ADVANCED FUNCTIONAL MATERIALS
www.afm-journal.de

Robust Phototaxis by Harnessing Implicit Communication in Modular Soft Robotic Systems

*H. A. H. Schomaker, S. Picella, A. Küng Garcia, L. C. van Laake, and J. T. B. Overvelde**

In robotics, achieving adaptivity in complex environments is challenging. Traditional robotic systems use stiff materials and computationally expensive centralized controllers, while nature often favors soft materials and embodied intelligence. Inspired by nature's distributed intelligence, this study explores a decentralized approach for robust behavior in soft robotic systems without knowledge of their shape or environment. It is demonstrated that only a few basic rules implemented in identical modules that shape the soft robotic system can enable whole-body phototaxis, navigating on a surface toward a light source, without explicit communication between modules or prior system knowledge. The results reveal the method's effectiveness in generating robust and adaptive behavior in dynamic and challenging environments. Moreover, the approach's simplicity makes it possible to illustrate and understand the underlying mechanism of the observed behavior, paying particular attention to the geometry of the assembled system and the effect of learning parameters. Consequently, the findings offer insights into the development of adaptive, autonomous robotic systems with minimal computational power, paving the way for robust and useful behavior in soft and microscale robots, as well as robotic matter, that operate in real-world environments.

1. Introduction

Unlike living systems, which are comprised primarily of soft tissues that adjust their shape during interactions with the environment, robotic systems generally use stiff materials.[1] As a result, these conventional robotic systems exhibit well-defined motion and excel in precision and speed within controlled environments. In contrast, natural systems have evolved a focus toward robustness to survive in our complex world. These natural systems have inspired researchers to create soft robots that deform when interacting with the environment, increasing the potential for improved performance in more complex and unstructured real-world environments.[2]

Inspired by the many invertebrates that exist in nature (e.g., sea stars, octopuses, and sea urchins), we aim to explore how distributing the robot's "brain" over the body could result in more robust behavior.[3–5] For example, sea urchins have hundreds of tube feet, spines, and pedicellaria with locally integrated perception and motor control. They can sense and process information independently while still leading to whole-body locomotion, and robust behavior emerges without a central brain and only a limited central nervous system.[6] There has been a growing effort to understand such distributed systems in biology, in part due to their potential applications in autonomous robotic systems.[3,4,7] In the field of robotics, distributed control is mostly studied in modular and swarm robotics, which utilizes synergy and redundancy to improve the system's adaptability, functionality, reliability, and robustness.[8–10] Importantly, these systems typically have reduced complexity of their controllers since behavior emerges from interactions between agents.[11,12] These systems show that similar to out-of-equilibrium systems in nature, global behavior can emerge from nonreciprocal interactions.[13,14] More application-driven robotics examples include adaptation to mechanical stimuli,[15,16] construction,[17] and locomotion.[18–21]

Previous research has found inspiration in emergent natural systems like amoeboids[19] and collective migration phenomena in cell biology,[18] to create more resource-efficient and adaptive systems (able to achieve complex tasks with minimal computational resources).[22] These systems often rely on coupled oscillators to adapt their behavior based on external stimuli.[23] While this continuing line of research shows that autonomy and adaptivity can be achieved with minimal computational requirements, it remains intrinsically dependent on sensitive parameters that have to be a priori optimized (e.g., through evolutionary algorithms) to produce their desired emergent properties. Instead, we focus on embodying a basic and stochastic learning algorithm throughout the robot that allows it to adjust to its environment and situation within limits specified by the robot's design

H. A. H. Schomaker, S. Picella, A. Küng Garcia, L. C. van Laake, J. T. B. Overvelde
Autonomous Matter Department
AMOLF
Science Park 104, Amsterdam 1098 XG, the Netherlands
E-mail: overvelde@amolf.nl
S. Picella, J. T. B. Overvelde
Institute for Complex Molecular Systems and Department of Mechanical Engineering
Eindhoven University of Technology
P.O. Box 513, Eindhoven 5600 MB, the Netherlands

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/adfm.202310932

DOI: 10.1002/adfm.202310932

and functional capabilities. This embodied computation is realized with active and identical modules constituting a single robot body.

Importantly, rather than optimizing the physical system for the specific task, we focus on a physical platform with minimal functionalities in order to understand to global behavior that arises from the local interaction. Therefore, we focus on the general principles involved in decentralized whole-body phototaxis in the hope of making the outcomes more generally applicable to other physical platforms. To constrain the information in the system to be strictly local, we exploit a form of implicit communication facilitated by sensing changes to a shared environment, similar to the implicit communication of stigmergy in nature.[24–26] Our approach demonstrates that a decentralized system can optimize its behavior without explicit communication between its modules in various situations and environments.

Our previous research on a distributed stochastic learning algorithm showed great promise for emergent decentralized locomotion control in a one-dimensional framework[5] (the robot was constrained to a circular track), illustrating the value of a short-term memory algorithm in changing environments, as well as demonstrating the system's ability to resist damage.[5] Here, we move from a one-dimensional to a two-dimensional domain (the robot can freely move on a surface), which considerably increases the system and task complexity, e.g., due to competition. Importantly, even though we use exactly the same algorithm as in previous work, the fact that rotation is not directly controlled (only indirectly through competition) results in a dynamic learning process that is more complicated than in one dimension. From a physics perspective, transitioning from a one-dimensional to a two-dimensional transforms the state space from fixed to dynamic. In a one-dimensional setting, the phase space is static, and the optimization problem is relatively straightforward because the system's orientation does not change. Conversely, in a two-dimensional environment, the optimization problem becomes dynamically evolving. The system's orientation, and consequently its interaction with the environment, changes over time. This introduction of rotational dynamics that individual units cannot directly control enables a much richer set of behaviors and emergent phenomena that were not possible in the one-dimensional case. In this novel setting with increased complexity, we focus on three fundamental questions: i) Does the modular robot exhibit robust learning behavior? ii) How general is the implemented distributed stochastic approach, and how does geometry or configuration impact the system's ability to perform a task? iii) How adaptable is the system? Can it robustly adapt to the environment and damage?

2. Experimental Setup

We start by introducing the modular robotic platform that we developed (**Figure** 1a,b) to try to answer these questions. Our robotic platform consists of physically connected modules that can move on a plane and can be assembled in square lattice configurations. Each robotic module in the lattice is identical and has its own microprocessor to control a stepper motor, measure the light intensity, and run a stochastic algorithm to change its behavior. Importantly, we aim to employ a stochastic algorithm in each unit such that the system as a whole performs phototaxis (i.e., moves to a light source). An individual unit is programmed to periodically expand and contract its four connection points every 2 s (Figure 1a), where the only parameter that each module can tune is the phase of actuation ϕ_i, corresponding to the phase of unit i. Note that an individual unit cannot displace itself if it is not connected to other units. By physically connecting multiple units (Figure 1c) by their soft elastomeric arms, the system as a whole starts to move and rotate depending on the phase differences between the units (Movie 1). As each unit can only sense the light intensity from its own sensor, it is unaware of what the other connected units are doing. Each unit can only change ϕ_i and operates on this one-dimensional search space. However, the corresponding collective behavior changes when the connected units change their phase or when changes occur in the environment. Due to the change in collective behavior, each unit must continuously re-evaluate its performance in its one-dimensional search space. In a way, these units, therefore, can also change the environment for the other units in the assembly.

We implement the same stochastic learning algorithm that we previously studied in one-dimensional experiments, where the robotic system was constrained to a circular track to limit its movement to a single dimension.[5] In this learning algorithm, each unit performs random experiments by perturbing its phase and only conditionally accepts these phases by comparing the difference in light intensity with its previous step. Specifically, units individually undergo learning cycles (n_{LC}) as shown in Figure 1d, each consisting of a total of N_{AC} regular actuation cycles (AC) with a constant phase. Each actuation cycle has a duration of $t_{\mathrm{AC}} = 2$ s and consists of an extension of the soft connector arms with a duration of 0.6 s, followed by contraction of the arms with the same duration, where it remains in its contracted state for the remainder of the actuation cycle. At the end of every learning cycle, the unit perturbs its phase ϕ_i to explore a new phase ϕ_i'. The perturbation is performed according to $\phi_i' = \phi_i^m + \epsilon\Delta\mathrm{S}$. Here, ϕ_i' is the phase that is evaluated the next n_{AC}, ϕ_i^m is the previous phase kept in memory, ϵ represents a uniformly distributed random variable between the interval $[-1, 1]$, and the parameter $\Delta\mathrm{S} = 0.1$ describes the maximum difference between two phases of consecutive learning cycles.

To implement the phase change, each learning cycle is followed by a phase adjustment actuation cycle (AC_a). Only the last part of the cycle is altered during the phase adjustment cycle as the unit implements a new phase for the next step. The amount of adjustment conditionally depends on the acceptance or rejection of the current phase. If the current phase is accepted, phase difference $\Delta\phi_i$ between the current and the next learning cycle is given by $\Delta\mathrm{S}$. In the case of rejection, we perturbed the phase in memory to find a new phase for the next learning cycle, resulting in a $\Delta\phi_i$ that is a combination of the difference between these two phases and new perturbation $\Delta\mathrm{S}$ for the next learning cycle. This results in a new ϕ_i' by extending or shortening t_{cycle} according to:

$$\Delta\phi_i = \begin{cases} \phi_i^m - \phi_i' + \epsilon\Delta\mathrm{S}, & \text{for } \Delta\mathrm{I}_i' < \Delta\mathrm{I}_i \\ \epsilon\Delta\mathrm{S}, \phi_i = \phi_i' & \text{for } \Delta\mathrm{I}_i' \geq \Delta\mathrm{I}_i \end{cases} \quad (1)$$

Here, each unit approximates the quality of its current ϕ_i' by evaluating the difference $\Delta\mathrm{I}'$ between two independent light measurements I_1 and I_2 measured during AC_1 and AC_4,

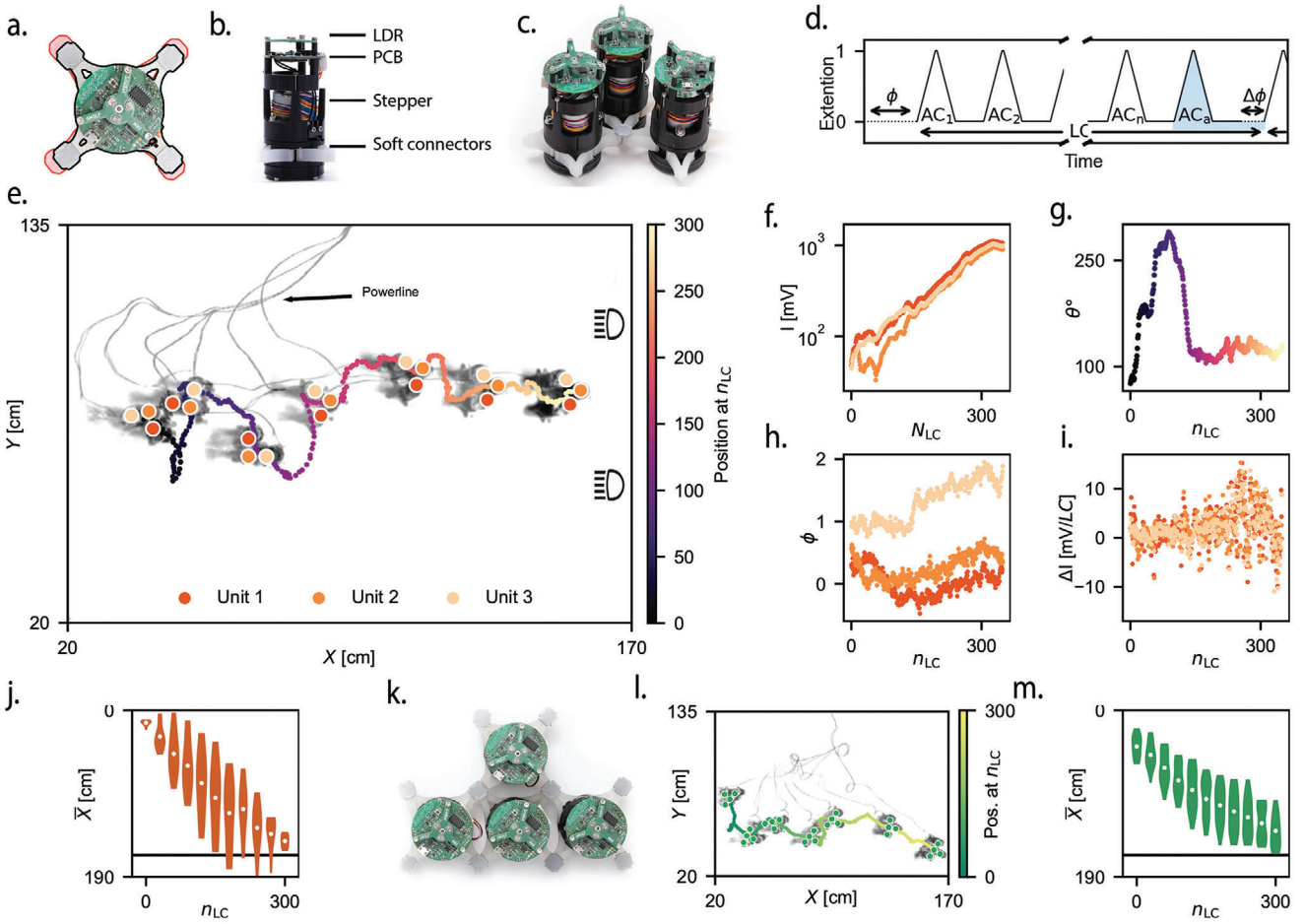

Figure 1. Robotic unit design and learning experiments for an assembled system in two different configurations. a) Realization of a unit and its corresponding extended (red outline) and contracted state (black outline). b) Side view of a robotic unit. c) Robot assembled from three units. d) Schematic representation of a learning cycle LC that occurs in each unit. e) Evolution of a single learning experiment with a three-unit system (d) during N_{LC} = 300, given by overlapping images of the experiments at every 50 n_{LC}. The colored line indicates the position of the center of mass at every learning step n_{LC}. For this experiment, the robot starts on the left, and a planar light source is placed on the right. f) The measured light intensity I in each unit. g) Orientation in degrees ($\theta°$) of the three-unit system during the single experiment. h) Absolute phase ϕ_{ii} for every learning cycle. i) The measured difference in light intensity ΔI. j) Distribution of the average X positions of the assembled robot for ten experiments with the three-unit configuration. k) Robot assembled in a configuration with four units. l) Evolution of a single learning experiment with the four-unit system during 300 learning cycles, under similar experimental conditions as in the three-unit experiment (e). m) Distribution of the X positions for ten experiments with the four-unit configuration.

respectively. Therefore, the units evaluate the rate of change in light intensity rather than only looking at the absolute light intensity. The units accept the current phase ϕ_i' if the difference in light intensity $\Delta I_i'$ is larger than ΔI_i stored in memory. Finally, in previous work, we established that to achieve adaptive behavior ΔI_i should change to $\Delta I_i'$ after every learning cycle.[5] Note that every unit independently changes its ϕ_i. Therefore, the units are not synchronized, resulting in asynchronous timing between the units for measurements and phase adjustment. More details on the learning are provided in Section 9 (Figure S1, Supporting Information).

3. Phototaxis in Experiments

To determine if a system of coupled units can perform phototaxis while moving on a surface, we first conduct experiments with the smallest two-dimensional configuration (i.e., with units that do not only lie on a line) as shown in Figure 1c. We place the assembled system on the left side of a flat rectangular surface with two LED panels on the right side (Figure 1e) and observe the system's behavior during N_{LC} = 300 learning cycles (Movie S2, Supporting Information). Even though the units operate entirely independently and asynchronously (the only electronic connection is the power supply), Figure 1e shows how the assembled units rotate and move while getting closer to the light source on the right. This is clearly demonstrated in Figure 1f, where we show that the average light intensity I increases for all units during the experiment.

Looking closer at the behavior of the assembly in Figure 1e, we can roughly distinguish four regimes. i) During approximately the first n_{LC} = 50 learning cycles, the measured ΔI does not seem to increase for all units (Figure 1f). Yet, we do observe a change of

orientation Θ° of the assembled system. This orientation is represented in degrees and measured with respect to the experimental canvas (Figure 1g). ii) Between approximately $50 \leq n_{LC} \leq 150$ learning cycles, the units start to move toward the light source while also undergoing significant rotation. iii) We find a straight trajectory for the last $n_{LC} = 50$ learning cycles, along with a stabilization of the orientation of the system (Figure 1g). While the phases of the individual units still vary over time, the phase differences between the units stabilize (Figure 1h). It appears that the assembly has reached stable behavior. iv) Around $n_{LC} = 300$, the units reach a peak of their change in light intensity ΔI as presented in (Figure 1i). This indicates that they arrived at the position of highest light intensity as also revealed by the drop in ΔI shown in Figure 1f. In general, Figure 1f–i demonstrates the emergence of robust global phototaxis of the system.

To investigate the robustness of the observed phototaxic behavior, we repeat the experiment of Figure 1f–i ten times, where we start from initially random phases and random orientations. Figure 1j shows the position distribution for all experiments for progressing learning cycles. Note that in this figure, the position of the light source is at $x = 165$, and a black line indicates the place with the highest light intensity I as measured in separate experiments (Figure S2c). Figure 1j demonstrates that the distributions converge to the location with the highest light intensity, thereby indicating that all assemblies move toward the point of highest light intensity and, thus, achieve robust phototaxis.

Next, to explore if phototaxis is also achievable for a robot with a different geometry, we assemble a robot by adding an additional unit (Figure 1k). We place the assembly on the left side of the test setup to repeat the same experiment as before and observe the first $N_{LC} = 300$ learning cycles in Figure 1l. Similar to the three-unit system, we find a robust directional motion toward the light source, with considerable rotation during the first part of the experiment and convergence to stable behavior for the second part.

Subsequently, we repeat the experiment ten times (Figure 1m) with random initial phases. Similarly, we find a displacement toward the light source for all ten experiments. Interestingly, we find slower average velocities for the four-unit system compared to the three-unit system and a smaller variance between experiments. These results emphasize the importance of previously raised questions on how general the implemented distributed stochastic approach is and what the impact is of the geometry or geometry on the system's ability to perform a task.

4. Understanding the Observed Behavior

Before answering how geometry affects the ability to perform phototaxis, we first return to the first question raised in the introduction and try to better understand how the system can achieve its behavior without explicit communication between modules. To get a better visualization and understanding of the underlying dynamic behavior from both the perspective of the individual units and the assembled system, we implement a coarse-grained mass–spring model that qualitatively captures the system's behavior.

To reduce the computational requirement for the model, we build single units using four masses and four springs, as shown in **Figure** 2a. Two additional diagonal springs actively drive the extension and contraction of the unit by varying their rest length periodically in time. The behavior over time is found by numerical integration of the equations of motion. Within the numerical integration, we perform a discrete event model to capture the decentralized nature of the learning behavior in experiments. A block diagram of the event model can be found in Figure S1. Further information on the model can be found in Section 9. Figure 2a presents the model representation of the three-unit configuration as depicted in Figure 1c. Similar to the experiments performed in Figure 1, we place a planar light source on the right side of the two-dimensional plane and model the light source as a two-dimensional scalar light intensity field, where the intensity has an inverse square relation with the distance to the light source $I \propto \frac{1}{d^2}$. Simulating the three-unit assembly, we find that the system starts to move in the direction of the light within $n_{LC} \approx 10$ learning cycles (Figure 2b and c). In agreement with the experiments, the assembled robot initially behaves in a regime dominated by large rotations that occur while learning to move toward the light, followed by a more consistent orientation and movement.

4.1. System Perspective

To find an accurate representation of how the system with the learning algorithm performs phototaxis, we evaluate the system displacement toward the light for all possible phase combinations as presented in Figure S3a. Now, a change of reference frame is applied to obtain the search space for a specific system orientation during the simulation. Figure 2d presents this rotated search space for three different snapshots of the simulation presented in Figure 2b and c, at different learning cycles. Furthermore, the respective rotations of these three phase combinations are represented in Figure 2e. We can make four main observations from the perspective of this system. First, the search space drastically changes due to the system's rotation, meaning it must continuously reevaluate its behavior. Second, we see that the maximal achievable displacement toward the light changes over the time due to the rotation of the system, from $\overline{V} \approx 1.0$ [mm per cycle] at 450 n_{LC} to $\overline{V} \approx 0.6$ [mm per cycle] at 950 n_{LC}. Third, the assembly finds the area close to optimal phototaxis in the phase space, demonstrated by these three orientations. Note that due to the stochastic nature of the optimization, the behavior inherently fluctuates rather than fixates on the optimal position in the phase space. Fourth, depending on the phases, the assembly unintentionally (without being observed by the light intensity sensor) changes its orientation. These changes in the search space indicate that the assembly cannot sustain a specific, and maybe faster, behavior over extended periods due to the inevitable rotation that coincides with the displacement toward the light source. These results raise the question of whether the system can stabilize its behavior over a longer time scale and what the effect of the stabilization is on the phototaxic behavior.

4.2. Orientation Perspective

Of course, as each unit can only sense a single-light intensity and there is no interunit communication, neither the system nor the units have any notion of their orientation. However, Figure 2d reveals that the maximum speed heavily depends on the system's

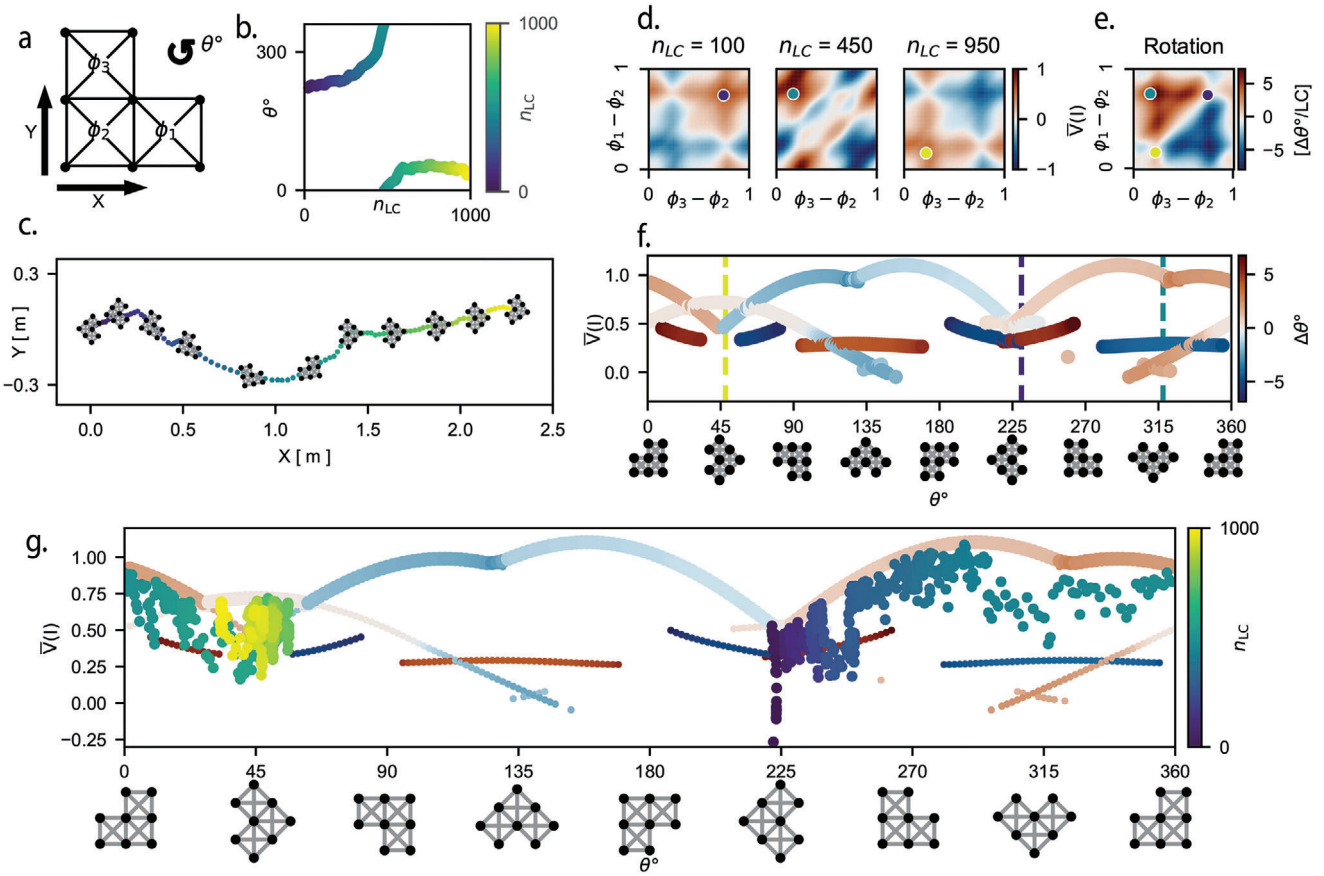

Figure 2. Course-grained mass–spring model to numerically study the qualitative behavior of the system. a) Mass–spring representation of a three-unit system as depicted in Figure 1c, with the active springs in the diagonal of the squares and passive springs around the edges. a) Orientation over n_{LC} for a single simulation of the three-unit system for a 1000 learning cycles. c) Trajectory of the same simulation, with a planar light source placed on the right side of the two-dimensional plane. The system's state is depicted every 100 n_{LC}. d) Evolution of the potential velocity mapped to the phase space from a system perspective measured in $\overline{V}$(I) [mm LC^{-1}] at three-time instances during the simulation of (c). The color map indicates the velocity in the direction of the light for all possible phase combinations of the system. The colored dots indicate the phase combination during the simulation of (c) colored by n_{LC}. e) The rotation ($\Delta\theta$) of the system for all phase combinations. The dots indicate the phases at the n_{LC} snapshots of (d). f) a projection of the local optima of the $\overline{V}$(I) [mm LC^{-1}] in the phase space for all system orientations. Obtained by rotating the reference frame of Figure S3a and numerically extracting the local optima for each orientation. The dashed vertical lines indicate the slices as represented in (d). The color of the optima indicates the change in angle $\Delta\theta$ for the phases that correspond to the local optima as can be found in (e). g) The same projection as represented in (f) with the optimal $\overline{V}$(I) for each orientation represented in an increased size compared to the local optima. The trajectory of the single simulation (c) is mapped onto this projection to visualize the long-term phototaxic behavior.

orientation. In fact, the system stabilizes at an orientation for which the maximum velocity in the direction of the light is lower than what this specific assembly can potentially achieve at different orientations (Figure 2d). This difference in maximum directional velocity for different orientations can be more clearly seen by looking at the fastest velocities for any given orientation, as shown in Figure 2f. This figure depicts the amplitude of phototaxis (displacement toward the light) for all the local optima in each orientation, obtained by rotating the global reference frame of Figure S3a and numerically finding the local optima for each orientation. The color indicates how much the system rotates during a learning cycle. A more detailed view of how this projection is created can be found in Movie 3. Note that Figure 2f solely resembles the potential (locally) optimal displacement of the assembly without considering any behavior related to the learning, but it helps us to visualize the potential optimal behavior the system could exhibit. From this perspective, we find a reflective symmetry between 45° and 225° and the existence of only two local optima for which the assembly does not rotate with peaks at 45° and 225°.

To better understand how the system moves through this three-dimensional search space (consisting of the two $\Delta\phi_i$ and the Θ°) and how close the behavior is to a (local) optimum, we project the displacement and orientation of the single simulation of the three-unit system into Figure 2g. We observe that the assembly seems to find the general areas of optimal phototaxis and tracks them over time as its orientation changes. Stability occurs when the units reach an orientation for which no rotation occurs, in this case, at ≈45°. Interestingly, the assembly spends most of its learning cycles on the two optima with minimal rotation at the respective 45° and 225° angles. Moreover, the assembly seems to move away from one to the other and stabilize its

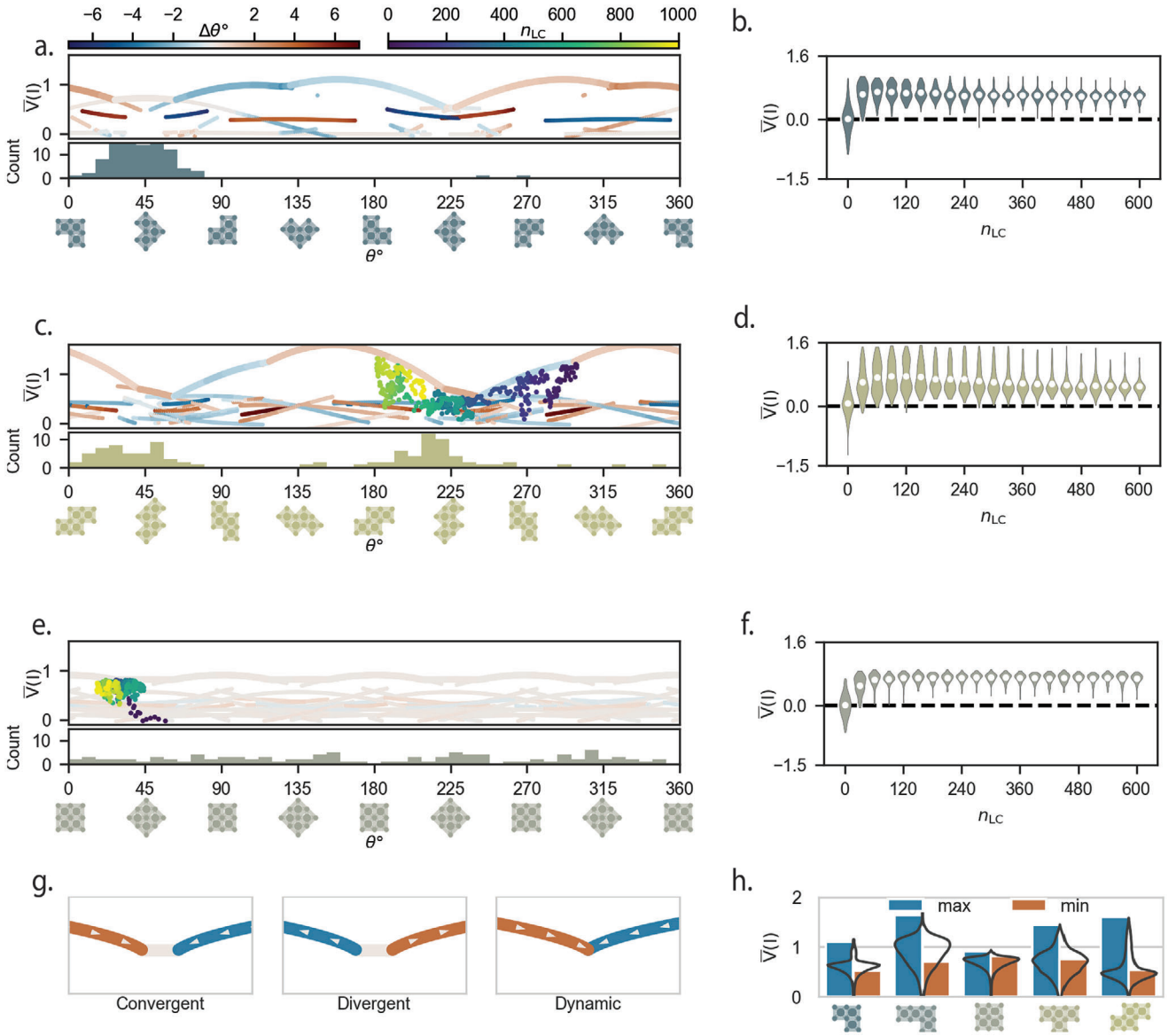

Figure 3. Effect of geometry on the long-term behavior of the system. Orientation perspective for a) a three-unit robot, and two four-unit robots with c) skew and e) square shape. The distribution underneath the orientation perspective indicates the final orientation after a 1000 N_{LC} for a 100 simulations. The corresponding evolution of the velocity $\overline{V}(I)$ [mm LC^{-1}] for the first 600 learning cycles are shown in (b), (d), and (f). g) Schematic representation of the different types of local optima in the orientation projection. h) Overview of the simulation results obtained for all robot shapes up to four units. The distribution (black line) of the average velocity ($\overline{V}(I)$ [mm LC^{-1}]) for 100 simulations at the 600th learning cycle, compared to the maximum and minimum velocity of the global optimal behavior.

orientation around 45°. These results could indicate stable orientations that emerge on longer timescales, dominated by the geometry of the assembly.

4.3. Long-Term Stability

With this improved understanding of the system's behavior from different perspectives, we explore the dynamic stability of the system over longer timescales. We repeat the simulation performed in Figure 2g 100 times for different initial conditions and starting orientations of the assembly. In **Figure 3**a, we present the distribution of the orientations of the assembly after at $n_{LC} = 1000$. In line with the results presented in Figure 2g, almost all simulations converge toward 45° (with only two outliers around 250°). Therefore, we conclude that the 45° angle serves as an "attractor" for the three-unit configurations. In contrast, the 225° angle is an "unstable fixed point" for long-term behavior. Although stable in the short term, the stochastic nature of the algorithm will eventually push the system away from the 225° angle.

To further investigate the influence of these long-term dynamics on phototaxic behavior, in Figure 3b, we show a distribution

of the average displacement of the system for all 100 simulations while moving toward the light source. Considering the analysis from Figure 2, we can identify the same three distinct regimes in the assembly's behavior as we have observed in experiments in Figure 1, but with additional insights. Firstly, for $\approx 0 \leq n_{LC} \leq 50$ learning cycles, the system quickly changes phases to improve its velocity in the direction of the light ($\overline{V}(I)$). Next, between $\approx 50 \leq n_{LC} \leq 250$, the system reaches a peak in $\overline{V}(I)$ due to changing orientation, where higher velocities are still possible as the system is still changing its orientation. Note that although these orientations allow for phase combinations with a high $\overline{V}(I)$, these phases also result in large rotations $\Delta\theta$ and are, therefore, unstable and cannot be maintained for longer time periods. Finally, for $n_{LC} > 250$, the variance of the distribution becomes smaller, and the $\overline{V}(I)$ converge across all simulations. This convergence occurs because more simulations reach the 45° angle, approaching the same stable orientation and phase combination.

4.4. Unit Perspective

It is important to consider that so far, we have visualized the potential behavior from the assembled system perspective. Yet, the individual units do not directly communicate. To understand the dynamics from the perspective of the individual units, we use the mapped search space as shown in Figure S3a of the three-unit system to visualize the one-dimensional search space of every unit over time in Figure S4. Each unit operates in a different one-dimensional search space, depending on its position in the assembly. Note that the units do not see the full search space of the whole system and can only probe the local one-dimensional search space by varying their phase. Importantly, the search space's mapping from phase to behavior fluctuates over time due to the rotation of the assembly and the continuously changing behavior of the surrounding units. Figure S4 demonstrates how all three units can adapt their phase ϕ_i to maintain the desired phototaxis behavior of the assembly. Surprisingly, the individual units stay close to the optimal behavior in their constantly changing search space.

5. Effect of Geometry on the Long-Term Behavior of the System

With this more explicit understanding of the long-term dynamics for a three-unit configuration, we now focus on getting a better idea of how the system's geometry influences long-term behavior. We, therefore, perform the same analysis for two distinct four-unit configurations. In Figure 3c–f, we show the learning behavior of the two four-unit configurations (skew shape and square shape) that, together with the three-unit system, can describe the main similarities and differences between geometries. First, Figure 3c describes the trajectory and global optima of the skew-shape configuration with one-fold rotational symmetry. Due to the one-fold rotational symmetry of the configuration, we find a repeating pattern in the global optima projection. If we follow the optima that cause counter-clockwise rotation (in blue) to the left and the optima that cause clockwise rotation (in red) to the right, we find two intersections at ≈50° and 230°. These intersections lead to the preferential orientation of the system in these two orientations, as seen from the distribution in Figure 3c. However, where the three-unit configuration has one stable optimum without rotation that all simulations converge toward, this four-unit geometry constantly transitions from rotation in one direction to the other, as there is no global optimum with zero rotation. This differs from the "convergent" behavior, as the phases must be "dynamically" adjusted to maintain phototaxis. This could also lead to a wider distribution of velocities, as observed when running multiple simulations (Figure 3d).

In contrast to the skew shape that does not have stable convergent orientations, the optimal behavior of the square shape is characterized by relatively little rotation (Figure 3e). The combination of this observation and the four-fold symmetry of the unit leads to a nearly flat projection of the local optima. Apparently, for any given orientation of the square geometry, an optimal phase combination exists that leads to a relatively stable and fast phototaxis. As a result, the square shape also does not exhibit the initial overshoot in velocity that is the result of stabilization of orientation (Figure 3f) that is observed for both the three-unit (Figure 3b) and the skew shape (Figure 3d).

In summary, from the results in Figure 3, we find three characteristic transitions between global optima that govern the long-term behavior of the system. We define the global optimum from the perspective of the system as the set of phases that result in the highest $\overline{V}(I)$ at a fixed orientation (as the system has no notion of orientation) (Figure 3g). First, the convergent behavior, where the geometry drives the system toward a global optimum without rotation, thereby stabilizing the behavior (i.e., no phase changes are required to maintain phototaxis). Second, a divergent transition is presented in Figure 3a. Although the system can maintain this orientation for extended periods, the stochastic nature of the control will eventually drive the system out of this orientation. Third, a dynamic behavior, as presented in Figure 3c, leads to persistent orientation fluctuations (i.e., constant phase changes are required to maintain phototaxis). Different geometries can yield combinations of the optima transitions mentioned above.

Figure 3h provides an overview of the difference between the maximum $\overline{V}(I)$ in the fastest and slowest orientation for all three- and four-unit configurations, along with the $\overline{V}(I)$ distribution at $n_{LC} = 1000$. From this overview, we can conclude that the phototaxic potential of the system heavily depends on its geometry. This difference can also be observed in their trajectories presented in Figure S5. Furthermore, we find that the system's $\overline{V}(I)$ distributions, in general, do not stabilize around their maximum $\overline{V}(I)$ because these orientations are accompanied with rotations of the system.

6. Effect of Learning Parameters on the Long-Term Dynamics of the System

Previous simulations on multiple shapes teach us that shape has an important role in the behavior of the assemblies and that the shape directly influences the short- and long-term behavior that the system can exhibit. However, the learning parameters can also affect the stability of the system's long-term behavior. In this section, we will further explore the effect that the learning step ΔS has on the potential to perform phototaxis.

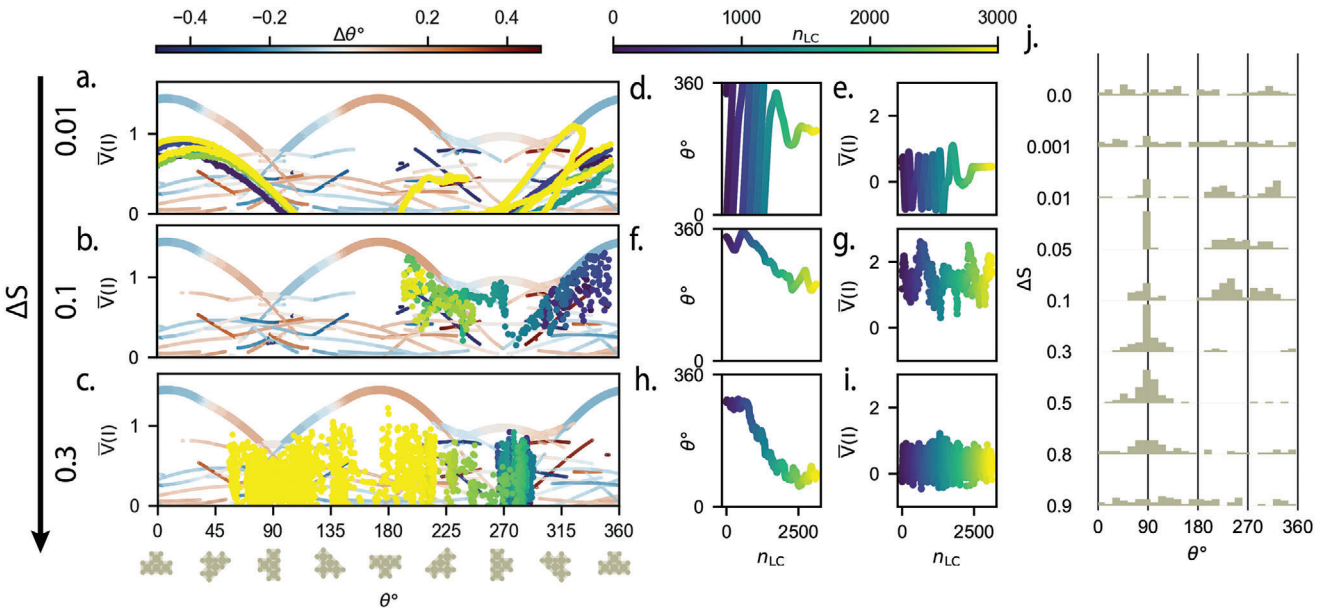

Figure 4. Effect of the learning step size on the behavior of the system. Evolution of the velocity $\overline{V}$(I) [mm LC^{-1}] of a T-shape robot in the orientation perspective, for a) $\Delta S = 0.01$, b) $\Delta S = 0.1$, and c) $\Delta S = 0.3$. The evolution of the robot's angle for each simulation is given in (d), (f), and (h). The evolution of the robot's average velocity $\overline{V}$(I) [mm LC^{-1}] for each simulation is given in (e), (g), and (i). j) Distribution of the angle at the 1000th learning cycle, for 100 simulations and for a range of learning step sizes ΔS.

The current minimal approach to learning is designed with generality in mind and is dominated by only two parameters, ΔS and the number of actuation cycles (N_{AC}). Both relate to the rate at which the system can adapt its behavior to the changing environment. On one hand, the number of actuation cycles dominates the signal-to-noise ratio of the light measurement by controlling the number of actuation cycles between the first and second light measurements. However, it also determines the amount of rotation between consecutive learning cycles, as the rotation scales linearly with the number of actuation cycles. On the other hand, the amplitude of the phase change between consecutive learning cycles, ΔS, directly controls the algorithm's adaptability (i.e., stochasticity). For this parameter, we predict that there is a direct trade-off between exploration and exploitation, given by high and low ΔS, respectively.

To demonstrate the influence that ΔS has on the system's behavior, **Figure** 4a–i shows three simulations of a T-shape configuration with $\Delta S = 0.01, 0.1$, and 0.3. Having mapped the optima in Figure 4a–c, we find one convergent stable orientation at 90° and two dynamically stable orientations at 220° and 320°.

First, for the relatively small learning step $\Delta S = 0.01$ (Figure 4a), we find that the system remains in the local optima and does not jump between local optima as the pay-off declines due to rotation. Figure 4d shows that the assembly keeps rotating until settling in a convergent local optimum around 225°. Apparently, the small values for the learning step suppress the stochastic exploratory nature of the algorithm. Second, in Figure 4b, we find the $\overline{V}$ for $\Delta S = 0.1$ to be more scattered, resulting in global behavior that jumps over the local optima to reach the global optima. As a result, we find less change in the orientation (Figure 4f) and a global behavior that settles around one of the dynamic global optima with a similar orientation as Figure 4a, but considerably higher $\overline{V}$ (Figure 4g). Third, for the simulation with $\Delta S = 0.3$ in Figure 4c, we observe even more scattered behavior. With this value of ΔS, the system seems unable to follow any specific optima and rather moves around the entire phase space. This results in an inability to track the dynamic global optima. Yet, the systems still slowly drift toward the convergent behavior at 90° (Figure 4h and i).

Looking at the long-term behavior of the T-shape configuration for different values of ΔS (Figure 4j), we find that, for $\Delta S = 0$, the end distributions are randomly spread out over all possible orientations. From $\Delta S > 0.01$, we start to see clear peaks in the end distribution, with centers around the convergent point at 90° and the two dynamic points around 220° and 320°. Between $0.1 \leq \Delta S \leq 0.3$, we find the distribution peaks of the dynamic points disappear, resulting in only a single-stable orientation at 90°. Last, starting from a $\Delta S > 0.8$, we find the onset of the disappearance of any stable orientations as the system starts to approach a state where it always chooses new random phases.

Although we observe phototaxis for all three ΔS, the rate at which phototaxis is achieved and the $\overline{V}$(I) after the behavior converges are considerably different. Furthermore, we investigate the scalability of the system in lattice configurations with increasing size (Figure S6a). Figure S6b demonstrates that the ΔS significantly impacts the system effectiveness of the phototaxis for larger systems. We see that, for both $\Delta S = 0.1$ and $\Delta S = 0.05$, the lattice configurations perform phototaxis in a system with up to at least 49 units. However, we do find that for the $\Delta S = 0.1$, the $\overline{V}$(I) declines more quickly with increasing system sizes compared to the $\Delta S = 0.05$. When evaluating the evolution of the $\overline{V}$(I) distributions for a lattice of 49 units (Figure S6c), we find that the

$\Delta S = 0.1$ learns more quickly. However, Figure S6c also shows that around $n_{LC} = 80$ the smaller $\Delta S = 0.05$ starts to outperform $\Delta S = 0.1$. From $n_{LC} = 100$ onward $\Delta S = 0.1$ simulations stop improving while the smaller ΔS simulations keep on improving their $\overline{V}(I)$.

These results indicate that the phase space of the physical system becomes more complex as the number of units increases. Therefore, the larger ΔS simulations are less effective in exploiting the local optima than the smaller ΔS simulations. While we did not test configurations larger than 49 units due to computational and practical constraints, we have no evidence to suggest that phototaxis would not occur in larger systems, albeit becoming less effective. However, we acknowledge that the efficiency, as indicated by the speed of movement toward the light, may vary with the system size and the learning ΔS parameter, which is an important consideration for scalability. Further research is needed to test how the current algorithm would work with an even larger system.

Thus, we find the long-term behavior of our system not only emerges from its geometry but is also heavily dependent on the chosen learning parameter. Furthermore, we find that phototaxis remains feasible for a wide range of the ΔS parameter (Figure S7), but that its value does influence the equilibrium behavior. These results further strengthen the claim of robustness in the current learning approach.

7. Experimental Validation

Having gained insights into the dynamic learning behavior of our soft modular system using simulations, we next turn to multiple experiments backed up by simulations that demonstrate the main characteristics and, specifically, the robustness of our approach that we also observed in simulations. We do this by i) changing the direction of light during an experiment, ii) increasing the complexity of the environment by adding obstructions, and iii) changing the geometry during the experiment by cutting the assembled robot in two.

7.1. Suddenly Changing the Direction of Light

As in the first experimental demonstration, we change the light direction during an experiment, which will require each unit to update its phases. The change in light direction is accomplished by mounting LED panels on both sides of the experimental setup and turning them ON and OFF during the experiment. In **Figure 5a**, we show a single experiment of the square configuration for $N_{LC} = 400$ learning cycles, where we manually change the direction of the light from left to right approximately halfway through the experiment at $n_{LC} = 23$. We can clearly observe how the robot first moves to the left, after which the direction changes to accommodate displacement in the opposite direction (Movie 4). The sudden change can be clearly seen in the discrete jump in the measured light intensity of the four units, which reduces to a value close to zero as shown in Figure 5b. As expected, the system requires some learning steps to adjust its behavior and reach a steady motion in the direction of the light.

Similarly, for a three-unit geometry (Figure 5c and d), we also observe that the system adapts its motion to the change in light direction. It is interesting to note that in this specific case, at least one of the units measures no light intensity ($I = 0$) after the manual change in light direction (see insert in Figure 5d), which means the sensor of this unit is outside of the range of the light source. However, it appears that the other units can compensate for this unit and move the system back to a measurable distance of the light.

Furthermore, Figure 5c demonstrates how the system orients itself toward its preferred orientation (similar to Figures 2e and 3a) (with the middle unit pointing in the direction of the light) and turning 180° when the light is switched. For the square shape of Figure 5a, we do not observe a clear switch of orientation as the light is changed. Next, we turn to simulations to find a more quantitative evaluation of the reorientation due to the switching of the light direction. From Figure 5e, we see that, in general, the trajectories of the square shape seem to find a straight line back and forward, whereas the three-unit system (Figure 5f) makes wide loops while rotating. Figure 5g indicates that throughout the simulation, all orientations are used by the square shape in contrast to the thee-unit configuration where we find that the orientations converge to 45° on the first half of the simulation and to 125° of the second. Figure 5i provides additional insights into the lack of reorientation compared to the converging behavior found in Figure 5j. Hence, it demonstrates the influence of the more symmetric square-shape geometry as opposed to the three-unit geometry, as also demonstrated by the orientation perspective in Figure 3e

Because the current system only has short-term memory and the units have no notion of orientation, we find similar learning behavior between the start of the experiment and the behavior after the light has switched. Therefore, switching the light can be viewed as nothing more than kicking the dynamical system out of its equilibrium, after which we find robust recovery to phototaxic behavior.

7.2. Operating in a More Complex Environment

So far, we have demonstrated the system's ability to generate robust phototaxic behavior, irrespective of the system's configuration. However, we have not considered more complex environments that can obstruct the motion and rotation of the system. As the second experimental demonstration of our approach's robustness, we will explore the system's behavior in more challenging environments. We achieve such environments by fixing impassable circular perspex disks with a height of 1 cm and a diameter of 25 cm to the surface, as shown in **Figure 6a**. These barriers restrict the system's movement while leaving the light measurements undisturbed.

We next perform a learning experiment using a square assembly (Figure 6a and b). Even though the system is capable of moving past the obstacles, the observed behavior is considerably affected by the obstruction. The effect on the phototaxic behavior is clearly visible by considering the $\overline{V}(I)$ in Figure 6b, which decreases every time the system hits an obstacle. For example, we find five interactions with the barriers in this single experiment. As a result, we observe sharp changes in the trajectory of the configuration. Still, for this specific environment, the system is capable of adjusting the behavior after each interaction (Movie 5).

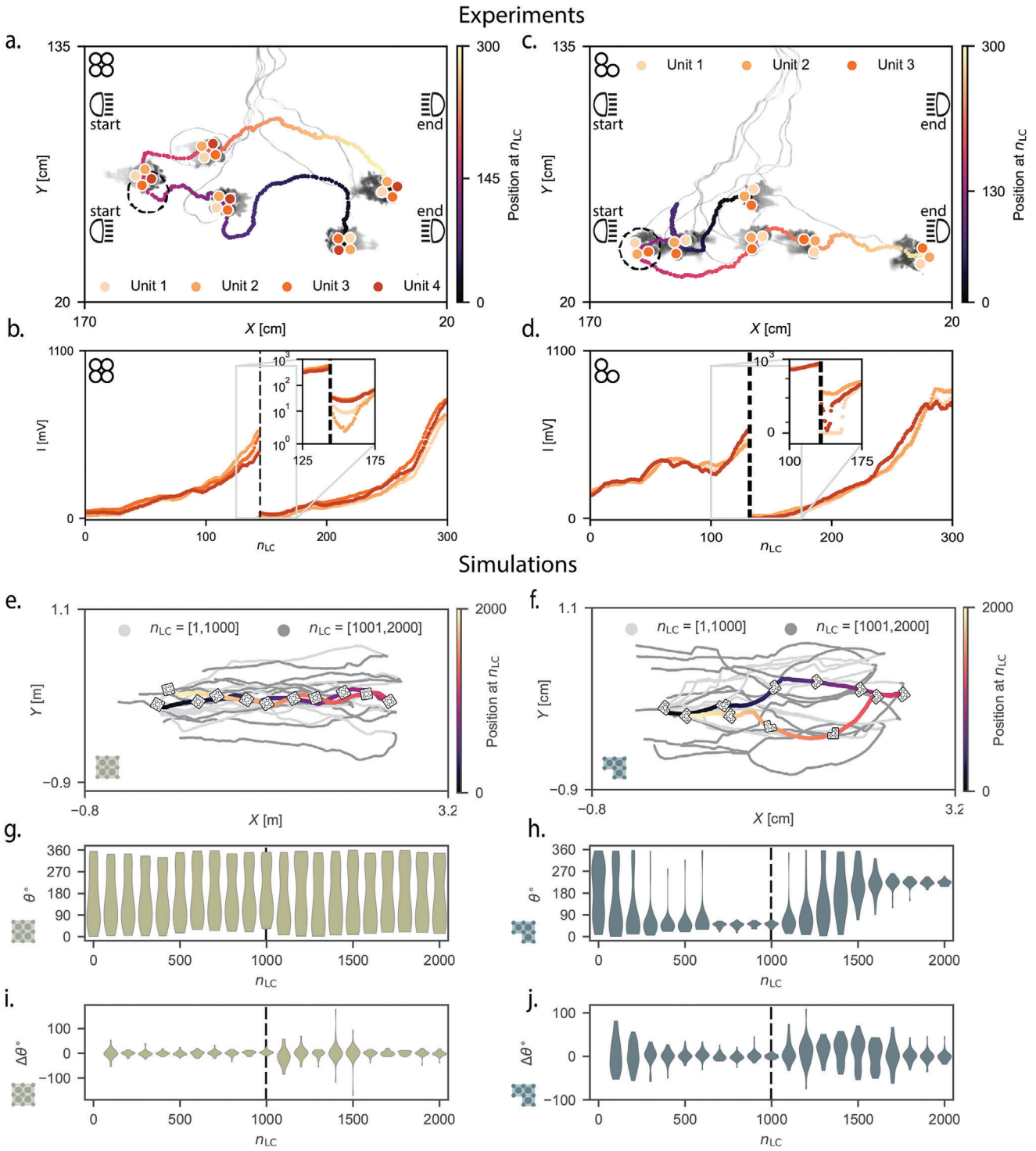

Figure 5. Response of an assembled robot to a sudden change in light direction. Experiment of a) a four-unit square robot with 350 N_{LC}. The light source is initially placed on the left side of the canvas and is switched to the right side of the canvas at the 145th learning cycle. The image consists of six overlapping photos of the experiment, each 50 N_{LC} apart. The dashed circle indicates the position of the robot when the light source is switched from left to right. The light intensity measured by the individual units is shown in (b). c) The same experiment as (a) for a three-unit assembly. The light source is switched at n_{LC} = 130, with the light measurements presented in (d). The vertical dashed line indicates the switching of the light source. Trajectory of simulations e) a four-unit square system and f) a three-unit system for 2000 N_{LC}. Out of the 30 simulations performed for each configuration, the trajectories of the first ten are represented in the figure. For each configuration, the first trajectory is highlighted along with snapshots of the system for every 200 n_{LC}. g and h) The distributions of the orientations Θ° for all 30 simulations of both configurations. Change of orientation ΔΘ° for all 30 simulations in (i) and (j), respectively.

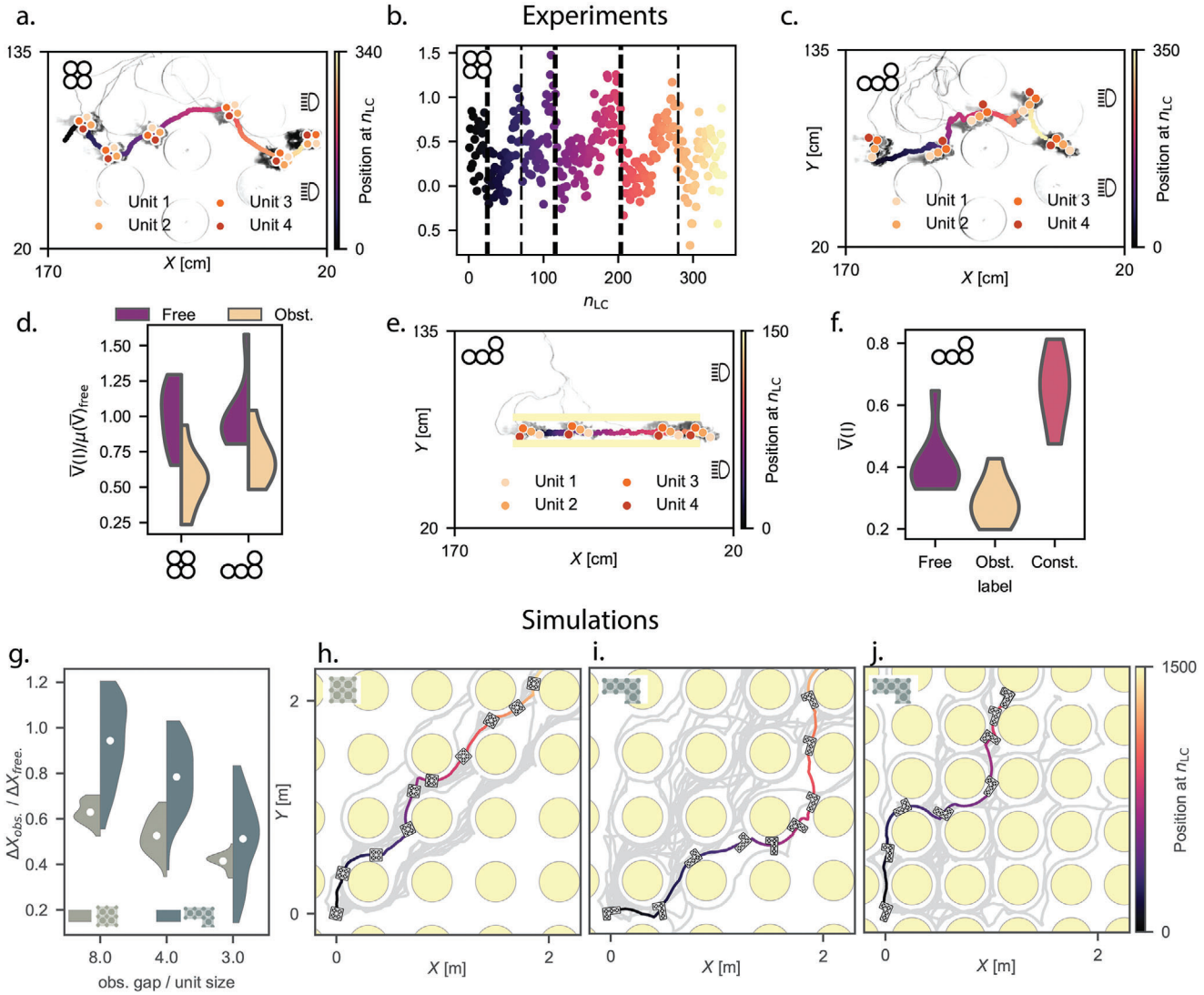

Figure 6. Adaptive behavior of assembled robots in a more complex environment. a) Single experiment of a square-shaped robot moving through circular obstacles. The circular obstacles restrict the motion of the robot, but do not block the light source that is placed on the right. The first five images indicate the moment of contact with an obstacle, and the last image represents the end of the experiment, as indicated by the dashed lines in **b**. b) The average velocity $\overline{V}(I)$ [cm LC^{-1}] of the system during the experiment performed in (a). c) an experiment with an L-shaped robot under the same conditions as in (a). d) Distribution of the $\overline{V}(I)$ [cm LC^{-1}] for both the square-shape and L-shape robots, with and without obstacles, and normalized by the average $\overline{V}(I)$ [cm LC^{-1}] without obstacles, each distribution contains eight experiments of 230 N_{LC} (at $n_{LC} = 230$ the first free experiments reaches the end of the canvas). e) A single experiment with the L-shape robot in a confined environment restricted by two bars. f) Average $\overline{V}(I)$ [cm LC^{-1}] for the L-shape robot in experiments without obstacles, with circular obstacles, and in the environment constrained by the two bars. Each distribution consists of 8 experiments with the same N_{LC} as (d). g) Distributions of the covered distance (in the direction of the light) [m] divided by the covered distance without obstacles for both the square-shaped and the L-shaped in simulations with obstacles (circles with a diameter of 20cm). The simulations last 1500 N_{LC}, and the *x*-axis represents the minimal distance between the outer edges of the obstacles in cm divided by the minimal length of a single unit (5 cm). h and k) Trajectories of the simulations of (g) for two different configurations and three different obstacle distances. The square shape with an obstacle gap distance of 20 cm is shown in (h) with the obstacles in yellow, the 30 trajectories in gray, and a single simulation highlighted with the color map for the n_{LC}. i and j) follow the same representation as (h) but for the L-shape with an obstacle gap distance of 20 and 15, respectively.

The ability of the system to adjust to more complex environments will likely depend on the geometry of the system. To test this, we also run experiments with an L-shape assembly operating in the same environment (Figure 6c). For the L-shape, we observe that the system requires more learning cycles N_{LC} to reach the position with the highest light intensity. To evaluate the influence of the obstacles on both the square and the L-shape geometries, in Figure 6d, we compare the distribution of the velocity over a learning cycle normalized by the average speed of experiments without obstacles for that specific shape. We find that this more complex environment influences the L-shape less. One explanation could be the difference in the symmetry between the two geometries, as we know from Figure 3e that the maximum $\overline{V}(I)$ of the square unit is less affected by rotation. The results

from Figure 6d could indicate that the obstacles constrain the system into orientations with higher maximum $\overline{V}(I)$, as Figure 3 already indicated that geometries often stabilize in orientations with sub-optimal $\overline{V}(I)$.

Although in the previous case, the environment has a negative effect on the average $\overline{V}(I)$ of both assemblies, to test if the L-shape assembly can also benefit from its surroundings, we next perform an experiment in an environment that contains two parallel placed bars (Figure 6e). In Figure 6f, we compare the average $\overline{V}(I)$[cm LC^{-1}] of the system in the direction of the light between no obstacles, circular obstacles, and the two bars, and interestingly observe that the velocity of the system is, on average higher when constrained by the two bars. These results show that, in some cases, the system could even utilize its environment to increase its displacement toward the light. This can be explained by the fact that the system's rotation is constrained, such that the modular robot can sustain otherwise unstable orientations with higher velocities. In addition, the friction from the bars could aid in an additional contact point.

To further extend our studies of the impact of geometry on the ability to maneuver the obstacles presented in the experiments above, we emulate the experiments with the implemented course-grained simulations. We specifically focus on how the distance between the obstacles influences the behavior. Figure 6g demonstrates how sparsely placed obstacles result in a better relative performance compared to no obstacles in the L-shape as opposed to the square shape, similar to the results of the experiments in Figure 6d. Interestingly, we find that, on average, the L-shape experiences more speed degradation due to smaller obstacle distances compared to the square shape. This is likely because the L-shape has to reorient to fit through the narrow opening between the obstacles, whereas the square shape still fits through in most of its orientations and only has to change phases without the need to rotate. When evaluating the trajectories for the obstacle distance of 20 cm (equal to the size of four units), we find more deviating trajectories for the L-shape (Figure 6i) compared to the square shape (Figure 6i). Furthermore, the L-shape seems to have a preferred direction to surpass the obstacles as we find more trajectories following the top-left part compared to the bottom-right in Figure 6i, which could potentially be explained by the asymmetry in the orientation perspective (Figure S8a). In Figure 6j, we can clearly see this reorientation of the L-shape geometry as it maneuvers in between the obstacles.

From these experiments and simulations, we conclude that the system can adapt to its environment to move in the light source direction, emphasizing the robust nature of the control strategy (Figure S9). Here, the obstacles can be viewed as changes to the equilibrium behavior of the dynamical system, which is different from the effect of changing the light direction. Where the changing direction of the light slowly changes the dimension of the orientation in the search space, the obstructions actually change the search space as a whole. Note that we have only considered a very limited number of environments, where in both cases, there were possibilities to increase the light intensity even after hitting an obstruction. It could be possible for systems to get stuck in environments, similar to how flies sometimes get stuck behind the glass when trying to fly outside. We expect this to occur when the light intensity decreases in all possible movement directions (even though there is still some stochasticity in the movement that could help escape these situations).

7.3. Changing the Geometry by Damaging the Robot

In previous sections, we observed that our system could adapt its behavior to changing objectives and more complex environments. As a final demonstration of robustness, we ran an experiment with a larger rectangular assembly comprised of eight units that we damaged during the experiment (**Figure** 7a–d). At n_{LC} = 50 we split the system in two by manually breaking the physical connection between the units. As a result, both geometries have to adapt their behavior independently to the change in geometry and relearn to move toward the light source (Movie 6.). We observe in Figure 7e that the system is able to increase its light intensity before and after the damage. When comparing the light readings between two units that are in a different section after the cut, we observe how the units can perform phototaxis separately, without any input, and overcome sudden damage, clearly highlighting the robustness of our decentralized approach.

Apart from cutting the system in two, one could also inflict damage to the extent that the system remains connected, but any number of units become inactive. To study the resilience to failure in the system, we also perform simulations with a 5 × 5 (25 units) square lattice. Interestingly, Figure S6d shows that the performance first slightly increases for a system with four inactive units. These results indicate a redundancy in the system. Although it has four non-actuating units (dead weight), the search space becomes less complex; therefore, the system can exploit a better solution in the phase space. Furthermore, we find that even though the $\overline{V}(I)$ gradually decreases with the increase in the number of inactive units, the system remains functional until there is only a single unit left.

8. Conclusion

In this work, we introduced a modular robotic platform to further study decentralized learning algorithms that allow them to adapt to more complex dynamic environments and configurations without needing a centralized controller or electrical connections between units. The current system relies on a form of "collective memory" where each unit explores its own behaviors (the one-dimensional phase parameter). The units evaluate these behaviors using their respective light sensor (LDR) to form a notion of their "environment" (the mapping of the behavior to the phototaxis of the unit). By adjusting its phase, each unit in our modular robotic system influences the collective state of the assembly. The resulting phototaxis depends on a distributed form of memory and coordination. This indirect communication mechanism is based on the partially shared environment, which is facilitated by the physical connection between units and affects the individual decision-making process in subsequent cycles. While this process bears a resemblance to stigmergy that also depends on decentralized behavior and indirect communication, it does not involve altering the terrain. Instead, it relies on the dynamic interplay of local memory and partially shared sensory inputs across the collective. This is exemplified by the

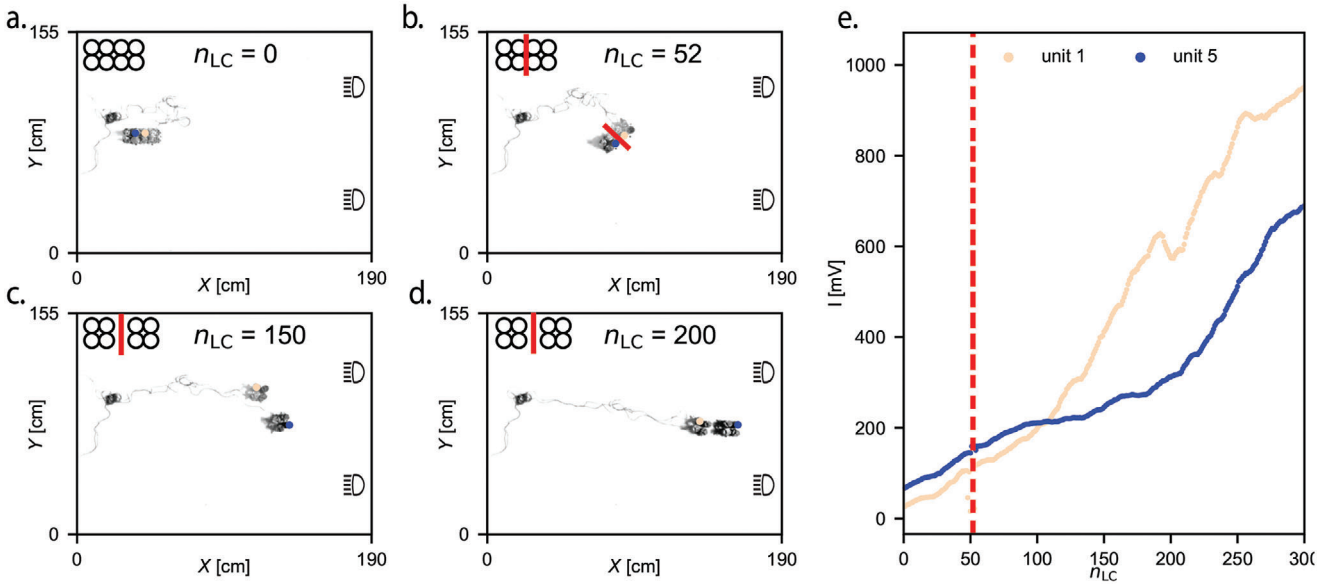

Figure 7. Separating a robot in two during an experiment. a–d) Depicts the system's time evolution for a single experiment for an eight-unit configuration that is cut in half during the experiment at the 50th learning cycle. e) Light measurements at each learning cycle for two units that are each in a separate part of the robot. The reddashed line indicates the moment the robot is separated in two.

partially shared environment that is constantly changing, as observed for our individual units in Figure S4.

In the current study, we increased the complexity of the task by moving from one-dimensional experiments[5] to a two-dimensional modular system. Even though different phase behavior is needed for different geometries to achieve phototaxis, no information or model was a priori needed. We demonstrated that the current short-term memory approach allows the system to adapt to changes in its objective, environment, and geometries. Furthermore, we discovered that relying on short-term memory in this two-dimensional task inherently gives rise to preferential orientations and rotations, thereby strongly influencing the long-term phototaxis of the system. Moreover, we found that these preferential states depend solely on the geometry but can be traversed differently by adapting the learning parameters. Because each unit is identical (both in mechanical behavior as well as in algorithm) and the system does not have any centralized brain, the ability of the system to adjust its behavior to the environment and assembled shape could be viewed as a material property of the system as a whole, which is often referred to as "robotic matter".[5,16,27–29]

Here, the simplicity and universality of the algorithm make it suitable for a wide range of future applications on the interface of soft robots, swarm intelligence, and nanorobotics,[30,34–36] where computational power is limited due to size and weight restrictions or for which the behavior and interactions are difficult to model a priori. Our algorithm could be transferred to the fluidic domain to create more autonomous electronics-free soft robotic systems.[31,32] Furthermore, the algorithm could be beneficial for micro- and nanosystems in medical applications, where the following chemical gradients aid in site-specific drug delivery.[37] Groups of aggregates can be deployed to explore environments where fault tolerance is of the utmost importance (e.g., space exploration and exploration of underground oil reservoirs).[33] However, to reach these applications, more knowledge needs to be gained on how decentralized behavior can be embedded in such systems and how specific objectives can be incorporated and emerge. It should be noted that current work focused mostly on gaining a better understanding of how implicit communication and collective memory can lead to robust behavior. Potential applications will likely bring about compromises, and additional work and redesign of the algorithm are likely needed to make our approach applicable to these different physical implementations. Our study provides a platform to study these principles, both numerically and experimentally.

9. Experimental Section

Fabrication of Units: The individual robotic units consisted of structural components custom-designed and 3D printed in polylactide (PLA). An overview of the assembly can be viewed in Figure S10 (Supporting Information). Two 3D-printed geared disks are positioned on a base disk. A top segment was designed to accommodate a stepper motor (28BYJ-48) and support the electronic components (custom-designed PCB). The robotic units have a 2.1 cm radius, stand 8.5 cm tall, and weigh 75 g when fully assembled. The "brain" of each unit consisted of a microcontroller (ESP32) mounted on a custom circular PCB. A second triangular PCB containing three symmetrically positioned Light Detector Resistors (NSL-19M51 Luna Optoelectronics) was placed in series, connected to the first PCB with 2 mm PCB spacers in between. The PCB was connected to the stepper motor through a stepper driver (ULN2003), all encased in a 3D-printed body and attached with inbus bolts (M2X16 and M3X12). The 3D-printed body consisted of four printed segments. The first was the top frame that housed the PCB, stepper driver, and stepper motor and sensors. A centrally placed brass axle (6 mm) connected the top frame to the bottom frame. A 3D-printed gear connected the stepper motor to two 3D-printed disks (top and bottom) aligned by the central brass axle and held in place by the top body and a 3D bottom body piece. The bottom frame also housed a metal bearing (NMB, Radial Ball Bearing) 10 × 4 × 4 to keep the 3D stepper gear in place. The rotation of the stepper translates into

the bottom and top disk rotations in opposite directions to each other. A Micro-switch (Omron Ultra Subminiature Basis Switch) verified the starting orientation of the disks mounted to the top frame and a notch in the top disk.

Physical connections between the units were needed to let the units interact with each other, allowing the system to displace its center of mass. Soft connectors were used to bind units with each other. The soft connectors were casted using custom-designed molds 3D printed out of PLA (Ultimaker 3). The connections were fabricated with two-component silicone (Smooth-on DS30, in opaque white). The soft connectors had two arms, each fixed to one of the disks of the units. The total length of the soft actuator was $L = 25$ mm. The radial position of the soft connectors was dictated by the poses of the two counter-rotating geared disks actuated by the stepper motor. Using this design, each unit can be connected to a maximum of four neighboring units using a 3D-printed pin connector shown in Figure S8d (Supporting Information).

Experimental Setup: The phototaxis experiments were performed under controlled light conditions on a white multiplex surface (190 cm × 155 cm) encased by a black multiplex frame, as presented in Figure S2b (Supporting Information). At the beginning of each experiment, all units are updated and turned on using a master-slave protocol over Wi-Fi, coordinated from a master ESP32 connected in a one-to-many configuration using the ESP-NOW library. After the initiation of an experiment, every unit gathers sensor and learning data individually. At the end of every learning cycle, the unit sent its collected data back to the master ESP32 connected to the lab computer. The master unit, therefore, collects the gathered data for all the units and writes this data to a .tsv file in real-time. Note that the data only needs to be gathered to visualize the results but is not needed to update the phases of the units. The visual and positional data of the experiments are collected using a GoPro (hero 4) positioned at the center of the white multiplex surface at the height of 1.11m. The GoPro was controlled remotely from the computer that collects the unit's sensor data. The GoPro was prompted to take a photo every time the units had completed a learning cycle.

The position data obtained with the GoPro was synchronized with outputs from the agents to fully reconstruct the dynamics of the system. For example, in the experiments where the behavior of a three-unit assembly in phase space was mapped (Figure S3b, Supporting Information), pictures were collected every ten actuation cycles per phase combination. This allowed for the averaging of unwanted biases in the actuation and a large enough displacement to validate the magnitude and direction of locomotion. For all other experiments, instead, one image was collected every learning cycle of the system. In order to take accurate data while inferring the position of the units over time, optical distortions and camera perspective were corrected. On top of that, the visual data were processed using Python 3.8 and the OpenCV package in three stages. First, the camera's perspective was corrected using four arucos in the corners of the canvas. By measuring the distance between the outer corners of the arucos, the pixel distance could be mapped to the actual distance on the canvas. Second, a color mask and circle detection were used to detect the positions of two 3D-printed discs with $d = 5$ cm and $d = 2.5$ cm that were placed on top of two of the units in the configuration. Last, the positions of the two discs were used to infer the center of the configuration and its rotation. The following protocol was followed to ensure the validity of our data. First, we define the working range of the canvas to be 20 cm from every edge of the canvas to avoid the influence of the borders during the experiment. Experiments were, therefore, cut off if the units surpassed this boundary. As a consequence, in Figure S11 (Supporting Information), it was found that not all experiments were of equal duration, as the experiment was stopped when the units reached the end of the canvas. Furthermore, more than one consecutive missing data point was not accepted. In the case of a single missing data point, linear interpolation was performed to fill the gap. Measured over all the performed experiments, linear interpolation was used for 0.059% of the position data points.

To automate the repetition of the experiments, a mechanism was implemented to consistently restore the initial position of the units at the end of each experiment. The solution consisted of a physical connection between the assembly and a pulling system positioned above the working area. A dedicated servo motor (DFROBOT DF15RSMG) was positioned above the experiment and was connected to the units using a nylon thread with $d =$ 1 mm. When an experiment was over, a flag signal was detected, and the servo motor was activated using the Python interface. As a result, units were pulled to a position approximately beneath the servo.

Unit Actuation Protocol: As displayed in Figure 1a, the experimental units extend and contract their silicone connectors by rotating the two PLA discs in opposite directions. In the contracted state, the distance between the center of the unit and the outer point of the connector measures ≈ 3.5 cm, while in the fully extended state, this distance becomes ≈3.85 cm. The actuation cycle (AC) comprises two stages.

To compare experiments and simulations, a parameter called α, which is set to 0.3, was used. Alpha represents the fraction of the total actuation cycle time (T_{AC}), which is 2 s. During this time, the unit extended its connection arms for ≈ 0.58 s (this time was denoted as D_1).

The remaining part of the actuation cycle ($1 - \alpha$) had two components. First, the unit contracted its four connection arms, which took roughly the same time as the extension. Second, the unit stayed stationary in the contracted state for a duration denoted as D_2.

The last actuation cycle was called the adjustment cycle (AC_a) and differed from the other cycles. In this cycle, the duration of time the unit stays in the contracted state (D_2) was modified to be $D_{ajd} = D_2 - \Delta\phi_i$. This adjustment changed the total duration of the adjustment cycle (AC_a), enabling the unit to start the next learning cycle (LC) with a new phase ϕ_i.

In between the two light intensities measures I_1 and I_2, there were $N_{\mathrm{AC}} = 6$ regular actuation cycles to reduce the noise measured in ΔI and to make the camera tracking feasible. Therefore, there were seven extensions and contractions during a single learning cycle (LC). At the end of every learning cycle, the unit sends its data related to the current learning cycle back to the central computer. Note that every unit operated completely independently from the other units and without synchronization.

Model–Mass–Spring System: To qualitatively capture and to gain more insight into the behavior observed in experiments, the soft modular robot was modeled as a minimal mass–spring system. The aim was to develop a model that is simple enough for quick computation, yet comparable enough to qualitatively capture the observed behavior. A perfect quantitative comparison was not aimed. To achieve this, the system was modelled as a two-dimensional structure consisting of one-dimensional linear elastic elements connected by joints of a certain mass, which interact with the environment through friction. This system decomposed all forces in x and y directions, resulting in a system with n_n nodes connected by e_n elements in a two-dimensional space with $2n_n$ degrees of freedom, where n is the number of units in the assembled robot.

Figure 2a illustrates the nodes and elements represented as masses and springs. Each unit in the assembled robot consisted of four nodes and six elements, where an element connects every pair of nodes. The diagonal elements of the module were periodically actuated to achieve symmetric expansion and shrinking. These elements were referred to as the active elements.

Model–Force Balance: A multimodule system coupled the sides of individual blocks together. Each block had four sides that could be connected to another module. Two connected modules shared the two nodes and the connecting element of the linked side, thereby reducing the number of nodes and elements. The mass of the units was equally distributed over its four adjacent nodes. The mass–spring system was governed by the following equation:

$$\boldsymbol{M\ddot{u}} + \boldsymbol{C\dot{u}} + \boldsymbol{Ku} = \boldsymbol{F} \tag{2}$$

In Equation (2) the $\boldsymbol{M\ddot{u}}$ component describes the forces of mass acceleration, $\boldsymbol{C\dot{u}}$ represents a damping component and the $\boldsymbol{Ku}$ component describes the elastic forces of the springs.

This study redefined Equation (2) as the following force balance equation. Taking the force balance, for each node, it was found

$$\boldsymbol{F}_{\mathrm{m}}(\boldsymbol{\ddot{u}}, \boldsymbol{u}) + \boldsymbol{F}_{\mathrm{f}}(\boldsymbol{\dot{u}}, \boldsymbol{u}) + \boldsymbol{F}_{\mathrm{k}}(\boldsymbol{u}) = \boldsymbol{F}_{\mathrm{e}} \tag{3}$$

In Equation (3), $\boldsymbol{F}_e$ represents the external forces in the system, and $\boldsymbol{F}_k$ is the elastic force of the elements exerted on the nodes. These elements, commonly called bars, are only subjected to axial strain. Both the shear and the bending of the elements do not generate a reaction force. In addition, $\boldsymbol{F}_f$ is the friction force experienced by the nodes, and $\boldsymbol{F}_m$ is the force on the nodes as the mass is accelerated. To be able to apply available solvers, this equation is transformed into a state-space form, where Equation (3) is transformed into a first-order differential equation

$$\boldsymbol{y}(t) = \begin{bmatrix} \dot{\boldsymbol{u}}(t) \\ \boldsymbol{u}(t) \end{bmatrix} \tag{4}$$

which is combined with Equation (3) to find the following derivative of $\boldsymbol{y}(t)$

$$\dot{\boldsymbol{y}}(t) = \begin{bmatrix} \ddot{\boldsymbol{u}}(t) \\ \dot{\boldsymbol{u}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_m{}^{-1}(\boldsymbol{F}_e - \boldsymbol{F}_f(\dot{\boldsymbol{u}}(t)) - \boldsymbol{F}_k(\boldsymbol{u}(t))) \\ \dot{\boldsymbol{u}}(t) \end{bmatrix} \tag{5}$$

Equation (5) was used for direct numerical integration with the RKF45 method of the SciPy python library. The RKF45 method implemented a fourth-order Runge–Kutta formula with an embedded fifth-order formula. By adaptively adjusting the step size, the RKF45 method efficiently balanced accuracy and computational cost, making it a reliable and widely used method for numerical integration of ordinary differential equations.

Next, the force equations for individual nodes described in Equation (3) were elaborated. First, the elastic force of the elements exerted on the springs F_k. F_k arising from the stress and strain relation of the elements was discussed. In the two-dimensional coordinate system, the reaction force exerted by the spring on the adjacent node n_i is given by

$$\boldsymbol{F}_{k\,ij} = \left(\|\boldsymbol{r}_j - \boldsymbol{r}_i\| - l_0\right) K_s \frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{\|\boldsymbol{r}_j - \boldsymbol{r}_i\|} \tag{6}$$

The constant strain within the bar results in a reaction force of the same magnitude and opposite direction for node n_j. Note that there was a difference in the initial length l_0 between the diagonal and border elements. Therefore, the difference in l_0 between the diagonal and border elements implied a factor of $\sqrt{2}$ difference in stiffness between these elements, as the current implementation assumed the same K_s for all elements.

The total reaction force on node n_i is the sum of forces for all adjacent elements

$$\boldsymbol{F}_{k,\,\mathrm{tot}\,i} = \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{F}_{k_{ij}} \boldsymbol{A}_{ij} \tag{7}$$

In Equation (7) A_{ij} is a unitary variable that denotes the presence of an element connecting nodes n_i and n_j. Where, A_{ij} is one if there is an element connecting the two nodes and zero otherwise. The parameters used to model the springs can be found in Table S1.

Next, the friction on the nodes was based on a general friction model.[5] This friction model assumed the total friction force F_f to be a sum of Stribeck, Viscous, and Coulomb friction components. Stribeck friction F_s referred to the negative slope of the friction force as the velocity V starts to increase. Viscous friction F_v results in friction proportional to the velocity. Coulomb friction F_c represents constant friction independent of the velocity. Last, there is friction at zero velocity, often referred to as breakaway friction F_{break}.

Since the friction force had the opposite direction as the direction of the velocity, a velocity-based transformation component is added to transform the friction from the local reference frame to the global reference frame. Furthermore, with the use of the estimated parameters obtained in friction experiments performed for the one-dimensional studies[5] (Table S2), the friction force simplifies to:

$$\boldsymbol{F}_{f_i} = -\left(F_C \cdot \tanh\left(\frac{\|\dot{\boldsymbol{u}}_i\|}{v_{Coul}}\right) + f\|\dot{\boldsymbol{u}}_i\|\right)\frac{\dot{\boldsymbol{u}}_i}{\|\dot{\boldsymbol{u}}_i\|} \tag{8}$$

The current system has individual modules that can expand and shrink back to their original shape. These deformations were achieved by simultaneously activating the diagonal elements in each unit. The current model implements actuation by redefining l_0 in Equation (6) as the time-dependent variable preferred spring length $L_p(t)$ in each diagonal element. The preferred spring length $L_p(t)$ of the diagonal elements was divined as the sum of the initial spring length l_0 plus the time-dependent actuation length $l_a(t+\phi)$. In order to make the differential equation of the force balance less stiff and speed up the simulation, a smooth version of the actuation function was implemented for $l_a(t+\phi)$ based on the actuation used in the experimental units. By making use of a piecewise sine function, the extension–contraction cycle of the experimental units was qualitatively approximated. The active spring length during an actuation cycle is given by Equation (9) with the parameters of Table S3.

$$l_a(t,\phi_i) = \begin{cases} \omega \cdot \sin(2\pi f\,(t+\phi_i)) & \text{if } t < \frac{T_{mot}}{4} \\ \omega \cdot \sin\left(2\pi f \frac{T_m}{4}\right) & \text{if } t \in \left[\frac{T_m t}{4},\ \frac{T_{mot}}{4} + D_1\right) \\ \omega \cdot \sin\left(2\pi f\,((t+\phi_i) - D_1)\right) & \text{if } t \in \left[\frac{T_m}{4} + D_1,\ \frac{3T_m}{4} + D_1\right) \\ \omega \cdot \sin\left(2\pi f \frac{3T_m}{4}\right) & \text{if } t \in \left[\frac{3T_m}{4} + D_1 t,\ \frac{3T_m}{4} + T_s\right) \\ \omega \cdot \sin\left(2\pi f\,((t+\phi_i) - D_1 - T_s)\right) & \text{if } t > \frac{3T_m}{4} + T_s \end{cases} \tag{9}$$

Furthermore, the same learning cycle (LC) as used for the experimental units was implemented apart from the number of regular actuation cycles between the two light measurements I_1 and I_2, as there was no noise in the light measurements during the simulations and because the only aim was to capture the qualitative behavior of system the number of regular actuations N_{AC} was reduced to three. Thereby the learning cycle LC of the system in the simulations consisted of four expansion–contraction cycles.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This project was funded by the European Union's Horizon 2020 research and innovation program under grant agreement No. 767 195. This work was part of the Dutch Research Council (NWO) and was performed at the research institute AMOLF.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The experimental and numerical data that support the findings of this work and the computer algorithms necessary for running the analysis have been deposited to Zenodo (doi:10.5281/zenodo.11092762).

Keywords

decentralized, embodied learning, phototaxis, physical intelligence, robotic matter

Received: September 11, 2023
Revised: March 6, 2024
Published online:

[1] F. Iida, C. Laschi, *Procedia Comput. Sci.* **2011**, *7*, 99.
[2] D. Rus, M. T. Tolley, *Nature* **2015**, *521*, 467.
[3] S. Heydari, A. Johnson, O. Ellers, M. J. McHenry, E. Kanso, *J. R. Soc., Interface* **2020**, *17*, 20190700.
[4] T. Kano, R. Yoshizawa, A. Ishiguro, *Bioinspir. Biomim.* **2017**, *12*, 046009.
[5] G. Oliveri, L. C. van Laake, C. Carissimo, C. Miette, J. T. Overvelde, *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2017015118.
[6] K. M. Stiefel, G. A. Barrett, *J. Marine Sci. Eng.* **2018**, *6*, 112.
[7] Y. Ozkan-Aydin, D. I. Goldman, M. S. Bhamla, *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2010542118.
[8] R. M. McKenzie, M. E. Sayed, M. P. Nemitz, B. W. Flynn, A. A. Stokes, *Soft Robot.* **2019**, *6*, 195.
[9] X. Sui, H. Cai, D. Bie, Y. Zhang, J. Zhao, Y. Zhu, *Appl. Sci.* **2020**, *10*, 294.
[10] N. J. Wilson, S. Ceron, L. Horowitz, K. Petersen, *Front. Robot. AI* **2020**, *7*, 44.
[11] G. Gardi, M. Sitti, *Phys. Rev. Lett.* **2023**, *131*, 058301.
[12] N. H. Nguyen, D. Klotsa, M. Engel, S. C. Glotzer, *Phys. Rev. Lett.* **2014**, *112*, 075701.
[13] T. H. Tan, A. Mietke, J. Li, Y. Chen, H. Higinbotham, P. J. Foster, S. Gokhale, J. Dunkel, N. Fakhri, *Nature* **2022**, *607*, 287.
[14] M. Fruchart, R. Hanai, P. B. Littlewood, V. Vitelli, *Nature* **2021**, *592*, 363.
[15] O. Peleg, J. M. Peters, M. K. Salcedo, L. Mahadevan, *Nat. Phys.* **2018**, *14*, 1193.
[16] B. Saintyves, M. Spenko, H. M. Jaeger, *arXiv preprint arXiv:2304.03125* **2023**.
[17] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, R. D'Andrea, *IEEE Control Syst. Mag.* **2014**, *34*, 46.
[18] S. Li, R. Batra, D. Brown, H.-D. Chang, N. Ranganathan, C. Hoberman, D. Rus, H. Lipson, *Nature* **2019**, *567*, 361.
[19] M. Shimizu, A. Ishiguro, T. Kawakatsu, in Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, IEEE, Piscataway, NJ **2005**, pp. 2982–2987.
[20] J. W. Romanishin, K. Gilpin, S. Claici, D. Rus, in Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), IEEE, Piscataway, NJ **2015**, pp. 1925–1932.
[21] W. Savoie, T. A. Berrueta, Z. Jackson, A. Pervan, R. Warkentin, S. Li, T. D. Murphey, K. Wiesenfeld, D. I. Goldman, *Sci. Rob.* **2019**, *4*, eaax4316.
[22] R. Pfeifer, M. Lungarella, F. Iida, *Science* **2007**, *318*, 1088.
[23] A. J. Ijspeert, A. Crespi, D. Ryczko, J.-M. Cabelguen, *Science* **2007**, *315*, 1416.
[24] J. Werfel, K. Petersen, R. Nagpal, *Science* **2014**, *343*, 754.
[25] J. Chen, M. Gauci, W. Li, A. Kolling, R. Groß, *IEEE Trans. Robot.* **2015**, *31*, 307.
[26] F. Berlinger, M. Gauci, R. Nagpal, *Sci. Robot.* **2021**, *6*, 50.
[27] A. Kotikian, C. McMahan, E. C. Davidson, J. M. Muhammad, R. D. Weeks, C. Daraio, J. A. Lewis, *Sci. Robot.* **2019**, *4*, eaax7044.
[28] M. A. McEvoy, N. Correll, *Science* **2015**, *347*, 1261689.
[29] S. Shankar, A. Souslov, M. J. Bowick, M. C. Marchetti, V. Vitelli, *Nat. Rev. Phys.* **2022**, *4*, 380.
[30] M. F. Reynolds, A. J. Cortese, Q. Liu, Z. Zheng, W. Wang, S. L. Norris, S. Lee, M. Z. Miskin, A. C. Molnar, I. Cohen, et al., *Sci. Rob.* **2022**, *7*, eabq2296.
[31] D. Drotman, S. Jadhav, D. Sharp, C. Chan, M. T. Tolley, *Sci. Robot.* **2021**, *6*, eaay2627.
[32] Q. He, R. Yin, Y. Hua, W. Jiao, C. Mo, H. Shu, J. R. Raney, *Sci. Adv.* **2023**, *9*, eade9247.
[33] L. Bayindir, *Neurocomputing* **2016**, *172*, 292.
[34] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, *Swarm Intell.* **2013**, *7*, 1.
[35] E. Osaba, J. Del Ser, A. Iglesias, X.-S. Yang, Soft computing for swarm robotics: new trends and applications, **2020**.
[36] M. Z. Miskin, A. J. Cortese, K. Dorsey, E. P. Esposito, M. F. Reynolds, Q. Liu, M. Cao, D. A. Muller, P. L. McEuen, I. Cohen, *Nature* **2020**, *584*, 557.
[37] G. Alici, *Expert Rev. Med. Devices* **2015**, *12*, 703.