White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of York.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/218844/

Version: Accepted Version

Article:

Buchanan Berumen, Edgar, Le Goff, Léni, Hart, Emma et al. (Accepted: 2024) Towards a Unified Framework for Software-Hardware Integration in Evolutionary Robotics. MDPI Robotics. ISSN: 2218-6581 (In Press)

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Article

Towards a Unified Framework for Software-Hardware Integration in Evolutionary Robotics

Edgar Buchanan [1]*, Léni K. Le Goff [2], Matthew F. Hale [4], Emma Hart [2], Agoston E. Eiben [3], Matteo De Carlo, Mike Angus[1], Robert Woolley [1], Jon Timmis[5], Alan F. Winfield [6] and Andy M. Tyrrell [1,]

[1] School of Physics, Engineering and Technology, University of York, UK
[2] School of Computing, Edinburgh Napier University, UK
[3] Department of Computer Science, Vrije Universiteit Amsterdam, NL
[4] Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK
[5] Department of Computer Science, Aberystwyth University, UK
[6] Bristol Robotics Laboratory, University of the West of England, Bristol, UK
* Correspondence: edgar.buchanan@york.ac.uk

Abstract: The discrepancy between simulated and hardware experiments, the reality gap, is a challenge in evolutionary robotics. While strategies have been proposed to address this gap in fixed-body robots, they are not viable when dealing with populations and generations where the body is in constant change. The continual evolution of body designs necessitates the manufacturing of new robotic structures, a process that can be time-consuming if carried out manually. Moreover, the increased manufacturing time not only prolongs hardware experimental durations but also disrupts the synergy between hardware and simulated experiments. Failure to effectively manage these challenges could impede the implementation of evolutionary robotics in real-life environments. The Autonomous Robot Evolution project presents a framework to tackle these challenges through a case study. This paper describes the main three contributions of this work: Firstly, it analyses the different reality gap experienced by each different robot or the heterogenous reality gap. Secondly, it emphasizes the importance of automation in robot manufacturing. And thirdly, it highlights the necessity of a framework to orchestrate the synergy between simulated and hardware experiments. In the long term, integrating these contributions into evolutionary robotics is envisioned to enable the continuous production of robots in real-world environments.

Keywords: Evolutionary Robotics; Evolution of things; Automation; Software-hardware Synergy; Reality Gap

Citation: Buchanan, E.; Le Goff, L. K.; Hale, M. F.; Hart, E.; Eiben, A. E.; De Carlo, M.; Angus, M.; Woolley, R.; Timmis, J.; Winfield, A. F.; Tyrrell, A. M. Towards a Unified Framework for Software-Hardware Integration in Evolutionary Robotics. *Robotics* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

Copyright: © 2024 by the authors. Submitted to *Robotics* for possible open access publication under the terms and conditions of the Creative Commons Attri- bution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

One of the ultimate goals of evolutionary robotics (ER) is the transition from evolutionary computation, or *digital evolution*, to the evolution of things, or *physical evolution* [1]. In *physical evolution*, entire robotic ecosystems will run autonomously with minimal human intervention, and robots will evolve in real-time and real space adapted to their task and surroundings. Over the last couple of decades, the ER field has made significant strides towards this goal, from the early beginnings with the evolution of controllers in physical robots [2–4] in the 90s, to the joint evolution of body designs and controllers, morpho-evolution, of robots [5–7], and to recent years, where small populations of robots are evolved in hardware [8–10]. Despite this progress, this paper highlights three challenges in ER that need to be addressed before the transition to the continuous production of evolved robots in real-world environments: *the software-hardware synergy balance*, *the heterogenous reality gap*, and *the continuous production of robots and the importance of automation*.

The software-hardware synergy balance: Even though *physical evolution* can be powerful by itself, this process can be further augmented with *digital evolution*. Firstly, due to the rapid evaluation in *digital evolution*, a wider range of diverse robot designs and controllers can be explored [11]. Secondly, damaging physical components can be prevented when the robots are firstly simulated [3,12]. Thirdly, waste of resources can be avoided when fabricating sub-optimal robots [3,12]. In most of the work in the literature, the results from the *digital evolution* are fabricated into physical robots [10,13–18], however, to close the software-hardware feedback loop, information needs to be propagated back to the *digital evolution*. Since these processes, *digital and physical evolution*, work at different time scales, the question arises of when to synchronize them and how to design the robot selection operator. The issue of long-time evaluations in *physical evolution* has been raised in previous literature [1,19–21].

The heterogenous reality gap: The resulting discrepancy between experiments in simulation and experiments in hardware, or the reality gap, has been explored with solutions proposed [20–23]. These solutions often assume that the designs of the robots are fixed and will not change. However, this is not the case when *digital and physical evolution* are integrated, in which case the body designs constantly change. The reality gap will change for every single different robot design and behaviour that evolves [16], and if not handled properly, this can impact the number of evaluations and the selection process during evolution. The question arises of how to cope with this constant change in the reality gap.

The continuous production of robots: The process of manually fabricating robots can be time-consuming and this impacts robot production throughput. In most of the experiments with physical robots in ER, the robots are manually constructed, with a handful of exceptions [8,18,24]. In contrast to conventional manufacturing automation, where the same product is produced repeatedly [25], in *physical evolution*, the product (in this case, robots) is constantly changing, and the automation facility should be flexible enough to handle the different robot designs evolved. Realistically, there will always be manufacturability constraints, which will also affect the types of robots that can be manufactured, also known as the *viable phenotype space* [26]. This then becomes a chicken-and-egg problem of how to design an autonomous fabrication system for relatively diverse robots.

The previous three challenges are interconnected, with their outcomes influencing one another, as shown in Figure 1. The selection and quantity of physical robots, along with their body designs, impact the autonomous robot fabrication process, which in turn determines the production throughput. This throughput affects the balance between the number of robots evaluated in hardware and those evaluated in simulation. The constantly shifting reality gap between robots will also impact the final score, which, in turn, affects the robot selection process.

The Autonomous Robot Evolution project[1] [24] is presented in this paper as a case study to explore in more detail the previous three challenges. Preliminary strategies to address these challenges are proposed along with initial results. Additionally, this paper offers three novel key contributions. First, it presents the full implementation of an autonomous robot fabrication process. Second, robots with different designs are physically evaluated and compared to their digital counterparts to assess the reality gap. Finally, it introduces the first implementation of an integrated *digital and physical evolution* system.

The structure of the paper is as follows. Section 2 describes the work in the literature that addresses the previous challenges. Section 3 illuminates the importance of these challenges and proposes some solutions to address them. Section 4 summarizes the work in this paper and describes further work.

2. Related Work

This section will describe approaches to address the challenges introduced: *the software-hardware synergy balance*, *the heterogeneous reality gap*, and *the continuous production of robots*.

[1] https://www.york.ac.uk/robot-lab/are/

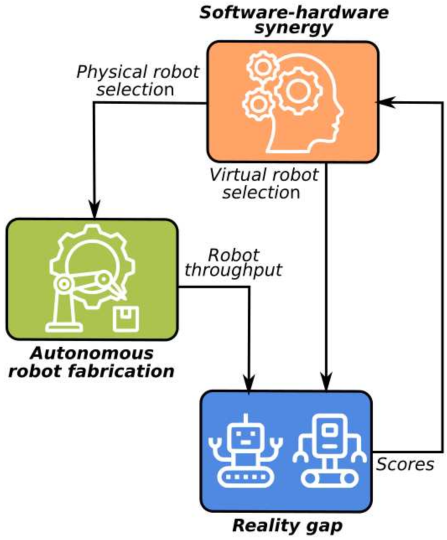

Figure 1. This diagram illustrates a minimal framework that integrates both *physical and virtual evolution*. Robot selection occurs within the software-hardware block. Physical robots are autonomously fabricated by a facility focused on minimizing fabrication time to maximize throughput. After fabrication, the robots are evaluated, with the reality gap information fed back into the software-hardware synergy for further optimization

2.1. The software-hardware synergy balance

The work in the literature regarding software-hardware synergy can be categorized as *simulation-only*, *sim-to-real*, *hardware-only* and *sim-and-hardware*. Most of the work on morpho-evolution in ER has been carried out as *simulation-only*, where robots are evolved in a simulated environment and never tested in hardware. This is because *digital evolution* offers the advantage of rapidly evaluating a wide range of diverse robot designs and controllers without the risk of damaging physical components, and it avoids wasting resources on fabricating suboptimal robots [3,11,12]. However, evaluating robots in hardware helps validate simulation results, and this information can significantly impact the performance of the algorithms.

Valuable feedback has been provided by hardware experiments conducted by authors in the literature who constructed and evaluated resulting robots from simulation in *sim-to-real*. The work by [13] is foundational, as it was the first instance where morphological evolution was conducted in simulation, followed by the fabrication, evaluation, and comparison of the resulting population with their simulated counterparts. This work highlighted discrepancies in behaviour, particularly differences in speed, which were attributed to inaccuracies in the simulation's friction model. Subsequently, authors such as [16–18,27] highlighted that discrepancies in results were produced by inaccurate friction coefficients. [14] suggested that, when conducting experiments, the actuation in simulation should be high enough to ensure that the same actuation in the physical robot is able to break the static friction, or the robot will be unable to move. Inappropriate modelling of limb-to-limb collision is suggested as a source of the reality gap by [16]. [17] mentioned morphological differences and inaccurate actuation between simulation and hardware, while [18] noted that broken connections in hardware introduce behavioural differences. All this prior knowledge could have been applied to evolve better robot designs, either by adjusting parameters in the simulator, as done by the authors in [10] (see more in *sim-to-real* below), or by modifying the selection operators in the evolutionary process (discussed further in Section 4).

This feedback can be dismissed entirely when experiments are conducted as *hardware-only*, where experiments are carried out in hardware with minimal contribution from simulation. The trade-off is that these experiments are time-demanding [1,19–21]. [8]

conducted experiments where a hundred robots were evolved across ten generations in hardware. The results indicate an improvement in performance after each generation. Nevertheless, the authors suggest that the combination of simulation and hardware will help to produce better and more viable solutions in hardware.

As far as the authors are aware, there has been only one attempt to integrate morpho-evolution in software and hardware (*sim-and-hardware*). [10] evolved robots with a pipeline approach where robots were evolved first in simulation and then manually assembled the best-performing robots. The performance of these robots was measured; this is referred to as the first pass. Then, the constraints in the simulation were manually updated, the initial population was reinitialized with the best-performing robots, and the robots were evolved again in simulation. A selected group of robots was then assembled physically; this is referred to as the second pass. The authors discovered that the behaviour of the robots in simulation better matched the behaviour seen with the physical robots after the second pass.

In conclusion, even though experiments in ER have been carried out in simulation and hardware, there is a gap in how experiments from these two domains can be integrated to exploit their benefits to create even better robot designs. This paper proposes an integration method and shows preliminary results.

2.2. The heterogenous reality gap

The reality gap, or the discrepancy between simulation and hardware results, has been a constant challenge, impacting robot performance and necessitating iterative adjustments in both simulation and hardware to minimize this gap [22]. Numerous approaches have been proposed in the literature to address the reality gap [20,21,23,28–31]. These approaches typically assume a static robot design; however, this assumption does not hold true in ER, where robot morphologies are constantly changing. The reality gap can be reduced by improving the simulator's fidelity. However, higher fidelity requires more computational resources, resulting in longer evaluation times, which is suboptimal in evolutionary robotics, where numerous evaluations are necessary for each process. Additionally, calibrating a simulator to accommodate a wide range of robot designs, each interacting with the environment in unique ways, becomes exponentially more challenging.

Experiments conducted in hardware within the ER domain, which involve evolving morphologies, have highlighted the existing reality gap [13,14,16–18]. However, the explanations provided by the authors do not clarify whether the reality gap is consistent across all robots or varies with each evolved robot. [16] described how the reality gap varied from robot to robot, possibly linked to both the morphology and behaviour of the robot. For example, robots exhibiting dragging behaviours experienced a higher reality gap than those performing gait behaviours. Another example is the collision between limbs, which is also linked to morphology. The reality gap variation between evolved robots is referred to in this paper as the *heterogenous reality gap*

A key distinction between the platforms described in [13,14,16–18,32] and the ARE platform is that evolved robots can have various configurations of different components, such as sensors, joints, and wheels. This is particularly important because it can amplify the effects of the heterogeneous reality gap, as each evolved robot interacts with the environment in unique ways. For example, some robots may walk, drag themselves, or roll across the terrain.

[12] introduced a method to minimize the reality gap for robots with varying numbers of limbs. In this approach, a repertoire of behaviours is generated in simulation. This repertoire is then used to train a model, which is subsequently deployed in the physical robot. The robot switches between behaviours to accommodate its current morphology.

In conclusion, evidence suggests that the reality gap varies between different robots when evaluated in hardware due to differences in their designs and behaviours. It is crucial to account for this varying reality gap when selecting robots for *physical evolution*, as it can significantly impact the evolutionary process, a topic that will be explored in more detail in later sections of this paper.

2.3. The continuous production of robots

In *physical evolution*, relatively large numbers of robots will be produced, and manually constructing these robots will be physically demanding. For this reason, the process of fabricating evolved robots needs to be automated. Automation, or the use of technology to perform tasks with minimal human intervention [25], has been widely deployed in industry to perform repetitive actions. The philosophy of fixed and programmable automation is to design the process around the product that is to be manufactured, changing the automation process only when the product changes. However, in ER, the products, or robots, are in constant change, and therefore flexible (soft) automation is required. Creating such a system is challenging because the restrictions from the system will impact the evolutionary space, and therefore the encodings should encapsulate this information, as discussed by [26]. This might be one of the reasons that in most literature, evolved robots are manually constructed [9,10,13,15–17,33].

The work by [8,34] was the first to automate the process of fabricating evolved robots. In this approach, a robotic arm assembles robots by performing handling operations such as rotation and placement and connecting components using hot melt adhesive. The robotic arm then places the robot in an arena where its fitness is computed using an overhead camera to assess performance. However, the system lacks sensing capabilities, and the fabrication process is relatively straightforward due to the modular component design.

The proof-of-concept of an autonomous fabricator of amorphous robots was shown in [24] and this system was further expanded in [26]. In this system, the amorphous shape of robots was 3D printed, and a robotic arm assembled all the components, including shape, brain, wheels, sensors, and joints. Similar to the previous approach, one of the limitations of this system is that there is no feedback during fabrication, and therefore errors during production can occur.

Feedback was introduced by [18,35] in a system for fabricating modular robots. A robotic arm assembles the body designs through magnetic connections. Each module is labelled with a tag, and a visual tracking system provides necessary feedback, such as distance and orientation errors, to perform assembly more accurately.

In conclusion, automation is crucial in ER, yet the number of approaches remains limited. There is a lack of analysis regarding the production of robots and its impact on ER. This paper examines the robot production throughput and its impact on ER in more detail.

2.4. Summary

Work shown in the literature has insinuated the existence of the three challenges mentioned in this paper; however, it has not gone into detail. It is important to do so to understand their impact and, with this understanding, propose appropriate solutions to mitigate these challenges.

In addition, even though at first glance each challenge might seem independent, they are closely linked. For instance, since the reality gap will change for each robot, this will necessitate a higher number of robots to be evaluated and with this an increase in the continuous production of robots, highlighting the importance of automation. Since one of the goals is to minimize the reality gap in the population of robots, simulation must focus on the regions of the most transferable robots.

3. Framework for the Hardware-Software integration in evolutionary robotics

The Autonomous Robot Evolution (ARE) project is used in this paper to explore and illuminate the importance of the challenges mentioned in the previous sections. ARE envisions the integration of *digital evolution* and *physical evolution*, combining the benefits from both domains.

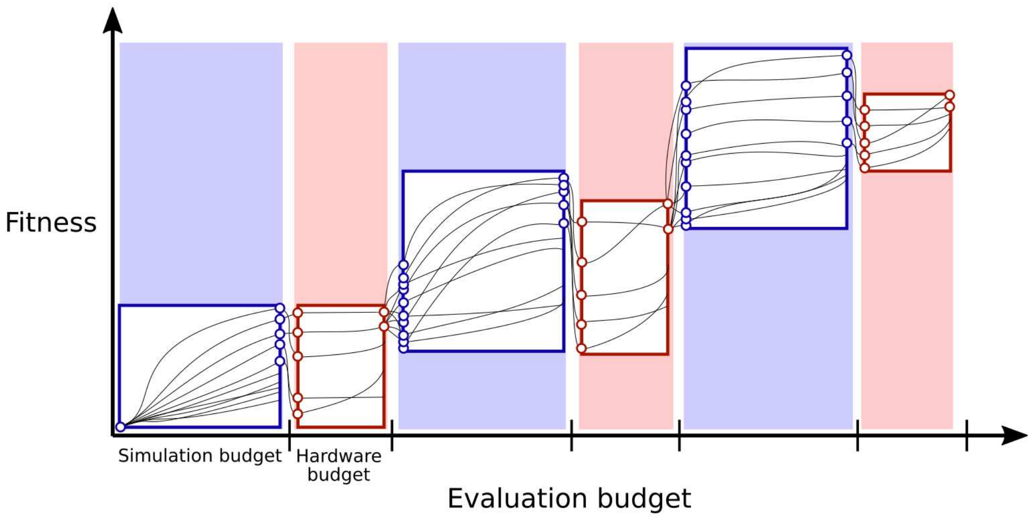

Figure 2. Long-term vision for the ARE hardware-software integration. A population of robots is first evolved in simulation and then a subset is selected to initialize a population in hardware. After evolution in hardware is completed a subset of robots is selected to initialize a new population of robots in simulation. This cycle repeats until the termination condition is met.

Ultimately, an optimal robot design will evolve to suit a given environment. If the environment changes, evolution will adapt the design to meet the new conditions. The ARE system is particularly relevant for remote, unknown locations and/or environments hazardous to humans.

Physical robots will be autonomous and fabricated by a robot fabricator, *RoboFab*. Details of each element will be described in more detail in the next sections.

3.1. *The software-hardware synergy balance*

The long-term term version of the software-hardware integration proposed by ARE is similar to the one implemented by [10], where evaluations in software and hardware are divided into stages. A budget is allocated to each stage, where the budget can be time, number of evaluations, or resources. This multi-stage approach facilitates tracking evolution progression with multiple checkpoints. This process is illustrated in Figure 2.

The ARE current implementation is illustrated in Figure 3. It works as an evolutionary process with four stages *reproduction*, *survival*, *evaluation*, and *selection*.

The field of parallel Evolutionary Algorithms (pEA) [36] has demonstrated the ability to increase solution diversity, accelerate optimization through parallelization, and enhance generalization through migration. For these reasons, the decision was made to begin the process by evolving a set of populations (islands) of robots in simulation. Ten independent evolutionary processes (islands) in simulation are launched each beginning with a different initial random population of 25 robots and a budget of 50 robots in total. The evolutionary algorithm (EA) used is an asynchronous version of Morpho-Evolution with Learning using Archive Inheritance (MELAI) [37]. This EA is a hierarchical optimization process consisting of two nested processes.

The outer process is a non-generational, asynchronous, pEA that optimizes the design of the robots. The inner process called the Novelty-based Increasing Population Evolutionary Strategy (NIP-ES), optimizes the controller. MELAI includes an archive that stores the best controllers to bootstrap NIPES. This exploration stage aims to develop a variety of high-fitness robots with diverse body designs and behaviours.

Once all islands have completed their evolution, 5 robots are selected from the 500 digital robots in the *survival stage* based on their fitness and body design novelty. These robots are then built and tested in the real world.

The *evaluation stage* has three steps:

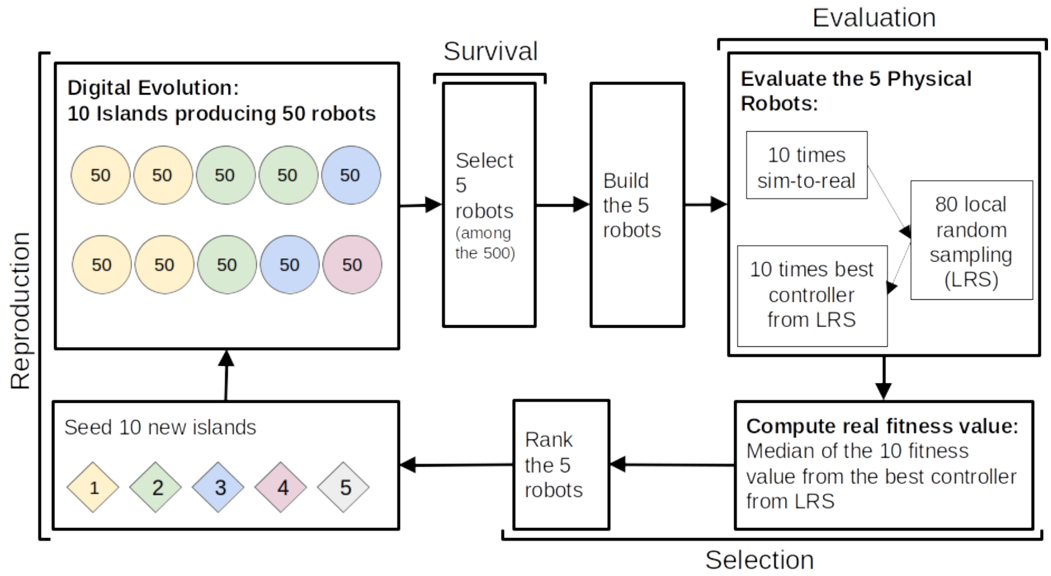

Figure 3. The diagram illustrates the complete hardware evolutionary algorithm proposed in this paper. The process begins with 10 independent evolutionary processes (islands) of *digital evolution*, each generating 50 designs. From these, 5 robots are selected to be built and evaluated in hardware. To bridge the reality gap, a 3-step process based on local random sampling (LRS) is used during the evaluation phase. After evaluation, the 5 robots are ranked according to their fitness value and evolvability score. This ranking determines which robots will initialise (seed) the next set of islands in the *digital evolution* phase.

1. The physical robot is evaluated 10 times using the controller from its digital twin; this is referred to as *sim-to-real*.
2. Next, 80 controllers are sampled locally around the controller from the digital twin, a step known as local random sampling (LRS). LRS uses a multivariate normal distribution centred on the controller from the digital twin, with a variance of 0.1.
3. Finally, the best controller from the LRS is re-evaluated 10 times. The fitness of the physical robot is the average fitness value of these 10 re-evaluations.

During the *selection stage*, the 5 physical twins are ranked based on two objectives: fitness value and evolvability score. The evolvability score measures a genome's ability to produce a wide range of robotic designs through mutation. To compute this score, each robot's genome is mutated 100 times to generate 100 new designs. The average distance between the original robot's design and these 100 new designs is calculated based on a morphological descriptor, a 3D matrix representing the placement of the robot's components. This average is the evolvability score.

In the *reproduction stage*, the robot with the highest rank initializes (seeds) the starting population for 4 islands in simulation, the second highest initializes 3 islands, the third 2 islands, and the fourth 1 island. The robots comprising the population at each island are a mutated version of the robot used to seed the island.

Given the relatively long time required to fabricate robots (more information in Section 3.2) and the extended duration of evaluations, the hardware budget is smaller, resulting in smaller robot populations. Because of this, this stage is exploitative, focusing on identifying the most transferable robots—those that experience smaller reality gaps. After the hardware budget is depleted, a selected group of best-performing robots is used to initialize a new stage in simulation.

This approach combines the benefits of the explorative aspect of simulation with the exploitative aspect of hardware, with the expectation that over time, the most diverse and transferable robots will evolve.

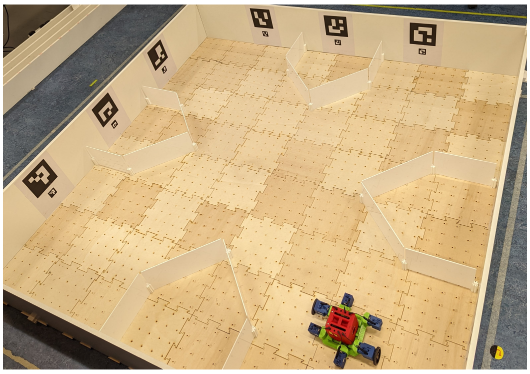

Figure 4. The environment used for physical experiments. The robot is shown in the starting location, where it was replaced for each evaluation.

In the ARE project, a software-hardware experiment was conducted with four stages in simulation and four stages in hardware. The environment used, shown in Figure 4, consisted of square arena with sides of approximately two meters and divided by fixed barriers into a cross shape, with the robot starting in one corner. The floor was made up of 64 equally sized titles. The uneven surface created by small gaps and ridges at the joins between tiles sometimes made it difficult for robots with only wheels to explore effectively. The robots themselves were made from the ARE hardware described previously [26]. The task chosen was exploration, with the fitness function defined by the percentage of the floor tiles visited at some point during the evaluation time.

The *sim-to-real* performance, or fitness from the physical twin without learning, for the 4 generations in hardware are shown in Figure 5a. From generation 0 to generation 1, there is a significant increase in fitness, but from generation 1 to generation 2, there is a reduction in fitness. This may be because the simulation stage moves to a region of robots that are not transferable. In addition, this region could be characterised by good learners. The key takeaway is that when running software-hardware experiments, careful attention must be paid to prevent this issue, which might be related to the selection process of the robots (further details in Section 4) or the allocated budget. More experiments are needed to confirm these observations. First, a larger number of generations is needed to better analyse trends where evolution favours learnable robots. Second, it would be beneficial to run experiments with a different task for comparison.

Despite the drop in fitness, there are indications that the overall quality of the evolved robots is improving, as shown in Figure 5b. This figure illustrates the progression of fitness over each generation for the sorted robots, from the lowest fitness (blue) to the highest fitness (orange) in their respective generation. For each rank, there is an increasing trend in fitness after each generation. This along with the work shown in [8,10] highlights the importance of evaluating populations of physical robots to find better robots.

In conclusion, this section highlights the benefits of integrating evolution in both software and hardware. Preliminary results suggest an improvement in performance with each hardware generation; however, these results also reveal challenges associated with this approach. Additional experiments with physical robots are needed to confirm these findings.

3.2. Autonomous fabrication of robots

As mentioned in Section 2.3 for *physical evolution*, manually fabricating and assembling the evolved robots is physically demanding. Therefore, a facility is required to carry out this labour. However, the design of this facility has direct implications for the body shapes that can be evolved and the robot throughput.

The ARE project introduced the autonomous robot fabricator (RoboFab), shown in Figure 6 and described next. At the heart of RoboFab is a robotic arm that moves components between various workstations and performs the main assembly process. Operating

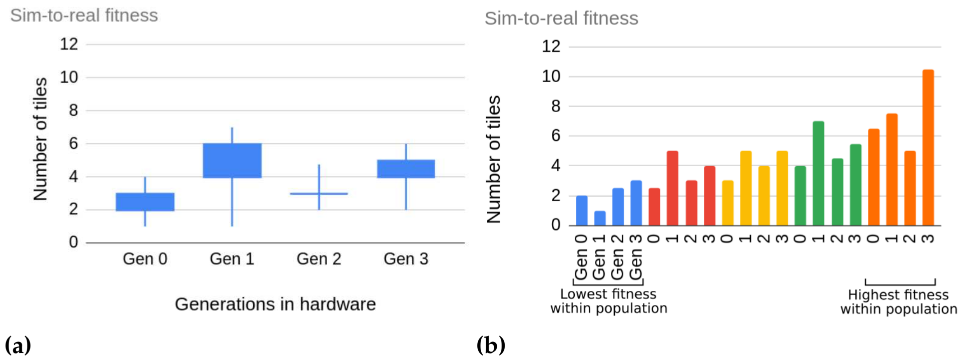

Figure 5. Figure (a) on the left shows the performance of *sim-to-real* across each generation in hardware. The performance improves from generation 0 to generation 1 but then decreases from generation 1 to generation 2. Figure (b) on the right displays the *sim-to-real* performance for all five robots across each generation. The bar plots are sorted based on the robots' performance within each generation: for example, blue bars represent robots with the lowest fitness in that generation, while orange bars indicate robots with the highest fitness. Overall, there is a positive trend, as even the robots with the lowest fitness show improvement across generations.

without a vision system or similar feedback, the robotic arm retrieves components from predetermined positions and assembles the robot "blindly" based on its genome. To the right and rear of RoboFab, two 3D printers produce custom body parts. On the left side, a component bank stores all necessary parts for easy access by the robotic arm, with each component held in a defined position for straightforward retrieval. The assembly fixture secures the robot during construction and rotates it, giving the robotic arm access to the correct attachment point for the next component. An example of an evolved robot can be found in Figure 7.

A critical factor in the practicality of the autonomous fabrication of robots is the time taken for each robot to be produced. Figure 8 illustrates the time taken for each robot's production, broken down into the key stages: 3D printing the skeleton, inserting the head component into the main body, and attaching the remaining organs. While component assembly takes between 3 and 6 minutes—comparable to the under 5-minute assembly time reported by [18]—removing the skeleton from the printer requires around 25 minutes to allow the print bed to cool. The overall construction time is primarily dominated by the 3D printing of the skeleton, which takes an average of 4 hours and 3 minutes and this is less than the up to 20 hours reported in [13], this reduction in time is likely due to differences in 3D printing settings, such as the larger nozzle size and increased layer height used in the ARE platform, though this comes at the cost of reduced print quality. The difference in print time is primarily influenced by the size of the body, with larger parts taking longer to produce. Consequently, the design decision to allow evolution the option of creating an amorphous robot directly impacts the time needed to fabricate a single robot.

Production time may well be the limiting factor in practice for evolving robots in hardware, and these initial results allow for some extrapolation to assess the feasibility of the system. In the kind of research lab setting expected for the RoboFab, it will likely run during working hours and can be expected to produce approximately ten robots per week. To give some idea of the time required to carry out meaningful evolution, an estimate for the minimum number of individuals to be produced is needed, for which previous examples can be used. [8] used a population size of ten for ten generations, giving a total of 100 individuals per run. For the experiments shown in Section 3.1, a total of 20 robots were evolved. It is important to note that more complex tasks and/or repeated evolutionary runs will require orders of magnitude more individuals. Table 1 summarizes the implications of these approximate numbers by estimating the real-world time required for various combinations of production capacity and individuals needed. It is crucial to highlight that

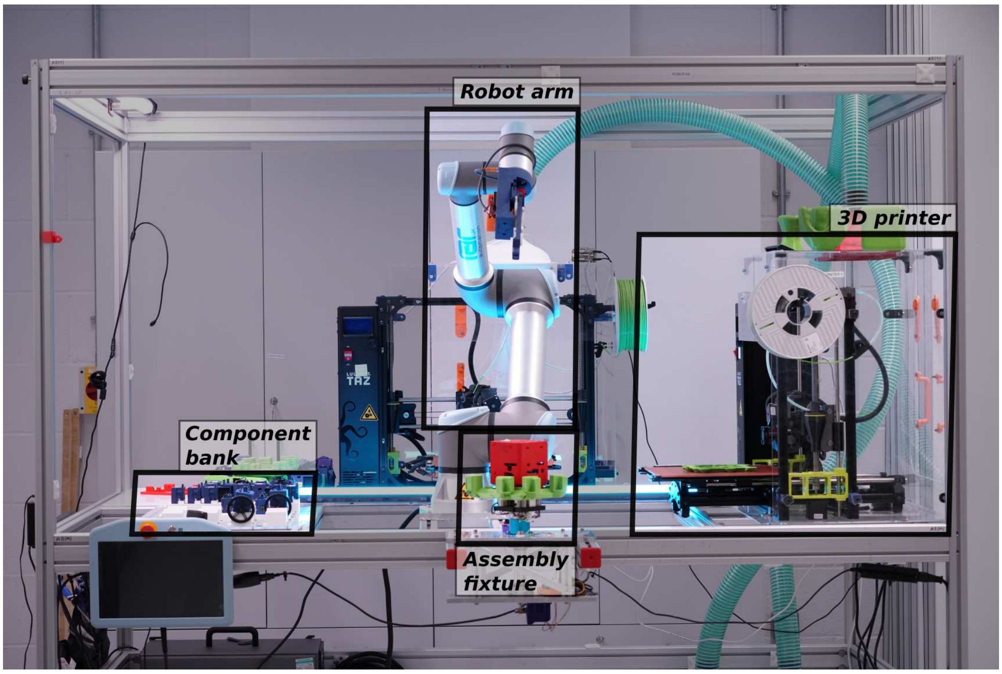

Figure 6. The RoboFab (short for Robot Fabricator) autonomously manufactures evolved robots. The main components are highlighted: two 3D printers produce body parts; the robotic arm assembles the evolved robots; the component bank provides storage for the components until they are needed; and the assembly fixture holds the new robot during the assembly process.

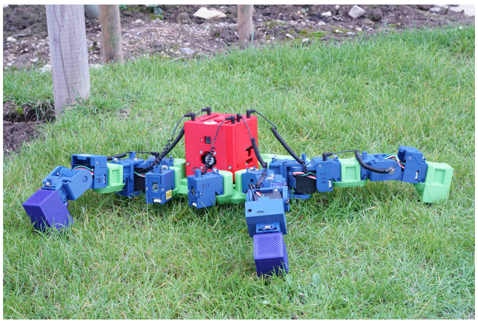

Figure 7. Example of an ARE robot.

if a relatively large number of robots, such as 1000, is required, it will take months or even years, making it unfeasible in practice for lab-based experiments. Therefore, careful design decisions for the robot fabricator need to be considered regarding the final application of this facility.

The previous estimate in production times assumes that there are no errors during production within the ARE system. However, similar to the system in [8], errors can occur during production. The authors in [8] claim that 95% of assemblies were successful, with some failures including connection failures and collisions during assembly. The errors of assembly in the ARE platform are summarized in Table 2 and described next.

As mentioned before, RoboFab is an open system and has no external feedback during assembly. Therefore, the majority of failures are related to failed attempts to grip components (gripping fault) that were humanly misplaced in their location. The rest of the errors were caused by inappropriate engineering design decisions such as the cables not connecting properly into the sockets (cable fault) or message interruption between the PC and gripper (communication fault). These reasons highlight the importance of feedback in the autonomous construction of robots and the importance of human supervision at the early stages of development.

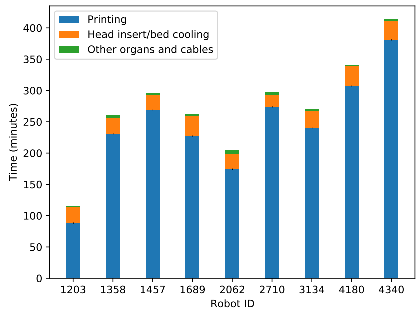

Figure 8. Time taken to fabricate each robot is divided by the main stages of construction: 3D printing the body, inserting the head into the body (which includes time for the print bed to cool), and attaching the remaining organs, components, and cables.

Table 1. Estimated duration of a robot evolution process depending on the total number of individuals and the (re)production capacity.

	1 RoboFab 1 3D printer	1 RoboFab 3 3D printer	2 RoboFabs 3 3D printer
Robots per day	2	6	12
20	2 weeks	1 week	4 days
100	10 weeks	4 weeks	2 weeks
200	5 months	7 weeks	4 weeks
1000	23 months	9 months	4 months

In the approach proposed by [18], a camera mounted on the robot arm can detect the tags on each module and thus provide an accurate estimation of the location of the components. Additionally, the attraction of the magnets in the connections facilitates alignment for attachment. The number of failures in the ARE and future platforms can be reduced with external feedback.

Even when feedback is incorporated into the system, it cannot be left unattended, at least in the early stages of development. Due to the sequential nature of the assembly process, a single failure can lead to a cascade of sequential failures. Therefore, a human is required to monitor the process.

In conclusion, even though the continuous autonomous fabrication of robots could be a possible solution to the high number of physical robots needed for *physical evolution*, there are two important aspects to consider. First, even with autonomous fabrication, the time it takes to construct a single robot is long and if this time were to be reduced some limitations would be introduced to the feasible evolutionary body design space [26]. A single experiment involving a couple hundred physical robots could take from a few weeks to half a year to complete, which could impact experimentation. Second, a fabrication system cannot be left unattended, at least during the early stages of development and for the near future a human needs to be part of the loop.

Table 2. Success and error frequency during assembly.

Success or error fault type	Number of robots
Success	11/20
Cable fault	3/20
Gripping fault	4/20
Assembly fixture fault	1/20
Communication fault	1/20

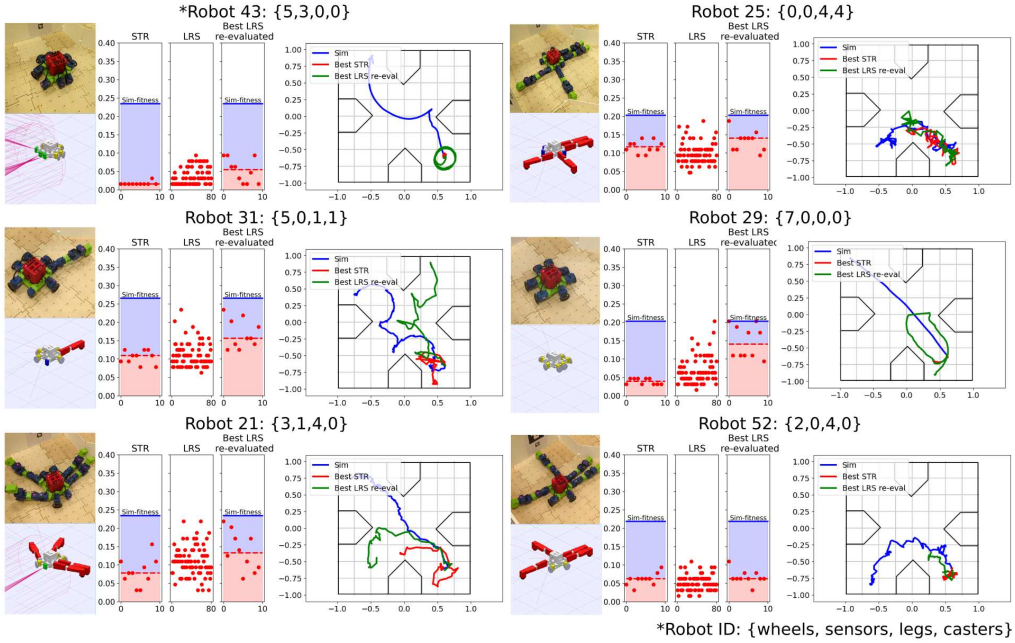

Figure 9. The heterogeneous reality gap. The fitness from the digital twin is shown with the blue line (*sim-fitness*). The *sim-to-real* (STR) column illustrates the evaluation of the physical twin with the controller from the digital twin where each dot represents an evaluation of the total 10 evaluations and the median is shown as a dashed line. The local random sampling (LRS) columns show the 100 evaluations of different controllers. The best controller found by LRS is evaluated 10 times as shown in the best LRS re-evaluated column.

3.3. *The heterogeneous reality gap*

Authors in [16] mentioned that the different shapes and behaviours of evolved robots produce varying reality gaps between limbed robots. For instance, the length of the limbs can lead to collisions, and robots with dragging behaviours experience a larger reality gap than those with gaits. In this paper, the different reality gaps experienced by various robots referred to as *the heterogeneous reality gap*, are further analyzed using a selection of evolved robots.

A sample of six evolved robots with different combinations of components was selected to illustrate the heterogeneity of the reality gap. These robots are shown in Figure 9. The analysis is divided into three stages introduced in Section 3.1 and summarized next.

For each robot, the fitness from the digital twin is first shown as the blue line (*sim-fitness*). Then, each physical twin is evaluated 10 times with the same controller used in the digital twin. This stage is referred to as the *sim-to-real* (STR) experiment (left column). The objective of this stage is to quantify the reality gap for each robot. In the second stage, a local random sampling (LRS) method is used to generate 100 new controllers based on the controller from the simulation (shown in the middle column). The best controller found in LRS is re-evaluated 10 times in the last stage to validate the results (third column).

The first observation is that the fitness in simulation for these six robots is very similar, ranging between 0.21 and 0.28 or between 14 and 18 tiles out of 64 tiles, despite their different shapes and component configurations. This fitness equivalence corresponds to the robot starting at the bottom-right corner of the arena and reaching one of the other corners, as illustrated by the blue trajectories. Robots with wheels experience smoother movement than robots with legs, as shown by the less noisy trajectories.

On average, robots with legs achieve higher fitness in STR than robots with wheels. A common behaviour observed in robots with wheels is that they get trapped at the edges of the tiles, altering their locomotion. This behaviour is replicable and is illustrated in the distribution of the fitness values, which are clustered around the median. In contrast, robots with legs generally experience a lower reality gap because their behaviours are more resilient to the environment. However, the behaviour of robots with legs is not as consistent as that of robots with wheels, as indicated by the distribution of fitness values.

LRS effectively provides different controllers with varying fitness values, as illustrated for each robot. The range of controllers is wide, and the best controller from LRS is selected for re-evaluation. All the robots experience an increase in fitness after LRS, indicating that learning is necessary for each robot and each reality gap they encounter. Robots with wheels appear to improve their fitness the most.

The relatively large variability in fitness observed in STR and after LRS is due to real-world testing, aligning with the findings of [8] where despite the use of elitism from the authors in *physical evolution*, robots still exhibit high variability because of uncertainties and the lack of perfect repeatability.

The most significant aspect of these results is their impact on the selection of the next generation of robots in simulation and/or hardware. For instance, based on the results shown in this section, it is likely that STR robots with legs will be selected for the next generation. However, if robots are selected after learning, then robots with wheels are more likely to be chosen. Each of these choices could drive the evolutionary process in different directions, affecting the hardware-software integration discussed in Section 3.1. The different possible metrics that can be used for the selection operator are further discussed in Section 4.

In conclusion, each robot with different component configurations experiences a different reality gap. This gap can be mitigated with a learning algorithm and each robot needs to run its independent learning process. However, when selecting the next generation of robots, the question arises of how to make the best selection.

4. Discussion

This paper identifies three major challenges in the transition from *digital evolution* to *physical evolution*. Although these challenges have been suggested in previous literature, they have not been comprehensively explored. Here, an more in-depth discussion of these challenges is provided, supported by experiments conducted on the ARE platform.

4.1. Software-hardware synergy

Physical evolution offers many benefits, but it also comes with several disadvantages. To balance these, it is crucial to integrate *digital evolution* into the system. However, this integration is not straightforward, and key initial design decisions can significantly influence the evolutionary path and the bias in robots evolved. One of the most critical design decisions is the selection operator, which determines which robots migrate between the virtual and hardware domains, and vice versa.

Robots can be selected based on various metrics, including:

- Fitness in hardware score without learning: the same copy of the controller of the digital twin is used with the physical twin and the fitness from the physical twin is used as a score during selection.
- Transferability score: The difference between digital twin fitness and physical twin fitness (or the reality gap) is used as a score.
- Fitness post-learning: the controller from the digital twin goes through a learning process in the physical twin to adapt the controller to the physical environment. Then the fitness post-learning is used as a score.
- Learnability score: The difference between the physical twin fitness pre-learning and the physical fitness post-learning is used as a score.

Each of these metrics offers unique insights into the robot's body design and controller, and they are not necessarily correlated. For instance, a robot without sensors may be more learnable due to the simplicity of its body design and open-loop controller. However, robots with sensors and closed-loop controllers tend to be more transferable because their behaviour can self-correct with the feedback from the environment. Some evidence of this was presented in Section 3.1. Additionally, the work in [38] demonstrated how learnability leads to a different robot design space compared to when learnability is not considered in *digital evolution*.

The choice of selection metrics affects not only the evolution of body designs but also the time spent on fabricating and evaluating the robots. For example, if the focus is on transferability, more time may be devoted to robot fabrication rather than physical evaluation. Conversely, if the emphasis is on learnability, more time will likely be spent evaluating physical robots.

Learnability is crucial in the evolutionary process. The authors in [38] demonstrated through simulations that considering learnability can save evaluations. In addition, results in Section 3.3 show that a simple learning method can enhance the performance of robots with different designs. This raises questions about how learnability will influence the dynamics of *physical evolution* and what future experiments might reveal. On one hand, learnability can help reduce the number of robots fabricated. On the other hand, inheritance can be used to bootstrap controllers in physical robots with similar features, thus saving on evaluations. This approach is similar to what the authors did with an inheritance archive [37,39] and Lamarckian learning [40,41].

In conclusion, achieving an effective synergy between software and hardware in *physical evolution* requires careful consideration of selection metrics. This choice not only shapes the evolutionary trajectory but also influences the balance between digital and physical evaluations.

4.2. Autonomous fabrication of robots

Population size is a crucial factor that significantly impacts the performance and efficiency of ER. Smaller populations are generally sufficient for simpler, well-understood problems, while larger populations are necessary for addressing more complex challenges. In the context of *physical evolution*, determining the optimal population size is particularly challenging due to the considerable time and resources required to fabricate and evaluate each robot.

One of the primary motivations behind autonomous robot fabrication is to reduce fabrication time, thereby enabling larger population sizes in physical evolution. However, even with automation, fabrication time can remain substantial—often taking hours per robot, depending on the platform [13,18,24]. As analyzed in Section 3.2, producing a few hundred robots could take anywhere from weeks to months. This suggests that initial populations in physical evolution are likely to be small and more focused on exploitation rather than exploration [8].

Fabrication time can be reduced in different ways including:

- Further parallelisation: experiments can be run on more platforms across different institutions.
- Platform design: the robot platform can be redesigned; for example, by reducing the robot's size.
- Manufacturing process: while 3D printing is time-consuming, alternative manufacturing processes can reduce fabrication time, though each introduces trade-offs. For instance, using a laser cutter increases the complexity of autonomous robot assembly. Resin casting requires mould changes, and CNC machining generates more material waste and limits design flexibility due to the constraints of cutting tools.
- Reused pre-printed parts: the main body of the robot can be 3D printed into multiple parts instead of a single piece. If the same piece is needed for a second robot then this piece can reused.

A long-term goal for *physical evolution* is achieving near-complete autonomy from human intervention. However, as demonstrated in both the literature [8] and this paper, this goal is not yet feasible due to the potential for fabrication faults. This paper also emphasizes the importance of incorporating feedback mechanisms within autonomous fabrication systems to minimize these faults. For example, the platform presented in [18] uses a camera to detect tags on each component, allowing for error correction when the gripper handles different parts. The main drawback is that adapting this specific system to work with amorphous robots, where there are no tags, would be challenging. In the event of any serious faults during fabrication, the process, including 3D printing and assembly, must be halted, and the issue reported to the operator for prompt resolution.

In conclusion, while autonomous fabrication can reduce manufacturing time, it remains high and may not be suitable for certain experimental purposes and different and new manufacturing technologies are required to reduce this time. Additionally, it is crucial to consider potential manufacturing faults and implement strategies to minimize them, while also accounting for any implications on the body design landscape.

4.3. The heterogeneous reality gap

When *digital evolution* and *physical evolution* are integrated into ER, the *heterogenous reality gap* is unavoidable as shown in Section 3.3. This gap means that each evolved robot will have a different degree of reality gap due to variations in body design and behaviour. As a result, a reality gap treatment effective for one robot may not be suitable for another.

To address this, an independent controller learner is required for each robot, enabling it to adapt its controller to its unique physical body and environment. The results presented in this paper suggest that the performance of the learner will vary depending on the robot's body configuration. A promising area for future research in ER is the potential to model a robot's reality gap and incorporate this model into the robot itself. This would enable the robot to not only learn about the discrepancies between simulation and hardware but also to adapt and account for component wear over time.

Another challenge arises when a learner is used for each evolved physical robot, leading to long evaluation times. Reducing this time would be advantageous. One approach to achieving this is by bootstrapping physical robots with relatively well-performing controllers, similar to the method used with simulated robots in [37]. Another approach is to incorporate surrogate models to reduce the number of evaluations with physical robots.

It is also important to consider the design of controllers used in ER. Developing a controller architecture that is effective across robots with different modes of locomotion, such as wheels and legs, is particularly challenging.

In conclusion, the *heterogenous reality gap* is a challenge in ER which plays an important role during the selection of robots for both *virtual and physical evolution*. Additionally, the use of independent learners for each physical robot to address this gap could lead to substantial time spent on evaluation.

5. Conclusions

One of the ultimate goals in Evolutionary Robotics is transitioning from *digital evolution*, where robots are evolved in simulation, to *physical evolution*, where robots are evolved in hardware. Recent work has made significant strides in this direction [8,10,24]. However, this paper discusses three key challenges of this transition and their possible mutual connections, analyzed using the Autonomous Robot Evolution platform.

1. Autonomous Fabrication of Robots: To increase the throughput of robot production, autonomous fabrication is essential, as ER typically requires large robot populations. However, it is crucial to consider the time required for fabrication and the system's ability to reliably produce robots with diverse shapes, while minimizing faults during assembly.

2. Soft-Hardware synergy: Effective integration of software and hardware is vital to leverage the advantages of both simulated and physical evolution. Careful consideration must be given to parameter design, as selection parameters will significantly influence the evolutionary process. Therefore, understanding and choosing the appropriate metrics is critical.

3. The heterogenous reality gap: The reality gap—differences between simulated and physical environments—will vary from robot to robot due to differences in body design and behaviour. As a result, a learning mechanism is needed to adapt each robot's controller

to its new environment. The downside is that this will increase the number of evaluations across populations, leading to longer overall evaluation times.

In summary, while substantial progress has been made in ER, addressing the challenges of software-hardware integration, managing the dynamic reality gap, and automating robot production is crucial to transition to the continuous production of evolved robots in real-world environments. Future work will focus on refining these strategies to further advance the field toward autonomous robotic ecosystems.

Author Contributions: The individual contribution for this article were as follows: Conceptualization: E.B., L.K.L.G., M.F.H., E.H., A.E.E., A.F.W., J.T., A.M.T.; methodology: E.B., L.K.L.G., M.F.H., E.H., A.E.E., A.F.W., J.T., A.M.T.; software: L.K.L.G., E.B., M.D.C., M.F.H.; validation: E.B., L.K.L.G., M.F.H., M. A., R. W.; formal analysis: E.B., L.K.L.G., M.F.H.; investigation: E.B., L.K.L.G., M.F.H.; visualization: E.B., L.K.L.G., M.F.H.; supervision: E.H., A.E.E., A.F.W., J.T., A.M.T.; project administration: E.H., A.E.E., A.F.W., J.T., A.M.T.; funding acquisition: E.H., A.E.E., J.T., A.F.W., A.M.T.; writing—original draft: E.B., L.K.L.G., M.F.H., J.T., A.M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by EPSRC ARE project, EP/R03561X, EP/R035733, EP/R035679, and the Vrije Universiteit Amsterdam.

Acknowledgments: The authors would like to acknowledge the Institute for Safe Autonomy (ISA).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ARE	The Autonomous Robot Evolution project
ER	Evolutionary Robotics
LRS	Local random sampling
STR	Sim-to-real

References

1. Eiben, A.E.; Hart, E.; Timmis, J.; Tyrrell, A.M.; Winfield, A.F. Towards autonomous robot evolution. *Software engineering for robotics* **2021**, pp. 29–51.
2. Lund, H.H.; Hallam, J. Evolving sufficient robot controllers. In Proceedings of the Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97). IEEE, 1997, pp. 495–499.
3. Nolfi, S.; Floreano, D. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*; MIT press, 2000.
4. Walker, J.; Garrett, S.; Wilson, M. Evolving controllers for real robots: A survey of the literature. *Adaptive Behavior* **2003**, *11*, 179–203.
5. Bongard, J. Evolutionary Robotics. *Communications of the ACM* **2013**, *56*, 74–85.
6. Vargas, P.; Paolo, E.D.; Harvey, I.; Husbands, P., Eds. *The Horizons of Evolutionary Robotics*; MIT Press, 2014.
7. Cheney, N.; Bongard, J.; Lipson, H. Evolving soft robots in tight spaces. In Proceedings of the Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation, 2015, pp. 935–942.
8. Brodbeck, L.; Hauser, S.; Iida, F. Morphological evolution of physical robots through model-free phenotype development. *PloS one* **2015**, *10*, e0128444.
9. Jelisavcic, M.; De Carlo, M.; Hupkes, E.; Eustratiadis, P.; Orlowski, J.; Haasdijk, E.; Auerbach, J.E.; Eiben, A.E. Real-world evolution of robot morphologies: A proof of concept. *Artificial life* **2017**, *23*, 206–235.
10. Kriegman, S.; Blackiston, D.; Levin, M.; Bongard, J. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences* **2020**, *117*, 1853–1859.
11. Nordmoen, J.; Veenstra, F.; Ellefsen, K.O.; Glette, K. Quality and diversity in evolutionary modular robotics. In Proceedings of the 2020 IEEE symposium series on computational intelligence (SSCI). IEEE, 2020, pp. 2109–2116.
12. Cully, A.; Clune, J.; Tarapore, D.; Mouret, J.B. Robots that can adapt like animals. *Nature* **2015**, *521*, 503–507.
13. Lipson, H.; Pollack, J.B. Automatic design and manufacture of robotic lifeforms. *Nature* **2000**, *406*, 974–978.
14. Hiller, J.; Lipson, H. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics* **2011**, *28*, 457–466.
15. Auerbach, J.; Aydin, D.; Maesani, A.; Kornatowski, P.; Cieslewski, T.; Heitz, G.; Fernando, P.; Loshchilov, I.; Daler, L.; Floreano, D. Robogen: Robot generation through artificial evolution. In Proceedings of the Artificial Life Conference Proceedings 14. MIT Press, 2014, pp. 136–137.

16. Samuelsen, E.; Glette, K. Real-world reproduction of evolved robot morphologies: Automated categorization and evaluation. In Proceedings of the Applications of Evolutionary Computation: 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings 18. Springer, 2015, pp. 771–782.
17. Kriegman, S.; Walker, S.; Shah, D.; Levin, M.; Kramer-Bottiglio, R.; Bongard, J. Automated shapeshifting for function recovery in damaged robots. *Proceedings of Robotics: Science and Systems (2019)* **2019**.
18. Moreno, R.; Faiña, A. EMERGE modular robot: A tool for fast deployment of evolved robots. *Frontiers in Robotics and AI* **2021**, *8*, 699814.
19. Matarić, M.; Cliff, D. Challenges in evolving controllers for physical robots. *Robotics and autonomous systems* **1996**, *19*, 67–83.
20. Sofge, D.; Potter, M.A.; Bugajska, M.D.; Schultz, A.C. Challenges and opportunities of evolutionary robotics. *arXiv preprint arXiv:0706.0457* **2007**.
21. Silva, F.; Duarte, M.; Correia, L.; Oliveira, S.M.; Christensen, A.L. Open issues in evolutionary robotics. *Evolutionary computation* **2016**, *24*, 205–236.
22. Jakobi, N.; Husbands, P.; Harvey, I. Noise and the reality gap: The use of simulation in evolutionary robotics. In Proceedings of the Advances in Artificial Life: Third European Conference on Artificial Life Granada, Spain, June 4–6, 1995 Proceedings 3. Springer, 1995, pp. 704–720.
23. Mouret, J.B.; Chatzilygeroudis, K. 20 years of reality gap: a few thoughts about simulators in evolutionary robotics. In Proceedings of the Proceedings of the genetic and evolutionary computation conference companion, 2017, pp. 1121–1124.
24. Hale, M.F.; Buchanan, E.; Winfield, A.F.; Timmis, J.; Hart, E.; Eiben, A.E.; Angus, M.; Veenstra, F.; Li, W.; Woolley, R.; et al. The are robot fabricator: How to (re) produce robots that can evolve in the real world. In Proceedings of the Artificial Life Conference Proceedings. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 2019, pp. 95–102.
25. Groover, M.P. *Fundamentals of modern manufacturing: materials, processes, and systems*; John Wiley & Sons, 2010.
26. Angus, M.; Buchanan, E.; Le Goff, L.K.; Hart, E.; Eiben, A.E.; De Carlo, M.; Winfield, A.F.; Hale, M.F.; Woolley, R.; Timmis, J.; et al. Practical hardware for evolvable robots. *Frontiers in Robotics and AI* **2023**, *10*.
27. Sun, Y.; Zong, C.; Pancheri, F.; Chen, T.; Lueth, T.C. Design of topology optimized compliant legs for bio-inspired quadruped robots. *Scientific Reports* **2023**, *13*, 4875.
28. Koos, S.; Mouret, J.B.; Doncieux, S. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation* **2012**, *17*, 122–145.
29. Glette, K.; Johnsen, A.L.; Samuelsen, E. Filling the reality gap: Using obstacles to promote robust gaits in evolutionary robotics. In Proceedings of the 2014 IEEE International Conference on Evolvable Systems. IEEE, 2014, pp. 181–186.
30. Chebotar, Y.; Handa, A.; Makoviychuk, V.; Macklin, M.; Issac, J.; Ratliff, N.; Fox, D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 8973–8979.
31. Salvato, E.; Fenu, G.; Medvet, E.; Pellegrino, F.A. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access* **2021**, *9*, 153171–153187.
32. Cellucci, D.; MacCurdy, R.; Lipson, H.; Risi, S. 1D printing of recyclable robots. *IEEE Robotics and Automation Letters* **2017**, *2*, 1964–1971.
33. Liao, T.; Wang, G.; Yang, B.; Lee, R.; Pister, K.; Levine, S.; Calandra, R. Data-efficient learning of morphology and controller for a microrobot. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 2488–2494.
34. Vujovic, V.; Rosendo, A.; Brodbeck, L.; Iida, F. Evolutionary developmental robotics: Improving morphology and control of physical robots. *Artificial life* **2017**, *23*, 169–185.
35. Moreno, R.; Liu, C.; Faina, A.; Hernandez, H.; Gomez, J. The EMeRGE modular robot, an open platform for quick testing of evolved robot morphologies. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2017, pp. 71–72.
36. Alba, E.; Tomassini, M. Parallelism and evolutionary algorithms. *IEEE transactions on evolutionary computation* **2002**, *6*, 443–462.
37. Le Goff, L.K.; Buchanan, E.; Hart, E.; Eiben, A.E.; Li, W.; De Carlo, M.; Winfield, A.F.; Hale, M.F.; Woolley, R.; Angus, M.; et al. Morpho evolution with learning using a controller archive as an inheritance mechanism. *IEEE Transactions on Cognitive and Developmental Systems* **2022**, *15*, 507–517.
38. Luo, J.; Stuurman, A.C.; Tomczak, J.M.; Ellers, J.; Eiben, A.E. The effects of learning in morphologically evolving robot systems. *Frontiers in Robotics and AI* **2022**, *9*, 797393.
39. Liu, S.; Yao, W.; Wang, H.; Peng, W.; Yang, Y. Rapidly evolving soft robots via action inheritance. *IEEE Transactions on Evolutionary Computation* **2023**.
40. Holzinger, A.; Blanchard, D.; Bloice, M.; Holzinger, K.; Palade, V.; Rabadan, R. Darwin, lamarck, or baldwin: Applying evolutionary algorithms to machine learning techniques. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). IEEE, 2014, Vol. 2, pp. 449–453.
41. Harada, K.; Iba, H. Lamarckian Co-design of Soft Robots via Transfer Learning. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference, 2024, pp. 832–840.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.