Evolving Modular Soft Robots without Explicit Inter-Module Communication using Local Self-Attention

Federico Pigozzi
DIA - University of Trieste
Trieste, Italy
federico.pigozzi@phd.units.it

Yujin Tang
Google Brain
Tokyo, Japan
yujintang@google.com

Eric Medvet
DIA - University of Trieste
Trieste, Italy
emedvet@units.it

David Ha
Google Brain
Tokyo, Japan
hadavid@google.com

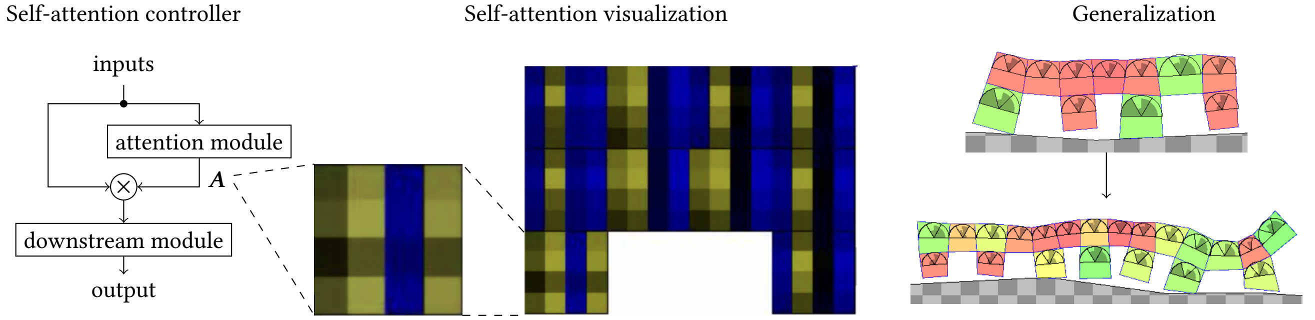


Figure 1: Overview of the proposed approach. We use the same neural controller (left picture) inside each voxel, with shared parameters. The middle picture is a biped with the attention matrices of the different voxels. Each controller uses self-attention to compute importance scores (A) among the inputs sensed by its voxel. We also find evolved controllers to generalize to unseen morphologies (right picture; color represents the ratio between the voxel current area and its rest area: red stands for contraction, green for expansion, yellow for no change).

ABSTRACT

Modularity in robotics holds great potential. In principle, modular robots can be disassembled and reassembled in different robots, and possibly perform new tasks. Nevertheless, actually exploiting modularity is yet an unsolved problem: controllers usually rely on inter-module communication, a practical requirement that makes modules not perfectly interchangeable and thus limits their flexibility. Here, we focus on Voxel-based Soft Robots (VSRs), aggregations of mechanically identical elastic blocks. We use the same neural controller inside each voxel, but without *any* inter-voxel communication, hence enabling ideal conditions for modularity: modules are all equal and interchangeable. We optimize the parameters of the neural controller—shared among the voxels—by evolutionary computation. Crucially, we use a local *self-attention* mechanism inside the controller to overcome the absence of inter-module communication channels, thus enabling our robots to truly be driven by the collective intelligence of their modules. We show experimentally that the evolved robots are effective in the task of locomotion: thanks to self-attention, instances of the same controller embodied in the same robot can focus on different inputs. We also find that the evolved controllers generalize to unseen morphologies, after a short fine-tuning, suggesting that an inductive bias related to the task arises from true modularity.[1]

CCS CONCEPTS

• **Computing methodologies** → *Mobile agents*; **Evolutionary robotics**; • **Theory of computation** → **Evolutionary algorithms**.

KEYWORDS

Neuroevolution, collective intelligence, soft robots, self-attention

ACM Reference Format:
Federico Pigozzi, Yujin Tang, Eric Medvet, and David Ha. 2022. Evolving Modular Soft Robots without Explicit Inter-Module Communication using Local Self-Attention. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3512290.3528762

GECCO '22, July 9–13, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9237-2/22/07.
https://doi.org/10.1145/3512290.3528762

[1] Videos of our results are available at https://softrobots.github.io/.

1 INTRODUCTION

Autonomous robotic ecosystems [5, 25] promise to accomplish cooperatives of robots working toward a common, human-aligned goal, ideally for tasks that are either too dangerous (e.g., rescue missions) or unfeasible for humans (e.g., digesting pollutants). In this setting, modularity plays a crucial role: unlike rigid, fully-integrated robots, modular robots rely on a large number of often identical independent modules that collectively complete an objective. As such, they have the flexibility to adapt to a wide range of morphologies and environments [38], possess fault tolerant properties [9], and can even self-assemble into novel morphologies that have not been envisioned by their original designers [51]. Indeed, modularity is a gift of nature to living organisms [20], and a product of evolution.

Modularity is also ubiquitous in the field of artificial intelligence: it appears in Graph Neural Networks (GNNs) [58], Cellular Automata (CA) [48], and multi-agent systems [70]. In evolutionary robotics, Voxel-based Soft Robots (VSRs) [27] are simulated aggregations of mechanically identical elastic blocks: as such, they have emerged as as a relevant formalism to model state-of-the-art robotic systems, e.g., soft robotics [56]. VSRs are appealing for investigating questions related to Artificial Life (AL) [37] and designing living organisms that evolve in vivo [36], bridging the sim-to-real gap [4]. In doing so, researchers mostly rely on Evolutionary Algorithms (EAs) [12], a family of algorithms that successfully married with the field of robotics [49, 61**? ?**].

Yet, in the path toward autonomous robotic ecosystems, full exploitation of modularity is still a roadblock [72], as the flexibility of modular robots depends on how the modules are "wired" together. Minimizing the need for communication among modules would permit to disassemble robots and to reassemble them differently, to cope with different tasks. Early controllers for VSRs relied on a single centralized neural network [65] or fixed, rigid inter-module communication between voxels [28, 44], effectively making their design closer to a single, large neural controller with shared weights. This rigid Message-Passing (MP) mechanism, while practical, makes the modules not perfectly interchangeable, thus limiting their flexibility. This leads to the question of whether VSRs will work *at all* if we remove *all* MP channels between modules. Any communication will thus take place through physical interactions in the environment initiated by each module, while simultaneously performing a task collectively: an outmost instance of morphological computation [26] (i.e., the "brain" offloading computation to the "body").

In this work, we explore the properties of such modular robots without MP channels. We use the same neural controller inside each voxel, but without *any* inter-voxel communication, hence realizing the ideal conditions for modularity: modules are all equal and interchangeable, enabling our robots to truly be driven by the collective intelligence of their modules. We leverage EAs to optimize the controller parameters, and experimentally test whether the resulting VSRs are effective in a locomotion task on hilly terrains. We initially find the lack of inter-module communication to be too severe of a handicap for VSRs that rely on traditional fully-connected neural network controllers. However, we find that a local self-attention mechanism—a form of adaptive weights—achieves superior performance since instances of the same controller embodied in the same robot can focus on different inputs. See Figure 1 for an overview of the proposed approach. Further, we also find that we can generalize to unseen morphologies after a short fine-tuning, suggesting that an inductive bias related to the task arises from true modularity. Through these findings, we envision this work to position itself as a stepping stone in the road toward autonomous robotic ecosystems.

2 RELATED WORK

Modularity allows robotic systems to present various kinematic configurations beyond what a fixed architecture can, and such robots are usually optimal to solve many robotic tasks [16, 60]. Modularity in robotic systems takes the form of fabricating physical parts that are interchangeable for a single robot, or designing independent robots that participate in a common task adaptively [18]. Due to the flexibility, versatility, and robustness to changing environmental conditions [72], we are witnessing an increasing number of works on modular robots [29], including soft ones [64]. For example, Kamimura et al. [34] proposed a method to automatically generate locomotion patterns for an arbitrary configuration, and Groß et al. [23] tackled object manipulation and transportation tasks using self-reconfigurable swarm-bots. The possibilities of such systems are encouraging, but the self-organizing and adaptive properties are even more inspiring. In a life-long gait learning task, Christensen et al. [9] showed that, given a body configuration, a modular system can automatically figure out the best gaits, while presenting morphology independence and fault tolerance. Taking a step further, Pathak et al. [51] demonstrated that a modular robotic system can generalize to unseen morphologies and tasks. Finally, Pigozzi and Medvet [53] partitioned modular robots into independent units on the basis of self-organizing neural controllers.

The concept of modularity appears also in the AL and machine learning communities, most noticeably in the area of neuroevolution and CA [**?**]. Indeed, we are witnessing a surge in works that incorporate these ideas into modular robots. While most robotic works are controller-focused, AL and ML researchers also explored the co-optimization of configuration and controller [3, 39**?**]. For instance, Cheney et al. [7] evolved soft robots with multiple materials through a generative encoding. Inspired by multi-cellular systems, Joachimczak et al. [31] evolved soft-bodied animats in both aquatic and terrestrial environments, showcasing the concept of metamorphosis in simulation. In another soft robot simulation work, partially damaged robots regenerated their original morphology through local cell interactions in a neural CA system [28]. Moreover, Sudhakaran et al. [63] grew complex functional entities in Minecraft through a neural CA-based morphogenetic process. Finally, there exists a whole body of literature on the evolution of virtual creatures with artificial gene regulatory networks [11].

Intuitively, the performance of a modularized system depends on the communication pattern, one therefore naturally wonders if there exist better MP graphs. Recent work suggests that GNNs possess self-organizing properties and are capable of learning rules for established CA systems [22], which hints at learning novel inter-module communication patterns. On the other hand, the very existence of inter-module MP prevents modular robots from being robust and truly interchangeable. Researchers thus set their eyes on the other end of the spectrum and explored the possibility of creating modular robots with more localized communication [50] or even without inter-module communications [33, 43, 54]. In this work, we evolve simulated VSRs for locomotion tasks, and demonstrate that it is possible to evolve a shared controller for each voxel that excludes communication with others.

3 METHODS

3.1 Background on Voxel-based Soft Robots

Voxel-based Soft Robots (VSRs) are a kind of modular soft robots composed as aggregations of elastic cubic blocks (*voxels*). Each voxel contracts or expands its volume; it is the overall symphony of volume changes that allows for the emergence of the high-level behavior of the robot. Hiller and Lipson [27] first formalized VSRs and proposed a fabrication method. In this work, we consider a 2-D variant of simulated (in discrete time and continuous space) VSRs [45], where cubes become squares. While disregarding one dimension certainly makes these simulated VSRs less realistic, it also eases the optimization of VSR design and facilitates research on their control [3], thanks to the smaller search space. However, the representations and the algorithms adopted in this paper are easily portable to the 3-D setting, and so are the considerations on the self-attention controller.

In the following, we outline the characteristics of VSRs relevant to this study, and refer the reader to [45] for more details. A VSR is completely defined by its *morphology* (i.e., the body) and its *controller* (i.e., the brain), that we detail in the following two subsections.

3.1.1 *VSR morphology.*

The morphology of a VSR describes how the voxels, i.e., elastic squares, are arranged in a grid topology. Each voxel is modeled as the assembly of spring-damper systems, masses, and distance constraints [45]. Each voxel is rigidly connected to its four adjacent voxels (if present).

Over time, voxels change their area according to (a) external forces acting on the voxel (e.g., other bodies, including other voxels and the ground) and (b) an actuation signal computed by the controller. The latter produces a contraction/expansion force that is modeled in the simulation as an instantaneous change in the resting length of the spring-damper systems of the voxel. The length change is linearly dependent on an actuation value residing in $[-1, +1]$, -1 being the greatest possible expansion and $+1$ being the greatest possible contraction.

Each voxel receives, at every time step of the simulation, the readings $\boldsymbol{s} \in \mathbb{R}^4$ of four sensors embedded in the voxels. Area sensors sense the ratio between the current area of the voxel and its resting area. Touch sensors sense whether the voxel is touching another body different from the robot itself (e.g., the ground) or not, and output a value of 1 or 0, respectively. Velocity sensors sense the speed of the center of mass of the voxel along the x- and y-directions. We normalize all sensor readings into $[0, 1]^4$ by soft normalization. After normalization, to simulate real-world sensor noise, we perturb every sensor reading with additive Gaussian noise with mean 0 and variance σ^2_{noise}. We set $\sigma_{\text{noise}} = 0.01$.

3.1.2 *Controller.*

We consider the distributed controller proposed in [44], consisting in a number of fully-connected, feed-forward Artificial Neural Networks (ANNs), one for every voxel. In particular, we adopt the "homogeneous" variant presented in [46], where the ANNs share the same parameters: Medvet et al. [46] proved that such homogeneous representation is comparable to one where parameters are different for every ANN, with the additional benefit of a more compact search space, similarly to what happens in most multi-agent reinforcement learning systems [70]. Moreover, parameter sharing makes the controller agnostic with respect to the morphology, putting ourselves on a vantage point to test generalization to unseen morphologies. With respect to the model of [44, 46], we here drop MP among voxels, as the focus of this work is on minimizing (and, possibly, dispensing with) inter-module communication in soft robots: we hence enable interchangeability of modules and thus the full exploitation of modularity. Moreover, we perform actuation every k_{act} steps, rather than at every time step as in [44, 46], to prevent VSRs from exploiting the emergence of high-frequency dynamics.

Every ANN takes as input the local sensor readings $\boldsymbol{s} \in \mathbb{R}^4$ and outputs the local actuation value $a \in \mathbb{R}$. We use a one-hot encoding of $\boldsymbol{s} \in \mathbb{R}^4$ to let the voxels know where they are in the body: the actual input of the ANN is hence built as follows. Let n be the number of voxels of a given VSR; for the i-th voxel, let $\boldsymbol{h}_i \in \mathbb{R}^n$ be a one-hot vector that is 1 at the i-th entry and 0 everywhere else. We encode $\boldsymbol{s}$ for each i-th voxel as $\boldsymbol{X} = \boldsymbol{s}\boldsymbol{h}_i^T \in \mathbb{R}^{4 \times n}$. $\boldsymbol{X}$ is then a matrix that is equal to $\boldsymbol{s}$ at the i-th column and 0 otherwise. In this way, a voxel can distinguish itself in the morphology. We apply this same pre-processing to every model considered in this paper.

We remark that one-hot encoding does not invalidate the claim that voxels are all identical: in practice, it just requires the "operator" (i.e., the robot assembler) to set the position of each voxel "in" the voxel controller itself, i.e., to do the proper configuration. Regardless of the ANN architecture, there is a unique vector $\boldsymbol{\theta} \in \mathbb{R}^p$ of parameters that specifies every ANN in the VSR. Thus, we optimize a VSR for a given task by optimizing the parameters $\boldsymbol{\theta}$.

3.2 Self-attention

Attention can be seen as an "adaptive weights" [19] mechanism that computes importance scores for the inputs. Attention mechanisms were first introduced in the context of machine translation [2, 42] to capture relationships in temporal sequences of data, and have thus prospered in natural language processing [13, 21]. Attention mechanisms have achieved state-of-the-art performance in domains (e.g., computer vision [35]) where data are not temporal but spatial [14, 71] or even sets [**?** **?**], also considering robotic settings [8, 66, 73]. There is indeed evidence that the nervous system modulates attention on every sensory channel [15]. As a result, being attention an adaptive weights mechanism, we argue it could supplant inter-module communication by computing importance scores that are tailored to the specific voxel, while being local (i.e., attending only to voxel-specific information) and shared (i.e., same parameters for every voxel).

Let $\boldsymbol{X} \in \mathbb{R}^{r \times u}$ be a sequence of u inputs of dimension r. In its general formulation, an attention module computes an *attention matrix* $\boldsymbol{A} \in \mathbb{R}^{r \times r}$, to get weighted inputs:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} \tag{1}$$

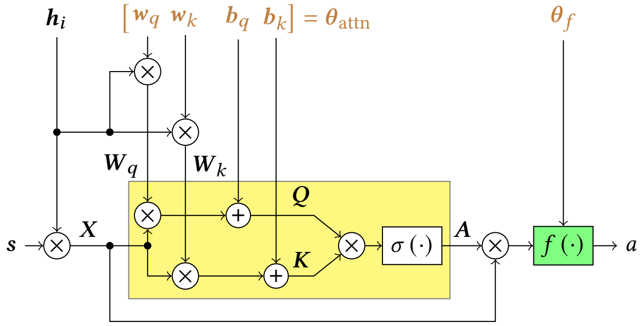


Figure 2: The architecture of our self-attention controller. Evolvable parameters are shown in brown, the attention module is shown in yellow, the downstream module is shown in green. Parameters are the same for every voxel.

to be fed to a downstream module $f(Y) = z \in \mathbb{R}$ for some desired output z.

While a great variety of attention mechanisms do exist in the literature [6], we resort on self-attention [68]. Self-attention is a generalized form of attention, and has already shown to achieve state-of-the-art results on continuous control tasks that exploit data other than temporal [66]. Self-attention computes:

$$\boldsymbol{A} = \sigma\left(\frac{1}{\sqrt{d}}\boldsymbol{Q}\boldsymbol{K}^T\right) \tag{2}$$

where $\sigma(\cdot)$ is a non-linear function that constrains $\boldsymbol{A}$ to be in a given range (e.g., tanh), $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{r \times d}$ (known as the Query and Key matrices) are the output of linear transformations of the form:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q + \boldsymbol{b}_q \tag{3}$$

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k + \boldsymbol{b}_k \tag{4}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{u \times d}, \boldsymbol{W}_k \in \mathbb{R}^{u \times d}$ are weight matrices, $\boldsymbol{b}_q \in \mathbb{R}^d, \boldsymbol{b}_k \in \mathbb{R}^d$ are bias vectors, and + denotes the matrix-vector addition (the vector is added to each row of the matrix). We take the dot product between $\boldsymbol{Q}$ and $\boldsymbol{K}$ to compute compatibility between two different representations of the inputs. The division in Equation (2) appears because the dot product grows with the operands dimensions.

3.2.1 Self-attention in VSR controller. We use the one-hot encoded sensor reading as input $\boldsymbol{X}$: then, $r = 4$ and $u = n$. The attention matrix is $\boldsymbol{A} \in \mathbb{R}^{4 \times 4}$: the attention is on the sensor readings and $\boldsymbol{A}$ dimension does not depend on the robot morphology. For self-attention to focus only on local information, we set on each i-th voxel $\boldsymbol{W}_q = \boldsymbol{h}_i\boldsymbol{w}_q^T$ and $\boldsymbol{W}_k = \boldsymbol{h}_i\boldsymbol{w}_k^T$, with $\boldsymbol{h}_i$ defined as in Section 3.1.2: in this way, we extract the column of $\boldsymbol{X}$ corresponding to the voxel. $\boldsymbol{w}_q, \boldsymbol{w}_k \in \mathbb{R}^d$ are evolvable vectors of parameters and are the same for all the voxels. We summarize the building blocks of our architecture in Figure 2.

After preliminary experiments, we set $\sigma(\cdot)$ to be tanh, $d = 8$, and f to be a Multi Layer Perceptron (MLP) with no hidden layers and tanh activation function (to ensure the output lies in $[-1, +1]$). Moreover, given that we use tanh as non-linearity in Equation (2), the entries of $\boldsymbol{A}$ lie in $[-1, +1]$, +1 being the highest compatibility between two inputs, −1 the least, and 0 no compatibility. Finally, our model differs from the original formulation of self-attention [68] in that we set Values to be the identity function, since, after preliminary experiments, we found them to be unnecessary.

In a self-attention model of this form, we optimize the parameters $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_{\text{attn}} \; \boldsymbol{\theta}_f\right]$, where the attention module parameters $\boldsymbol{\theta}_{\text{attn}}$ are the concatenation of $\boldsymbol{w}_q, \boldsymbol{w}_k, \boldsymbol{b}_q, \boldsymbol{b}_k$, and the downstream module parameters $\boldsymbol{\theta}_f$ are the weights and biases of the downstream MLP. Then, $\boldsymbol{\theta}$ is the genotype of our EA, that we detail in the next subsection.

3.3 Evolutionary algorithm

We perform optimization by means of Evolutionary Computation (EC); in particular, we resort to a Genetic Algorithm (GA) [12]. Indeed, Risi and Stanley [55] used GAs to effectively evolve complex neural architectures consisting of heterogeneous modules, and Such et al. [62] proved GAs to be competitive with state-of-the-art reinforcement learning algorithms.

Our GA iteratively evolves a fixed-size population of n_{pop} individuals according to a $\mu + \lambda$ generational model, i.e., with overlapping. We initialize individuals at the very first iteration by uniformly sampling the interval $[-1, +1]^p$. Then, at each iteration, we select parents with tournament selection of size n_{tour} and build the offspring applying Gaussian mutation, with probability p_{mut}, or extended geometric crossover, with probability $1 - p_{\text{mut}}$. For mutation, given a parent $\boldsymbol{\theta} \in \mathbb{R}^p$, we add Gaussian noise to get $\boldsymbol{\theta}' = \boldsymbol{\theta} + \sigma_{\text{mut}}\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ and $\boldsymbol{I}$ being the diagonal matrix of size $p \times p$. For extended geometric crossover, given two parents $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^p$, the new individual is born as $\boldsymbol{\theta}' = \boldsymbol{\theta}_1 + \boldsymbol{\alpha}(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) + \sigma'_{\text{mut}}\boldsymbol{\epsilon}$, where each element α_i of $\boldsymbol{\alpha}$ is chosen randomly with uniform probability in $[-0.5, 1.5]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Then, we merge offspring and parents and retain only the best half of individuals, that will constitute the population at the next iterations. Evolution terminates after n_{evals} fitness evaluations have been computed.

After preliminary experiments and exploiting our previous knowledge, we set $n_{\text{pop}} = 100$, $n_{\text{tour}} = 5$, $\sigma_{\text{mut}} = 0.35$, $\sigma'_{\text{mut}} = 0.1$, $p_{\text{mut}} = 0.2$, and, unless otherwise specified, $n_{\text{evals}} = 30\,000$.

4 EXPERIMENTS

Our goal is to answer the following questions with an experimental analysis:

RQ1 Are VSRs evolved with self-attention effective at solving a locomotion task? If so, are they robust to environmental changes?

RQ2 Why does self-attention work?

We evaluate our method on two different VSR shapes, namely a 4×3 rectangle with a 2×1 rectangle of missing voxels at the bottom-center, that we call *biped* ▦, and a 7 × 2 rectangle with empty voxels at the odd x positions in the bottom row, that we call *comb* ▦. The dimension of $\boldsymbol{X}$ is hence 4 × 10 for the biped shape and 4 × 11 for the comb shape (the second operand in the multiplication being the number of voxels); as a result, the self attention controller has $|\boldsymbol{\theta}| = 73$ parameters for biped ($|\boldsymbol{\theta}_f| = 41$ pertaining to the downstream MLP and $|\boldsymbol{\theta}_{\text{attn}}| = 32$ for the attention module) and 77 for comb ($|\boldsymbol{\theta}_f| = 45$ and $|\boldsymbol{\theta}_{\text{attn}}| = 32$).

For all the experiments, we considered the task of locomotion. The goal is to travel as fast as possible on a terrain along the positive

x direction, in a fixed amount of simulated time t_{final}. We use as fitness the average velocity $\overline{v}_x$ of the center of mass of the VSR during the simulation. We set $t_{\text{final}} = 30$ s. Locomotion is indeed a classic task in evolutionary robotics [49], but usually consists in running along a flat surface. We hereon consider a hilly terrain, consisting of bumps that are randomly procedurally generated with an average height of 1 m and an average distance of 10 m. We randomize the seed for the procedural generation at every fitness evaluation and re-evaluate individuals retained from the previous iteration, so that evolution does not unfairly favor individuals with an "easy" terrain and to make adaptation more challenging.

We implemented the experimental setup in the Java programming language, relying on JGEA[2] for the evolutionary optimization and 2D-VSR-Sim [45] for the simulation of VSRs. For the latter, we set $\Delta t = \frac{1}{60}$ s for the time step, and all other parameters to default values (as a result, all voxels share the same mechanical properties). After preliminary experiments, we set $k_{\text{act}} = 20$ (i.e., one actuation every ≈ 0.33 s). We made the code publicly available at https://github.com/pigozzif/AttentionVSRs.

For each experiment, we performed 5 evolutionary runs by varying the random seed for the EA. We remark that, for a given VSR and terrain, the simulations are instead deterministic. We carried out all statistical tests with the Mann-Whitney U rank test for independent samples.

4.1 Results

4.1.1 RQ1: effectiveness and robustness with self-attention. In order to verify the effectiveness of evolved VSRs equipped with our self-attention model, we measure their performance in two different cases: in the same conditions they were evaluated during the evolution and in slightly different environmental conditions aimed at testing an individual generalization abilities. In both cases, we use $\overline{v}_x$ as performance index: in the former, it is the value of the fitness function itself, while in the latter it is the average over 10 unseen hilly terrains, obtained with 10 different predefined random seeds.

As baseline, we compare the self-attention model (hereon Attention) with:

(a) a "communication-less" MLP (hereon MLP), that takes the same input as Attention;
(b) a "communication-based" MLP (hereon MLP-Comm), that takes as input the local sensor readings and the 4 values generated by the four adjacent voxels (if any, or zeros, otherwise) at the previous time step. Then, it outputs the local actuation and the 4 values that will be used by the adjacent voxels at the next time step. This model is the same as [44, 46] and, by virtue of the MP mechanism, is an instance of a communication-based controller.

Both models have the same architecture as the self-attention downstream MLP (see Section 3.2): as a result, MLP has 41 parameters for biped and 45 for comb, while MLP-Comm has 405 for biped and 440 for comb. MLP-Comm and MLP differ from Attention in that they do not employ an attention mechanism to obtain importance scores for the inputs. For the optimization, we use the same EA of Section 3.3.

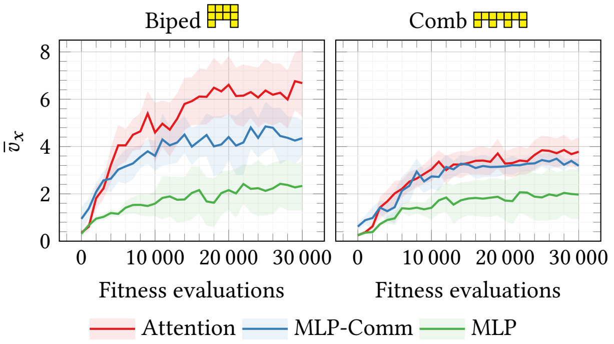


Figure 3: Median ± standard deviation (solid line and shaded area) of the average velocity for the best individuals found during each evolutionary run, obtained with three controller models and two shapes. Attention is never worse than the baselines.

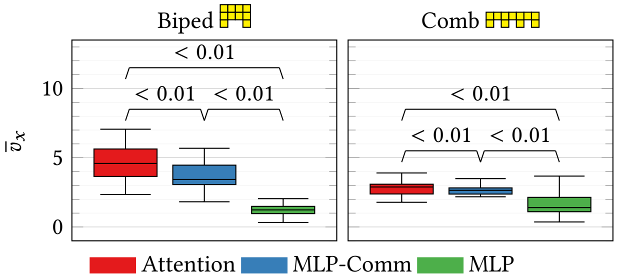


Figure 4: Distribution of the average velocity across 10 unseen hilly terrains for the best individuals found during each evolutionary run, obtained with two controller types and two shapes. Attention outperforms the baselines in terms of re-assessment.

We summarize the results in Figure 3, which plots $\overline{v}_x$ in terms of median ± standard deviation for the best individuals over the course of evolution. Moreover, Figure 4 shows $\overline{v}_x$ for the best individuals over the re-assessment terrains. For the same shape, it also reports the p-value for the statistical test against the null hypothesis of equality between the medians.

From the figures, we find that Attention outperforms both baselines. The curves of Figure 3 also suggest that all models settle on a plateau and that continuing evolution would unlikely bring better results. Attention individuals also perform better in terms of re-assessment: they achieve significantly better $\overline{v}_x$ on unseen terrains. Moreover, MLP-Comm outdoes a poor-performing MLP, as attested by the low $\overline{v}_x$ scores, especially for the biped shape. We visually inspected the evolved behaviors for the best individuals and found them to be highly adapted to a locomotion task on hilly terrain; indeed, bipeds hop on their legs as equines do and combs propagate leg movements from posterior to anterior as millipedes do. We made videos available at https://softrobots.github.io[3]. With

[2] https://github.com/ericmedvet/jgea.

[3] Number 1 and number 2.

no communication, a "vanilla" MLP controller fails; it does not enable communication-less VSRs. To reach a decent performance, we must add a MP communication mechanism. Intuitively, the MP mechanism is very important for emerging behavior to arise with an MLP-based controller. Self-attention instead does enable truly communication-less VSRs: it performs better or comparatively to both a communication-less and a communication-based MLP. We believe the reason for this to be that self-attention is a form of adaptive weights, tailoring importance scores to the inputs and, indirectly, to the particular voxel.

In addition to the evolved behaviors, we also visualized the attention matrices. Indeed, interpretability is one of the major strengths of self-attention and allows humans to get an insight into the robot inner decision mechanisms. To illustrate that point, Figure 5 presents a locomotion time-lapse for a sample VSR: at each time step, it displays the robot state in the top row and the corresponding attention matrices in the bottom row; as happens with our distributed controller, there is one such attention matrix for every voxel in the morphology. Columns and rows of the attention matrices correspond to sensors, in particular: first for the touch sensor, second and third for the x- and y-velocity sensors, fourth for the area sensor. The color of each attention score tells the strength of the compatibility between inputs, and provides an interpretation for what is driving the behavior of the VSR.

From the visualization, we realize attention scores are consistent with humans intuition and common sense. From Figure 5a to Figure 5b, attention shifts from the touch sensor of the rear leg to the touch sensor of the front leg. In Figure 5c, the VSR stops at a hole in the terrain, and the attention matrix witnesses a radical shift as a consequence. In Figure 5d and Figure 5e, the VSR starts walking again by first focusing on the front leg, and then on the rear leg, initiating locomotion once again. We found the other individuals to present similar attention patterns.

We conducted an ablation study to assess whether evolution found a trivial solution for self-attention or not. Indeed, AL researchers are all aware of the many uncanny and "creative" convergences that artificial evolution is capable of [40]. In particular, we test whether the attention update over time is needed or not, i.e., if there is a "one attention to rule them all" case. To this end, we conducted the following procedure. Given an evolved VSR with Attention, we take a snapshot of it at every second of the simulation, alongside its attention matrix at that time step. For every such snapshot, we simulate it on a fixed unseen hilly terrain for 30 s with the attention matrix frozen (i.e., it is the same as the one of the snapshot). We repeated this procedure for the best individual of every evolutionary run. We summarize the results in Figure 6 in terms of $\overline{v}_x$ (median ± standard deviation), with the time (in s) at which we took the corresponding snapshot on the x-axis. We see that performance drops dramatically, meaning that self-attention is indeed a fundamental piece in the architecture and that it is non-trivial.

Through that evidence, we can answer positively to RQ1: VSRs evolved with self-attention can solve the task of locomotion and outperform an MLP baseline without relying on inter-module communication, while being fairly able to generalize to unseen terrains.

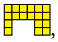
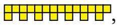
4.1.2 RQ2: why does self-attention work? We hypothesize the reasons for self-attention effectiveness to be:

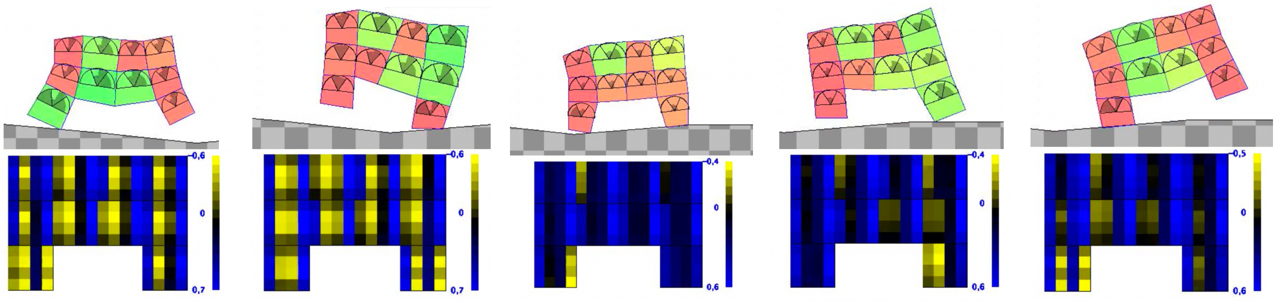
(a) self-attention evolves to represent an *inductive bias*, i.e., what tasks the controller is naturally suited at tackling, and
(b) self-attention evolves to be an *instinctive* component (akin to reflexes in biology), that complements decision-making with a fast time-scale of adaptation.

We first focus our attention on the inductive bias hypothesis. In the field of machine learning, an "inductive bias" refers to the set of assumptions made by an algorithm in order to generalize to novel and unseen data [47]; in our case, data refer to tasks. If self-attention evolved an inductive bias, as others did point out [73], it would represent features that are general for the task of locomotion, and are thus suitable for generalization.

To test the inductive bias hypothesis, we conduct an experiment of generalization to unseen morphologies. Given the attention module $\boldsymbol{\theta}_{\text{attn}}$ evolved on a specific morphology, we wonder whether we can freeze it and use it as an off-the-shelf "feature extractor" for a different morphology—as mentioned in Section 3.2, $\boldsymbol{\theta}_{\text{attn}}$ dimension is agnostic with respect to the morphology. If self-attention evolved an inductive bias, it would be useful to control the new morphology. In particular, for a frozen self-attention, we fine-tune the downstream module. The goal of fine-tuning is to have a VSR that is effective and converges faster than evolving from scratch. If that were the case, self-attention would be useful to quickly assemble new VSRs for a different task, starting from pre-optimized components. This fine-tuning procedure is of crucial importance, since we expect one-shot generalization to fail as a consequence of the embodied cognition paradigm [52, 59], which posits a deep entanglement between the morphology and the controller of an embodied agent.

For every best individual $\boldsymbol{\theta}^\star = \left[\boldsymbol{\theta}^\star_{\text{attn}}\ \boldsymbol{\theta}^\star_f\right]$ of an evolutionary run from the experiments of Section 4.1.1, we freeze its attention module parameters $\boldsymbol{\theta}^\star_{\text{attn}}$ and fine-tune its downstream module parameters $\boldsymbol{\theta}^\star_f$ using the same GA of Section 3.3 (i.e., mutation and crossover operate on $\boldsymbol{\theta}^\star_f$ only), yet starting from a different initial population. In detail, the initial population is composed of $\boldsymbol{\theta}^\star$ and $n_{\text{pop}} - 1$ mutations of it, obtained by copying $\boldsymbol{\theta}^\star_{\text{attn}}$ and re-initializing $\boldsymbol{\theta}_f$ by sampling uniformly $[-1, +1]^{|\boldsymbol{\theta}_f|}$.

We conducted an experimental campaign of 5 evolutionary runs, lasting $n_{\text{evals}} = 20\,000$ fitness evaluations each, with one larger version of each of the two shapes used in the previous experiments. We fine-tuned the best individual of every run as explained above, freezing and transferring its attention module to the corresponding smaller or larger morphology, i.e., from biped-small to biped-large , from biped-large to biped-small, from comb-small to comb-large , and from comb-large to comb-small. For fine-tuning, we set $n_{\text{evals}} = 10\,000$ fitness evaluations as termination criterion, as our goal is to quickly adapt to a new task starting from pre-optimized components (i.e., the attention module). We summarize the results in Figure 7: for each morphology, we compare $\overline{v}_x$ (in terms of median ± standard deviation) for fine-tuning (Fine-tune) and from-scratch optimization (From-scratch). For a fair comparison, we plot the results for From-scratch over 30 000 fitness evaluations, since every Fine-tune run is the outcome of



(a) (b) (c) (d) (e)

Figure 5: Time-lapse showing locomotion for a sample VSR and the corresponding attention matrices. The color of each voxel encodes the ratio between its current area and its rest area: red for < 1, yellow ≈ 1, green > 1; the circular sectors drawn at the center of each voxel indicate the current sensor readings. The color of each attention score tells the strength of the compatibility between inputs, and provides an interpretation for what is driving the behavior of the VSR.

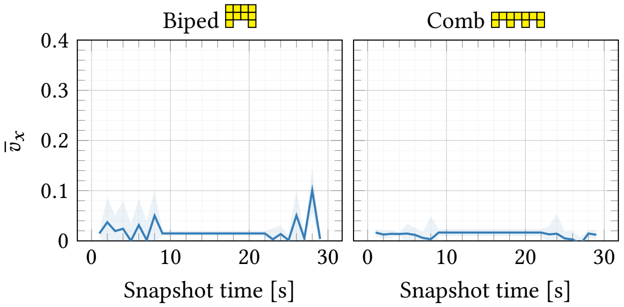


Figure 6: Median ± standard deviation (solid line and shaded area) of the average velocity for the best individuals found during each evolutionary run, with the attention matrix frozen at different time steps of the simulation (on x axis in s), obtained with the two shapes. Self-attention evolves to be a necessary component of the controller, and ablating it results in abysmal performance.

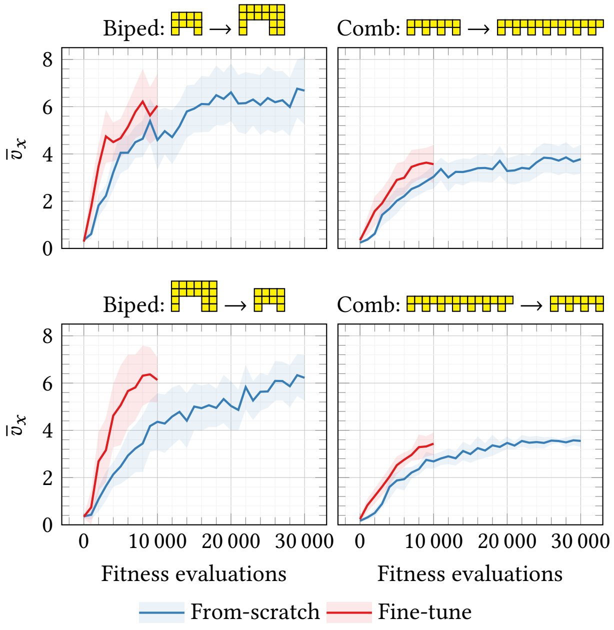


Figure 7: Median ± standard deviation (line and shaded area) of the average velocity for the best individuals found during each evolutionary run, with two controller types (color), with optimization from scratch (dashed line) or fine-tuning a pre-optimized attention (solid line) on a different morphology. Attention evolves an inductive bias that, generally, represents invariant features for the locomotion task.

an evolution with 20 000 fitness evaluations plus fine-tuning with 10 000.

From Figure 7, we see that Fine-tune succeeds in converging much faster. Moreover, the curves suggest that Fine-tune would benefit from a longer re-optimization; however, we remark that the goal of this study is not to reach the best performance, but to converge cheaply. We visually inspected the behaviors of the Fine-tune individuals and found them to be adapted for a locomotion task and consistent with the behaviors observed in Section 4.1.1. We made videos available at https://softrobots.github.io for the best Fine-tune individuals[4].

We conclude that it is often possible to re-use an evolved attention module for a smaller or larger morphology, after a fine-tuning stage. That result is hopeful in the context of future autonomous robotic ecosystems: we expect new robots to be assembled from pre-optimized components, so that they are, at the same time, proficient at new tasks and computationally cheaper than optimizing from scratch.

We believe the result is relevant, as, to date, [38] is the only other work addressing generalization to different soft robot morphologies by means of EC. Albeit ground-breaking, Kriegman et al. [38] achieved generalization by computing the fitness function over

[4] Numbers 3, 4, 5, and 6.

three different morphologies at every evaluation, which might not be feasible in a fully autonomous setting. We achieve generalization with no penalty during evolution, at the cost of introducing a fine-tuning stage. Finally, we remark it could be argued that other works, e.g., [30, 51, 69], achieved generalization to unseen morphologies via techniques more sample-efficient than EC, most notably Reinforcement Learning (RL). While those advances are indeed noteworthy, there are reasons to use EC in the first place: empirically, we find that RL algorithms are notoriously more unstable than evolution; indeed, there is a recent body of literature that shows how even simple EAs can achieve performance competitive to state-of-the-art RL systems [55, 57, 62], at the benefit of less complexity and less sensitivity to hyperparameters.

It is worth noting that evolution and learning can complement each other very effectively [17]. For example, in evolutionary robotics, learning augments evolution by allowing newborn controllers to adapt more quickly to their bodies [24, 41]. It is clear that self-attention is not a form of learning: in particular, there is no memory or state, as every attention matrix is computed just from current observations. We argue that self-attention evolves to have an instantaneous time-scale of adaptation; similarly to reflexes, it immediately reacts to sensory perceptions. Under this light, self-attention belongs to the "System 1" mode of thought: as put forward in the seminal work of Kahneman [32], System 1 thinking is fast, instinctive, and emotional. System 1 then exists at the level of the sub-conscious. That hypothesis is in line with the nature of attention in biology [10, 67].

To validate that hypothesis, we perform an experiment by slowing down the dynamics of attention. In particular, given the attention matrix $\boldsymbol{A}$ as defined in Equation (2), that depends only on current input, we define the matrix $\boldsymbol{A'}^{(k)}$ as:

$$\boldsymbol{A'}^{(k)} = \begin{cases} \alpha \boldsymbol{A} + \eta \boldsymbol{A'}^{(k-1)} & \text{if } k > 0 \\ \boldsymbol{A} & \text{if } k = 0 \end{cases} \tag{5}$$

where $\alpha, \eta \in [0, 1]$. α acts as a *learning rate*, weighting the current attention, while η acts as a *forget rate*, weighting the attention compounded from the previous time steps. In the following, we substitute $\boldsymbol{A}$ with $\boldsymbol{A'}^{(k)}$ in Equation (1), i.e., $\boldsymbol{A'}^{(k)}$ is the effective attention matrix at time step k, i.e., at time $t = k\Delta t$. As a side note, if we enforce $\alpha + \eta = 1$, Equation (5) resembles the "associative weights" memory of Ba et al. [1].

In the following, we evolve α and η as part of the genotype. To enforce $\alpha, \eta \in [0, 1]$, when mapping a genotype $\boldsymbol{\theta}$ into a robot phenotype, we set both to be the absolute value clipped at 1 of the corresponding genes. In doing so, we let evolution discover what is the fittest value for α and η and, as a result, what is the optimal balance between current information and past information in self-attention. In a certain sense, we are "meta-evolving" self-attention.

With those settings, we conducted an experimental campaign of 5 evolutionary runs with the shapes of Section 4.1.1. We found the resulting best individuals to be not significantly different from those of Section 4.1.1, in terms of effectiveness and qualitative analysis of self-attention, so we simply show in Figure 8 the evolution of α and η. We notice a clear trend: α evolves to be 1, while η evolves to approach 0. In other words, self-attention evolves to keep all the present information and retain little from the past. Those results mostly confirm our hypothesis that evolution leads self-attention to have an instantaneous time-scale of adaptation, akin to an instinct.

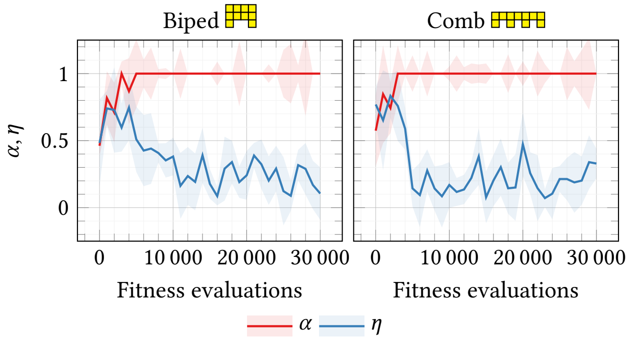


Figure 8: Median ± standard deviation (solid line and shaded area) of the α and η values of Equation (5) for the best individuals found during each evolutionary run, obtained with two shapes. Self-attention evolves to quickly forget past information and attend to the present.

Through those tests, we can answer to RQ2: self-attention works because it evolves to represent an inductive bias that is useful for generalization; at the same time, self-attention evolves to be an instinctive adaptation mechanism, allowing the controller to react quickly to changes in the inputs.

5 CONCLUSION

In the path toward autonomous robotic ecosystems, full exploitation of modularity remains a roadblock [72]. Minimizing the need for communication among robot modules would greatly facilitate the disassembly of robots in components and their reassembly in different forms, to cope, e.g., with different tasks.

Considering the case of VSRs, that, a priori, enjoy a high degree of modularity (in both bodies and brains), those points is precisely what the paper demonstrates:

(i) we can evolve controllers that dispense with inter-module communication for a locomotion task on hilly terrains;
(ii) we can achieve generalization to unseen morphologies, after a short fine-tuning with evolution.

To do so, we employ a local self-attention mechanism—a form of adaptive weights—to let every voxel adapt to its local perceptions, and optimize the controller parameters (which are the same for all the voxels) with EC. On the other side, an MLP baseline is not proficient in the same task. We also observe that self-attention evolves to be an instinctive adaptation mechanism, allowing the controller to quickly react to input changes.

Yet, it is unclear how our results extend beyond the scope of modular soft robots, including modular rigid robots. Future work will develop on these ideas to investigate how critical the soft body dynamics are in supporting full controller modularity, and how to leverage self-attention to evolve communication patterns tailored to different voxels.

ACKNOWLEDGMENTS

The experiments in this work were conducted using virtual machines provided by Google Cloud Platform granted to F.P.

REFERENCES

[1] Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. *Advances in Neural Information Processing Systems* 29 (2016), 4331–4339.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Jagdeep Bhatia, Holly Jackson, Yunsheng Tian, Jie Xu, and Wojciech Matusik. 2021. Evolution gym: A large-scale benchmark for evolving soft robots. *Advances in Neural Information Processing Systems* 34 (2021).

[4] Douglas Blackiston, Emma Lederer, Sam Kriegman, Simon Garnier, Joshua Bongard, and Michael Levin. 2021. A cellular platform for the development of synthetic living machines. *Science Robotics* 6, 52 (2021), eabf1571.

[5] Edgar Buchanan, Léni K Le Goff, Wei Li, Emma Hart, Agoston E Eiben, Matteo De Carlo, Alan F Winfield, Matthew F Hale, Robert Woolley, Mike Angus, et al. 2020. Bootstrapping Artificial Evolution to Design Robots for Autonomous Fabrication. *Robotics* 9, 4 (2020), 106.

[6] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 5 (2021), 1–32.

[7] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. 2014. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution* 7, 1 (2014), 11–23.

[8] Jinyoung Choi, Beom-Jin Lee, and Byoung-Tak Zhang. 2017. Multi-focus attention network for efficient deep reinforcement learning. In *Workshops at the thirty-first AAAI conference on artificial intelligence.*

[9] David Johan Christensen, Ulrik Pagh Schultz, and Kasper Stoy. 2013. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robotics and Autonomous Systems* 61, 9 (2013), 1021–1035.

[10] Asher Cohen and Robert D Rafal. 1991. Attention and feature integration: Illusory conjunctions in a patient with a parietal lobe lesion. *Psychological science* 2, 2 (1991), 106–110.

[11] Sylvain Cussat-Blanc, Kyle Harrington, and Wolfgang Banzhaf. 2019. Artificial gene regulatory networks—a review. *Artificial life* 24, 4 (2019), 296–328.

[12] Kenneth A De Jong. 2006. *Evolutionary Computation: A Unified Approach.* MIT Press.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Jon Driver. 2001. A selective review of selective attention research from the past century. *British Journal of Psychology* 92, 1 (2001), 53–78.

[16] AE Eiben. 2021. Real-World Robot Evolution: Why Would it (not) Work? *Frontiers in Robotics and AI* (2021), 243.

[17] AE Eiben and Emma Hart. 2020. If it evolves it needs to learn. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion.* 1383–1384.

[18] Andres Faiña. 2021. Evolving Modular Robots: Challenges and Opportunities. In *ALIFE 2021: The 2021 Conference on Artificial Life.* MIT Press.

[19] Andrea Ferigo, Giovanni Iacca, Eric Medvet, and Federico Pigozzi. 2021. Evolving Hebbian Learning Rules in Voxel-based Soft Robots. (2021).

[20] Jerry A Fodor. 1983. *The modularity of mind.* MIT press.

[21] Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[22] Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. 2021. Learning graph cellular automata. *Advances in Neural Information Processing Systems* 34 (2021).

[23] Roderich Groß, Elio Tuci, Marco Dorigo, Michael Bonani, and Francesco Mondada. 2006. Object transport by modular robots that self-assemble. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2558–2564.

[24] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. 2021. Embodied Intelligence via Learning and Evolution. *arXiv preprint arXiv:2102.02202* (2021).

[25] Matthew F Hale, Mike Angus, Edgar Buchanan, Wei Li, Robert Woolley, Léni K Le Goff, Matteo De Carlo, Jon Timmis, Alan F Winfield, Emma Hart, et al. 2020. Hardware design for autonomous robot evolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 2140–2147.

[26] Helmut Hauser, Auke J Ijspeert, Rudolf M Füchslin, Rolf Pfeifer, and Wolfgang Maass. 2011. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics* 105, 5 (2011), 355–370.

[27] Jonathan Hiller and Hod Lipson. 2012. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics* 28, 2 (2012), 457–466.

[28] Kazuya Horibe, Kathryn Walker, and Sebastian Risi. 2021. Regenerating Soft Robots Through Neural Cellular Automata.. In *EuroGP.* 36–50.

[29] Toby Howison, Simon Hauser, Josie Hughes, and Fumiya Iida. 2020. Reality-assisted evolution of soft robots through large-scale physical experimentation: a review. *Artificial Life* 26, 4 (2020), 484–506.

[30] Wenlong Huang, Igor Mordatch, and Deepak Pathak. 2020. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning.* PMLR, 4455–4464.

[31] Michał Joachimczak, Reiji Suzuki, and Takaya Arita. 2016. Artificial Metamorphosis: Evolutionary Design of Transforming, Soft-Bodied Robots. *Artificial Life* 22, 3 (2016), 271–298. https://doi.org/10.1162/ARTL_a_00207 PMID: 27139940.

[32] Daniel Kahneman. 2011. Thinking fast and slow. *New York: Farrar, Straus and Giroux* (2011).

[33] Shadi Tasdighi Kalat, Siamak G Faal, and Cagdas D Onal. 2018. A decentralized, communication-free force distribution method with application to collective object manipulation. *Journal of Dynamic Systems, Measurement, and Control* 140, 9 (2018), 091012.

[34] Akiya Kamimura, Haruhisa Kurokawa, E Toshida, Kohji Tomita, Satoshi Murata, and Shigeru Kokaji. 2003. Automatic locomotion pattern generation for modular robots. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 1. IEEE, 714–720.

[35] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169* (2021).

[36] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. 2020. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences* 117, 4 (2020), 1853–1859.

[37] Sam Kriegman, Nick Cheney, and Josh Bongard. 2018. How morphological development can guide evolution. *Scientific reports* 8, 1 (2018), 13934.

[38] Sam Kriegman, Amir Mohammadi Nasab, Douglas Blackiston, Hannah Steele, Michael Levin, Rebecca Kramer-Bottiglio, and Josh Bongard. 2021. Scale invariant robot behavior with fractals. *arXiv preprint arXiv:2103.04876* (2021).

[39] Gongjin Lan, Matteo De Carlo, Fuda van Diggelen, Jakub M Tomczak, Diederik M Roijers, and Agoston E Eiben. 2021. Learning directed locomotion in modular robots with evolvable morphologies. *Applied Soft Computing* 111 (2021), 107688.

[40] Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J Bentley, Samuel Bernard, Guillaume Beslon, David M Bryson, et al. 2020. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial life* 26, 2 (2020), 274–306.

[41] Jie Luo, Aart Stuurman, Jakub M Tomczak, Jacintha Ellers, and Agoston E Eiben. 2021. The Effects of Learning in Morphologically Evolving Robot Systems. *arXiv preprint arXiv:2111.09851* (2021).

[42] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[43] Georg Martius, Ralf Der, and Nihat Ay. 2013. Information driven self-organization of complex robotic behaviors. *PloS one* 8, 5 (2013), e63400.

[44] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Giulio Fidel. 2020. Evolution of distributed neural controllers for voxel-based soft robots. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference.* 112–120.

[45] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. 2020. 2D-VSR-Sim: A simulation tool for the optimization of 2-D voxel-based soft robots. *SoftwareX* 12 (2020).

[46] Eric Medvet, Alberto Bartoli, Federico Pigozzi, and Marco Rochelli. 2021. Biodiversity in evolved voxel-based soft robots. In *Proceedings of the Genetic and Evolutionary Computation Conference.* 129–137.

[47] Tom M Mitchell. 1980. *The need for biases in learning generalizations.* Department of Computer Science, Laboratory for Computer Science Research ....

[48] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. 2020. Growing Neural Cellular Automata. *Distill* (2020). https://doi.org/10.23915/distill.00023 https://distill.pub/2020/growing-ca.

[49] Stefano Nolfi and Dario Floreano. 2000. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines.* MIT press.

[50] Dai Owaki, Shun-ya Horikiri, Jun Nishii, and Akio Ishiguro. 2021. Tegotae-Based Control Produces Adaptive Inter-and Intra-limb Coordination in Bipedal Walking. *Frontiers in neurorobotics* 15 (2021), 47.

[51] Deepak Pathak, Chris Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. 2019. Learning to control self-assembling morphologies: a study of generalization via modularity. *arXiv preprint arXiv:1902.05546* (2019).

[52] Rolf Pfeifer and Josh Bongard. 2006. *How the body shapes the way we think: a new view of intelligence.* MIT press.

[53] Federico Pigozzi and Eric Medvet. 2022. Evolving Modularity in Soft Robots through an Embodied and Self-Organizing Neural Controller. *Artificial Life* (2022).

[54] J Pena Queralta, Cassandra McCord, Tuan Nguyen Gia, Hannu Tenhunen, and Tomi Westerlund. 2019. Communication-free and index-free distributed formation control algorithm for multi-robot systems. *Procedia Computer Science* 151 (2019), 431–438.

[55] Sebastian Risi and Kenneth O Stanley. 2019. Deep neuroevolution of recurrent and discrete world models. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 456–462.

[56] Daniela Rus and Michael T Tolley. 2015. Design, fabrication and control of soft robots. *Nature* 521, 7553 (2015), 467.

[57] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).

[58] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[59] Lawrence Shapiro. 2019. *Embodied cognition*. Routledge.

[60] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. 2008. *Springer handbook of robotics*. Vol. 200. Springer.

[61] Karl Sims. 1994. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 15–22.

[62] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567* (2017).

[63] Shyam Sudhakaran, Djordje Grbic, Siyan Li, Adam Katona, Elias Najarro, Claire Glanois, and Sebastian Risi. 2021. Growing 3D Artefacts and Functional Machines with Neural Cellular Automata. *arXiv preprint arXiv:2103.08737* (2021).

[64] Xin Sui, Hegao Cai, Dongyang Bie, Yu Zhang, Jie Zhao, and Yanhe Zhu. 2020. Automatic generation of locomotion patterns for soft modular reconfigurable robots. *Applied Sciences* 10, 1 (2020), 294.

[65] Jacopo Talamini, Eric Medvet, Alberto Bartoli, and Andrea De Lorenzo. 2019. Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots. In *Artificial Life Conference Proceedings*. MIT Press, 574–581.

[66] Yujin Tang, Duong Nguyen, and David Ha. 2020. Neuroevolution of self-interpretable agents. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 414–424.

[67] Anne Treisman, Alfred Vieira, and Amy Hayes. 1992. Automaticity and preattentive processing. *The American journal of psychology* (1992), 341–362.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[69] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. 2018. Nervenet: Learning structured policy with graph neural networks. In *International Conference on Learning Representations*.

[70] Annie Wong, Thomas Bäck, Anna V Kononova, and Aske Plaat. 2021. Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches. *arXiv preprint arXiv:2106.15691* (2021).

[71] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677* (2020).

[72] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian. 2007. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 43–52.

[73] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. 2018. Deep reinforcement learning with relational inductive biases. In *International Conference on Learning Representations*.