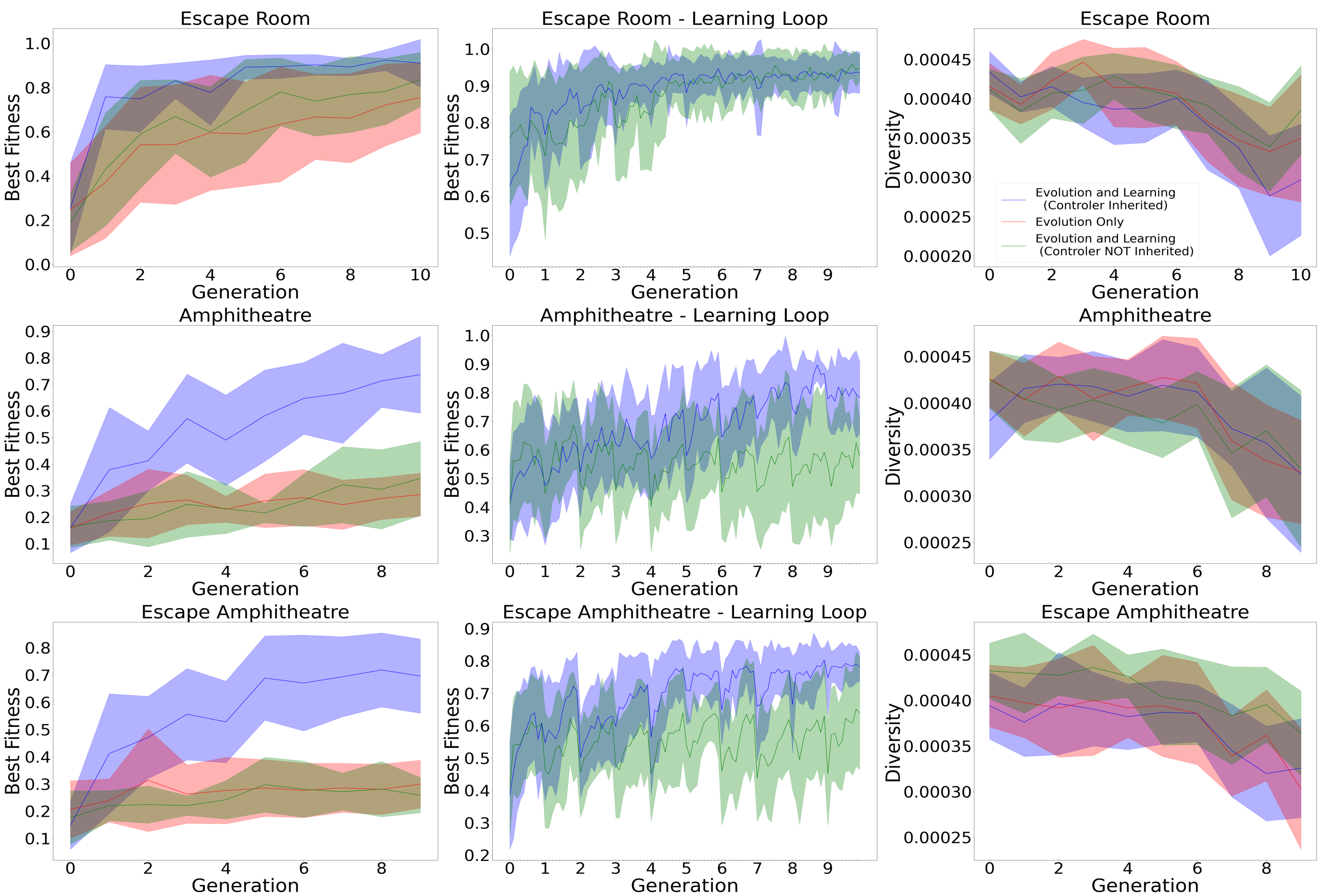

Fig. 7. Plots of evolution and learning performance: Three approaches: evolution and learning with controller inherited, evolution and learning without controller inherited and evolution only, in escape room, amphitheatre and escape amphitheatre: the best fitness in both of the evolution (column 1) and learning loops (column 2) and diversity curves are plotted. Best fitness plots show the mean of the fitness of the best individual per generation over 20 replicates (solid line), and the standard deviation. Diversity describes the morphological variety of the population per generation, showing mean diversity (solid line) and standard deviation over 20 replicates.

generations for each generation of the outer loop. There is a statistically significant difference between the two methods at generation 10, with the learning with inheritance method outperforming the learning (no inheritance) approach. Again the difference in performance become clearer as the difficulty of the task increases.

The final column shows the change in the diversity metric measured in the outer evolution loop. This illustrates the change in diversity of body-plans over time of the three approaches, calculated using the metric described in Section IV-B2. The morphological diversity of the three approaches are very similar, indicating the performance difference is mainly associated with the difference in learning approaches rather than by morphological differences.

In summary, in all environments, the addition of a guided learning mechanism that includes inheritance improves performance, but does not increase the morphological diversity of the population. Significant difference in performance is observed even after one generation with the learning with inheritance method, indicating that controllers benefit from learning at very early stage of evolution. As the difficulty of the environments increases, the advantage of evolution and

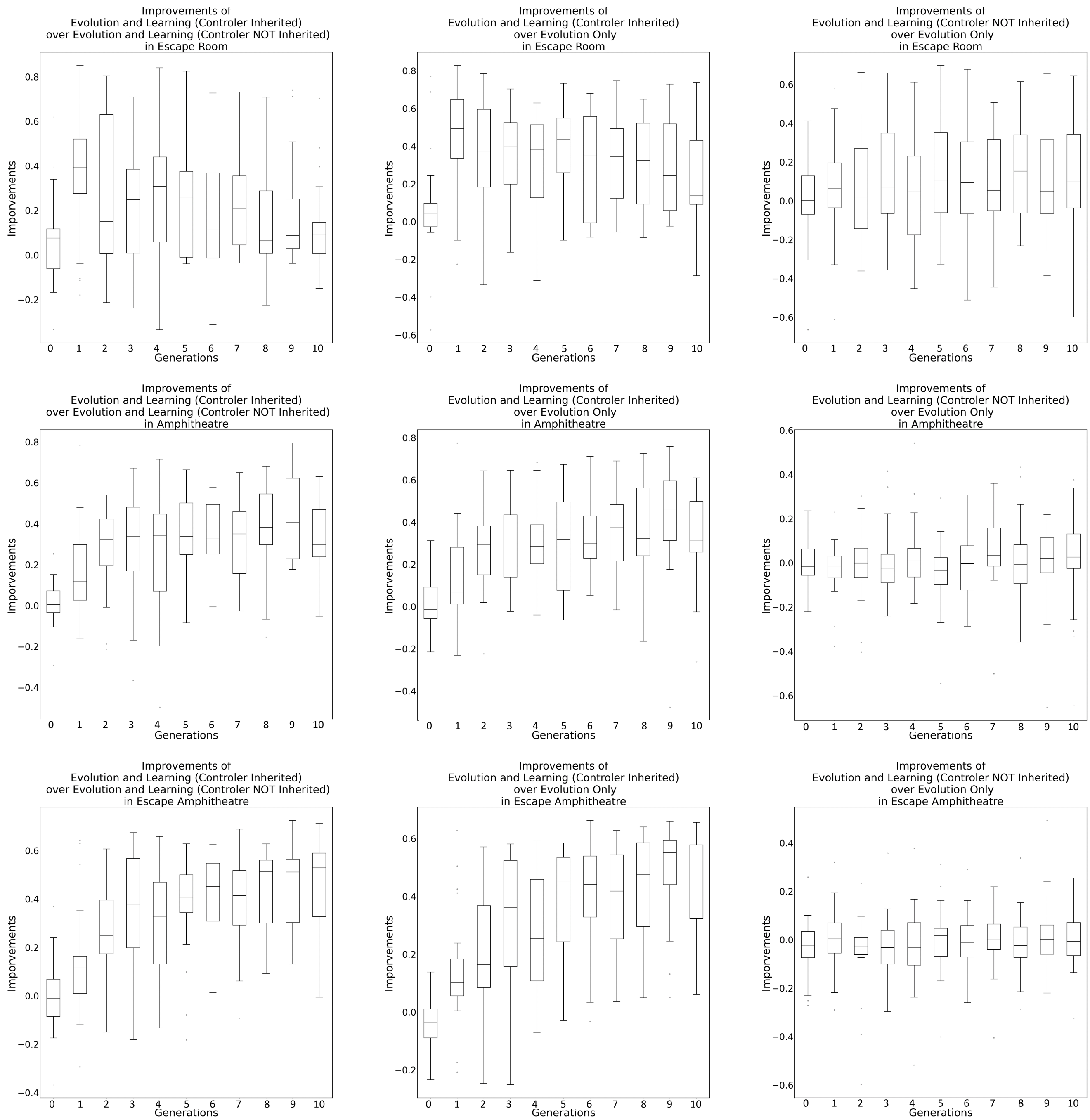

Fig. 8. The improvements of evolution and learning approach with controller inheritance over evolution and learning approach without controller inheritance and evolution only. The improvement is calculated by the fitness score of former approach minus the latter approach. For example, improvement of evolution and learning (controller inherited) over evolution and learning (controller NOT inherited) is the fitness of evolution and learning (controller inherited) minus evolution and learning (controller NOT inherited) at each generation for the 20 replicates.

learning with controller inherited become stronger. Overall, all of the evidence shows that DLES (evolution and learning with controller inherited) is the superior method.

B. An analysis of evolved robots

Examples of individuals generated in each the three scenarios are presented in Figure 9. A demo video of evolved robots working in all three scenarios can be found on https://doi.org/10.6084/m9.figshare.23735742.v2.

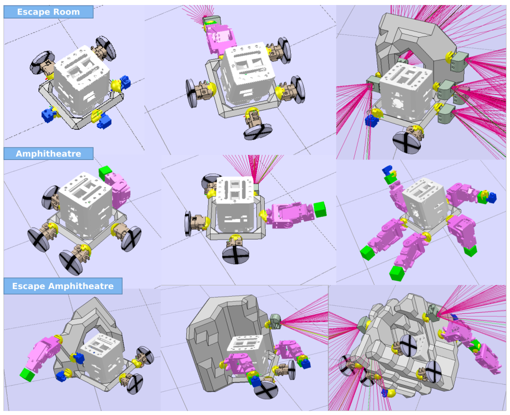

Fig. 9. Robots generated in various scenarios: First, second and third row are robots generated in escape room, amphitheatre and escape amphitheatre respectively.

In the escape room, robots need to have the ability to make turns to move around the surrounding walls. Joints or casters attached on sides can help to change the direction of motion in order to avoid being stuck by walls. Since the floor is flat in escape room, wheels, joints and casters can all be used to drive effective 2D motion. In the amphitheatre, joints are more important for locomotion as there are steps requiring an individual to have the ability to overcome height changes in its path. Joints are used to tilt the body when the locomotion is driven by wheels or casters. Joints can also be used as legs to drive locomotion directly as well. In the escape amphitheatre, the challenges in both escape room and amphitheatre exist. Robots need joint to provide 3D locomotion ability and casters/wheels to move around surrounding walls.

Figure 10, Figure 11 and Figure 12shows the component distribution of individuals with fitness greater than 0.3 in each environment for evolution and learning with controller inherited, evolution and learning without controller inherited and evolution only. A fitness value higher than 0.3 is considered to be a 'working individual' as the robot is moving towards the target in the right direction.

It can be seen that when the controller is inherited, body-plans gradually adapt to different scenarios. In the escape room, all of the components can contribute towards providing effective functionality. For instance, wheels, joints and casters can provide 2D locomotion, sensors can help to find the target, while joints and casters can help to get around the walls. Thus there is a good deal of flexibility in terms of finding a suitable morphology, which makes the evolutionary process less challenging. Also, due to the fact that robot always starts in the same place facing in the same direction, it might be possible to generate a behaviour that gets to the target with pure luck for simple arena such as escape room. In the harder arenas, such as the amphitheatre and the escape amphitheatre, the need for other types of components starts to become apparent. In the controller inherited approach, it is obvious that senors, joints and casters are more often used in the designs than in the other two cases (evolution and learning without controller inheritance and evolution only).

The results imply that the mechanism which uses evolution and learning with inheritance facilitates the emergence of morphologies that are better adapted to the environment in which a task is learned. The results can be interpreted as demonstrating the emergence of morphological intelligence [29], i.e. in which the approach produces body-plans with components that can overcome specific challenges in each arena.

C. Parameter Influence: Evolution and Learning Budgets

In Section V-A, evolution and learning with controller inherited approach of DLES has shown superior performance. In this section, the contribution of each of the parameters of evolution and learning with controller inherited approach of DLES are studied. Detailed parameters are listed in Table II. Experiments with each setup were replicated five times in the ablation study (in contrast to the experiments in the previous section which were repeated 20 times for statistical significance). All experiments are conducted in the escape amphitheatre since it is the most difficult scenario for robots to be successful.

The parameters studied are listed in Table II. For each setup in Table II, one parameter is changed while keeping all the other parameters constant. Setup 1 and setup 4 study the effect of changing the size of the outer evolutionary loop population, setup 4 and setup 5 study the effect of changing the number of generations in the outer loop, setup 2 and setup 3 study the effect of changing the size of the learning population, and setup 1 and setup 2 study the effect of changing the number of learning generations in the inner loop. The results are shown in Figure 13, Figure 14, Figure 15 and Figure 16.

From the figures, it can be seen that the benefit of increasing the computational budget (e.g. via increasing the outer loop population size, number of outer loop generation, learning population size and number of learning generations) rapidly diminishes. The final experimental setup used (50 evolution population size, 10 evolution generations, 25 learning population size and 10 learning generations) is determined by these results, and concurs with similar results found by others, e.g. [5], [9], [24].

VI. Conclusion

In this paper, a dual loop evolution structure (DLES) for robot evolution with learning in a rich morphological space is proposed. DLES enables the evolution of robots that exhibit

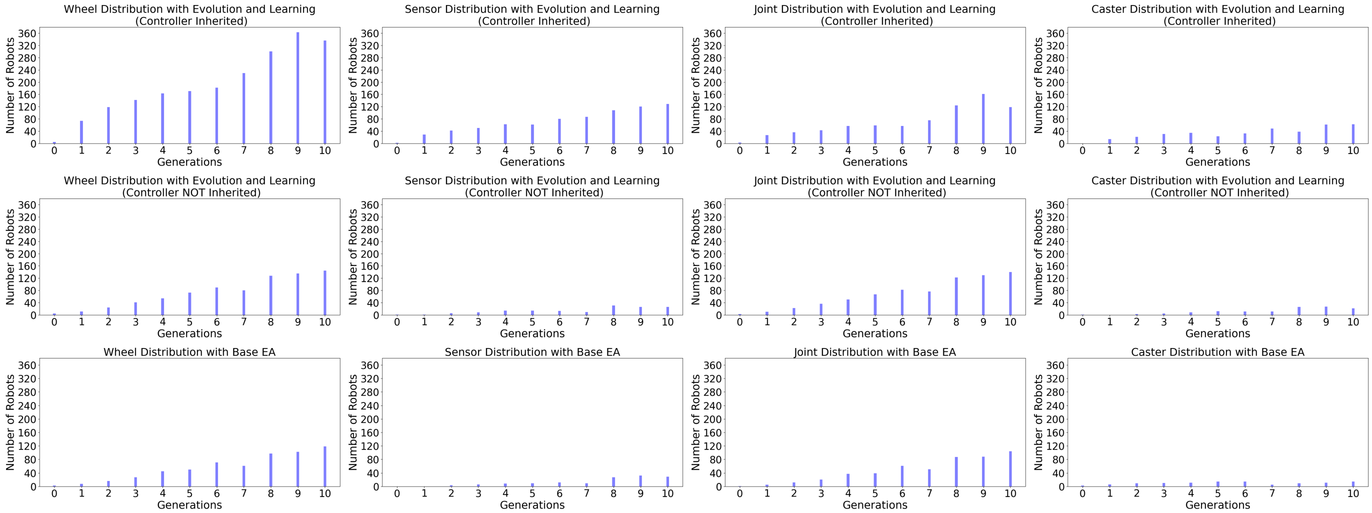

Fig. 10. Component distribution of individuals with fitness greater than 0.3 in the escape room. The fist row of plots show the component distribution for evolution and learning with controller inherited. The second row of plots are the distribution for evolution and learning without controller inherited. The third row of plots are the distribution for evolution only. The threshold of 0.3 fitness value is applied considering individuals that function properly.

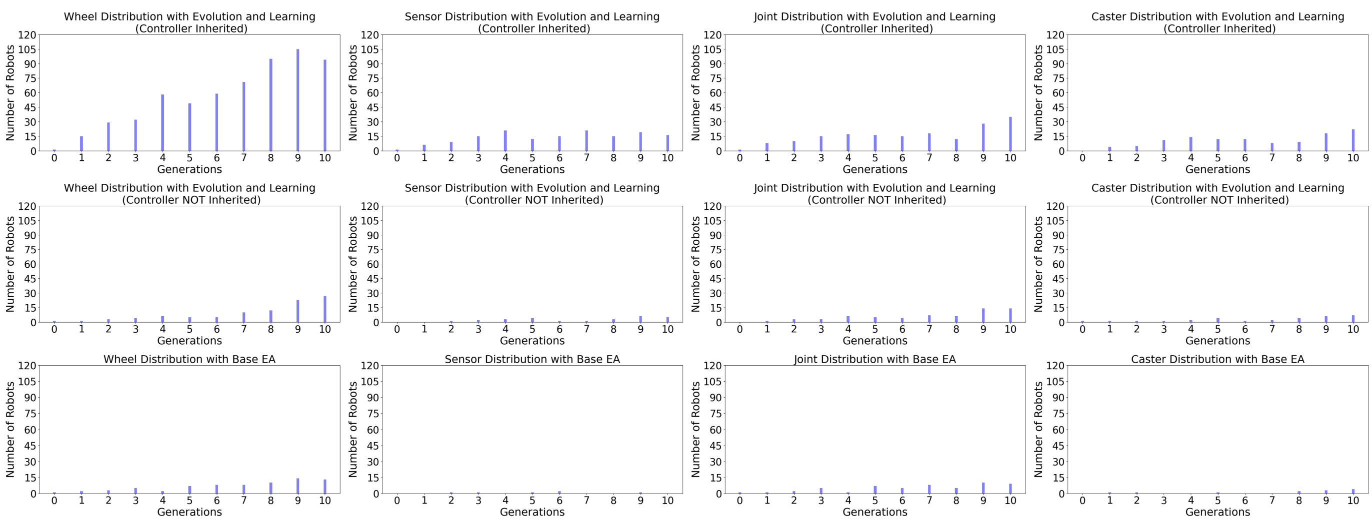

Fig. 11. Component distribution of individuals with fitness greater than 0.3 in the amphitheatre. The fist row of plots show the component distribution for evolution and learning with controller inherited. The second row of plots are the distribution for evolution and learning without controller inherited. The third row of plots are the distribution for evolution only. The threshold of 0.3 fitness value is applied considering individuals that function properly.

a diverse array of forms adapted to a specific environment by augmenting an evolutionary loop with a learner. Specifically three approaches are compared on three locomotion tasks: evolution and learning with controller inherited, evolution and learning without inheriting the controller, and evolution only. The results show that evolution and learning with inheritance of the controller results in more efficient and more effective performance than the other two approaches. We argue that augmenting evolution with individual learning is essential when trying to evolve robots in complex morphological spaces with closed loop control due to the challenges in matching a neural controller to a new morphology. It appears that inheriting the learned controller is mandatory if there is to be a benefit from the additional cost associated with learning. In this respect, the results concur with previous work e.g. [11] that also found a benefit in inheriting learned controllers, rather than just selecting for controllers that are capable of being improved. Similarly, [11] used a design-space that evolved robots in simulation that could also be physically created. However, it is important to note that other work that evolved in a simpler design-space that is only ever simulated (e.g. [12]) demonstrated that while learning is important, inheritance of the learned controller is not necessary (i.e. a Baldwin effect is observed). We postulate that in very complex design spaces, inheriting the learned controller effectively provides a mechanism to enable evolution to proceed more rapidly, by directly

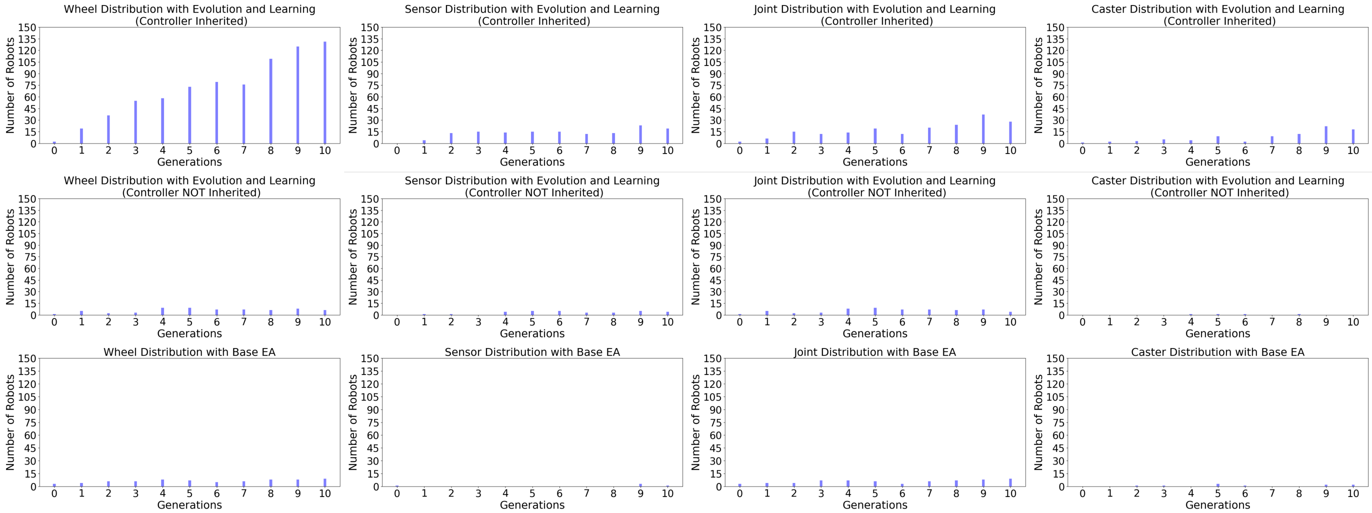

Fig. 12. Component distribution of individuals with fitness greater than 0.3 in the escape amphitheatre. The fist row of plots show the component distribution for evolution and learning with controller inherited. The second row of plots are the distribution for evolution and learning without controller inherited. The third row of plots are the distribution for evolution only. The threshold of 0.3 fitness value is considering individuals that function properly.

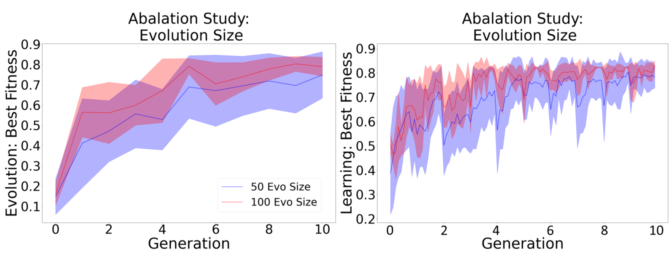

Fig. 13. DLES with different settings: the effect of changing outer evolution population size.

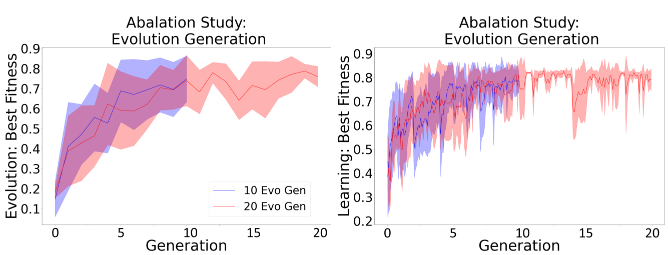

Fig. 14. DLES with different settings: the effect of changing outer evolution generation.

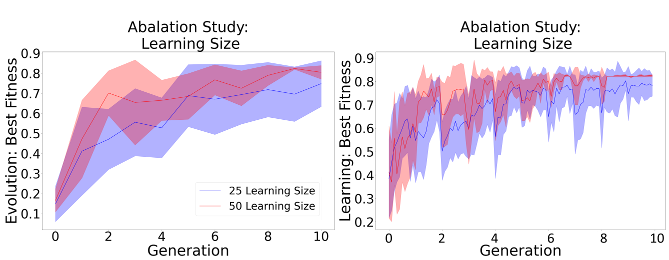

Fig. 15. DLES with different settings: the effect of changing inner learning population size.

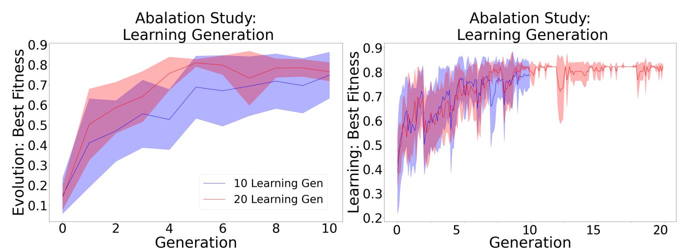

Fig. 16. DLES with different settings: the effect of changing inner learning generation.

influencing selection of high performing learned controllers that can be passed to future generations. It was also observed that the evolution and learning with inheritance mechanism enables the population to rapidly adapt its ‘morphological preference’ over time to match the environment, i.e. in its selection of suitable sensors and actuators, in contrast to the other approaches. This might be viewed as the emergence of morphological intelligence [29]. The components of the framework are general enough that the same framework can be used to evolve other types of robotic systems. For example, the CPPN representation used here to represent bodies and brains could be applied to a completely modular system (where the skeleton is formed from choosing between a set of pre-formed parts as in [11] and also to soft robotics systems (e.g. [30]).

An obvious extension to this work would be to consider how to further augment the learning process with knowledge learned in past generations across populations. In this work the learner is seeded with a single inherited controller, but this could be adapted to make use of additional information, i.e. taking inspiration from some of the literature in the cultural learning field [31]. Determining what information is useful to inform future generations remains a topic for research. Finally, the work is motivated by the desire to evolve robots that can be physically built to conduct tasks in the real world. Therefore we intend to evaluate the best robots evolved in simulation in order to assess the reality gap between simulated and physical versions. As first noted in [7], we expect that an additional period of individual learning will be necessary for every physical robot built to cross an inevitable reality gap.

ACKNOWLEDGEMENTS

This work is funded by Engineering and Physical Sciences Research Council (EPSRC) Autonomous Robot Evolution project, EP/R03561X, EP/R035733, EP/R035679, UK, and the Vrije Universiteit Amsterdam, NL, and Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), China. The authors would like to acknowledge the Institute for Safe Autonomy (ISA) [2].

REFERENCES

[1] K. Sims, "Evolving 3D morphology and behavior by competition," *Artificial Life*, vol. 1, no. 4, pp. 353–372, 1994.

[2] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, no. 974-978, 2000.

[3] S. Kriegman, D. Blackiston, M. Levin, and J. Bongard, "A scalable pipeline for designing reconfigurable organisms," *Proceedings of the National Academy of Sciences*, vol. 117, no. 4, pp. 1853–1859, 2020.

[4] L. Brodbeck, S. Hauser, and F. Iida, "Morphological evolution of physical robots through model-free phenotype development," *PloS one*, vol. 10, no. 6, p. e0128444, 2015.

[5] M. Jelisavcic, M. De Carlo, E. Hupkes, P. Eustratiadis, J. Orlowski, E. Haasdijk, J. E. Auerbach, and A. E. Eiben, "Real-world evolution of robot morphologies: A proof of concept," *Artificial life*, vol. 23, no. 2, pp. 206–235, 2017.

[6] R. Moreno and A. Faiña, "Emerge modular robot: a tool for fast deployment of evolved robots," *Frontiers in Robotics and AI*, vol. 8, p. 699814, 2021.

[7] M. F. Hale, E. Buchanan, A. F. Winfield, J. Timmis, E. Hart, A. E. Eiben, M. Angus, F. Veenstra, W. Li, R. Woolley *et al.*, "The are robot fabricator: How to (re) produce robots that can evolve in the real world," in *The 2019 Conference on Artificial Life*. MIT Press, 2019, pp. 95–102.

[8] E. Buchanan, L. K. Le Goff, W. Li, E. Hart, A. E. Eiben, M. De Carlo, A. F. Winfield, M. F. Hale, R. Woolley, M. Angus *et al.*, "Bootstrapping artificial evolution to design robots for autonomous fabrication," *Robotics*, vol. 9, no. 4, p. 106, 2020.

[9] L. K. Le Goff, E. Buchanan, E. Hart, A. E. Eiben, W. Li, M. De Carlo, A. F. Winfield, M. F. Hale, R. Woolley, M. Angus *et al.*, "Morpho-evolution with learning using a controller archive as an inheritance mechanism," *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[10] A. E. Eiben and E. Hart, "If it evolves it needs to learn," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1383–1384.

[11] K. Miras, M. De Carlo, S. Akhatou, and A. Eiben, "Evolving-controllers versus learning-controllers for morphologically evolvable robots," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 2020, pp. 86–99.

[12] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[13] L. K. Le Goff, E. Buchanan, E. Hart, A. E. Eiben, W. Li, M. de Carlo, M. F. Hale, M. Angus, R. Woolley, J. Timmis *et al.*, "Sample and time efficient policy learning with cma-es and bayesian optimisation," in *ALIFE 2020: The 2020 Conference on Artificial Life*. MIT Press, 2020, pp. 432–440.

[14] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic programming and evolvable machines*, vol. 8, no. 2, pp. 131–162, 2007.

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

[16] J. Baldwin, "A new factor in evolution," *The american naturalist*, vol. 30, no. 1, pp. 441–451, 1896.

[17] A. Eiben, N. Bredeche, M. Hoogendoorn, J. Stradner, J. Timmis, A. Tyrrell, and A. Winfield, "The triangle of life: Evolving robots in real-time and real-space," in *Proc. of the 12th European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)*, P. Lio, O. Miglino, G. Nicosia, S. Nolfi, and M. Pavone, Eds. MIT Press, 2013, pp. 1056–1063.

[18] F. Veenstra and K. Glette, "How different encodings affect performance and diversification when evolving the morphology and control of 2d virtual creatures," in *ALIFE 2020: The 2020 Conference on Artificial Life*. MIT Press, 2020, pp. 592–601.

[19] S. Kriegman, N. Cheney, and J. Bongard, "How morphological development can guide evolution," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[20] J. Hiller and H. Lipson, "Automatic design and manufacture of soft robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 457–466, 2012.

[21] E.-L. Ruud, E. Samuelsen, and K. Glette, "Memetic robot control evolution and adaption to reality," in *2016 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2016, pp. 1–7.

[22] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[23] L. K. L. Goff and E. Hart, "On the challenges of jointly optimising robot morphology and control using a hierarchical optimisation scheme," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 1498–1502.

[24] M. Jelisavcic, K. Glette, E. Haasdijk, and A. Eiben, "Lamarckian evolution of simulated modular robots," *Frontiers in Robotics and AI*, vol. 6, p. 9, 2019.

[25] M. F. Hale, M. Angus, E. Buchanan, W. Li, R. Woolley, L. K. Le Goff, M. De Carlo, J. Timmis, A. F. Winfield, E. Hart *et al.*, "Hardware design for autonomous robot evolution," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 2140–2147.

[26] J. E. Auerbach and J. C. Bongard, "Evolving complete robots with cppn-neat: the utility of recurrent connections," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 1475–1482.

[27] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial life*, vol. 15, no. 2, pp. 185–212, 2009.

[28] N. Cheney, J. Bongard, V. SunSpiral, and H. Lipson, "Scalable co-optimization of morphology and control in embodied machines," *Journal of The Royal Society Interface*, vol. 15, no. 143, p. 20170937, 2018.

[29] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2007.

[30] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson, "Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding," *ACM SIGEVOlution*, vol. 7, no. 1, pp. 11–23, 2014.

[31] E. Hart and L. K. Le Goff, "Artificial evolution of robot bodies and control: on the interaction between evolution, learning and culture," *Philosophical Transactions of the Royal Society B*, vol. 377, no. 1843, p. 20210117, 2022.

[2] Website: https://www.york.ac.uk/safe-autonomy/