Morphology Choice Affects the Evolution
of Affordance Detection in Robots

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

Stephanie Woodman
Yale University
New Haven, Connecticut, USA
stephanie.woodman@yale.edu

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

Rebecca Kramer-Bottiglio
Yale University
New Haven, Connecticut, USA
rebecca.kramer@yale.edu

Josh Bongard
University of Vermont
Burlington, Vermont, USA
jbongard@uvm.edu

		passable		impassable	
		affordance detection		**affordance detection**	
		correct	incorrect	correct	incorrect
action	correct				
	incorrect				

Figure 1: Our robots (here, a starfish-shaped robot) predict their affordances (such as the passability of an aperture given their bodies) and take appropriate actions (attempt to pass through passable apertures). For the sake of the former, voxels vote (on the basis of their local sensor readings) for impassability (red color) or passability (blue color). We evolve robots that can discriminate between passable and impassable apertures while moving in the correct direction (on the other side of the aperture if it is passable, in front of the aperture if it is impassable).

ABSTRACT

A vital component of intelligent action is affordance detection: understanding what actions external objects afford the viewer. This requires the agent to understand the physical nature of the object being viewed, its own physical nature, and the potential relationships possible when they interact. Although robotics researchers have investigated affordance detection, the way in which the morphology of the robot facilitates, obstructs, or otherwise influences the robot's ability to detect affordances has yet to be studied. We do so here and find that a robot with an appropriate morphology can evolve to predict whether it will fit through an aperture with just minimal tactile feedback. We also find that some robot morphologies facilitate the evolution of more accurate affordance detection, while others do not if all have the same evolutionary optimization budget. This work demonstrates that sensation, thought, and action are necessary but not sufficient for understanding how affordance detection may evolve in organisms or robots: morphology must also be taken into account. It also suggests that, in the future, we may optimize morphology along with control in order to facilitate affordance detection in robots, and thus improve their reliable and safe action in the world.

CCS CONCEPTS

- **Computing methodologies** → *Evolutionary robotics*; • **Theory of computation** → *Evolutionary algorithms*; • **Computer systems organization** → *Neural networks*.

Federico Pigozzi, Stephanie Woodman, Eric Medvet, Rebecca Kramer-Bottiglio, and Josh Bongard

KEYWORDS
Affordance detection, embodied cognition, soft robots

1 INTRODUCTION AND RELATED WORKS

A vital component of cognition is affordance detection [10]: understanding what actions external objects afford the viewer (i.e., the agent). These include the viewer's body, its surrounding environment, and how these can potentially interact. If we were to deploy robots in hazardous and exotic environments [12, 26], we would need to endow such robots with a sense of what their bodies and environments afford them for their safe action. Consider the case of a robot that must pass through an aperture: not every aperture is passable for every robot body, and attempting to pass through an impassable aperture would be disastrous for the robot itself and the agents interacting with it. Thus, we want robots that behave according to correct predictions of affordances projected by objects in their environment.

Natural agents evolved the innate ability to detect affordances [34]. Indeed, neuroscience has argued that no cognition is otherwise possible without understanding the interactions between one's body and the environment [16, 20]. Not surprisingly, researchers have striven to endow artificial agents with the same ability. Different communities developed different concepts and terminologies: self-modeling in robotics [2, 3, 6], world models [11] in machine learning, intrinsic motivation [27], empowerment [17] and information-driven measures [23] in the cognitive sciences. More specifically, Slocum et al. [33], extending the work of [1] on the simplest behavior that raises issues of genuine cognitive interest, evolved robots to detect affordances by visually deciding which openings their bodies could and could not fit through.

Still, whether and how morphology affects a robot's ability to detect affordances is yet an unexplored issue, despite the *embodied cognition* paradigm [28], positing a deep entanglement between the brain, the body, and the environment, suggests that morphology can facilitate or obstruct the ability to evolve a behavior for an animal or robot.

Here, we set out to study morphology choice in the evolution of affordance detection and consider voxel-based soft robots [13] as our instance of embodied agents. Voxel-based soft robots are aggregations of cubic blocks of soft material: given that the interactions between a soft robot's body and its environment are hard to predict [31], affordance detection is likely to be more arduous for soft robots. These robots, equipped with minimal tactile feedback, decide whether or not their bodies can fit through an aperture, and, if so, then attempt to pass through it. We experiment with three morphologies and optimize the parameters of their closed-loop controllers with an Evolutionary Algorithm (EA) [7] since EAs allow for the exploration of search spaces unencumbered by a priori assumptions.
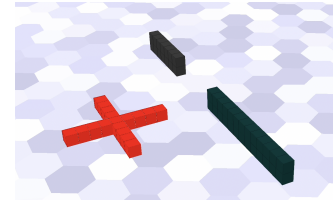

Figure 2: A snapshot from the Voxcraft simulator. Red voxels are active voxels (i.e., actuators), while black voxels are immovable and do not actuate.

We show experimentally that the choice of morphology plays a key role: some morphologies facilitate the evolution of affordance detection, while others do not. In particular, simpler morphologies may be less suitable for detecting affordances. Looking forward, we envision that the joint optimization of morphology and control may facilitate the evolution of affordance detection in robots and thus improve their reliable and safe action in the world.

2 EMBODIED AGENT MODEL

As embodied agents, we consider voxel-based soft robots. Soft robots have a number of advantages over rigid robots, such as deforming their bodies [32] for, e.g., squeezing through tight spaces [4] or underwater locomotion [5]. However, the interactions between a soft robot's body and its environment are more complex and harder for the robot to predict [31]. For example, a soft robot may be able to deform itself to fit through a narrow aperture but not be able to determine what that deformation is. For this reason, we herein investigate the evolution of soft robot affordance detection, as it is likely to be more difficult compared to rigid robots.

First introduced by Hiller and Lipson [13], voxel-based soft robots are aggregations of cubic blocks (the *voxels*) of soft material. Actuation is volumetric: the voxels alter their volume according to a control signal and contact forces with other bodies. Behavior at the robot level emerges from the combined action of voxels contracting and expanding. By virtue of their many degrees of freedom, these robots are expressive yet challenging to control.

2.1 Mechanical model: the Voxcraft simulator

We simulate voxel-based soft robots and their environment using the Voxcraft soft-body physics engine [21], the GPU-accelerated version of Voxelyze [14]. Voxcraft simulates each voxel as a point mass connected to up to six neighboring point masses with Euler-Bernoulli beams. Every voxel has a specific temperature that, at time step t, changes according to:

$$\Delta T = \alpha a_t \qquad (1)$$

where $\alpha \in \mathbb{R}$ is a fixed amplitude across the robot and $a_t \in \mathbb{R}$ is the temperature change (actuation) output by the controller embedded in the voxel. A beam, in turn, sets its resting length to the average of the current temperatures of the two voxels it connects: the higher the two temperatures, the more the beam stretches, the lower the two temperatures, the more the beam contracts. The robot's environment contains objects constructed from voxels with unvarying temperatures (Figure 2).

Following Kriegman et al. [18], we set $\alpha = 14.4714$ and simulate materials of $10\,\mathrm{kg} \cdot \mathrm{m}^{-3}$ density, Young's modulus of $104\,\mathrm{Pa}$, Poisson's ratio of 0.5, coefficient of static friction of 1, and coefficient of dynamic friction of 0.5. Each voxel is $0.01\,\mathrm{m}$ in length.

2.2 Sensing

We equip the voxels with touch, floor, and velocity sensors, and a central pattern generator. Touch sensors perceive whether the voxel is contacting a body other than the floor (e.g., a wall) or not and return 1 if yes, and −1 if not. Floor sensors perceive whether the voxel is touching the floor or not and return 1 if yes, and −1 if not. Velocity sensors perceive the x- and y-velocity of the center of mass of the voxel. Finally, a central pattern generator—relevant to animal and robotic locomotion alike [15]—inputs the value of $\sin(-2\pi t)$ at time step t. Each voxel thus has 5 sensors.

2.3 Controller

We formulate the controller architecture to enable the robot to predict affordances (such as the passability of an aperture) and take appropriate actions (attempt to pass through passable apertures) by local processing of sensed interactions between its voxels and external objects.

We do so by beginning with the distributed controller proposed in [24], consisting of artificial neural networks (a closed-loop system), one for every voxel. We adopt the "homogeneous" variant presented in [25], where the neural networks share the same parameters: because of the more compact search space, such homogeneous representation is comparable to one where parameters are different for every voxel. Moreover, as seen in Pigozzi et al. [29], parameter sharing makes the controller architecture agnostic with respect to the morphology: the number of input and output neurons and, thus, the number of parameters does not depend on the arrangement and number of voxels in the morphology.

There are 17 input and 2 output neurons (Figure 3). At time step t, every neural network outputs $a_t \in [-1, 1]$ and $v_t \in [-1, 1]$, a *vote* that is a confidence score of the voxel on what the affordance is at the robot level, according to its local observation.

Every neural network takes as input its local observation $x_t \in \mathbb{R}^{17}$: the 5 sensor values described in Section 2.2, but also, to introduce a form of communication among the voxels, in the a_{t-1} outputs from its six adjacent neighbors of the previous time step and the v_{t-1} outputs from its six adjacent neighbors of the previous time step. If a neighbor is absent, we set those inputs to 0.

In preliminary experiments, we found that memory helped the evolution of successful behaviors. So, in the reported experiments, we implement each neural network as an Elman network, the simplest instance of stateful neural network [9]:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h) \tag{2}$$

$$y_t = \sigma(W_y h_t + b_y) \tag{3}$$

where $h_t \in \mathbb{R}^{n_{\mathrm{mem}}}$ is a memory vector and $\sigma(\cdot)$ is an element-wise activation function. An Elman network is a three-layer neural network with a recurrence in the second layer. The first layer has one input neuron for every input; the second layer (Equation (2)) computes a non-linear combination of the inputs and the memory of the previous time step h_{t-1} to output h_t, which it then feeds to the third
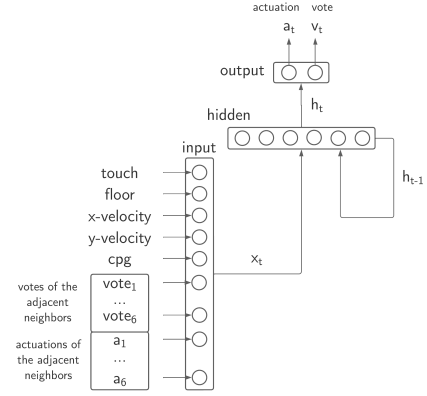

Figure 3: The architecture of the Elman network. There is one such network embedded within every voxel. We list the outputs on top and the inputs to the left: these include the vote and the actuation of the six adjacent neighboring voxels of the previous time steps.

layer (Equation (3)) for the final output. h_t then becomes the memory vector for the next time step and the network unfolds over time, hence its recurrent nature. $W_x \in \mathbb{R}^{n_{\mathrm{mem}} \times 17}$, $W_h \in \mathbb{R}^{n_{\mathrm{mem}} \times n_{\mathrm{mem}}}$, $b_h \in \mathbb{R}^{n_{\mathrm{mem}}}$, $W_y \in \mathbb{R}^{2 \times n_{\mathrm{mem}}}$, and $b_y \in \mathbb{R}^2$ are the evolvable parameters of the network. We summarize the architecture of our Elman network in Figure 3.

After preliminary experiments, we set $n_{\mathrm{mem}} = 6$, $h_0 = 0$, and tanh as activation function (to ensure the output lies in $[-1, 1]$).

We optimize a robot for a task by optimizing the vector $\theta = \begin{bmatrix} W_x & W_h & b_h & W_y & b_y \end{bmatrix} \in \mathbb{R}^p$ of parameters that specify every neural network in the robot. For our setting, $p = 158$.

2.3.1 Voting mechanism. At every time step t, the robot performs affordance detection by predicting whether the aperture in front of it is passable or not. It does so as follows.

(a) If voxel i's vote $v_{i,t}$ is greater than 0, this denotes the voxel predicting that the aperture is passable, and we set $v_{i,t}$ to 1. Otherwise, we set $v_{i,t}$ to 0, denoting a prediction that the aperture is impassable;

(b) We compute the majority vote across all voxels and store it in v_t. If there is a split vote, we set v_t to 1.

3 EXPERIMENTAL PROCEDURE

Our goal is to answer the following research questions:

RQ1 can affordance detection be evolved for soft robots?
RQ2 If yes, do some robot morphologies allow for the evolution of more accurate affordance detection than others?

To this end, we evolved three different robot morphologies to perceive whether an aperture is passable and whether they take the appropriate action based on this conclusion (Section 3.2).
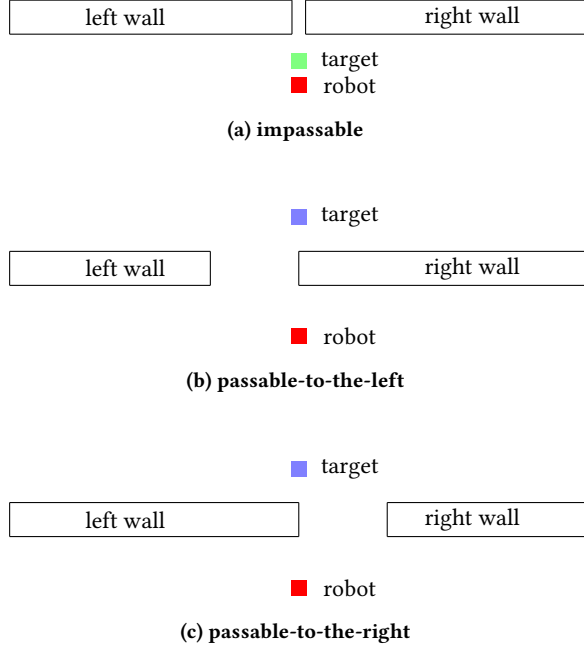
(a) impassable

(b) passable-to-the-left

(c) passable-to-the-right

Figure 4: The robot, with body length l_{body} voxels, faces three different environments. In the first, it must detect that the aperture is impassable, and avoid attempting to move through it: it should minimize its distance from the green point at the end of the simulation. In the second and third environments, the robot should detect that the aperture is passable, pass through it, and minimize its distance from the blue point at the end of the simulation.

3.1 Robots

We experimented with three different robot shapes: a 5×5 square of voxels (the "flatworm") , a 9×9 four-legged "starfish" , and a 6×6 "gecko" . The controller inputs, outputs, and parameters are the same for all voxels in all three shapes. The three shapes entail different morphologies: the flatworm has no salient features on its body, the starfish has four limbs, while the gecko has many protrusions that it can potentially use as hooks (hence the name, because geckos climb on trees through tiny hooks on their finger palms).

3.2 Task

We initially place the robot in front of an aperture formed by two walls (Figure 4). We evaluate the robot multiple times in the presence of apertures of differing widths and positions. The robot's task is to move to a target position on the far side of the aperture if it judges it passable, and to a target position before the aperture (but not at) if it judges it impassable. We build the walls with voxels having the same material properties as the robot. We design the task such that the decision to navigate the aperture or not is dependent on the aperture width and the robot's body length.

If we evaluated each θ in one environment with a passable aperture and another with an impassable one, the robot could evolve degenerate behaviors rather than affordance detection. For example, the robot could evolve to reach out toward just one of the two walls. Since the wall would be at a different relative position from the robot in the two environments, different parts of the robot would collide with the wall in the two environments. The robot could evolve the ability to predict whether the aperture is passable just from this information, but fail to generalize to new environments. So, we evaluated each θ in three environments as shown in Figure 4.

For a robot with a body length l_{body} voxels long, we set the two walls l_{body} voxels in front of the robot's most anterior voxel. In the first environment, we set the aperture width to 1 voxel, making it impassable, and center it in front of the robot. In the second environment, we set the aperture width to $l_{body} - 1$ voxels, making it passable, because the robot can move and deform itself. We offset the aperture to the left by $\lceil l_{body}/2 \rceil$ voxels. The third environment is the same as the second except that we offset the aperture $\lceil l_{body}/2 \rceil$ voxels to the right.

3.3 Fitness functions

As we wish to evolve robots that attempt to pass through passable apertures, and also explicitly predict whether apertures are passable or not, we must formulate two separate fitness objectives. We thus introduce a bi-objective evolutionary optimization problem with a "locomotion" fitness objective f_{loc} and an "affordance detection" fitness objective f_{aff}.

For the i-th of the three environments, we measure the distance d_i of the robot's center of mass from the relevant target position at the end of the simulation. Given that our EA requires bounded objective values (see Section 3.4), we set $d_i = d_{max}$ if $d_i > d_{max}$. After preliminary experiments, we found $d_{max} = 5$ to be sufficient. Then, $d_i \in [0, 5]$.

For the i-th of the three environments, we also measure:

$$\text{acc}_i = 1 - \frac{1}{t_{final}} \sum_{t=t_{contact}}^{t_{final}} |v_t - g_i| \qquad (4)$$

where t_{final} is the total number of time steps in the simulation, $t_{contact}$ is the time step at which the robot first touches the walls, v_t is the robot's prediction about the aperture's passability (see Section 2.3.1), and $g_i \in \{0, 1\}$ is the ground truth for the i-th environment (0 for impassable and 1 for passable) and is available only to the fitness function, not the robot. The summation on the right of Equation (4) counts the number of time steps that the robot's majority vote (see Section 2.3.1) differs from the ground truth. Thus, Equation (4) is the accuracy for the binary classification problem of discriminating between passable and impassable environments. Then, $\text{acc}_i \in [0, 1]$. Given that we equip the robot with minimal tactile sensors (see Section 2.2), no vote on the passability of an aperture is meaningful without first touching it. With this in mind, we start counting votes only after the robot has perceived its first contact with the walls at $t_{contact}$; in this way, dividing by t_{final} indirectly rewards the individuals for approaching the walls as soon as possible. If the robot never touches the walls, we set $\text{acc}_i = 0$.

After evaluating the robot in all three environments, we set f_{loc} and f_{aff} to the worst of the robot's three actions and the worst of

its three attempts at affordance detection, respectively:

$$f_{\text{loc}} = \max_{i=1,2,3} d_i \tag{5}$$

$$f_{\text{aff}} = \min_{i=1,2,3} a_i \tag{6}$$

to reward robustness across all environments.

Finally, we found some individuals achieved high fitness while ignoring the aperture by crashing into the walls to climb over them, or by circumventing the walls altogether. To punish these exploitative behaviors, we assigned the worst possible fitness values (i.e., $d_i = 5$ and $\text{acc}_i = 0$) to any individual that, at any time step, has more than half of its voxels outside of the bounding volume of the environment. The bounding volume of the environment is $3l_{\text{body}}$ voxels wide, $\left\lceil \frac{l_{\text{body}}}{3} \right\rceil + 1$ voxels tall, and $5l_{\text{body}}$ voxels deep.

3.4 Evolutionary algorithm

EAs are effective multi-objective optimizers [36] and have proved competitive on continuous control tasks [30, 35]. We employed the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [8], an established bi-objective EA. NSGA-II evolves a population of individuals by iterations according to a $\mu+\lambda$ generational model. We initialize individuals at the very first iteration by uniform sampling from the interval $[-1, 1]^p$. At each iteration, NSGA-II sorts the population into Pareto shells according to the two objectives f_{loc} and f_{aff}. As the genetic operator, we used Gaussian mutation of mean $\mathbf{0} \in \mathbb{R}^p$ and step-size σ_{mut}; as the selection operator, we used crowded tournament selection of size n_{tour}. We iterate until n_{evals} fitness evaluations have been computed.

The output of the above procedure is a set of Pareto-optimal individuals, namely, those belonging to the first Pareto shell: of these, we label as "specialists" those that are the best in one particular objective. We then have two specialist individuals $\boldsymbol{\theta}_{\text{loc}}^*$ and $\boldsymbol{\theta}_{\text{aff}}^*$ according to our two objectives f_{loc} and f_{aff}. We are also interested in individuals that perform well for both objectives. Then, we label as "knee" the individual that is the closest (in the fitness space) to the center of the bounding box of the first Pareto shell. We then have one knee individual $\boldsymbol{\theta}^{\text{k}}$. After preliminary experiments and relying on our experience, we set $\mu = \lambda = 50$, $\sigma_{\text{mut}} = 0.35$, $n_{\text{tour}} = 5$, and $n_{\text{evals}} = 10\,000$ (corresponding to 200 iterations).

3.5 Experimental settings

For every experiment in this work, we performed 10 evolutionary runs by varying the random seed for the EA. We performed all statistical tests with the Mann-Whitney U rank test for independent samples. As soft-body physics engine, we employed Voxcraft [21], written in C++ and CUDA, and we developed a Python wrapper for the evolutionary optimization. Simulation frequency is 200 Hz and we set each simulation to 50 s of simulated time, for a total of $t_{\text{final}} = 10\,000$ time steps. All simulations are deterministic for a given random seed and controller. Each run took approximately 35 h on a Linux node with 4 cores at 2.10 GHz, 10 GB of CPU RAM, and 3 Nvidia Tesla GPUs. The code is publicly available at https://github.com/pigozzif/Voxcraft-python.git.

4 RESULTS

In the following, we provide results for both research questions.

4.1 RQ1: can affordance detection be evolved for soft robots?

Before delving into whether and how morphology affects a robot's ability to detect affordances, we first verify if we can evolve affordance detection in soft robots. To test this question, we conducted quantitative analysis by comparing the behaviors of robots controlled by random or evolved controllers, and qualitative analysis by inspecting their behaviors. For this question, we only considered the starfish robot.

4.1.1 Quantitative analysis. As performance indexes, we measure the fitness values for the specialist individuals, namely f_{loc}^* for $\boldsymbol{\theta}_{\text{loc}}^*$ and f_{aff}^* for $\boldsymbol{\theta}_{\text{aff}}^*$, but also, the fitness values for the knee individual, namely $f_{\text{loc}}^{\text{k}}$ and $f_{\text{aff}}^{\text{k}}$ for $\boldsymbol{\theta}^{\text{k}}$.

As baselines, we compare our methodology with random controllers. After preliminary experiments, we found controllers that were random in both the outputs (actuation and affordance detection) to perform very poorly: the vast majority of them failed to touch the walls and perform any affordance detection at all. Thus, for a fairer comparison, we devised two controls, one for every fitness function: one control where we randomize the controller output for actuation and one control where we randomize the controller output for affordance detection. We optimize the two controls using the same EA of Section 3.4 and compare them with evolution where neither of the outputs is randomized (i.e., our approach). As a result, we have three different treatments:

EvolvedBoth: our approach, where neither of the outputs is randomized;

EvolvedAffordance: at every time step and for every voxel of every individual, we discard the output a_t controlling temperature change (and, thus, locomotion) and substitute it with a random number in $[-1, 1]$;

EvolvedLocomotion: at every time step and for every voxel of every individual, we discard the output v_t voting for affordance and substitute it with a random number in $[-1, 1]$.

We remark that all three have the same controller architecture and thus the same size of the search space.

We summarize the results in Figure 5, which plots f_{loc} and f_{aff} in terms of median ± standard deviation over the course of evolution, for the specialists and the knee individuals. Moreover, we compare in Figure 6 the Pareto shells at the beginning, the halfway, and the end of evolution for three exemplar runs of the BothEvolved treatment. The utopia point (the one optimizing both objectives) sits in the upper-left corner.

Our EvolvedBoth treatment evolves affordance detection specialists that are better at affordance detection than random controllers, and locomotion specialists that are better at locomotion than random controllers. The picture is similar if we look at the knee individuals. The lines of Figure 5 also suggest that all treatments settle on a plateau and that more evolution would unlikely occur.

From Figure 5, we spot different patterns for the two fitness functions. Focusing on the affordance detection fitness (lower side of Figure 5), the BothEvolved treatment outclasses the two controls in terms of detection accuracy: by the end of evolution, its specialist individuals can, on average, correctly detect whether the aperture is passable or not at least 76.26 % of the time steps. That is significantly
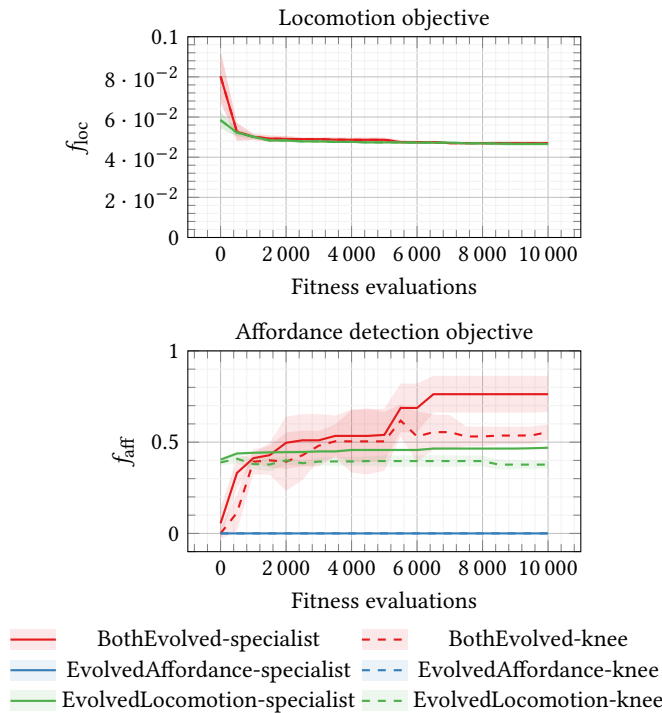
Figure 5: Median ± standard deviation (solid line and shaded area) for the two fitness values for the 10 specialist individuals and 10 knee individuals drawn from the 10 evolutionary runs, obtained with one experimental (EvolvedBoth) and two control (EvolvedAffordance and EvolvedLocomotion) treatments. The blue line on the upper plot is well above the upper bound and would have skewed the other lines. For the locomotion objective, specialists and knees mostly coincide. Our EvolvedBoth treatment evolves affordance detection significantly better than the two controls.

better than what the EvolvedLocomotion treatment evolves ($p < 0.0001$). If we look at the knee individuals, the affordance detection fitness is lower, but the p-value is still significant if we compare them with the EvolvedLocomotion treatment. Thus, evolution can find individuals that detect affordances significantly better than random controllers (the EvolvedAffordance line is stuck at the bottom because its specialists are unable to even reach the walls) and that, to some extent, are effective in both objectives.

Looking at the locomotion fitness (upper side of Figure 5), both the EvolvedBoth and the EvolvedLocomotion treatments evolve effective individuals (p-value not significant): they take less than 2 000 fitness evaluations to evolve the behavior of reaching the target. Moreover, performance is similar for specialists and knee individuals. We do not show the EvolvedAffordance line (the blue line) as it would be stuck at such a high value to skew the remaining lines. We visually inspected the behaviors evolved by EvolvedAffordance and, not surprisingly, found them to drift away over the floor until well exceeding the $d_{\max}$ upper bound we set in Section 3.3. Thus, we have to evolve also the locomotion behavior.

From Figure 6, we conclude that evolution is indeed taking place in our EvolvedBoth treatment: as evolution progresses, individuals in the Pareto shell from early during the evolutionary process are dominated by individuals in later Pareto shells.

4.1.2 Qualitative analysis. We analyze the evolved behaviors both in terms of locomotion and in terms of affordance detection.

Locomotion behaviors. For locomotion, an individual shows successful behavior if it reaches the target position and, in passable environments, correctly passes through the aperture in the walls.

We plot a top-view for the positions at the end of a simulation for 5 of the specialist individuals θ^*_{loc} in Figure 7: for the same individual, we then have a red point for the impassable environment, a blue point for the environment passable to the left, and a green point for the environment passable to the right. Moreover, the dashed lines stand for the position of the walls.

From the figure, we conclude that effective locomotion behavior did indeed evolve. Moreover, knee individuals behave nearly the same as specialists. After visual inspection, not only we found them to dodge the walls and slip in the aperture, but also to eventually bend their trajectories to come as close as possible to the target. With the exception of one run, which resulted in a specialist that does not reach the target, but passes through the aperture and then drifts away, most of them do reach and touch the target.

Affordance detection behavior. Our robots rely on minimal tactile feedback for sensing their environment. Then, an individual shows successful affordance detection behavior if a majority of the voxels not only stand for the correct vote even if only a few of them are sensing the walls but also if they retain the correct vote over time, possibly even after losing contact with the walls. Our distributed controller, having one neural network embedded inside each voxel, allows us to observe in real-time how voting shifts over time across the voxels.

We visually inspected the behaviors of the specialist individuals θ^*_{aff} and depict the time-lapse for an exemplar specialist in Figure 8 and include more at https://affordancesofrobots.github.io/affordancesofrobots/. Red voxels vote for impassability, whereas blue voxels vote for passability. In addition, we border a voxel in yellow if its touch sensor is firing (i.e., it is contacting a wall). Frame 1 corresponds to the snapshot at the first time step and we take the others at regular intervals of 1 500 time steps (approximately 7.5 s of simulated time).

We found the behaviors of the vast majority of specialist and knee individuals to consist of the same voting pattern. Those robots evolved to vote for passability by default and to switch to impassability after detecting the pattern of sensory signals corresponding to the impassable environment, or vice versa.

Referring to Figure 8, this successful voting pattern unfolds as follows:

Frame 1: At the beginning, starting off apart from the walls, the robot receives no tactile feedback from its environment. At this point, individuals do not necessarily cast the correct vote (blue color in Figure 8). The robot moves forward.

Frame 2: Once the robot contacts the walls with its arm, the touch sensor located on the tip starts firing and sends its feedback to the neural network embedded in the tip voxel.
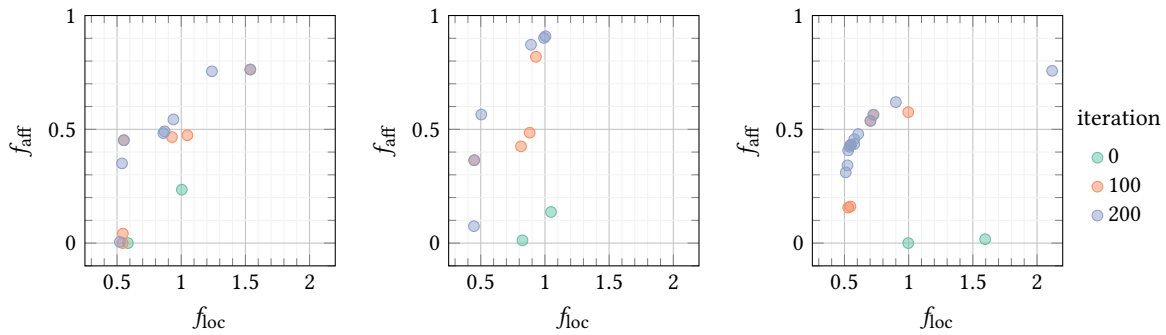
Figure 6: Pareto shells at different iterations (corresponding to the start, the halfway, and the end of evolution) for three exemplar evolutionary runs of the BothEvolved treatment. The utopia point (the one optimizing both fitness functions) sits in the upper-left corner. As evolution progresses, Pareto shells approach the utopia point more and more.
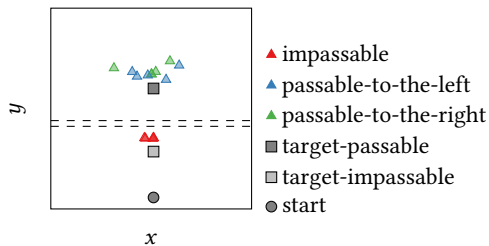

Figure 7: Top-view of the final positions of 5 among the locomotion specialists at the end of the simulation. The dashed lines stand for the position of the walls. In passable environments, the specialists evolve to eventually reach the target. In impassable environments, the robots stay on the near side of the walls: most of the red points overlap.

Frame 3: After a short "burn-in" period has elapsed, where voting happens randomly, we see the correct vote (the red color in Figure 8) propagate from the tip voxels to the rest of the body until flipping the majority towards the correct vote.

Frame 4: Crucially, a majority of the voxels retain the correct vote even after the touch sensor at the tip stops firing.

Through that evidence, we can answer positively to RQ1: we can evolve soft robots with the ability to detect their affordances.

4.2 RQ2: do some robot morphologies allow for the evolution of more accurate affordance detection than others?

In order to investigate whether and how morphology affects the evolution of affordance detection, we performed 10 additional evolutionary runs for each of the two additional morphologies: the "flatworm" and the "gecko" (see Section 3.1). We analyze and compare the results with different morphologies both quantitatively and qualitatively using the same methodology of Section 4.1. By virtue of the distributed controller, all three morphologies entail the same size of the search space, since the controller architecture does not depend on the number of inputs and outputs, as well as the number and arrangement of voxels.
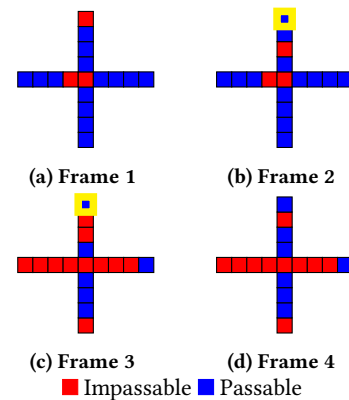

Figure 8: Time-lapse for the voting of an exemplar specialist in the impassable environment. Voxels bordered in yellow are in contact with the walls at that time. Frame 1 corresponds to the snapshot at the first time step and we take the others at regular intervals of 1 500 time steps (approximately 7.5 s of simulated time). Frame 1: when receiving no tactile feedback, the robot votes for passability. Frame 2: the robot contacts the wall and the touch sensor on the tip sends its feedback. Frame 3: the correct vote propagates from the tip to the rest of the body. Frame 4: a majority of the voxels retain the correct vote even after the robot loses contact with the wall.

We summarize the results in Figure 9, which plots f_{loc} and f_{aff} in terms of median ± standard deviation over the course of evolution, for both the specialist and the knee individuals.

4.2.1 Affordance detection. For f_{acc}, the three morphologies differ both in terms of efficiency (i.e., speed of convergence) and in terms of performance of the final specialist and knee individuals. On one side, geckos quickly converge to a very high fitness value, to the point of outclassing the starfishes; on the other side, the f_{acc}^* for flatworms plateaus at around 50 %: this is not statistically different from mean f_{acc}^* obtained by random controllers. Starfishes evolved f_{acc} values between those of flatworms and geckos. We found the p-values to be significant among different morphologies
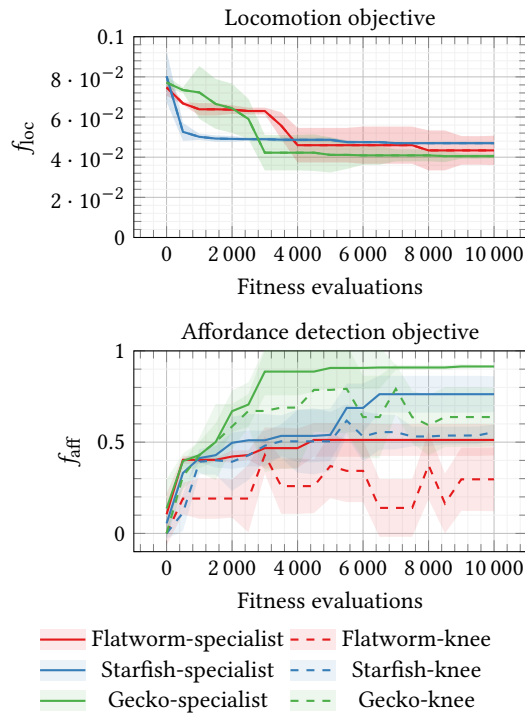
Figure 9: Median ± standard deviation (solid line and shaded area) of the two fitness values for the 10 specialist individuals and the 10 knee individuals drawn from the 10 evolutionary runs, obtained with three morphologies. For the locomotion objective, specialists and knees mostly coincide. The flatworm struggles to evolve affordance detection behavior, while the gecko outperforms the other morphologies.

for both specialists and knee individuals. Finally, we repeated the same EvolvedLocomotion control experiments of Section 4.2 for the gecko and flatworm morphologies: geckos are statistically better than their random counterparts on f_{acc}, while the same is not true for the flatworms (p-value not significant).

These results suggest that robots with more complex morphologies (e.g., the gecko), where we define complexity simply by the number of parts comprising the robot, might evolve more accurate affordance detection than those with simpler morphologies (e.g., the flatworm). To gain deeper insights into this intuition, we visually inspected the evolved behaviors. We found geckos to exhibit voting patterns similar to those discussed for the starfishes in Section 4.1.2: one or a few tips perceive the wall in front of them, propagate the correct vote across the body which retains it even in case the tips lose contact with the walls. Contrarily, we did not find signs of evolution in the flatworms: their voting patterns are random and chaotic. https://affordancesoftrobots.github.io/affordancesoftrobots/ shows one instance of this.

4.2.2 Locomotion. The difference among morphologies is less pronounced when we look at the locomotion fitness (upper side of Figure 9): the flatworm shape is thus not deficient per se, as it can evolve to locomote, but it is deficient for affordance detection.

These results suggest that different morphologies can facilitate the evolution of affordance detection behavior. One intuitive explanation is that, in the flatworm, it is more difficult for each voxel to distinguish itself: indeed, recalling that absent neighbors provide inputs of 0, a voxel can distinguish itself by the pattern of absent neighbors, but the majority of voxels in the flatworm actually have the same neighborhood pattern. Thus, different environments result in patterns of sensing that are more difficult to discriminate. Alternatively, a robot composed of more parts, or that is able to radically change its shape, may obstruct the evolution of affordance detection because self-contortions can more easily or rapidly alter the kinds of affordances afforded by objects in its environment. Altogether, these insights suggest that morphology plays an important role in the evolution of affordance detection. Albeit we here considered only a few robot body plans, and they remained fixed during evolutionary optimization, in the future, joint optimization of morphology and control may facilitate the evolution of affordance detection in robots and thus improve their reliable and safe action in the world.

5 CONCLUDING REMARKS

In the field of embodied cognition, no work to date has considered how morphology choice affects the ability of robots to detect their affordances: what actions their bodies and the environment afford them to accomplish. Considering the case of voxel-based soft robots—whose softness makes the interactions between their bodies and the environment less predictable—that is precisely the question this paper addresses. We optimize voxel-based soft robots to decide whether or not their bodies can fit through an aperture, and, if so, then attempt to pass through it. We optimize the parameters of their closed-loop controllers with an Evolutionary Algorithm (EA) and experiment with three morphologies. Our results show that different morphologies can facilitate the evolution of affordance detection behavior: some are effective, while others are not and we conjecture the reason to be their differing degrees of complexity. This work demonstrates that sensation and control are necessary but not sufficient for understanding how affordance detection evolves in organisms or robots: we must also take into account morphology.

Future work will validate these simulated results in reality, using soft voxel actuators [19]. New developments in stretchable electronics could enable sim2real transfer of the sensing [22], by patterning circuits that have light or contact sensors that would act as floor or neighbor sensors, while IMUs could calculate velocity.

REFERENCES

[1] Randall D Beer et al. 1996. Toward the evolution of dynamical neural networks for minimally cognitive behavior. *From animals to animats* 4 (1996), 421–429.
[2] Josh Bongard, Victor Zykov, and Hod Lipson. 2006. Resilient machines through continuous self-modeling. *Science* 314, 5802 (2006), 1118–1121.

[3] Boyuan Chen, Robert Kwiatkowski, Carl Vondrick, and Hod Lipson. 2022. Fully body visual self-modeling of robot morphologies. *Science Robotics* 7, 68 (2022).

[4] Nick Cheney, Josh Bongard, and Hod Lipson. 2015. Evolving soft robots in tight spaces. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*. ACM, 935–942.

[5] Francesco Corucci, Nick Cheney, Francesco Giorgio-Serchi, Josh Bongard, and Cecilia Laschi. 2018. Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics* 5, 4 (2018), 475–495.

[6] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

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

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[9] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[10] James J Gibson. 1977. The theory of affordances. *Hilldale, USA* 1, 2 (1977), 67–82.

[11] David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems* 31 (2018).

[12] M Hale, Edgar Buchanan Berumen, Alan Winfield, Jon Timmis, Emma Hart, Gusz Eiben, Wei Li, and Andy Tyrrell. 2019. The are robot fabricator: How to (re) produce robots that can evolve in the real world. In *International Society for Artificial Life: ALIFE2019*. York, 95–102.

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

[14] Jonathan Hiller and Hod Lipson. 2014. Dynamic Simulation of Soft Multimaterial 3D-printed Objects. *Soft Robotics* 1, 1 (2014), 88–101.

[15] Auke Jan Ijspeert. 2008. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks* 21, 4 (2008), 642–653.

[16] Julian Kiverstein and Matt Sims. 2021. Is free-energy minimisation the mark of the cognitive? *Biology & Philosophy* 36, 2 (2021), 25.

[17] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. 2005. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, Vol. 1. IEEE, 128–135.

[18] 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).

[19] Sam Kriegman, Amir Mohammadi Nasab, Dylan Shah, Hannah Steele, Gabrielle Branin, Michael Levin, Josh Bongard, and Rebecca Kramer-Bottiglio. 2020. Scalable sim-to-real transfer of soft robot designs. *2020 3rd IEEE International Conference on Soft Robotics, RoboSoft 2020* (may 2020), 359–366. https://doi.org/10.1109/ROBOSOFT48309.2020.9116004 arXiv:1911.10290

[20] Jakub Limanowski and Felix Blankenburg. 2013. Minimal self-models and the free energy principle. *Frontiers in human neuroscience* 7 (2013), 547.

[21] Sida Liu, David Matthews, Sam Kriegman, and Josh Bongard. 2020. Voxcraft-sim, a GPU-accelerated voxel-based physics engine. https://github.com/voxcraft/voxcraft-sim. https://doi.org/10.5281/zenodo.3835152

[22] Shanliangzi Liu, Dylan S. Shah, and Rebecca Kramer-Bottiglio. 2021. Highly stretchable multilayer electronic circuits using biphasic gallium-indium. *Nature Materials* (feb 2021), 1–8. https://doi.org/10.1038/s41563-021-00921-8

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

[24] 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.

[25] 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.

[26] Geoff Nitschke and David Howard. 2021. Autofac: The perpetual robot machine. *IEEE Transactions on Artificial Intelligence* 3, 1 (2021), 2–10.

[27] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation* 11, 2 (2007), 265–286.

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

[29] Federico Pigozzi, Yujin Tang, Eric Medvet, and David Ha. 2022. Evolving Modular Soft Robots without Explicit Inter-Module Communication Using Local Self-Attention. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 148–157.

[30] 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.

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

[32] Dylan S Shah, Joshua P Powers, Liana G Tilton, Sam Kriegman, Josh Bongard, and Rebecca Kramer-Bottiglio. 2021. A soft robot that adapts to environments through shape change. *Nature Machine Intelligence* 3, 1 (2021), 51–59.

[33] Andrew C Slocum, Douglas C Downey, and Randall D Beer. 2000. Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In *From animals to animats 6: Proceedings of the sixth international conference on simulation of adaptive behavior*. Citeseer, 430–439.

[34] Thomas A Stoffregen. 2018. Affordances as properties of the animal-environment system. In *How shall affordances be refined? Four perspectives*. Routledge, 115–134.

[35] 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).

[36] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and evolutionary computation* 1, 1 (2011), 32–49.