


Chemotaxis of an elastic flagellated microrobotChaojie Mo 

*Aircraft and Propulsion Laboratory, Ningbo Institute of Technology, Beihang University, Ningbo 315100, People's Republic of China
and State Key Laboratory of Fluid Power and Mechatronic Systems, Department of Engineering Mechanics, Zhejiang University,
Hangzhou 310027, People's Republic of China*

Qingfei Fu

*School of Astronautics, Beihang University, Beijing 100191, People's Republic of China
and Aircraft and Propulsion Laboratory, Ningbo Institute of Technology, Beihang University, Ningbo 315100, People's Republic of China*

Xin Bian 

*State Key Laboratory of Fluid Power and Mechatronic Systems, Department of Engineering Mechanics, Zhejiang University,
Hangzhou 310027, People's Republic of China*



(Received 30 July 2022; accepted 29 September 2023; published 23 October 2023)

Machine learning algorithms offer a tool to boost mobility and flexibility of a synthetic microswimmer, hence may help us design truly smart microrobots. In this work, we design a two-gait microrobot swimming in circular or helical trajectory. It utilizes the coupling between flagellum elasticity and resistive force to change the characteristics of swimming trajectory. Leveraging a deep reinforcement learning (DRL) approach, we show that the microrobot can self-learn chemotactic motion autonomously (without heuristics) using only several current and historical chemoattractant concentration and curvature information. The learned strategy is more efficient than a human-devised shortsighted strategy and can be further greatly improved in a stochastic environment. Furthermore, in the helical trajectory case, if additional heuristic information of direction is supplemented to evaluate the strategy during the learning process, then a highly efficient strategy can be discovered by the DRL. The microrobot can quickly align the helix vector to the gradient direction using just several smart sequential gait switchings. The success for the efficient strategies depends on how much historical information is provided and also the steering angle step size of the microrobot. Our results provide useful guidance for the design and smart maneuver of synthetic spermlike microswimmers.

DOI: [10.1103/PhysRevE.108.044408](https://doi.org/10.1103/PhysRevE.108.044408)**I. INTRODUCTION**

Biological microswimmers live in the regime of low Reynolds number [1], where the viscous force dominates the inertial force at microscale. As a result, they propel themselves using strategies completely different from that of macroscopic organisms. Evolution has led the microscopic organisms to develop effective propellers such as wriggling flagellum and rotating helix [2], which can overcome and even exploit the overwhelming viscous forces [3,4]. In particular, bacterial flagellum is one of the most renowned propeller at microscale [5,6]. To understand life at microscale, it is crucial to investigate not only the biological structures of the self-propelling organisms, but also their propulsion mechanism from the perspective of fluid dynamics [7,8]. It is for this reason that the microscopic propulsion has drawn lots of attention and fruitful results have been reported in the past decades [3,4]. Furthermore, studies on microswimmers may also teach us to design intelligent synthetic microswimmers. These microrobots may be used for cargo deliveries and

biomedical manipulations in microfluidics and even *in vivo* systems, hence offer a great potential for noninvasive drug delivery and medical treatment. Many synthetic microswimmers have been invented and further successfully applied: catalytic Janus particles exploiting the diffusiophoresis or thermophoresis process to accomplish self-propulsion [9–12]; catalytic nanomotors propelled by generating bubble jet [13]; self-propelling droplet by the Marangoni stress in a surfactant solution [14]; rotators breaking kinetic symmetry and actuated by external magnetic field [15]; biohybrid microswimmers imitating sperm cells [16–18]; and neutrophil-based microrobots that can actively deliver cargo to malignant glioma through chemotactic motion [19]. These examples evidenced microswimmers as a promising research direction due to the great potential. However, if the drug delivery is to be accomplished, then there are at least two issues to resolve: the microswimmer must survive the immune attack and be able to cross biological barriers, and it must maneuver precisely to the target through a complex environment. In this work, we address ourselves to the maneuvering problem.

There are many ways to steer a microswimmer toward a specific direction, such as chemotaxis [20,21], magnetotaxis

*bianx@zju.edu.cn

[13,22–24], phototaxis [25,26], gravitaxis [27,28], viscotaxis [29,30], and so on. Among these, the magnetotaxis is one of the most frequently adopted in laboratory due to its noninvasive characteristics and high efficiency on maneuvering. But the disadvantage of magnetotaxis is also apparent: It needs a large system to generate the rotational magnetic field and can usually manipulate only one microswimmer at a time [24]. In contrast, biological microswimmers, such as sperm cells [31], *Escherichia coli* [32], and green algae [33], often follow the chemotactic process to swim by themselves towards the target or to seek food. This has inspired researchers to envisage and design synthetic microswimmers (e.g., catalytic colloidal swimmers and droplet swimmers) to implement chemotaxis in specific environments [20,21,34]. However, it remains open how a synthetic flagellated microswimmer can be designed and programmed to implement chemotaxis.

It has been found that realistic sperm cells change direction using either nonzero average flagellum curvature or second harmonics [35–37]. A sperm cell usually swims in circular or helical trajectory [38]. It compares the concentration information along its swimming trajectory and modulates the flagellar beat to change the trajectory curvature and torsion. Eventually, the sperm cell steers toward the gradient direction in drifting circles or deformed helices [31,39]. This inspires us to design a flagellated microrobot that can implement chemotaxis in a similar way. However, it is unrealistic to fabricate a microrobot that can be controlled in a way as sophisticated as a sperm cell. The less complex is the principle the more feasible is the realisation. In this work, we propose a simple *in silico* elastic flagellated microrobot that can be controlled through the beating frequency alone. We will first validate our idea of the microrobot with computational fluid dynamics simulation and then investigate how such a microrobot can be steered toward a specific direction through chemotaxis.

Recently, there have been many emerging efforts to design intelligent microswimmers using machine learning techniques, especially the reinforcement learning (RL) algorithms. We summarize three main application categories of RL approach on the intelligent microswimmer. (1) The RL approach is employed to design locomotion gaits at low Reynolds number. For example, it has been shown by researchers that with a simple Q -learning method the Najafi-Golestanian swimmer [40] and multilink microswimmers [41] can self-learn propulsion. When the structure of the swimmer becomes complex, the RL algorithm can discover new classes of swimming gaits that are more efficient than a human-designed one. (2) The RL approach is utilized to discover smart steering strategy to navigate through complex environment. For example, Alageshan *et al.* [42] studied the path-planning problem of a microswimmer through a complex turbulent flow field. They employed a multiswimmer adversarial Q -learning algorithm to find the optimised steering strategy towards a specified target. Gunnarson *et al.* [43] studied the navigation of a swimmer in a time-varying vortical flow field, where they feed the background flow information (velocity or vorticity) to a deep neural network that determines the swimmer's action. As a result, the swimmers successfully discovered efficient policies to reach the target. Yang *et al.* [44] kept a Janus particle to rotate randomly to perceive the obstacles around

itself, and thereafter applied a deep reinforcement learning (DRL) algorithm to train the Janus particle. They showed that the Janus particle guided by the deep convolutional Q network can act smartly to bypass the obstacles and swim toward its target. There are many more researches [45–48] in this category. (3) RL approach can also be used to train the microswimmer to learn klinotactic behavior. Colabrese *et al.* [49] and Gustavsson *et al.* [50] used the Q -learning method to train active gravitactic microswimmers to accomplish counter-gravity navigation through 2D Taylor-Green vortex flow and 3D chaotic flow field, respectively. Hartl *et al.* [51] studied the self-learned chemotaxis of a Najafi-Golestanian swimmer in 1D space. They decouple the task into two parts: to teach the swimmer to swim, and to train the swimmer to determine the gradient direction of the chemoattractant concentration field and steer itself to that direction. They applied the neural evolution of augmenting topologies (NEAT) technique to optimize not just the weights of the neural network but also the topology. Very simple architectures of the neural network have been found to accomplish the chemotaxis task. Note that the three application categories sometimes overlap with each other. For example, in the work of Zou [52], where the targeted navigation of a three-beads swimmer is studied using a DRL, the locomotory gaits and steering strategy are learned simultaneously using the DRL. And in the works of Colabrese *et al.* [49] and Gustavsson *et al.* [50], the RL approach also guides the gravitactic microswimmers to navigate through complex flow fields, hence the second category also fits for these cases. In summary, the RL algorithms have been proven to be a very powerful tool for intelligent control of swimmers [53,54]. Nevertheless, most of the studies either completely neglect the swimmer's structure details (swimmers are assumed to be active material points) or adopt very simple microswimmers (multibeads swimmers or Janus particle). In this article, we will incorporate the RL technique to investigate the maneuvering strategy of an elastic flagellated microrobot.

We will show that with little information the microrobot can autonomously self-learn chemotactic motion by gait switching. The microrobot makes decision according to the current and several historical records of chemoattractant concentration and curvature information. The reward function is also calculated using these information, thus rendering the learning an autonomous process. The learned strategy is always better than a human-devised shortsighted strategy. In 3D motion, the learned strategy has some probability to fail, but stochasticity can significantly improve its performance and guarantee a successful chemotactic behavior. If accurate information about direction is supplemented to calculate a reward function that enhances direction alignment, then highly efficient strategies can be discovered by the DRL. The success for such efficient strategies depends on how much historical information is provided for the microrobot and also the steering angle step size of the microrobot. Our results provide useful guidances for the design and smart maneuver of synthetic spermlike microswimmers.

The article is organized as follows. In Sec. II we design the elastic flagellated microrobot and validate our idea using smoothed dissipative particle dynamics (SDPD) simulation. In Sec. III we introduce the simplified microrobot swimming

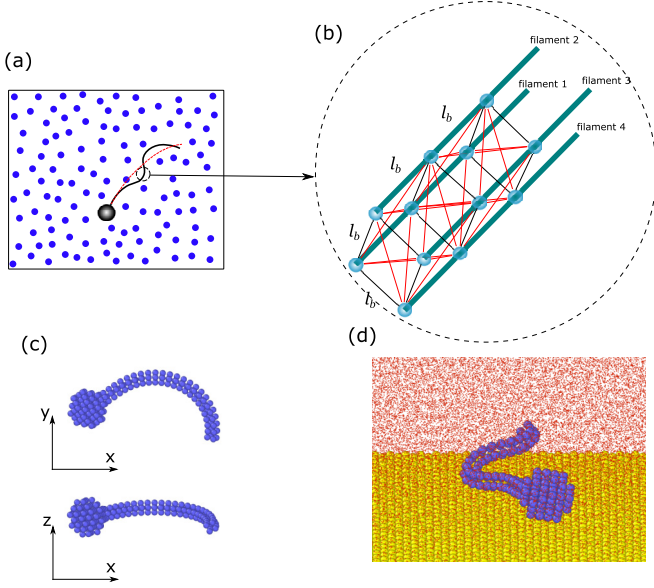


FIG. 1. Schematic and snapshot of the microrobot simulation model. (a) The spermlike microrobot is immersed in fluids simulated by SDPD. The dashed red line is the static state of the flagellum, which is curved. (b) The flagellum is modelled using discrete bead-spring model. The black lines are orthogonal springs, the light red lines are diagonal springs, the thick green lines are the actuating springs. (c) The shape of the microrobot at rest. (d) Snapshot of the SDPD particle model: blue spheres constitute the microrobot; small red dots are fluid particles; yellow spheres on the bottom are wall particles.

model, the implementation of chemotaxis through comparison of current and historical signals, the human-devised shortsighted strategy, the DRL approach, and the simulation parameters used in our study. In Sec. IV the results using an autonomous reward function is presented and discussed. In Sec. V, the results using a direction alignment-based reward function is presented and discussed. We draw conclusions and provide some discussions in Sec. VI.

II. AN ELASTIC FLAGELLATED MICROROBOT

We consider a spermlike microrobot propelled by beating an elastic flagellum. The flagellum is curved in a plane different from the beating plane. We employ SDPD to simulate the propulsion of such a microrobot in fluids. The author's previous SDPD modeling of a 2D sperm cell [55,56] is extended to a 3D swimmer model developed by Rode *et al.* [57]. The schematic of the model is shown in Fig. 1. The spermlike microrobot has a sphere head connected to a long flagellum. Both the microrobot and the surrounded fluid are discretized by SDPD particles. And the SDPD particles constituting the microrobot are connected by harmonic springs [Fig. 1(b)]. This makes the head quasi-rigid, but the flagellum is elastic due to its slender shape. As shown in Fig. 1(b), the flagellum is constituted by four filament, the springs on the first and the third filament have varying equilibrium length, while the springs on the second and fourth filaments have static

equilibrium length:

$$\begin{aligned} l_1^i &= l_b + A \sin(kl_b i - \omega t) + b_1, \\ l_2^i &= l_b + b_2, \\ l_3^i &= l_b - A \sin(kl_b i - \omega t) - b_1, \\ l_4^i &= l_b - b_2. \end{aligned} \quad (1)$$

Here l_b is the spring length when the flagellum is straight and at rest, A is the actuation amplitude, i is the segment index, k is the wave number, ω is the beating frequency, and b_1 and b_2 are constants used to impose intrinsic curvature. When $b_1 = 0$ and $b_2 = 0$ the flagellum beats in a sinusoidal way [57]. If $b_1 \neq 0$ and $b_2 = 0$, then the flagellum is curved at rest, but it is in the same plane as the beating. This makes the beating asymmetric but remains planar. If b_2 is also nonzero, then the flagellum is curved in a plane different with the beating plane [Fig. 1(c)]. The beating is nonplanar in this case [Fig. 1(d)]. In this article we consider only the case $b_1 > b_2 > 0$.

The elasticity of the flagellum is characterized by the Sperm number $S_p = L(\xi_{\perp} \omega / \kappa_f)^{1/4}$, where L is the length of the flagellum, ξ_{\perp} is the resistive force coefficient in the direction normal to the flagellum, κ_f is the bending stiffness. The S_p number is the ratio of the viscous force to the elastic force. Larger S_p indicates more flexible flagellum. Since S_p depends on the beating frequency, we expect that by changing only ω , the beating pattern will also change as a result of the coupling effects of the viscous and elastic forces. Therefore, the swimming trajectory will have different curvatures and torsions, and it is possible to implement chemotaxis by controlling the beating frequency alone. See the Appendix for more details about the simulation.

We first show the simulation results of a microrobot swimming in bulk fluid at different beating frequencies. As shown in Fig. 2(a), the trajectories of the microrobot are helical. The trajectories have the same starting point but different radius and pitch. Moreover, when the beating frequency increases the trajectory flips from being left-handed to right-handed, thus the net migration direction of the microrobot (i.e., the helix vector) reverses. This phenomenon is analogous to the bidirectional propulsion of curved elastic filament in 2D [58,59]. We find that the swimming velocity of the microrobot increases with beating frequency, as shown in Fig. 2(b). We further demonstrate that the evolution of curvature and torsion of the trajectories in Figs. 2(c) and 2(d), respectively. A larger beating frequency leads to a smaller curvature but a larger torsion. The sign of the torsion may change if the beating frequency is large enough, which corresponds to the reverse of the helix vector.

If the microrobot swims near a wall, then it is attracted to the wall due to the dipolar force flow field it sets up [3]. The corresponding simulation results are presented in Fig. 3. Note that the parameters are all the same as in Fig. 2, except the presence of a wall on the bottom. As shown in Fig. 3(a), in all cases the microrobot is first attracted to the wall, and then it swims in circular trajectories within a plane parallel to the wall. The circular trajectories are different for different beating frequencies. A larger frequency leads to a smaller distance to the wall, presumably due to the stronger dipolar flow field generated at the higher frequency. The evolution

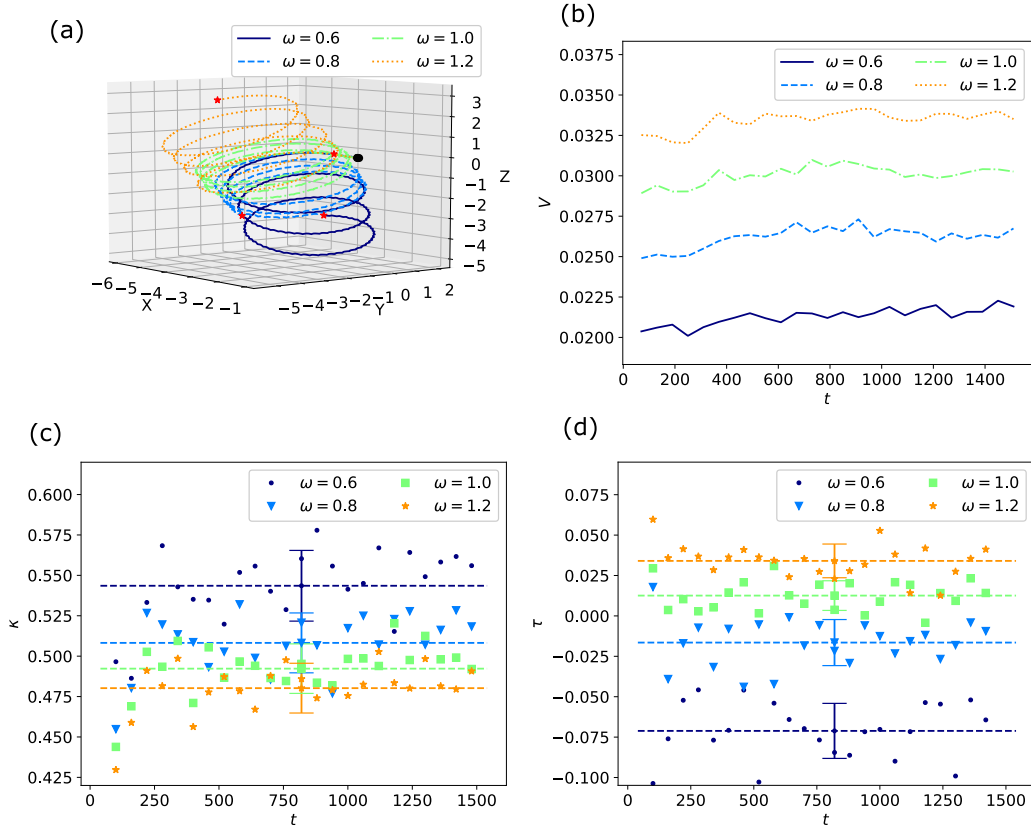


FIG. 2. Simulation result of a curved flagellated microrobot swimming in bulk fluid. (a) The swimming trajectories. The black dot marks the starting point of the trajectories, the red stars are the end points. (b) The swimming velocity. (c) The curvature of the trajectories. (d) The torsion of the trajectories. The dashed lines and the error bars mark the mean and standard error respectively.

of the swimming velocities is shown in Fig. 3(b) and the evolutions of the curvatures and torsions of the trajectories are demonstrated in Figs. 3(c) and 3(d), respectively. We observe clearly that for each set of parameters there is always a transient stage corresponding to the process of being attracted to the wall. After the transient stage, v , κ , and τ become nearly constant. The stable swimming velocity increases with the beating frequency. Similar to the case in bulk fluid, a higher frequency also leads to a smaller curvature when the wall is present. But unlike the bulk case, the stable torsion becomes zero eventually, which indicates a planar trajectory as a result of the wall confinement.

The results above unveil that the curvature and torsion of the swimming trajectory can be controlled through the beating frequency alone. Therefore, from the perspective of steering a flagellated microrobot, it is not necessary to change the intrinsic curvature of the flagellum independently to steer the microrobot. To independently control the intrinsic curvature would require either additional actuation units to be installed inside the flagellum, or to use some stimuli-responsive materials to fabricate the flagellum, either of which raises the difficulty in implementation at microscale. As an alternation, we can rely on the passive body-environment interaction to steer the microrobot by changing its beating frequency alone. This fact inspires us to explore how chemotaxis can be implemented by controlling the frequency of an elastic flagellated microrobot alone. Since only the velocity, curvature and torsion are important to describe the swimming trajectory, in the

next section, we will neglect the hydrodynamics and adopt a simplified swimming trajectory model to investigate the chemotaxis problem of the microrobot.

Note that the SDPD simulations presented in this section serve merely as a validation of our idea to steer the microrobot through the beating frequency alone. The accurate relationships among the beating frequency, the flagellum material properties, the flagellum geometric parameters, and the trajectory characteristics are not investigated in detail. In the next sections we will further take the liberty to pose many hypothetical combinations of different trajectory parameters and explore the conditions of efficient steering strategy to be discovered. The question that how a microrobot can swim in those exact trajectory parameters is left for future study.

III. MODELS AND MATERIALS

In the biological world, sperm cells swim by generating waves of deformation propagating along their flagella and change directions through nonzero average curvatures [31,35] and/or second harmonics [37]. Although it has been shown that the first steering mechanism has higher effectiveness [36], there is a similar signal transferring process in both mechanisms. In short, the chemoattractants in the environment bind the receptors on the surface of the sperm cells and initiate a signaling cascade to change the intracellular concentration of Ca^{2+} . Moreover, the Ca^{2+} concentration influences the activity of the dynein motors, which further modulate the beating

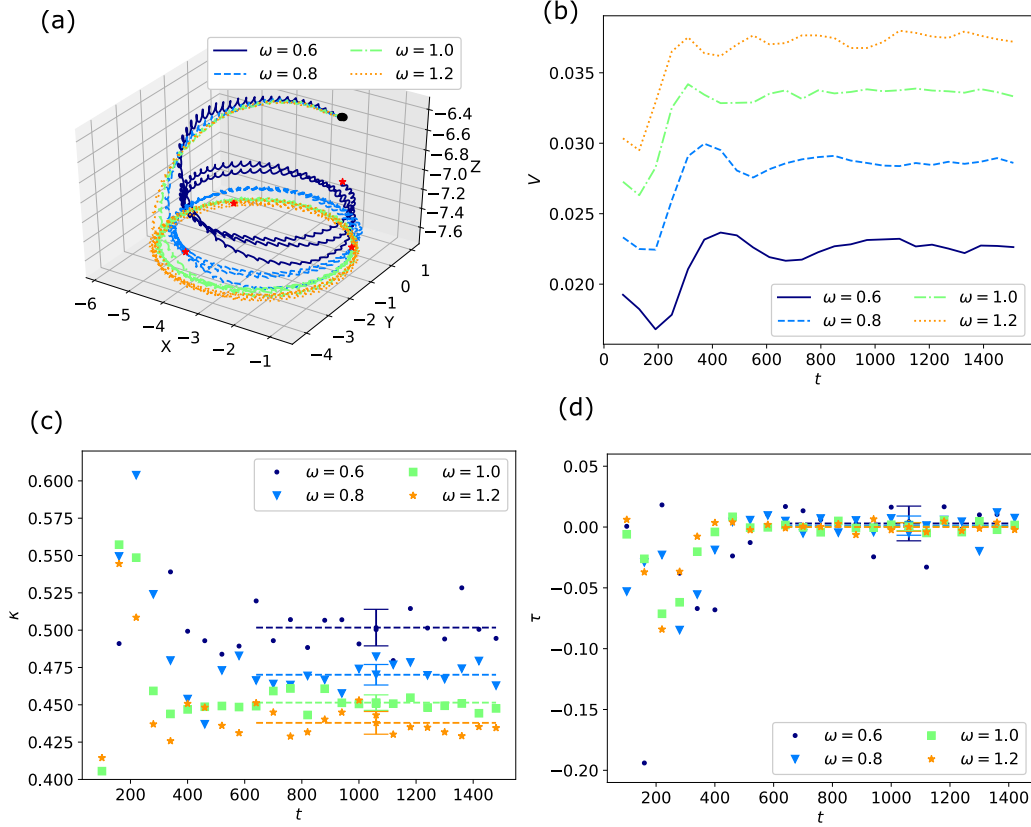


FIG. 3. Simulation result of a curved flagellated microrobot swimming near a wall. (a) The swimming trajectories. The black dot marks the starting point of the trajectories, the red stars are the end points. The wall is on the bottom parallel to the $x - y$ plane. (b) The swimming velocity. (c) The curvature of the trajectories. (d) The torsion of the trajectories. The dashed lines and the error bars mark the mean and standard error respectively.

waveform of the flagella. The sperm cells rely on the whole process above to change directions. Phenomenologically, the chemotactic signaling network initiates a series of complex tasks: temporally comparing the chemoattractant concentrations along the swimming trajectory; determining the gradient direction of the concentration field; and steering the sperm to swim toward the gradient direction upwards. In the works of Friedrich *et al.* [31,39,60], an adaptive dynamical system has been designed to imitate the behavior of the chemotactic signaling network and capture the essence of the chemotaxis behavior.

In this work, we aim to investigate the chemotaxis of a microrobot. To fabricate complex mechanics at microscale is of great challenge, we do not expect that a microrobot can maneuver itself in a way as sophisticated as a sperm cell does. In contrast, we assume the microrobot to be as simple as possible. Therefore, the flagellated microrobot described in the above section is considered here, and we suppose that it can be controlled by changing its beating frequency alone. It is also assumed that the microrobot can sense and record the local concentration field at discrete time steps. We explore how the microrobot can implement chemotaxis through comparing the current and historical concentration information and varying its beating frequency. Furthermore, we suppose the microrobot can switch between only two beating frequencies, making it essentially a two-gait microrobot (Table I).

A. Simplified microrobot model

For simplicity, the swimming model from Friedrich and Jülicher [31,39] is employed here. The interacting details between the flagellum and the fluid are neglected and the position of a sperm cell is represented by the average position of the center of mass in one beating cycle. The swimming trajectory is then described by the Frenet-Serret equation:

$$\dot{\mathbf{r}} = v\mathbf{t}, \quad \dot{\mathbf{t}} = v\kappa\mathbf{n}, \quad \dot{\mathbf{n}} = -v\kappa\mathbf{t} + v\tau\mathbf{b}, \quad \dot{\mathbf{b}} = -v\tau\mathbf{n}, \quad (2)$$

where \mathbf{r} is the position vector of the sperm cell, \mathbf{t} is the unit tangential vector, \mathbf{n} is the unit normal vector, \mathbf{b} is the unit binormal vector, v is the swimming speed, and κ is the curvature of the swimming trajectory that will be modulated by an agent. The reason for the employment of this model is twofold: first, its dynamics is sufficiently complex, which imitates the realistic behaviors of the microrobot; second, it does not involve difficult computations of fluid-structure

TABLE I. Two gaits of the microrobot.

Parameters	Gait 1	Gait 2
Beating frequency f	f_1	f_2
Swimming velocity v	v_1	v_2
Trajectory curvature κ	κ_1	κ_2
Trajectory torsion τ	τ_1	τ_2

interactions so that we can devote more efforts to discover the intelligent steering strategies.

Equation (2) can be integrated numerically. It is first re-framed to be [61]

$$\dot{\mathbf{r}} = v(t)\mathbf{t}, \quad \dot{\mathbf{i}} = \boldsymbol{\Omega} \times \mathbf{t}, \quad \dot{\mathbf{n}} = \boldsymbol{\Omega} \times \mathbf{n}, \quad \dot{\mathbf{b}} = \boldsymbol{\Omega} \times \mathbf{b}, \quad (3)$$

where $\boldsymbol{\Omega}$ is angular velocity:

$$\boldsymbol{\Omega}(t) = v(t)[\tau(t)\mathbf{t}(t) + \kappa(t)\mathbf{b}(t)]. \quad (4)$$

Then the position can be updated using Euler scheme, the rotation angle is also determined using Euler scheme (or Euler-Maruyama scheme if the curvature and torsion fluctuate), the direction vectors can be updated using the Rodrigues rotation formula.

The microrobot can switch between two gaits with different swimming trajectory parameters (Table I). Note that in the simplified swimming model, the beating frequency parameter becomes redundant. Since we intend to investigate how the relationship among κ_1 , κ_2 , τ_1 , and τ_2 impacts on the maneuvering strategy, we will also pose some hypothetical value combinations for them later.

There are two time steps in this problem: the time step ΔT which is the time interval between two consecutive actions of switching gaits, and the time step Δt which is the time step for the Euler integration scheme. If Δt is fixed, then it is difficult to guarantee that the division between ΔT and Δt is an integer. This may interfere the accurate timing of the actions and leads to error accumulation. To overcome this problem, we select a preferred Δt but also allow the value to float around the preferred one to make sure that $\Delta T/\Delta t$ is always an integer.

B. Environment information, action frequency, and memory capacity

We consider a microrobot that swims in a field of chemoattractant, which is described by the concentration $c(\mathbf{r})$. The microrobot is aware of the concentration in its current position and can also remember it for a period of time. When the microrobot is about to take action, it first put the current concentration information into its memory, and use all the concentration information in the memory to make decision. We use f_T to denote the action frequency, N_T to denote the number of discrete time points that the microrobot can remember. Then $\Delta T = 1/f_T$, $N_T \Delta T$ is the total time period the microrobot can remember. In a realistic chemotactic process, a sperm cell swims in a helical or circular trajectory to sample the concentration field. It perceives periodic stimulus from the environment and regulates the curvature and torsion of the trajectory accordingly so that it can bend the trajectory toward the chemoattractant source. In light of this, we need to allow our microrobot to remember the past information at least for one average period of the helical swimming. Therefore, it is straightforward to set $N_T \Delta T = 2\pi/\omega_0$, with ω_0 being the average frequency of the helical or circular swimming:

$$\omega_0 = \frac{1}{2}[v_1(\kappa_1^2 + \tau_1^2)^{1/2} + v_2(\kappa_2^2 + \tau_2^2)^{1/2}]. \quad (5)$$

Then the most simple and nontrivial case is $N_T = 2$, which means that the microrobot can only remember information at two time points, and make decision twice every period. In this

Algorithm 1. A shortsighted maneuvering strategy.

```

1:  $t = 0$ 
2: while  $t < t_{\text{lifc}}$  do
3:   Integrate and update  $\mathbf{r}$ ,  $\mathbf{t}$ ,  $\mathbf{n}$ ,  $\mathbf{b}$  from  $t$  to  $t + \Delta T$ 
4:   Update  $t = t + \Delta T$ 
5:   Get the concentration value at the current position  $c(t)$ 
6:   if  $c(t) == \max([c(t), c(t - \Delta T), \dots, c(t - (N_T - 1)\Delta T)])$ 
7:     then
8:       if  $r_1 < r_2$  then
9:         Set  $v = v_1$ ,  $\kappa = \kappa_1$ ,  $\tau = \tau_1$ 
10:      else
11:        Set  $v = v_2$ ,  $\kappa = \kappa_2$ ,  $\tau = \tau_2$ 
12:      end if
13:    else if  $c(t) == \min([c(t), c(t - \Delta T), \dots, c(t - (N_T - 1)\Delta T)])$ 
14:      then
15:        if  $r_1 < r_2$  then
16:          Set  $v = v_2$ ,  $\kappa = \kappa_2$ ,  $\tau = \tau_2$ 
17:        else
18:          Set  $v = v_1$ ,  $\kappa = \kappa_1$ ,  $\tau = \tau_1$ 
19:        end if
20:      else
21:        Do nothing
22:    end if
23:  end while

```

article, we intend to keep the microrobot as simple as possible, so only two cases: $N_T = 2$ and $N_T = 4$ are considered.

C. A shortsighted maneuvering strategy

When the microrobot is swimming in helical trajectory, the centerline of the trajectory is

$$\mathbf{R}_c = \mathbf{r} + r_i \mathbf{n}, \quad (6)$$

where r_i is the radius:

$$r_i = \frac{\kappa_i}{\kappa_i^2 + \tau_i^2}. \quad (7)$$

Therefore, if the microrobot changes its curvature and torsion from κ_1 , τ_1 to κ_2 , τ_2 in an instant, then there is a displacement on the center of the helix:

$$\Delta \mathbf{R}_c = (r_2 - r_1) \mathbf{n}. \quad (8)$$

If $\Delta \mathbf{R}_c$ is along the gradient direction of chemoattractant, then the center of the helical trajectory is slightly shifted toward the gradient direction. In light of this, we can devise a simple strategy (Algorithm 1) that guides the microrobot to swim toward higher concentration. The basic principle is that if the concentration value at the current position is the largest one compared with all the historical records from the past period, then the normal vector is likely pointing toward the negative gradient direction, thus the microrobot should switch to the gait with smaller trajectory radius. If the current concentration value is the smallest one, then the normal vector is likely pointing toward the gradient direction. In this case, the microrobot should switch to the gait with larger trajectory radius. In this strategy the microrobot moves its trajectory center toward higher concentration whenever it sees an opportunity, that is why we call it a shortsighted strategy. We expect that DRL

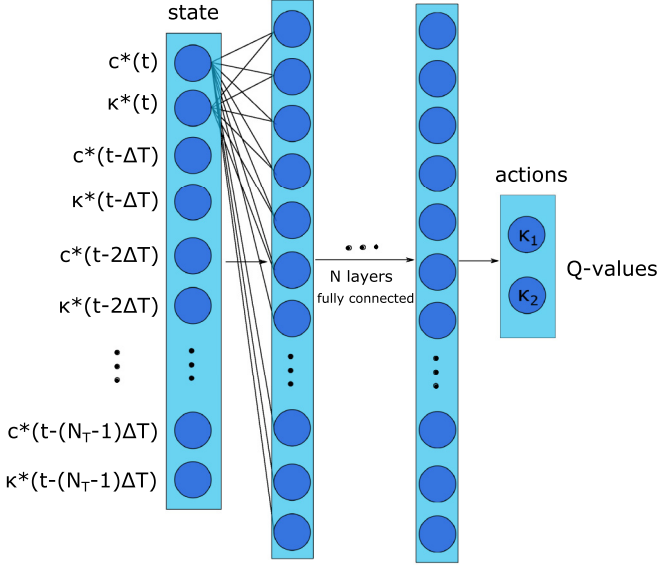


FIG. 4. Schematic of the deep Q network. The inputs nodes represent the states of the swimmer including current and historical states. The output nodes represent the legal actions, their values are their corresponding Q values.

could discover strategy that is more far-sighted and moves the microrobot faster toward the gradient direction.

D. Deep Q network as decision-making agent

In the context of reinforcement learning, the concentration information together with the microrobot's curvatures form the state s . A so-called agent can take in the state information and propose the next action a for the microrobot to maximise a reward function. The proposed action is then utilized to update the dynamics of the microrobot. As we employ an artificial neural network (deep Q network) to represent the agent, as shown in Fig. 4, the input vector consists of a temporal sequence of the concentrations and trajectory curvatures:

$$\begin{aligned} &c(t), \kappa(t), \\ &c(t - \Delta T), \kappa(t - \Delta T), \\ &\vdots \\ &c(t - (N_T - 1)\Delta T), \kappa(t - (N_T - 1)\Delta T). \end{aligned} \quad (9)$$

Therefore, a state s has $2N_T$ elements in total. Note that the torsion and velocity information are not necessary, since they have unique reflection to the curvature. The output nodes represent the legal actions, and their values are the Q values. The input vector is connected to the output neurons through several layers of hidden neurons. All the adjoining layers are fully connected. The tanh activation function is used for the input and hidden layers while the linear activation function is used for the output layer. A double deep Q -learning procedure [62] is applied to update the weights of the connections θ_i aiming to approximate the maximum action-value function $Q^*(s, a)$ via the deep Q network. The reward is represented as $R(s, a, s')$ when the microrobot takes the action a and the state transfers from s to s' . The network should predict a target

Q value:

$$Q_{\text{target}} = R(s, a, s') + \gamma Q^- [s', \text{argmax}_a Q(s', a; \theta_i), \theta_i^-], \quad (10)$$

where γ is the discount-rate parameter, and Q^- is a target network. At state s' , the training network Q is used to select which action has the largest Q value, but the actual Q value is evaluated using the target network Q^- . The target network is updated less frequently than the update of the training network. It is updated by simply copying the weights of the training network. The Q value predicted by the deep Q network is $Q_{\text{predicted}}(s, a, \theta_i)$. Therefore, the loss function can be defined as

$$L_i(\theta_i) = E_\pi [Q_{\text{target}} - Q_{\text{predicted}}(s, a; \theta_i)]^2, \quad (11)$$

which is to be minimized. The experience replay technique [63] is applied to randomize the experience data and alleviate the autocorrelation problem. Therefore, a replay buffer of finite size is created at the beginning of the learning. Every ΔT the agent's experience $e_t = (s, a, r, s')$ is stored into this buffer. During learning, a minibatch of experience is randomly drawn from the buffer to update the Q network using Eq. (11). The experience is replayed every time as the simulated swimmer reaches the end of its lifespan (t_{life}). The finish of a replay marks the end of an episode of the learning. Afterwards, a new swimmer with random initial position and direction is created to start a new episode.

During the simulation of each swimmer, the ϵ -greedy policy is used by the agent to select its actions, which balances the exploration and exploitation. An action is selected according to the probability:

$$\pi(a|s) = \begin{cases} \epsilon/2 + 1 - \epsilon, & \text{if } a^* = \text{argmax}_a Q(s, a), \\ \epsilon/2, & \text{otherwise,} \end{cases} \quad (12)$$

where $\pi(a|s)$ is the probability to select action a at state s . Note that the first row is the probability for the optimal action (predicted to be optimal by the Q network) to be selected, the second row is the probability of the other action to be selected. The value of ϵ starts with 1.0 and slowly anneals to ϵ_{min} during the learning.

The values of input vector are normalized before they are actually taken to the network:

$$c^*(t - i\Delta T) = \left[c(t - i\Delta T) - \frac{1}{N_T} \sum_{j=0}^{N_T-1} c(t - j\Delta T) \right] \frac{\bar{\kappa}}{|k_c|}, \quad (13)$$

$$i = 0, 1, \dots, N_T - 1,$$

where $\bar{\kappa}$ is the characteristic trajectory curvature $\bar{\kappa} = (\kappa_1 + \kappa_2)/2$ and k_c is the typical gradient of the concentration field. In a linear gradient field k_c is simply the constant gradient of the field. If the field has many different gradients, then we can choose the average of the absolute values of the gradients to be $|k_c|$. The N_T records of the curvature information are also normalized:

$$\kappa^*(t - i\Delta T) = 2 \frac{\kappa(t - i\Delta T) - \bar{\kappa}}{|\kappa_1 - \kappa_2|}, \quad i = 0, 1, \dots, N_T - 1. \quad (14)$$

We train the microrobot in a linear concentration field:

$$c = k_c y + c_0, \quad (15)$$

and test the learned strategy in the same linear concentration field.

E. Reward function

We still need a reward function to evaluate each action during the learning. Assuming that the inputted state of the agent is s at time t , the agent takes the action a so that the state transfers to s' at time $t + \Delta T$, we define the reward function as

$$\begin{aligned} R_1(s, a, s') &= \frac{1}{|k_c||r_1 - r_2|} \\ &\times \left[\frac{1}{N_T} \sum_{j=2}^{N_T-2} c(t - j\Delta T) - \frac{1}{N_T} \sum_{j=0}^{N_T-1} c(t - j\Delta T) \right] \\ &= \frac{c(t + \Delta T) - c[t - (N_T - 1)\Delta T]}{N_T |k_c||r_1 - r_2|}. \end{aligned} \quad (16)$$

Here, the agent determines the reward in an autonomous way: It utilizes only the current and historical concentration information, no external information is needed. Therefore, the same information the swimmer has gathered for decision-making is also used to infer the reward and evaluate its strategy.

For 3D helical swimming problem, we can also use the inner product of the helix vector \mathbf{h} and the gradient direction vector \mathbf{e}_y , to define another reward function:

$$R_2(s, a, s') = \mathbf{h} \cdot \mathbf{e}_y, \quad (17)$$

where \mathbf{h} can be calculated by

$$\mathbf{h} = \begin{cases} \sin \theta_0 \mathbf{t} + \cos \theta_0 \mathbf{b}, & \theta_0 \geq 0, \\ -\sin \theta_0 \mathbf{t} - \cos \theta_0 \mathbf{b}, & \theta_0 < 0, \end{cases} \quad (18)$$

with $\theta_0 = \tan^{-1}(\tau/\kappa)$ being the helix angle. In R_2 we have assumed that there is an omniscient observer who knows the accurate direction of the helix vector and the chemoattractant gradient and uses these information to evaluate the maneuvering strategy of the microrobot.

With R_1 the learning process is more like that a microrobot explores the environment and utilizes the information it gathers by itself to discover effective maneuvering strategy. We use R_1 to investigate whether the microrobot can learn chemotaxis autonomously, analogous to the way biological swimmers develop chemotaxis behavior through generations of evolution. With R_2 the learning process is more like that the microrobot explores and we use our knowledge (heuristics) to help the microrobot to evaluate and select better strategies. We use R_2 to investigate conditions for the DRL to discover efficient strategy and provide guidance for the design of microrobot.

F. Simulation parameters

We mainly consider four cases for the microrobot: (I) the microrobot swims near a wall, therefore the torsion is fixed at 0, and the motion is planar; (II) the microrobot swims in bulk fluid, and the torsion has different sign at the two

TABLE II. Basic parameters for the simulations.

Parameters	Values	Values in SI units
Curvature at gait 1 κ_1	5.5	$0.055 \mu\text{m}^{-1}$
Curvature at gait 2 κ_2	7.5	$0.075 \mu\text{m}^{-1}$
Velocity at gait 1 v_1	2.1	$210 \mu\text{m}\text{s}^{-1}$
Velocity at gait 2 v_2	1.9	$190 \mu\text{m}\text{s}^{-1}$
Case I: Planar motion		
Torsion at gait 1 τ_1	0	0
Torsion at gait 2 τ_2	0	0
Case II: Flipping torsion		
Torsion at gait 1 τ_1	1	$0.01 \mu\text{m}^{-1}$
Torsion at gait 2 τ_2	-1	$-0.01 \mu\text{m}^{-1}$
Case III: Negative torsion ($\tau_1/\kappa_1 \approx \tau_2/\kappa_2$)		
Torsion at gait 1 τ_1	-5.7	$-0.057 \mu\text{m}^{-1}$
Torsion at gait 2 τ_2	-7.7	$-0.077 \mu\text{m}^{-1}$
Case IV: Positive torsion		
Torsion at gait 1 τ_1	7.7	$0.077 \mu\text{m}^{-1}$
Torsion at gait 2 τ_2	5.7	$0.057 \mu\text{m}^{-1}$
Integration time step Δt	0.002	0.002 s
Physical integration time in learning t_{life}	80	80 s
Physical integration time in tests t_{life}	80	80 s
Concentration gradient parameter k_c	1	$100 \mu\text{m}^{-1}$
Concentration constant c_0	20	20

gaits; (III) the microrobot swims in bulk fluid, and torsion is nonzero but has the same negative sign at the two gaits; (IV) the microrobot swims in bulk fluid, and torsion is nonzero but has the same positive sign at the two gaits. We distinguish case II to IV using the sign of the torsion, but later we will show that the sign of the torsion is not essential; we will clarify the true relevant factors affecting the chemotactic motion. The simulation parameters are summarized in Table II, while the DRL training parameters are summarized in Table III. If not stated otherwise, then values in the two tables are adopted. Note that we focus on how a two-gait microrobot can implement chemotaxis, the accurate value of the curvature, torsion, and velocity are not essential. We have assumed that the microrobot has an average velocity of $200 \mu\text{m}^{-1}$, and an average curvature $0.065 \mu\text{m}^{-1}$ —the same as a typical sea urchin sperm cell [60]. Then we pose hypothetical parameter values for the curvature, torsion, and velocity at different gaits (Table II),

TABLE III. Basic parameters for the DRL trainings.

Parameters	Values
Learning rate α	0.01
Learning-rate decay	0.1
Discount-rate parameter γ	0.9 for R_1 , 0.1 for R_2
Minimum ϵ -greedy parameter ϵ_{min}	0.1
Number of hidden dense layers N_{hidden}	4
Number of nodes in each hidden layer N_n	32
Episodes	1600
ϵ decaying rate	0.998
Batch size	128
Update frequency of the target network	25

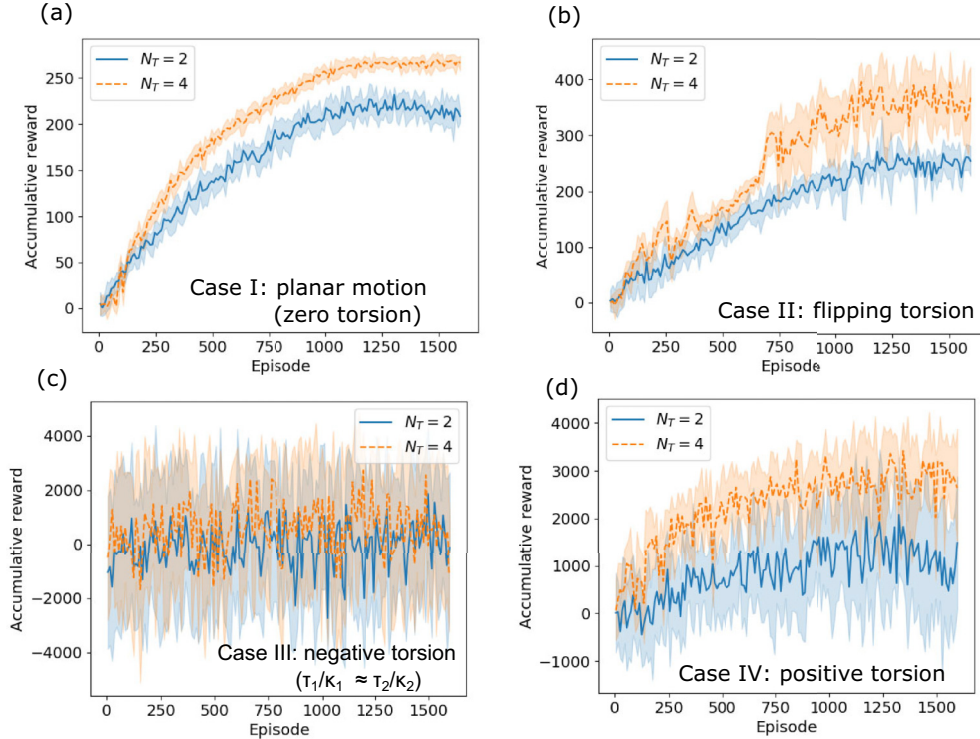


FIG. 5. The accumulative rewards during the deep reinforcement learning processes. (a) Case I: planar motion (zero torsion); (b) Case II: flipping torsion; (c) Case III: negative torsion; (d) Case IV: positive torsion. Reward function: R_1 .

following the rule we found in our SDPD simulation: Higher curvature corresponds lower torsion and lower velocity. In simulation, the length scale is rescaled using $100 \mu\text{m}$ as a length unit. All simulations and trainings run with Python 3.6.13 and Tensorflow 2.6.0 on the Windows 11 desktop installed with Intel Core i7-1165G7 CPU and NVIDIA GeForce RTX 3080 GPU.

IV. RESULTS WITH REWARD FUNCTION R_1

A. Planar motion results

When the microrobot swims near a wall the attraction of the wall constrain the swimming trajectory to be circular and parallel to the wall. This corresponds to case I in Table II with $\tau = 0$. In this case, the deep reinforcement learning processes are shown in Fig. 5(a) for both $N_T = 2$ and $N_T = 4$.

As already explained above, N_T denotes how many records of perception are used as input information for the decision-making machinery. We set $N_T \in \{2, 4\}$ and perform training for 1600 episodes. During the learning processes the accumulative rewards of the microrobot increases with the episode and reaches a stable value at the end of the learning. The stable reward is larger in the case of $N_T = 4$ than in the case of $N_T = 2$, since the microrobot can perform maneuver in a more sophisticated way at $N_T = 4$.

We first examine the results of $N_T = 2$. In Fig. 6, we show how a microrobot trained with DRL switches between the two gaits and swims toward higher concentration of chemoattractant. For comparison, we also show the results using the periodically alternating curvature pattern and the shortsighted strategy. In the alternating pattern, the curvature changes every

$N_T \Delta T/2$. As shown in Fig. 6(a), most of the time, both the DRL and the shortsighted strategy follow closely the alternating pattern. Figure 6(b) shows that the microrobot swims in drifted circles and the centerline of the trajectory is an arc when the alternating pattern is adopted. However, the microrobot would swim back to its initial position if the pattern were not broken at some point. Both the shortsighted strategy and DRL break the alternating pattern periodically, but the timing is different. The shortsighted strategy breaks the pattern when the centerline of the trajectory starts to go toward lower concentration. While the DRL breaks the pattern earlier and more frequently. Moreover, Fig. 7(a) shows the centerlines of many test microrobots with random initial position and direction, while Fig. 7(b) shows their final gains. The overall swimming directions of the microrobots guided by the DRL and the shortsighted strategy are not perfectly in line with the gradient direction of the chemoattractant. The DRL achieves visually worse alignment of trajectory with the gradient direction, as shown in Fig. 7(a), but the centerline is straighter. It sacrifices the angle alignment to achieve straighter centerline. The average final gains of the DRL is also always higher than that of the shortsighted strategy. This indicates that the DRL algorithm has discovered a better timing to break the alternating curvature pattern.

Second, we examine the case of $N_T = 4$. Again, we compare the results among the alternating curvature pattern, the shortsighted strategy and the DRL. In the alternating pattern, the curvature changes every $N_T \Delta T/2$. As shown in Fig. 8(a), the DRL and the shortsighted strategy still follow the alternating pattern most of the time, but only break the pattern occasionally. The alternating pattern still leads the microrobot

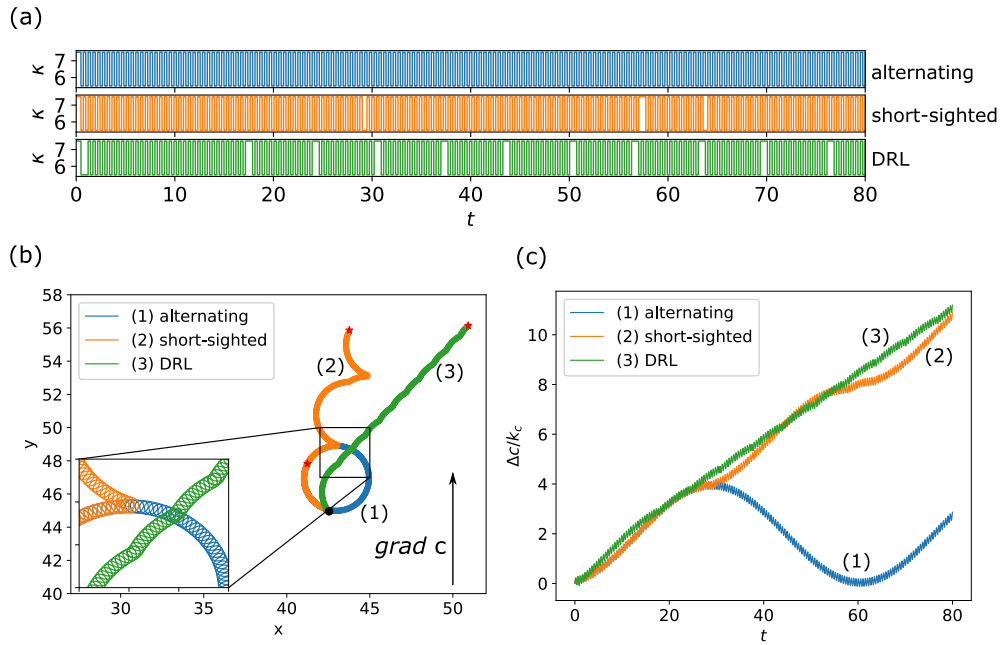


FIG. 6. (a) Evolution of κ ; (b) swimming trajectories, the black dot marks the starting point, the red stars mark the end points; (c) gains of chemoattractant $[\Delta c = c(t) - c(0)]$. $N_T = 2$.

back to its original position as shown in Fig. 8(b), whereas DRL and the shortsighted strategy are able to guide the microrobot to swim toward higher concentration steadily. In this case, The DRL tends to break the alternating pattern less frequently than the shortsighted strategy, thus the centerline is more curvy for the DRL trajectories [Fig. 8(b)]. But at this time, the DRL has sacrificed the straightness of the centerline to gain better angle alignment to the gradient direction. The overall swimming speed towards the gradient direction is still higher for the DRL. Therefore, the DRL has a better performance for the gains of chemoattractant than the shortsighted strategy as shown in Fig. 8(c). Furthermore, Fig. 9 shows the centerlines of many test microrobots and their final gains. It is apparent that microrobots guided by the DRL always have better angle alignment and slightly higher final gains than that guided by the shortsighted strategy.

To better understand the two sets of results above, let us further analyze how each strategy takes effect. We first note that the alternating pattern always leads to an arc or a circle for the centerline. The origin of this phenomenon is as follows. The curvature of the trajectory and the swimming velocity are alternating every $N_T \Delta T/2$, during which the tangential of the trajectory does not turn exactly 360° after one period. The angle deviation is constant every period. And every time the curvature alternates, the center of the curvature deviates in the direction of the normal of the swimmer's trajectory. Over long time of many alternating gaits, the centers of the curvatures form its own trajectory, which is called centerline here. The centerline of the trajectory turns out to be an arc, or a full circle if given sufficient time. In light of this, a microrobot may use the following strategy to gain net migration toward the gradient direction: following the alternating

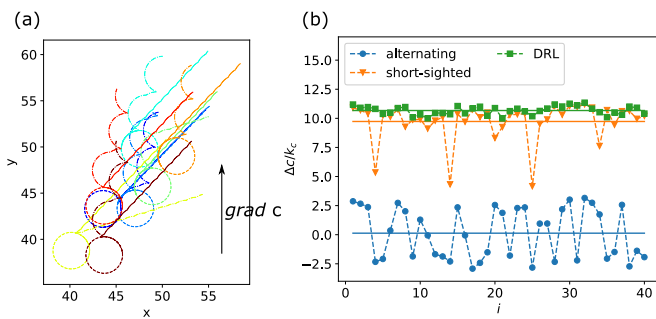


FIG. 7. (a) Centerlines of some test swimming microrobots: solid lines for the DRL; dash-dot lines for the shortsighted strategy; dashed lines for the alternating pattern. (b) The final gains $[\Delta c = c(t_{\text{life}}) - c(0)]$ of 40 microrobots. The solid lines mark the average values, respectively. $N_T = 2$.

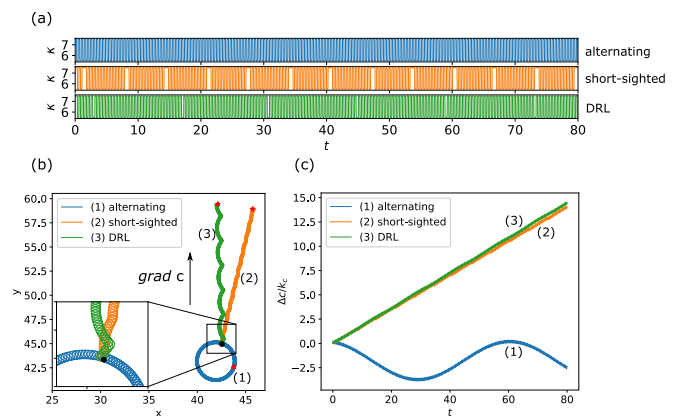


FIG. 8. (a) Evolution of κ ; (b) swimming trajectories, the red dot marks the starting point, the black dots mark the end points; (c) gains of chemoattractant $[\Delta c = c(t) - c(0)]$. $N_T = 4$.

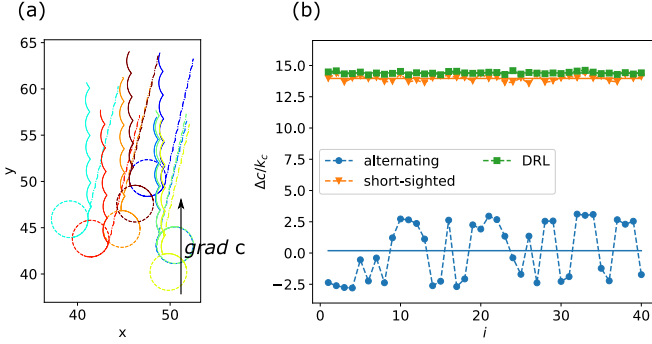


FIG. 9. (a) Centerlines of some test swimming microrobots: solid lines for the DRL; dash-dot lines for the shortsighted strategy; dashed lines for the alternating pattern. (b) The final gains [$\Delta c = c(t_{\text{life}}) - c(0)$] of 40 microrobots. The solid lines mark the average values, respectively. $N_T = 4$.

pattern most of the time but breaking the pattern occasionally. Therefore, the centerline of the trajectory would be prevented from becoming a full circle, but becoming a curvy line formed by many broken arcs bending towards the gradient direction upwards. This is exactly the strategy the DRL has discovered with $N_T = 2$ and $N_T = 4$. We observe that the centerlines in Figs. 7(a) and 9(a) are formed by connecting arcs.

B. 3D motion results

If the microrobot is swimming in bulk fluid, then its trajectory is helical as shown in Fig. 2. In the 2D case studied above, when no action is taken the microrobot will just swim in circular trajectory. But in 3D, when no action is taken the microrobot will still migrate toward the direction of the helix vector. The target of the 3D problem is to control the microrobot to bend its helical trajectory toward the gradient direction (align the helix vector to the gradient direction). We study three 3D cases with different torsion parameters as given in Table II. The deep reinforcement learning processes of the three cases are shown in Figs. 5(b)–5(d). As can be seen from the figure, in cases II and IV the accumulative rewards are increasing with the learning episode, indicating that the agent has learned effective strategies to control the microrobot to swim toward higher chemical concentration. But in case III nothing is learned, the accumulative reward shows no sign of increase and fluctuates dramatically. The difference on the learning results in different cases is related to the ability of the microrobot to change its helix vector. When the microrobot switch its gait the helix vector also changes, that is how the microrobot can bend its trajectory. But at different κ and τ parameter regime, the ability of the microrobot to change the helix vector is different. We can use the steering action step size ΔA to characterize this ability:

$$\Delta A = \cos^{-1} [\mathbf{h}(\tau_1/\kappa_1) \cdot \mathbf{h}(\tau_2/\kappa_2)]. \quad (19)$$

Figure 10 presents the contour of the value of ΔA at different τ_1/κ_1 and τ_2/κ_2 . Brighter colors indicate better steering capability for the microrobot. When τ_1 and τ_2 has different sign, the microrobot always has better steering capability, since in this case the direction of the helix vector can be nearly reversed by switching gait. Case II is exactly this case, with

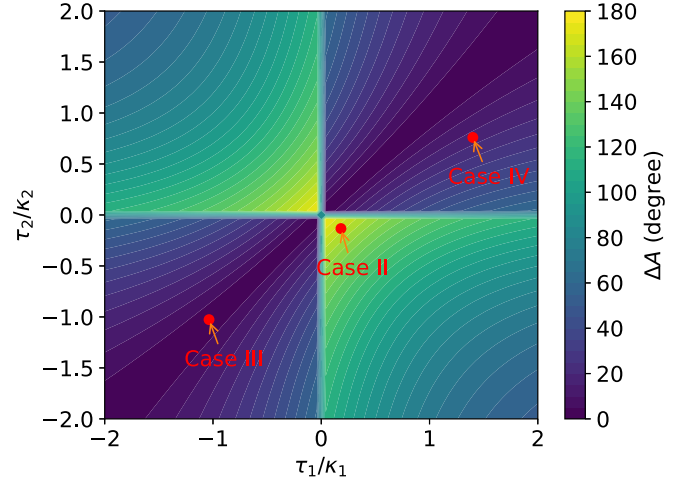


FIG. 10. Steering action step size of the microrobot at different τ_1/κ_1 and τ_2/κ_2 .

$\Delta A_{\text{II}} = 162.1^\circ$ it can reverse its net migration direction (the helix direction) by one simple gait switching. That is why in Fig. 5 the fluctuation of reward is the lowest in case II compared to cases III and IV. In contrast, on the line $\tau_2/\kappa_2 = \tau_1/\kappa_1$ the helix vector cannot be changed by gait switching, the microrobot has zero steering capability. In case III we have $\tau_2/\kappa_2 \approx \tau_1/\kappa_1$ and $\Delta A_{\text{III}} = 0.27^\circ$, so nothing is learned in this case [Fig. 5(c)]. In case IV, $\Delta A_{\text{IV}} = 17.2^\circ$, the microrobot cannot reverse the helix direction with one gait switching but it still has some steering capability. In principle, it can use many sequential gait switchings to bend the helical trajectory toward the gradient direction. Therefore, even though the fluctuation is high, effective strategy is still learned in Fig. 5(d).

We note that case IV has better final reward compared with case II due to the larger helix pitch [pitch $2\pi h_0 = 2\pi\tau/(\kappa^2 + \tau^2)$]. The steering capability is just one aspect to determine the final reward. The other aspect is the helix pitch. In case II, the steering capability is better, but the pitch is very small as a result of the small torsion, hence the microrobot cannot migrate fast. In case IV, even though the steering capability is worse, the pitch is very large. Once the microrobot successfully aligns its helix vector with the gradient direction it can migrate very fast toward that direction thus achieving very high reward.

We further examine the learned strategies in cases II and IV. In each case, we perform 80 tests (each 40 tests for $N_T = 2$ and $N_T = 4$) with random initial position and direction and compare the results with the shortsighted strategy (Algorithm 1) results and the alternating pattern results. The comparison of the final gains [$\Delta c = c(t_{\text{life}}) - c(0)$] is first shown in Fig. 11. We see that for case II, the strategies discovered by DRL has better average performance than the shortsighted strategy. In case IV, the DRL strategy is also better but the final gain fluctuates significantly. In many tests, the final gain is negative indicating that the microrobot is migrating against the gradient direction, the strategies fails to guide the microrobot to swim upward the chemoattractant concentration. Therefore, our tests have shown that, the DRL can discover strategies that are not worse than the shortsighted one and can guide the microrobots to enhance chemotaxis

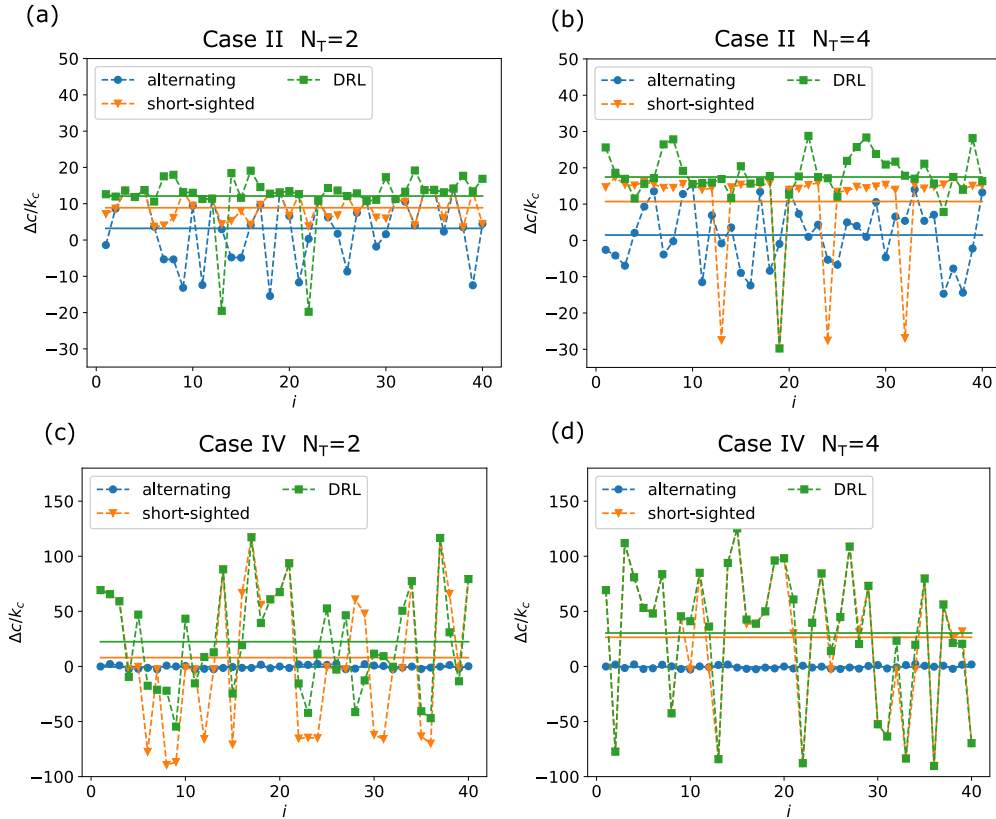


FIG. 11. Tests of the learned strategies. $\Delta c = c(t_{\text{life}}) - c(0)$. (a) Case II $N_T = 2$; (b) Case II $N_T = 4$; (c) Case IV $N_T = 2$; (d) Case IV $N_T = 4$. The solid lines mark the mean values.

statistically. But the enhancement is actually quite weak, and does not show significant improvement compared with the simple shortsighted strategy. As a matter of fact, in case IV at $N_T = 4$ the DRL has discovered a strategy that is almost the same as the shortsighted strategy, so most of the DRL data points in Fig. 11(d) overlap with the shortsighted data points. Both the shortsighted and the DRL strategy has high probability to fail guiding the microrobot to steer toward the gradient direction.

We examine the trajectories from the tests intensively. Figure 12 shows a typical test from case II. The evolutions of κ (a), the swimming trajectories (b), the gains of chemoattractant (c), and the angle between the helix vector and the gradient direction θ_h (d) are all presented. For clarity, only the first 10 s result is shown. For the DRL strategy, gait switching only happens at the very beginning. After that no further action is taken, the microrobot just swims in perfect helix and swims in a direction different from the gradient direction. But the angle difference θ_h is $< 90^\circ$, thus the gain is increasing anyway. The alternating pattern and the shortsighted strategy guide the microrobot to swim in a complex curvilinear trajectory. The trajectory can still produce net migration even though in a slower way than the helical trajectory. Figure 13 shows a typical test from case IV. For the DRL strategy, gait switching only happens at the beginning, the microrobot adjusts the helix vector to make a crude alignment with the gradient direction and stay unchanged for the remaining time. The DRL strategy actually is the same as the shortsighted strategy, therefore the DRL trajectory and the shortsighted

trajectory overlap with each other. For the alternating pattern, the centerline of the trajectory is drawing another helix with very small pitch, it fails to produce discernible net migration.

Looking at Figs. 12(d) and 13(d) we can understand how the DRL strategy works. The alternating pattern causes the θ_h to fluctuates periodically in a quite wide range, therefore the DRL first follows the alternating pattern and at the point with a small θ_h it breaks the pattern and stops the gait switching, a better alignment between the helix vector and the gradient direction is thus obtained. However, the efficiency of this strategy depends on the fluctuation range of θ_h . And this range in turn depends on the initial condition. If when θ_h fluctuates the smallest point is always larger than 90° , then the DRL will probably fail to guide the microrobot swim upward the chemoattractant concentration.

Moreover, the strategy discovered by DRL suffers from the problem of local optima. It adjusts the helix vector by gait switching only at the beginning, once it finds a relatively small θ_h it stops the attempt to make improvement, even though a smaller θ_h can still be obtained if further exploration is conducted. To overcome the local optima, we can include some stochasticity to the swimming process.

C. Improvement utilizing stochasticity

We consider three different kinds of stochasticity: (1) randomness in the decision; (2) noise at the sensing of the chemoattractant concentration; (3) fluctuations of the curvature and torsion.

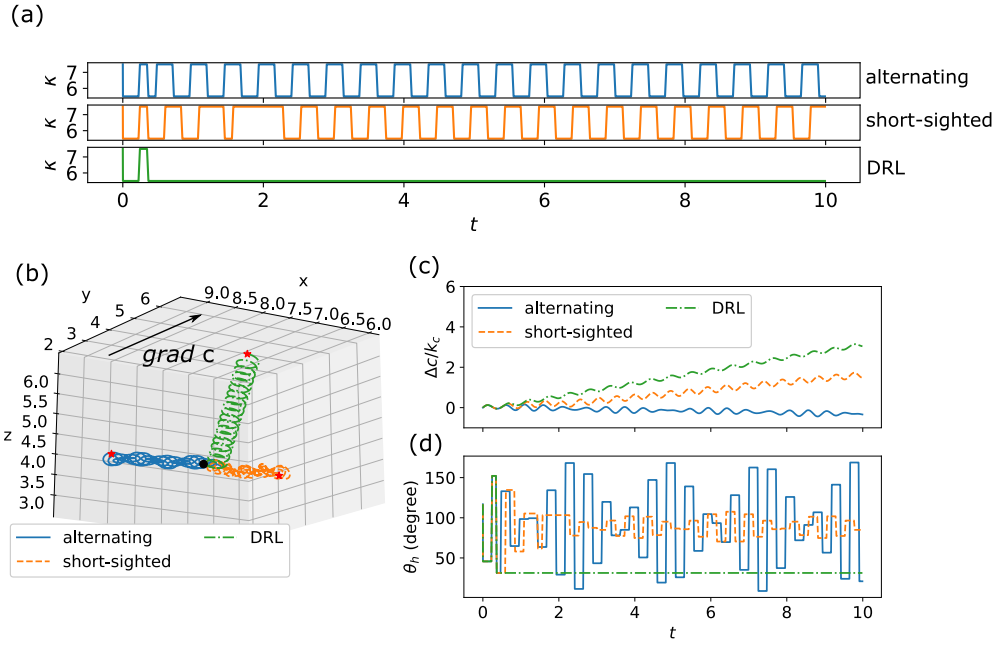


FIG. 12. (a) Evolutions of κ ; (b) swimming trajectories, the black dot marks the starting point, the red stars mark the end points; (c) gains of chemoattractant [$\Delta c = c(t) - c(0)$]; (d) the angle between the helix vector and the gradient direction θ_h . $N_T = 4$. For clarity only the initial 10 s swimming result is shown.

To include randomness to the decision, we use the ϵ -greedy method: with the probability ϵ the decision is made randomly, with the other $1 - \epsilon$ probability the decision is made following the deterministic strategy (DRL or shortsighted). To include noise at the sensing of chemoattractant concentration, we assume that every time the microrobot senses the local concentration it records a value $c(t) = c_t(t) + \delta c$, where c_t is the true value and δc is a Gaussian distributed random number

with average 0 and standard deviation ξ [51]. To include noise to the control of curvature and torsion, we assume κ and τ are fluctuating: $\kappa(t) = \kappa_t(t) + \kappa_\sigma(t)$, $\tau(t) = \tau_t(t) + \tau_\sigma(t)$, with $\kappa_t \in \{\kappa_1, \kappa_2\}$, $\tau_t \in \{\tau_1, \tau_2\}$, $\kappa_\sigma(t)$, and $\tau_\sigma(t)$ obeys Gaussian probability distribution: $\langle \kappa_\sigma(t) \kappa_\sigma(t') \rangle = \delta(t - t') \sigma_\kappa^2$, $\langle \tau_\sigma(t) \tau_\sigma(t') \rangle = \delta(t - t') \sigma_\tau^2$. We set $\sigma^* = 2\sigma_\kappa / (\kappa_1 + \kappa_2) = 2\sigma_\tau / (\tau_1 + \tau_2)$.

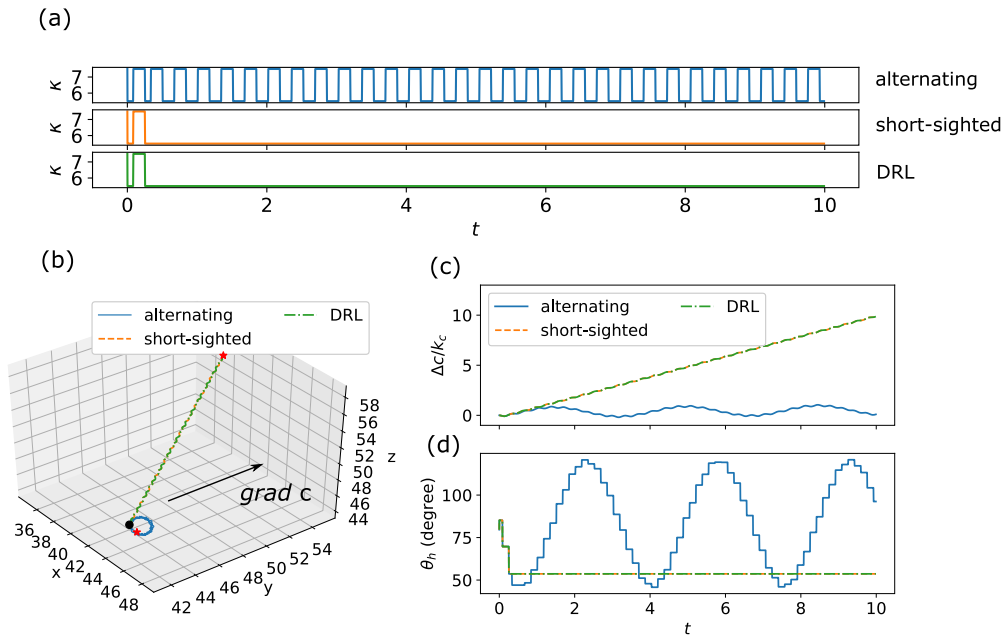


FIG. 13. (a) Evolutions of κ ; (b) swimming trajectories, the black dot marks the starting point, the red stars mark the end points; (c) gains of chemoattractant [$\Delta c = c(t) - c(0)$]; (d) the angle between the helix vector and the gradient direction θ_h . $N_T = 4$. For clarity only the initial 10 s swimming result is shown.

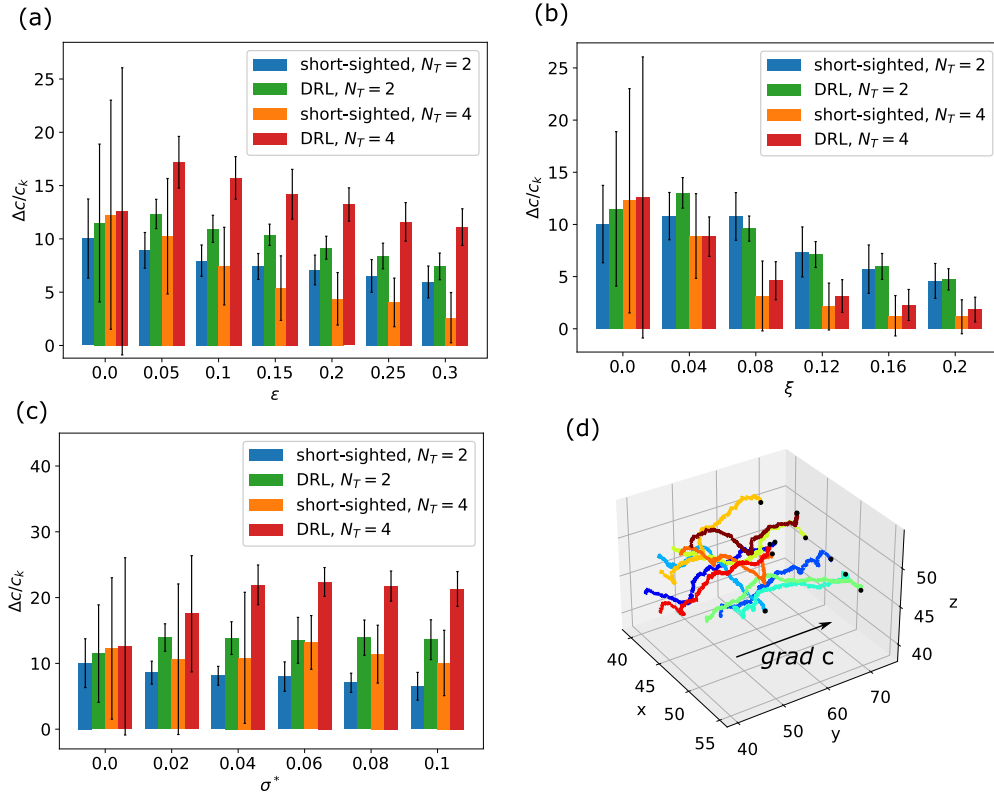


FIG. 14. The effects of stochasticity on the DRL and shortsighted strategy in case II. Each bar represents the average and standard error of 40 tests with the same simulation parameters but different initial conditions. $\Delta c = c(t_{\text{life}}) - c(0)$. (a) Considering randomness in the decision making process. (b) Considering noise at the sensing of chemoattractant concentration. (c) Considering fluctuation of curvature and torsion. (d) Illustrative swimming trajectories in the presence of stochasticity ($N_T = 4$, $\sigma^* = 0.06$). The black dots mark the endpoints.

These three kinds of stochasticity are considered separately. The parameters ϵ , ξ , and σ^* are varied to examine the effects of stochasticity. We focus mainly on cases II and IV, since for case I, the strategy discovered by DRL is very efficient, inclusion of stochasticity will only impede its performance. And as explained above, in case III the microrobot cannot be steered. We show the results for case II in Fig. 14. In most of the simulations, inclusion of some stochasticity can increase the average final gain of the microrobot, and decrease the standard error significantly. The DRL strategy is almost always better than the shortsighted one. If the stochasticity is included through randomness in decision (a) or through curvature and torsion fluctuation (c), then $N_T = 4$ is always better than $N_T = 2$. The best performance is achieved with DRL at $N_T = 4$. But if the stochasticity is included through noise at the sensing of chemoattractant concentration (b), the best performance is achieved with DRL at $N_T = 2$. Figure 14(d) shows some illustrative swimming trajectories guided by DRL with $\sigma^* = 0.06$ and $N_T = 4$. It can be seen that, with the help of stochasticity, all the test microrobots with random initial position and direction succeed in changing their net migration direction to roughly align with the gradient direction. Figure 15 shows the results of case IV. As explained above, in case IV, the steering capability of the microrobot is low, the microrobot has to use many sequential gait switchings to adjust its helix vector. When there is no stochasticity, it is very often that the microrobot fails to adjust itself to swim upward the chemoattractant concentration. This

leads to a very large standard error on the final gains. But with the inclusion of stochasticity the average final gain can increase significantly [Figs. 15(a)–15(c)]. The standard error also decrease to be very small, especially at $N_T = 4$. The simulations with $N_T = 4$ also have better final gains than that with $N_T = 2$. For the shortsighted strategy at $N_T = 2$, even with the help of stochasticity it still has some probability to fail to guide the microrobot to swim upward the chemoattractant concentration. But with $N_T = 4$ both the DRL strategy and the shortsighted strategy can achieve very good performance. When the stochasticity is included through randomness in the decision (a), the best performance is achieved by the shortsighted strategy. When the stochasticity is included through noise at the sensing of chemoattractant concentration or the fluctuations of curvature and torsion, the best performance is achieved by the DRL strategy. Figure 15(d) shows some illustrative swimming trajectories guided by DRL with $\xi = 0.08$ and $N_T = 4$, we can clearly see that all the test microrobots with random different initial and direction succeed in swimming upward the chemoattractant concentration.

The reason why noise is beneficial can be explained as follows. During the the learning process, all the environmental information the agent has is the several chemoattractant concentration on the swimming trajectory. This information is not enough to characterize the environment. The agent cannot learn to accurately infer its true state in the environment with the limited information and make appropriate actions.

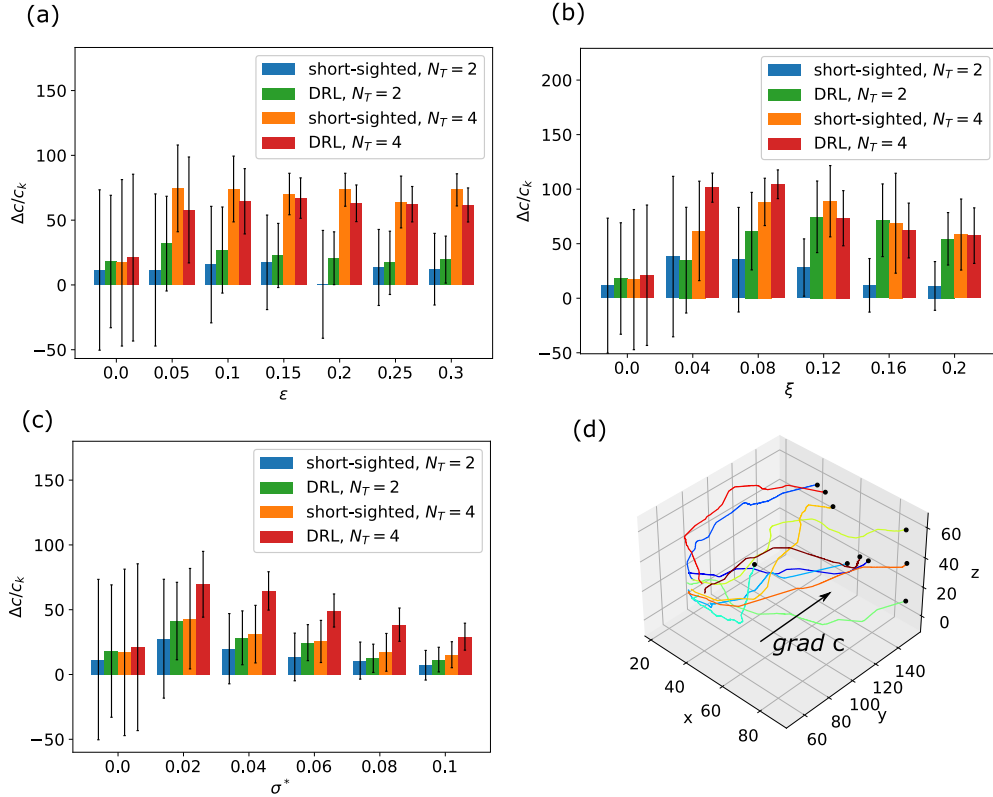


FIG. 15. The effects of stochasticity on the DRL and shortsighted strategy in case IV. Each bar represents the average and standard error of 40 tests with the same simulation parameters but different initial conditions. $\Delta c = c(t_{\text{life}}) - c(0)$. (a) Considering randomness in the decision making process. (b) Considering noise at the sensing of chemoattractant concentration. (c) Considering fluctuation of curvature and torsion. (d) Illustrative swimming trajectories guided by DRL in the presence of stochasticity ($N_T = 4$, $\xi^* = 0.08$). The black dots mark the endpoints.

In such case, a stochastic policy will be better than a deterministic policy. By adding noise to the decision, sensing or mobility, the decision-making process essentially becomes stochastic, hence the performance is improved. Recent study on the run-and-tumble process also suggests that weak noise can be beneficial to the chemotactic motion [64]. In the next section we will use reward function R_2 , which includes the directional information between the helix vector and the chemoattractant gradient. We will show that by including this information the agent can learn to accurately infer its state in

the environment, hence very efficient deterministic policy can be discovered.

V. RESULTS WITH REWARD FUNCTION R_2

We change the reward function to R_2 to investigate whether DRL can also discover efficient strategy if additional heuristic directional information is supplemented to evaluate the strategy in learning process. The learning processes using reward function R_2 are summarized in Fig. 16. Since R_2 is only applicable

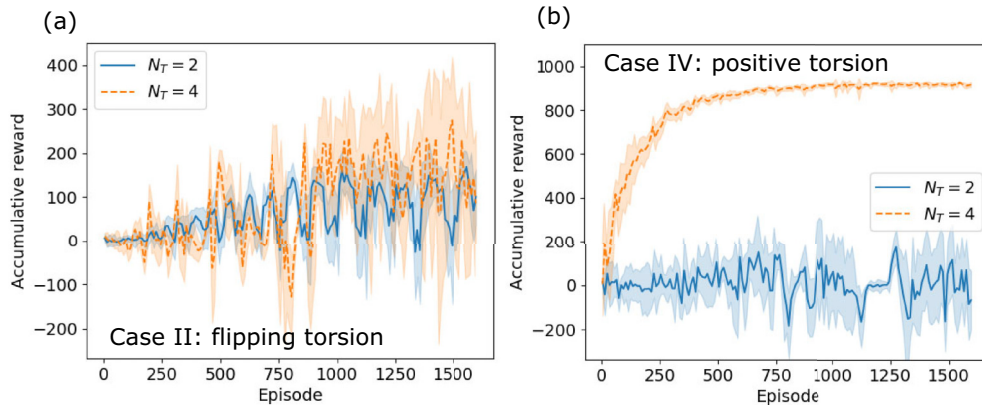


FIG. 16. The accumulative rewards during the deep reinforcement learning processes. (a) Case II: flipping torsion; (b) Case IV: positive torsion. Reward function: R_2 .

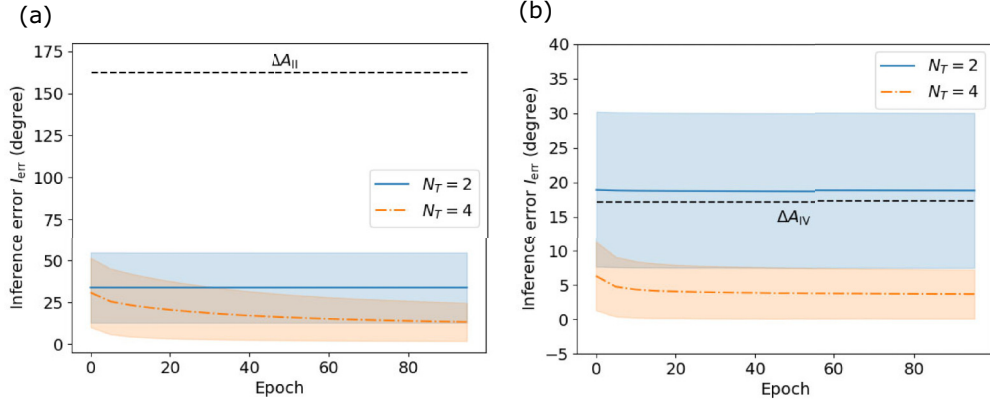


FIG. 17. The inference angle error I_{err} of a neural network: (a) in case II; (b) in case IV.

to 3D cases, and in case III the swimmer has very low steering capability, only cases II and IV are examined. In case II, the accumulative reward increases slightly with the learning episode at both $N_T = 2$ and $N_T = 4$, but the fluctuation is very large. We can conclude that the learning fails. In case IV, only at $N_T = 4$ is the accumulative reward increasing steadily. At $N_T = 2$, nothing is learned. Judging from the high accumulative reward and very small standard error, the DRL has obtained a very efficient strategy at $N_T = 4$. The difference of the learning result in different cases presented in Fig. 16 is not just related to the steering ability as in the R_1 case, but also to the ability of the microrobot to infer the relation between the current helix direction and the gradient direction using the neural network. With R_2 as reward function, the agent needs to first infer the angle between the helix direction and the gradient direction θ_h and take appropriate action that can improve the alignment (decrease θ_h). The inference is conducted through the neural network, but the neural network may not be able to infer the accurate direction relation using the limited state information. We perform additional supervised learnings (regression) to evaluate the ability of a neural network to infer θ_h using the state information [Eq. (9)]. The network structure is the same as Fig. 4 except that we change the output node to output normalized θ_h (with tanh activation function). Data gathered from microrobots swimming with completely

random action strategy is used to train the network. We show the test error between the predicted θ_h by the network and the true θ_h in Fig. 17. In case II (a), at $N_T = 2$, the error is about 33.8° , at $N_T = 4$ the error is about 13.3° . In case IV (b), at $N_T = 2$, the error is about 18.9° , at $N_T = 4$ the error decreases to about 3.5° . A necessary condition for an effective strategy to be learned is that the inference angle error I_{err} is smaller than the steering angle step size ΔA [Eq. (19)]. The steering angle step size in case II and IV are plotted as dashed lines in Figs. 17(a) and 17(b), respectively. In case II, the steering angle step size is very large, the swimmer cannot perform sophisticated steering action, even though I_{err} is smaller than ΔA_{II} , the agent fails to learn effective strategy to improve the vector alignment. In case IV, ΔA_{IV} is smaller than I_{err} at $N_T = 2$, we can say that the observation error is larger than the change caused by an action, the agent cannot learn an effective strategy to guide the microrobot to improve the vector alignment. But at $N_T = 4$, ΔA_{IV} is larger than I_{err} , an effective strategy can be learned. Moreover, since both ΔA_{IV} and the I_{err} are relatively small, the microrobot can conduct precise action to make improvement. Therefore, we have very high reward and small standard error at $N_T = 4$ in case IV [Fig. 16(b)].

We test the learned strategy at $N_T = 4$ in case IV. We perform 40 tests with random initial position and direction and

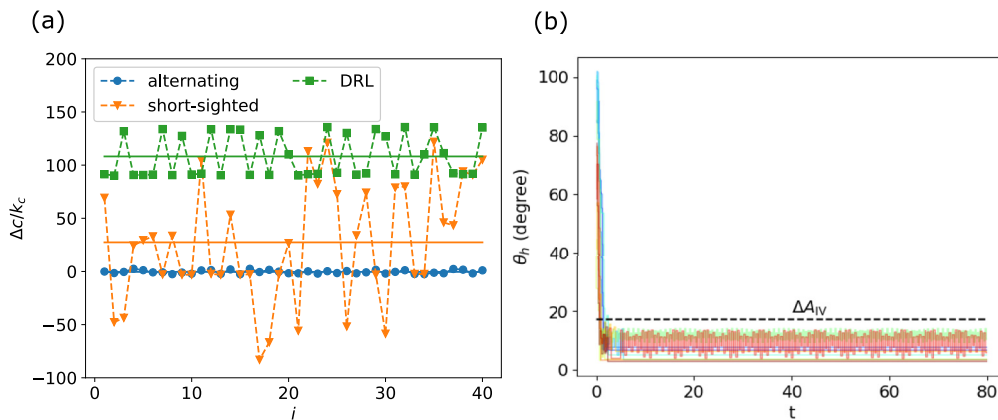


FIG. 18. Tests of the learned strategy at $N_T = 4$ in case IV. (a) Final gains of the chemoattractant [$\Delta c = c(t_{\text{life}}) - c(0)$]. The solid lines mark the mean values. (b) Evolution of θ_h of 10 test microrobots guided by DRL strategy.

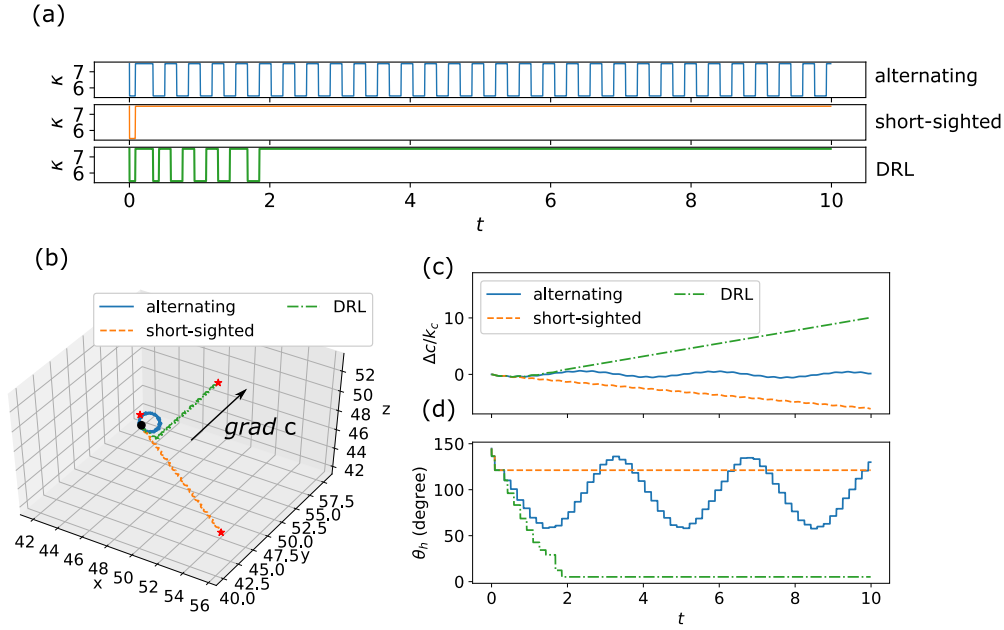


FIG. 19. (a) Evolutions of κ ; (b) swimming trajectories, the black dot marks the starting point, the red stars mark the end points; (c) gains of chemoattractant [$\Delta c = c(t) - c(0)$]; (d) the angle between the helix vector and the gradient direction θ_h . $N_T = 4$. For clarity only the initial 10 s swimming result is shown.

compare the DRL strategy results with the shortsighted strategy results and the alternating pattern results. The comparison are shown in Fig. 18(a). It can be seen that, the DRL strategy is much better than the other strategies, it produces very high average final gains, all the test microrobots succeed in swimming upward the chemoattractant concentration. Figure 18(b) shows the evolution of θ_h for 10 test microrobots guided by DRL. In all cases, θ_h decreases dramatically to below the steering angle step size ΔA_{IV} , suggesting that the microrobots achieve the best alignment between the helix vector and the gradient direction very fast. Figure 19 shows the gait switchings (a), the swimming trajectories (b), the evolutions of the gain (c), and the evolutions of θ_h from a test microrobot. When the microrobot is guided by the DRL strategy it conducts gait switching only at the beginning, in 2 s it successfully adjusts its helix vector to be almost the same as the gradient direction

and migrates toward this direction very fast. Therefore, a very efficient strategy has been found by DRL.

To clarify the requirements for a swimmer that an effective strategy can be learned, we perform more simulations with $\tau_0 = (\tau_1 + \tau_2)/2$ varying from -6.7 to 6.7 , and $\tau_1 - \tau_2$ fixed at 2. The learning results are shown in Fig. 20. If $N_T = 2$, then all the learnings fail. If $N_T = 4$, then only the case $\tau_0 = -6.7$ completely fails. But when $\tau_0 = \pm 0.7$ the accumulative rewards only slightly increase with large relative fluctuations. These results can be explained using Fig. 21. In this figure both the inference angle error I_{err} of the neural network, and the steering angle step size ΔA are presented at different τ_0 . Note that ΔA suddenly increases to very large with τ_0 near zero, since the torsion can change sign here. This very large ΔA means that the swimmer cannot make sophisticated steering action to enhance the alignment between helix vector

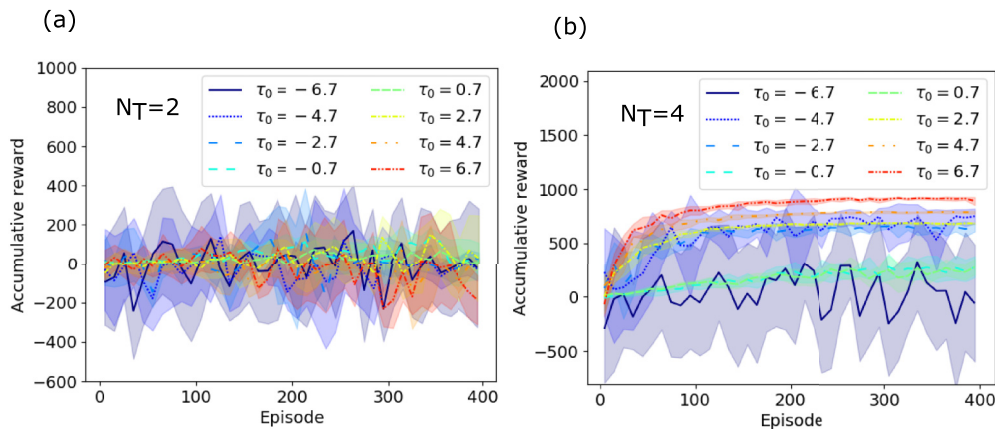


FIG. 20. The accumulative rewards during the deep reinforcement learning processes. (a) $N_T = 2$. (b) $N_T = 4$. The torsion parameter $\tau_0 = (\tau_1 + \tau_2)/2$, $\tau_1 - \tau_2 = 2$, the total learning episodes is 400, ϵ decaying rate is 0.992, update frequency of the target network is 15.

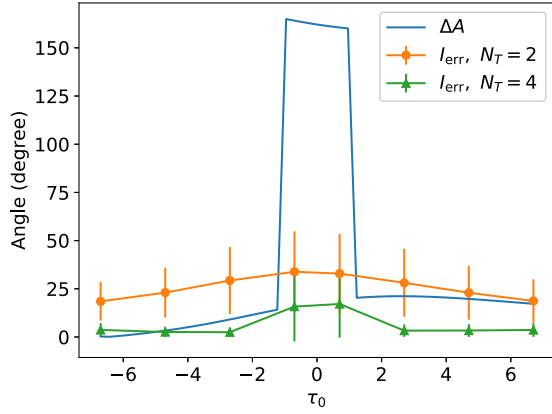


FIG. 21. Relationship between the inference angle error I_{err} and the steering angle step size ΔA at different $\tau_0 = (\tau_1 + \tau_2)/2$.

and the gradient direction. Besides, when τ_0 is near zero the helix pitch is small, the migration ability of the swimmer is low. Therefore, the cases $\tau_0 = \pm 0.7$ all perform poorly in Fig. 20. For all other cases at $N_T = 2$ we have $\Delta A < I_{\text{err}}$, this means the inference angle error of the neural network is larger than the steering angle step size. The observation error of the agent is larger than the change caused by an action, the agent cannot learn an effective strategy to guide the microrobot to improve the vector alignment. Therefore, all cases at $N_T = 2$ fails. And at $N_T = 4$, we have $\Delta A > I_{\text{err}}$ for $\tau_0 = -4.7, -2.7, 2.7, 4.7, 6.7$, all the corresponding cases shown in Fig. 20 succeed.

Based on these observation, we have summarized the following requirements for a swimmer to be able to learn efficient strategy: (1) The number of historical information provided for the microrobot should be enough for the microrobot to accurately infer the angle θ_h between the helix vector and the gradient direction (this requires $N_T \geq 4$ in our simulation); (2) The steering angle step size ΔA should be larger than the inference error of θ_h . Otherwise, the microrobot do not have enough observation accuracy to evaluate the actions. (3) The steering angle step size ΔA should also be moderate and appropriate. If ΔA is too large, then the microrobot cannot perform precise steering. If ΔA is too small, then the microrobot has very low steering capability hence cannot steer efficiently.

VI. SUMMARY AND CONCLUSIONS

A primary research on a three beads swimmer has already demonstrated that its chemotactic motion in 1D space can be learned by the swimmer through a DRL algorithm [51]. In this work, a specific but more realistic microswimmer in 2D or 3D space is considered. We design an elastic flagellated microrobot that swims in helical trajectory similar to a sea urchin sperm cell. This microrobot can utilize the coupling between flagellum elasticity and resistive force to change the curvature and torsion of the swimming trajectory. The qualitative relation between the beating frequency and the trajectory curvature and torsion is first investigated using SDPD simulations. Then we envisage a microrobot that can switch between two gaits with different curvature, torsion and velocity. We

investigate the chemotactic motion of this microrobot in four cases: planar motion (zero torsion), flipping torsion, negative torsion and positive torsion.

In reality, the chemotactic motion of a sperm cell is implemented through a very complex signaling network inside the cell. In this work we replace this biological signaling network with an artificial neural network to acts as a decision-making agent. We allow the agent to record the local chemoattractant concentrations and curvatures as states on the swimming trajectory, and further define the increment of average concentration as reward (R_1). The agent is trained via the DRL algorithm. It is found that the microrobot can self-learn to implement chemotactic motion in a planar motion case, a flipping torsion case, and a positive torsion case. In the negative torsion case, due to the specific parameter choice we have $\tau_1/\kappa_1 \approx \tau_2/\kappa_2$, the microrobot has very low steering capability hence fails to learn any useful maneuvering strategy. Indeed, very little information is needed to accomplish the chemotactic motion. At minimum, two or four signal records are enough. And the learned strategy is always statistically better than the human-designed shortsighted strategy. However, the learned strategy do not guarantee accurate alignment between the net migration direction and the chemoattractant gradient direction. In the case of 3D motion, there is also some probability for the strategy to fail. Nevertheless, some stochasticity, which is ubiquitous in realistic environment, can significantly improve the performance of the learned strategy and ensure the microrobot to swim upward the chemoattractant concentration.

If we do not restrict the microrobot to learn autonomously (using only the current and historical concentration information to evaluate the strategy) but supplement the accurate heuristic direction information (the helix direction and the gradient direction) to evaluate the strategy, then very efficient strategy can be learned. The microrobot can learn to quickly align the helix vector to the gradient direction using just several smart sequential gait switchings. The learned strategy is much better than the human-devised strategy and can guarantee a very good alignment between the net migration direction and the gradient direction. However, the success of the DRL also depends on the value of the steering angle step size and the inference angle error of the neural network. Three conditions should be satisfied: (1) Enough number of historical information should be fed to the neural network to accurately infer the angle θ_h between the helix vector and the gradient direction; (2) The steering angle step size ΔA , which is the angle change of the helix vector when the gait is switched, should be larger than the inference error of θ_h , so that the microrobot can have enough observation accuracy to evaluate its actions and learn to improve its strategy. (3) The steering angle step size ΔA should also be moderate, neither too large nor too small. If ΔA is too large, then the microrobot cannot perform precise steering. If ΔA is too small, then the microrobot has very low steering capability. These results provide useful guidance for the design of the microrobot.

To summarize, our results show that chemotactic behavior can be learned autonomously by microrobots through DRL, and the DRL approach can help the microrobot discover very efficient controlling strategy. It is possible to use the DRL approach to design smart synthetic flagellated microswimmers

that can self-learn to adjust itself to complex environments and develop “intelligent” behaviors.

From the perspective of a realistic microrobot, our study also reveals a feasible scheme to design and control a microrobot: As long as a microrobot can switch among several finite gaits with different characteristics along the swimming trajectory, we can rely on DRL to discover efficient and robust strategy to control the microrobot. It is difficult to fabricate microrobots with a sophisticated controlling mechanism, but in appropriate conditions simple controlling mechanism combined with DRL can still produce efficient maneuvering behaviors to navigate a complex environment.

The code necessary to reproduce the findings of this study can be accessed at [65].

ACKNOWLEDGMENTS

C.M. thanks Dr. Dmitry A. Fedosov at Forschungszentrum Jülich for useful discussions on the SDPD simulation of the microswimmer. X.B. acknowledges the starting grant from 100 Talents Program of Zhejiang University and the grant of Innovative Research Foundation of Ship General Performance (Contract No. 31422121). Q.F. and C.M. acknowledge the grants from National Natural Science Foundation of China (Grants No. 12272026 and No. 12302323).

C.M. and X.B. conceived the research project. C.M. performed the simulations and analysed the obtained data. All authors participated in the discussions and writing of the manuscript.

We declare we have no competing interests.

APPENDIX: SDPD SIMULATION OF A SWIMMING MICROROBOT

The smoothed dissipative particle dynamics (SDPD) method [66–68] is employed to resolve the fluid-structure interaction problem. It is a particle-based numerical approach that discretizes the Navier-Stokes equations in the Lagrangian framework and includes thermal fluctuations consistently. A version of SDPD conserving the angular momentum [69] is adopted, which is important for fluid-particle models to produce physically accurate results [69,70]. In SDPD, each particle can be considered as a small fluid volume (or Lagrangian discretization point) characterised by a position \mathbf{r}_i , velocity \mathbf{v}_i , and mass m_i . In addition, each particle possesses a spin velocity $\boldsymbol{\psi}_i$ and moment of inertia I_i for the enforcement of angular momentum conservation [69].

Every two SDPD particles (indexed by i and j) interact through four pairwise forces, including conservative \mathbf{F}_{ij}^C , dissipative forces \mathbf{F}_{ij}^D caused by translational velocity, dissipative forces \mathbf{F}_{ij}^R caused by rotational velocity, and random forces $\tilde{\mathbf{F}}_{ij}$. They are given by

$$\begin{aligned}\mathbf{F}_{ij}^C &= \left(\frac{P_i}{d_i^2} + \frac{P_j}{d_j^2} \right) F_{ij} \mathbf{r}_{ij}, \\ \mathbf{F}_{ij}^D &= -\gamma_{ij} [\mathbf{v}_{ij} + (\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}) \mathbf{e}_{ij}], \\ \mathbf{F}_{ij}^R &= -\gamma_{ij} \frac{\mathbf{r}_{ij}}{2} \times (\boldsymbol{\psi}_i + \boldsymbol{\psi}_j), \\ \tilde{\mathbf{F}}_{ij} &= \sigma_{ij} \left(d \overline{\mathbf{W}}_{ij}^s + \frac{1}{3} \text{tr}[d \mathbf{W}_{ij}] \mathbf{1} \right) \cdot \frac{\mathbf{e}_{ij}}{\Delta t},\end{aligned}\quad (\text{A1})$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and $\mathbf{e}_{ij} = \mathbf{r}_{ij}/r_{ij}$. Particle number density d_i is computed as $d_i = \sum_j W_{ij}$, where the smoothing kernel function $W_{ij} = W(r_{ij})$ vanishes beyond a cutoff radius r_c . It also defines a nonnegative function F_{ij} through the equation $\nabla_i W_{ij} = -\mathbf{r}_{ij} F_{ij}$. Particle mass density is given by $\rho_i = m_i d_i$. The pressure P_i is determined by an equation of state $P_i = P_0 (d_i/d_0)^v - P_b$, where d_0 is the average number density, P_0 and v are parameters controlling the sound speed $c = \sqrt{P_0 v/d_0}$, and P_b relates to the background pressure. $d \mathbf{W}_{ij}$ is a matrix of independent Wiener increments, $d \overline{\mathbf{W}}_{ij}^s$ is its traceless symmetric part, and Δt is the time step. The dissipative and random coefficients γ_{ij} and σ_{ij} are given by

$$\gamma_{ij} = \frac{20\eta}{7} \frac{F_{ij}}{d_i d_j}, \quad \sigma_{ij} = 2\sqrt{k_B T \gamma_{ij}}, \quad (\text{A2})$$

where η is the dynamic viscosity, T is temperature, and k_B is the Boltzmann constant.

The evolution of particle positions, translational and angular velocities is obtained by integration of the following equations of motion:

$$\begin{aligned}\dot{\mathbf{r}}_i &= \mathbf{v}_i, \\ m_i \dot{\mathbf{v}}_i &= \sum_j \mathbf{F}_{ij} = \sum_j (\mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R) + \tilde{\mathbf{F}}_{ij}, \\ \dot{\boldsymbol{\psi}}_i &= \frac{1}{2I_i} \sum_j \mathbf{r}_{ij} \times \mathbf{F}_{ij},\end{aligned}\quad (\text{A3})$$

using the velocity-Verlet algorithm [71].

In this work, the smoothing kernel is represented by the quintic spline function [72]

$$W(q) = w_0 \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5, & 0 \leq q < 1, \\ (3-q)^5 - 6(2-q)^5, & 1 \leq q < 2, \\ (3-q)^5, & 2 \leq q < 3, \\ 0, & q \geq 3, \end{cases} \quad (\text{A4})$$

where $q = r/h$ and $w_0 = 1/(120\pi h^3)$ in three dimensions (3D), $w_0 = 7/(478\pi h^2)$ in two dimensions (2D), and h is the smoothing length $h = r_c/3$.

The fluid, the microrobot, and the wall (if present) are all discretized using SDPD particles (Fig. 1). In simulations, the position of the wall particles are fixed, while the micro-

TABLE IV. Basic parameters for the SDPD simulations.

Parameters	Values
Cutoff radius r_c	1.0
Mass density ρ	12.0
Dynamic viscosity η	100
Average number density d_0	30
SPH particle mass m	1
Moment of inertial of SPH particles I	1
P_0 in the EoS	6400
Hydrostatic pressure $P_0 - P_b$	200
Exponent in the EoS ν	7
Boltzmann energy $k_B T$	$1e - 6$
Time step Δt	0.002
Total number of time steps N_{tot}	800 000
Spring coefficient k_l	16 000
Normal spring length l_b	0.437
Actuation amplitude A	0.16
Intrinsic curvature parameter b_1	0.05
Intrinsic curvature parameter b_2	0.015
Wave number k	0.942
Flagellum length L	10
Radius of the head R_h	1.2
Beating frequency ω	0.6, 0.8, 1.0, 1.2
Initial distance to the wall (if present) L_w	2.4
Size of the simulation box $L_x \times L_y \times L_z$	$25 \times 25 \times 20$
Relaxation time τ_r	10

robot particles are further connected by springs as depicted by Fig. 1(b) in the main text.

The basic parameters for our SDPD simulations are summarized in Table IV. Note that we have set the Boltzmann energy $k_B T$ to be very small to exclude the effect of thermal fluctuations. The simulation box is with periodic boundaries. At the beginning, the microrobot is straight and is positioned at the center of the xy plane. The head is pointing to the $-x$ direction. The first and the third filament of the flagellum are in the xy plane, while the second and the fourth filament are in the xz plane. After the simulation starts, the beating

amplitude of the flagellum (A) and the curvature parameters (b_1 and b_2) increase from 0 to their specified values following the rule: $C(t) = C[1 - \exp(-t/\tau_r)]$, where τ_r is a relaxation time. During a simulation the center of mass of the microrobot is recorded to generate the swimming trajectory for further usage.

If the length of the flagellum L and the swimming velocity V is used to estimate the Reynolds number $\text{Re} = \rho LV/\eta$, then we have $0.02 < \text{Re} < 0.05$ in our simulations. Therefore, the low Reynolds number condition is satisfied.

We perform a beam deflection simulation to measure the bending stiffness of the flagellum κ_f . In this simulation, the head of the microrobot is fixed, and no actuation, nor intrinsic curvature, is imposed to the flagellum. The flagellum is straight and pointing to the x direction. Then a force $f\mathbf{e}_z$ is applied to all the flagellum particles. The uniform load is $q = N_f f/L$, where N_f is the total number of particles constituting the flagellum. The flagellum is deflected under the uniform load. The deflection distance is given by the deflection formula:

$$\delta_z = \frac{qx'^2}{24\kappa_f}(6L^2 - 4Lx' + x'^2), \quad x' = x - x_h + R_h, \quad (\text{A5})$$

where x_h is the position of the center of the head, R_h is the radius of the head. The bending stiffness of the flagellum κ_f can be obtained by fitting the above formula to the deflection curve obtained from simulation. Using the parameters in Table IV, and set $q = 0.1$, the bending stiffness is measured to be $\kappa_f = 7938.6$.

The normal resistive force coefficient ξ_{\perp} of the flagellum can be measured by simulating a straight flagellum moving in bulk fluid in the direction normal to the flagellum. If the velocity is v and the total resistive force is F_r , then the resistive force coefficient is F_r/Lv . We set $F_r = 1840$ in our simulation, then the resultant moving velocity is $v = 0.366$. The resistive force coefficient is $\xi_{\perp} = 502.7$. Therefore, the sperm number $\text{Sp} = L(\xi_{\perp}\omega/\kappa_f)^{1/4}$ is $\text{Sp} \in [4.41, 5.25]$ in our simulations.

- [1] E. M. Purcell, Life at low Reynolds number, *Am. J. Phys.* **45**, 3 (1977).
- [2] H. C. Berg and R. A. Anderson, Bacteria swim by rotating their flagellar filaments, *Nature (London)* **245**, 380 (1973).
- [3] E. Lauga and T. R. Powers, The hydrodynamics of swimming microorganisms, *Rep. Prog. Phys.* **72**, 096601 (2009).
- [4] J. Elgeti, R. G. Winkler, and G. Gompper, Physics of microswimmers—Single particle motion and collective behavior: A review, *Rep. Prog. Phys.* **78**, 056601 (2015).
- [5] L. J. Fauci and R. Dillon, Biofluidmechanics of reproduction, *Annu. Rev. Fluid Mech.* **38**, 371 (2006).
- [6] R. Liu and H. Ochman, Stepwise formation of the bacterial flagellar system, *Proc. Natl. Acad. Sci. USA* **104**, 7116 (2007).
- [7] J. Happel and H. Brenner, *Low Reynolds Number Hydrodynamics with Special Applications to Particular Media* (Martinus Nijhoff Publishers, Leiden, Netherlands, 1983).
- [8] S. Kim and S. J. Karrila, *Microhydrodynamics: Principles and Selected Applications* (Butterworth-Heinemann, Oxford, UK, 1991).
- [9] W. F. Paxton, K. C. Kistler, C. C. Olmeda, A. Sen, S. K. S. Angelo, Y. Cao, T. E. Mallouk, P. E. Lammert, and V. H. Crespi, Catalytic nanomotors: Autonomous movement of striped nanorods, *J. Am. Chem. Soc.* **126**, 13424 (2004).
- [10] J. R. Howse, R. A. L. Jones, A. J. Ryan, T. Gough, R. Vafabakhsh, and R. Golestanian, Self-motile colloidal particles: From directed propulsion to random walk, *Phys. Rev. Lett.* **99**, 048102 (2007).
- [11] H. R. Jiang, N. Yoshinaga, and M. Sano, Active motion of a Janus particle by self-thermophoresis in a defocused laser beam, *Phys. Rev. Lett.* **105**, 268302 (2010).
- [12] G. Volpe, I. Buttinoni, D. Vogt, H. J. Kümmerer, and C. Bechinger, Microswimmers in patterned environments, *Soft Matter* **7**, 8810 (2011).

- [13] S. Sanchez, A. A. Solovev, S. Schulze, and O. G. Schmidt, Controlled manipulation of multiple cells using catalytic microbots, *Chem. Commun.* **47**, 698 (2011).
- [14] G. Li, Swimming dynamics of a self-propelled droplet, *J. Fluid Mech.* **947**, E1 (2022).
- [15] P. Tierno, R. Golestanian, I. Pagonabarraga, and F. Sagués, Controlled swimming in confined fluids of magnetically actuated colloidal rotors, *Phys. Rev. Lett.* **101**, 218304 (2008).
- [16] R. Dreyfus, J. Baudry, M. L. Roper, M. Fermigier, H. A. Stone, and J. Bibette, Microscopic artificial swimmers, *Nature (London)* **437**, 862 (2005).
- [17] T. Sanchez, D. Welch, D. Nicastro, and Z. Dogic, Microtubule bundles, *Science* **333**, 456 (2011).
- [18] B. J. Williams, S. V. Anand, J. Rajagopalan, and M. T. A. Saif, A self-propelled biohybrid swimmer at low Reynolds number, *Nat. Commun.* **5**, 3081 (2014).
- [19] H. Zhang, Z. Li, C. Gao, X. Fan, Y. Pang, T. Li, Z. Wu, H. Xie, and Q. He, Dual-responsive biohybrid neutroblots for active target delivery, *Sci. Robot.* **6**, eaaz9519 (2021).
- [20] S. Saha, R. Golestanian, and S. Ramaswamy, Clusters, asters, and collective oscillations in chemotactic colloids, *Phys. Rev. E* **89**, 062316 (2014).
- [21] C. Jin, C. Kru-ger, and C. C. Maass, Chemotaxis and autochemotaxis of self-propelling droplet swimmers, *Proc. Natl. Acad. Sci. USA* **114**, 5089 (2017).
- [22] D. Ahmed, C. Dillinger, A. Hong, and B. J. Nelson, Artificial acousto-magnetic soft microswimmers, *Adv. Mater. Technol.* **2**, 1700050 (2017).
- [23] L. Baraban, D. Makarov, R. Streubel, I. Mönch, D. Grimm, S. Sanchez, and O. G. Schmidt, Catalytic Janus motors on microfluidic chip: Deterministic motion for targeted cargo delivery, *ACS Nano* **6**, 3383 (2012).
- [24] L. Amoudruz and P. Koumoutsakos, Independent control and path planning of microswimmers with a uniform magnetic field, *Adv. Intell. Syst.* **4**, 2100183 (2022).
- [25] B. Dai, J. Wang, Z. Xiong, X. Zhan, W. Dai, C. C. Li, S. P. Feng, and J. Tang, Programmable artificial phototactic microswimmer, *Nat. Nanotechnol.* **11**, 1087 (2016).
- [26] C. Lozano, B. T. Hagen, H. Löwen, and C. Bechinger, Phototaxis of synthetic microswimmers in optical landscapes, *Nat. Commun.* **7**, 12828 (2016).
- [27] A. I. Campbell and S. J. Ebbens, Gravitaxis in spherical Janus swimming devices, *Langmuir* **29**, 14066 (2013).
- [28] B. Ten Hagen, F. Kümmel, R. Wittkowski, D. Takagi, H. Löwen, and C. Bechinger, Gravitaxis of asymmetric self-propelled colloidal particles, *Nat. Commun.* **5**, 4829 (2014).
- [29] B. Liebchen, P. Monderkamp, B. T. Hagen, and H. Löwen, Viscotaxis: Microswimmer navigation in viscosity gradients, *Phys. Rev. Lett.* **120**, 208002 (2018).
- [30] C. Datt and G. J. Elfring, Active particles in viscosity gradients, *Phys. Rev. Lett.* **123**, 158006 (2019).
- [31] B. M. Friedrich and F. Jülicher, Chemotaxis of sperm cells, *Proc. Natl. Acad. Sci. USA* **104**, 13256 (2007).
- [32] H. C. Berg, D. A. Brown, Chemotaxis in *E. coli* analysed by 3D tracking, *Nature (London)* **239**, 500 (1972).
- [33] H. I. Choi, J. Y. H. Kim, H. S. Kwak, Y. J. Sung, and S. J. Sim, Quantitative analysis of the chemotaxis of a green alga, *Chlamydomonas reinhardtii*, to bicarbonate using diffusion-based microfluidic device, *Biomicrofluidics* **10**, 014121 (2016).
- [34] M. N. Popescu, W. E. Uspal, C. Bechinger, and P. Fischer, Chemotaxis of active janus nanoparticles, *Nano Lett.* **18**, 5345 (2018).
- [35] D. Eshel and C. J. Brokaw, New evidence for a “biased baseline” mechanism for calcium-regulated asymmetry of flagellar bending, *Cell Motil. Cytoskel.* **7**, 160 (1987).
- [36] A. Gong, S. Rode, U. B. Kaupp, G. Gompper, J. Elgeti, B. M. Friedrich, and L. Alvarez, The steering gaits of sperm, *Philos. Trans. Roy. Soc. B: Biol. Sci.* **375**, 20190149 (2020).
- [37] G. Saggiorato, L. Alvarez, J. F. Jikeli, U. B. Kaupp, G. Gompper, and J. Elgeti, Human sperm steer with second harmonics of the flagellar beat, *Nat. Commun.* **8**, 1415 (2017).
- [38] A. Gong, S. Rode, G. Gompper, U. B. Kaupp, J. Elgeti, B. M. Friedrich, and L. Alvarez, Reconstruction of the three-dimensional beat pattern underlying swimming behaviors of sperm, *Eur. Phys. J. E* **44**, 87 (2021).
- [39] B. M. Friedrich and F. Jülicher, Steering chiral swimmers along noisy helical paths, *Phys. Rev. Lett.* **103**, 068102 (2009).
- [40] A. C. H. Tsang, P. W. Tong, S. Nallan, and O. S. Pak, Self-learning how to swim at low Reynolds number, *Phys. Rev. Fluids* **5**, 074101 (2020).
- [41] K. Qin, Z. Zou, L. Zhu, and O. S. Pak, Reinforcement learning of a multilink swimmer at low Reynolds numbers, *Phys. Fluids* **35**, 032003 (2023).
- [42] J. K. Alageshan, A. K. Verma, J. Bec, and R. Pandit, Path-planning microswimmers can swim efficiently in turbulent flows, *Phys. Rev. E* **101**, 043110 (2020).
- [43] P. Gunnarson, I. Mandralis, G. Novati, P. Koumoutsakos, and J. O. Dabiri, Learning efficient navigation in vortical flow fields, *Nat. Commun.* **12**, 7143 (2021).
- [44] Y. Yang, M. A. Bevan, and B. Li, Efficient navigation of colloidal robots in an unknown environment via deep reinforcement learning, *Adv. Intell. Syst.* **2**, 1900106 (2020).
- [45] E. Schneider and H. Stark, Optimal steering of a smart active particle, *Europhys. Lett.* **127**, 64003 (2019).
- [46] M. Buzzicotti, L. Biferale, F. Bonaccorso, P. C. D. Leoni, and K. Gustavsson, Optimal control of point-to-point navigation in turbulent time dependent flows using reinforcement learning, in *AIxIA 2020—Advances in Artificial Intelligence*, edited by Matteo Baldoni and Stefania Bandini (Springer International Publishing, Cham, 2021), pp. 223–234.
- [47] S. Muiños-Landin, K. Ghazi-Zahedi, and F. Cichos, Reinforcement learning of artificial microswimmers, *Sci. Robot.* **6**, eabd9285 (2021).
- [48] M. Nasiri and B. Liebchen, Reinforcement learning of optimal active particle navigation, *New J. Phys.* **24**, 073042 (2022).
- [49] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Flow navigation by smart microswimmers via reinforcement learning, *Phys. Rev. Lett.* **118**, 158004 (2017).
- [50] K. Gustavsson, L. Biferale, A. Celani, and S. Colabrese, Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning, *Eur. Phys. J. E* **40**, 110 (2017).
- [51] B. Hartl, M. Hübl, G. Kahl, and A. Zöttl, Microswimmers learning chemotaxis with genetic algorithms, *Proc. Natl. Acad. Sci. USA* **118**, e2019683118 (2021).
- [52] Z. Zou, Y. Liu, Y. N. Young, O. S. Pak, and A. C. H. Tsang, Gait switching and targeted navigation of microswimmers via deep reinforcement learning, *Commun. Phys.* **5**, 158 (2022).

- [53] G. Zhu, W.-Z. Fang, and L. Zhu, Optimizing low-Reynolds-number predation via optimal control and reinforcement learning, *J. Fluid Mech.* **944**, A3 (2022).
- [54] S. Verma, G. Novati, and P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, *Proc. Natl. Acad. Sci. USA* **115**, 5849 (2018).
- [55] C. Mo and D. A. Fedosov, Competing effects of inertia, sheet elasticity, fluid compressibility, and viscoelasticity on the synchronization of two actuated sheets, *Phys. Fluids* **33**, 043109 (2021).
- [56] C. Mo and D. A. Fedosov, Hydrodynamic clustering of two finite-length flagellated swimmers in viscoelastic fluids, *J. R. Soc. Interface* **20**, 20220667 (2023).
- [57] S. Rode, J. Elgeti, and G. Gompper, Sperm motility in modulated microchannels, *New J. Phys.* **21**, 013016 (2019).
- [58] Z. Liu, F. Qin, and L. Zhu, Actuating a curved elastic filament for bidirectional propulsion, *Phys. Rev. F* **5**, 124101 (2020).
- [59] Z. Liu, F. Qin, L. Zhu, R. Yang, and X. Luo, Effects of the intrinsic curvature of elastic filaments on the propulsion of a flagellated microrobot, *Phys. Fluids* **32**, 041902 (2020).
- [60] J. A. Kromer, S. Märcker, S. Lange, C. Baier, and B. M. Friedrich, Decision making improves sperm chemotaxis in the presence of noise, *PLoS Comput. Biol.* **14**, e1006109 (2018).
- [61] S. Lange and B. M. Friedrich, Sperm chemotaxis in marine species is optimal at physiological flow rates according theory of filament surfing, *PLoS. Comput. Biol.* **17**, e1008826 (2021).
- [62] H. V. Hasselt, A. Guez, and D. Silver, Deep reinforcement learning with double Q learning, *Proceedings of the 13th AAAI Conference on Artificial Intelligence* (AAAI, Washington, DC, 2016), pp. 2094–2100.
- [63] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, Playing atari with deep reinforcement learning, *arXiv:1312.5602* (2013).
- [64] R. O. Ramakrishnan and B. M. Friedrich, Learning run-and-tumble chemotaxis with support vector machines, *Europhys. Lett.* **142**, 47001 (2023).
- [65] <https://github.com/mokchie/chemotaxis.git>.
- [66] P. Español and M. Revenga, Smoothed dissipative particle dynamics, *Phys. Rev. E* **67**, 026705 (2003).
- [67] X. Bian, S. Litvinov, R. Qian, M. Ellero, and N. A. Adams, Multiscale modeling of particle in suspension with smoothed dissipative particle dynamics, *Phys. Fluids* **24**, 012002 (2012).
- [68] M. Ellero and P. Espanol, Everything you always wanted to know about SDPD (but were afraid to ask), *Appl. Math. Mech.* **39**, 103 (2018).
- [69] K. Müller, D. A. Fedosov, and G. Gompper, Smoothed dissipative particle dynamics with angular momentum conservation, *J. Comput. Phys.* **281**, 301 (2015).
- [70] X. Y. Hu and N. A. Adams, Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows? *Phys. Fluids* **18**, 101702 (2006).
- [71] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, UK, 2017).
- [72] M. Ellero and R. I. Tanner, SPH simulations of transient viscoelastic flows at low Reynolds number, *J. Non-Newton. Fluid Mech.* **132**, 61 (2005).