Long-distance migration with minimal energy consumption in a thermal turbulent environment

Ao Xu,[1,2,*] Hua-Lin Wu,[1] and Heng-Dong Xi[1,2]

[1] *School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China*

[2] *Institute of Extreme Mechanics, Northwestern Polytechnical University, Xi'an 710072, China*

(Dated: January 13, 2023)

Abstract

We adopt the reinforcement learning algorithm to train the self-propelling agent migrating long-distance in a thermal turbulent environment. We choose the Rayleigh–Bénard turbulent convection cell with an aspect ratio (Γ, which is defined as the ratio between cell length and cell height) of 2 as the training environment. Our results showed that, compared to a naive agent that moves straight from the origin to the destination, the smart agent can learn to utilize the carrier flow currents to save propelling energy. We then apply the optimal policy obtained from the $\Gamma = 2$ cell and test the smart agent migrating in convection cells with Γ up to 32. In a larger Γ cell, the dominant flow modes of horizontally stacked rolls are less stable, and the energy contained in higher-order flow modes increases. We found that the optimized policy can be successfully extended to convection cells with a larger Γ. In addition, the ratio of propelling energy consumed by the smart agent to that of the naive agent decreases with the increase of Γ, indicating more propelling energy can be saved by the smart agent in a larger Γ cell. We also evaluate the optimized policy when the agents are being released from the randomly chosen origin, which aims to test the robustness of the learning framework, and possible solutions to improve the success rate are suggested. This work has implications for long-distance migration problems, such as unmanned aerial vehicles patrolling in a turbulent convective environment, where planning energy-efficient trajectories can be beneficial to increase their endurance.

I. INTRODUCTION

Humans have long been fascinated with flight, and we can learn how to fly efficiently from birds. Some soaring birds can fly long distances during their trips without flapping their wings, and they spend the greatest effort only during the take-off or landing stage. For example, Weimerskirch et al. [1] showed that frigate birds can stay aloft for up to 48 days during transoceanic flight. Williams et al. [2] recorded that an Andean condor flew for over 5 hours without flapping, which covers 172 kilometers. Croxall et al. [3] revealed that the fastest gray-headed albatrosses can make global circumnavigations in just 46 days. It was not until 1885, when Lancaster published his pioneer observations and deductions [4], that the mystery of flying birds not flapping their wings is gradually solved. The secret of birds is that they can utilize warm rising atmospheric currents (also known as *thermals*) to

* Author to whom correspondence should be addressed: axu@nwpu.edu.cn

reduce the expenditure of energy. Thermals are part of the convection flows that develop in the convective layer of the atmosphere (i.e., the troposphere). During sunny days, heat from the sun warms the earth and the earth warms the air above it. Warm air expands and lighter air rises, and the resulting column of rising air is called *thermals*.

Not only birds but also gliders and unmanned aerial vehicles (UAVs) can utilize the updrafts of thermals to increase endurance and save energy. For example, MacCready [5] determined the optimal gliding speed to fly between thermals to maximize speed and energy gain. Allen et al. [6] estimated a UAV with a nominal endurance of 2 hours can achieve a 12-hours increase in the summer and a 6-hours increase in the winter. To investigate the use of the convective lift, various thermal models have been developed, such as chimney models and bubble models [7]. Chimney thermals are continuous columns of rising air, which extend from the ground surface to the highest level of the troposphere [8]. Bubble thermals are closed updraft masses that form a rising vortex ring near the ground, and the updraft at the core of the vortex ring is provided by the buoyancy of the air [9]. When the air leaves the bubble core, it cools down and loses buoyancy, thus moving downward on the outside of the vortex ring to complete a cycle. Both the chimney and bubble thermal models describe simplified situations, where there is no turbulent motion or the fluctuations are modeled as Gaussian white noise. However, in the troposphere, the wind field exhibits strong turbulent fluctuations. Akos et al. [10] found that turbulent fluctuations of the environment bring challenges in identifying effective thermal soaring strategies. Laurent et al. [11] further pointed out that turbulence leaves an imprint on all modes of flight, and they revealed the analogy between the flight trajectories of a golden eagle and the trajectories of particles carried by turbulent flows. They also reinforced the need to fully incorporate turbulence into understanding the movement and behaviors of the flying object.

To model the flow patterns of the wind in strong convective weather, a paradigmatic turbulent convection system, known as Rayleigh-Bénard (RB) convection, can describe turbulent flows driven by buoyancy forces [12–14]. The control parameters of the RB system mainly include the Prandtl number (Pr, defined later in the paper), the Rayleigh number (Ra), and the cell aspect ratio (Γ). The Pr describes the thermophysical properties of the fluid and $Pr = 0.71$ for the air. The Ra describes the ratio of buoyancy forces relative to the viscous forces due to temperature differences, and $10^{18} \leq Ra \leq 10^{22}$ in the atmosphere [13]. The Γ characterizes the geometric information of the convection system, and $\Gamma \approx 100$ for mesoscale convective system [15]. In the RB turbulent convection, important coherent structures include small-scale thermal plumes, large-scale circulation rolls, and the very-large-scale superstructure. The thermal plumes are detached from the hot or cold boundary layers, it then collides and merges, further self-organize into large-scale circulation rolls. If the convection system extends several times the distance in the horizontal direction than that in the vertical direction, thermal plumes form a web of connected ridgelike structures of cold downwelling and hot upwelling fluids, also known as the superstructure of thermal turbulence [16, 17].

Adopting the RB turbulent environment, Reddy et al. [18] numerically trained a glider to rise on thermals using reinforcement learning algorithms. The trained glider can ascend from low altitude to high altitude in a spiral form, which has a similar pattern to soaring birds in nature [19]. They analyzed the changes in the glider's flight strategy when the turbulent intensity varied. Thereafter, they equipped a glider with a two-meter wingspan and trained the glider in the field to navigate atmospheric thermals autonomously [20]. In Reddy et al.'s works [18, 20], the main goal is to train the glider to ascend higher; whilst for practical application of UAVs, a more frequently encountered scenario is to fly from one position to another. To minimize energy consumption during the point-to-point migration in a thermal turbulent environment, Xu et al. [21] optimized the trajectory for a self-propelling agent in the RB turbulent convection, such that the agent can utilize the kinetic energy of the thermal turbulence as much as possible. Compared with the straight-line propelling trajectory, the optimized trajectory allows the agent to save around two-thirds of its energy consumption.

In the previous work on soaring within the RB turbulent environment [18, 21], the simulated RB convection cells have an aspect ratio of $\Gamma \leq 2$. However, migration often occurs in a large-aspect-ratio convection system and covers a long distance. In this work, our motivation is to train the self-propelling agent to migrate in a large-aspect-ratio RB cell that has multiple circulation rolls. The rest of this paper is organized as follows. In Section II, we introduce numerical details for the simulation of the turbulent environment, including the mathematical model and the in-house numerical solver for the RB convection. In Section III, we present details of the dynamics of the self-propelling agent and the reinforcement learning algorithm to train the agent to find an energy-efficient trajectory. In Section IV, we first present general flow patterns in the RB convection, followed by training results for the agent migrating in a $\Gamma = 2$ cell, and then test the agent migrating in larger Γ cells. In Section V, the main findings of this work are summarized.

II. SIMULATION OF THE TURBULENT ENVIRONMENT

II.1. Mathematical model for the RB turbulent thermal convection

We simulate the turbulent environment in the RB convection cells based on the Boussinesq approximation. We assume the fluid flow is incompressible, and we treat the temperature as an active scalar that influences the velocity field through the buoyancy. The viscous heat dissipation and compression work are neglected, and all the transport coefficients are assumed to be constants. Then, the governing equations for the RB thermal convection can be written as

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0}\nabla P + \nu \nabla^2 \mathbf{u} + g\beta_T (T - T_0)\hat{\mathbf{y}} \tag{2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha_T \nabla^2 T \tag{3}$$

where $\mathbf{u} = (u, v)$, P and T are the velocity, pressure, and temperature of the fluid, respectively. ρ_0 and T_0 are reference density and temperature, respectively. $\hat{\mathbf{y}}$ is the unit parallel to gravity. With the scaling

$$\begin{aligned} &\mathbf{x}^* = \mathbf{x}/H, \quad t^* = t/\sqrt{H/(\beta_T g \Delta_T)}, \quad \mathbf{u}^* = \mathbf{u}/\sqrt{\beta_T g H \Delta_T}, \\ &P^* = P/(\rho_0 g \beta_T \Delta_T H), \quad T^* = (T - T_0)/\Delta_T \end{aligned} \tag{4}$$

Then, Eqs. 1, 2, 3 can be rewritten in dimensionless from as

$$\nabla \cdot \mathbf{u}^* = 0 \tag{5}$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla \mathbf{u}^* = -\nabla P^* + \sqrt{\frac{Pr}{Ra}} \nabla^2 \mathbf{u}^* + T^* \tilde{\mathbf{y}} \tag{6}$$

$$\frac{\partial T^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla T^* = \sqrt{\frac{1}{PrRa}} \nabla^2 T^* \tag{7}$$

Here, H is the cell height and it is chosen as the characteristics length. $t_f = \sqrt{H/(\beta_T g \Delta_T)}$ is the free-fall time and it is chosen as the characteristic time. Δ_T is the temperature difference between heating and cooling walls. The two dimensionless parameters are the Ra and the Pr, which are defined as

$$Ra = \frac{g\beta_T \Delta_T H^3}{\nu \alpha_T}, \quad Pr = \frac{\nu}{\alpha_T} \tag{8}$$

II.2. The lattice Boltzmann method for thermal convection

We adopt the lattice Boltzmann (LB) method to simulate thermal convection. The advantages of the LB method include easy implementation and parallelization as well as high computing efficiency [22]. Specifically, we chose a D2Q9 model for the Navier–Stokes equations to simulate fluid flows and a D2Q5 model for the energy equation to simulate heat transfer. To enhance the numerical stability, the multi-relaxation-time collision operator is adopted in the evolution equations of both density and temperature distribution functions. The evolution equation of the density distribution function is written as

$$f_i(\mathbf{x}+\mathbf{e}_i\delta_t, t+\delta_t) - f_i(\mathbf{x},t) = -\left(\mathbf{M}^{-1}\mathbf{S}\right)_{ij}\left[\mathbf{m}_j(\mathbf{x},t) - \mathbf{m}_j^{(\mathrm{eq})}(\mathbf{x},t)\right] + \delta_t F_i' \tag{9}$$

where f_i is the density distribution function. $\mathbf{x}$ is the fluid parcel position, t is the time, δ_t is the time step. $\mathbf{e}_i$ is the discrete velocity along the ith direction. The forcing term F_i' on the right-hand side of Eq. 9 is given by $\mathbf{F}' = \mathbf{M}^{-1}\left(\mathbf{I}-\mathbf{S}/2\right)\mathbf{M}\tilde{\mathbf{F}}$, and the term $\mathbf{M}\tilde{\mathbf{F}}$ is given as [23] $\mathbf{M}\tilde{\mathbf{F}} = [0,\ 6\mathbf{u}\cdot\mathbf{F},\ -6\mathbf{u}\cdot\mathbf{F},\ F_x,\ -F_x,\ F_y,\ -F_y,\ 2uF_x - 2vF_y,\ uF_x + vF_y]^T$ where $\mathbf{F} = \rho g\beta_T(T-T_0)\hat{\mathbf{y}}$. The macroscopic density ρ and velocity $\mathbf{u}$ are obtained from $\rho = \sum_{i=0}^{8} f_i$, $\mathbf{u} = \left(\sum_{i=0}^{8}\mathbf{e}_i f_i + \mathbf{F}/2\right)/\rho$.

The evolution equation of the temperature distribution function is written as

$$g_i(\mathbf{x}+\mathbf{e}_i\delta_t, t+\delta_t) - g_i(\mathbf{x},t) = -\left(\mathbf{N}^{-1}\mathbf{Q}\right)_{ij}\left[\mathbf{n}_j(\mathbf{x},t) - \mathbf{n}_j^{(\mathrm{eq})}(\mathbf{x},t)\right] \tag{10}$$

where g_i is the temperature distribution function. The macroscopic temperature T is obtained from $T = \sum_{i=0}^{4} g_i$. More numerical details of the LB method and validation of the in-house solver can be found in our previous work [24–26].

II.3. Simulation settings for the turbulent thermal convection

We consider a two-dimensional RB cell with length L and height H. The top and bottom walls of the cell are kept at constant cold and hot temperatures, respectively; while the other two vertical walls are adiabatic; all four walls impose no-slip velocity boundary conditions. We set the cell aspect ratio ($\Gamma = L/H$) as $2 \le \Gamma \le 32$, and we fix the Prandtl number as $Pr = 0.71$ (corresponds to the working fluids of air) and the Rayleigh number as $Ra = 10^8$. Although the Ra is far less than that in the atmosphere because of limited computing resources to simulate ultra-high Ra convection, we note the RB convection at

$Ra = 10^8$ already exhibits strong turbulent fluctuations and the flows fall in the 'hard turbulence' regime [27]. We also checked the turbulent database and confirmed that statistically stationary states have been reached and the initial transient effects of the simulations are washed out.

III. DYNAMICS AND CONTROL OF THE SELF-PROPELLING AGENT

III.1. Kinematic model of the self-propelling agent

The dynamics of the self-propelling agent can be described as

$$\mathbf{u}_{\text{agent}}(t) = \mathbf{u}_{\text{fluid}}(t) + \mathbf{u}_{\text{propel}}(t) \tag{11}$$

$$\mathbf{x}_{\text{agent}}(t + dt) = \mathbf{x}_{\text{agent}}(t) + \mathbf{u}_{\text{agent}}(t) \cdot dt \tag{12}$$

Here, dt is the time step, $\mathbf{u}_{\text{agent}}$ and $\mathbf{x}_{\text{agent}}$ denote the velocity and position of the agent, respectively. $\mathbf{u}_{\text{fluid}}$ denotes the velocity of the carrier fluids, and $\mathbf{u}_{\text{propel}}$ denotes the velocity generated by the agent. We assume that, without control, the velocity of the agent equals that of the carrier flows; whilst, with control, the velocity of the agent is the superposition of the carrier flow and agent's propulsion. Similar dynamics of the agent have been previously adopted by Krishna et al. [28]. To mimic the limited propelling ability of the agent in real-world scenarios, we restricted the maximum propelling velocity of the agent $\|\mathbf{u}_{\text{agent}}\|$ to be less than one-third of the largest carrier flow velocity. On the other hand, a more complex kinematic model for the self-propelling agent, such as the one that includes inertial and rotational dynamics [29–33], fluttering and tumbling [34], multimodal locomotion [35] of the propelling agent can be considered in the future work.

III.2. Optimal control via the reinforcement learning

We adopt the RL algorithm to optimize the control of the agent to migrate in an energy-efficient trajectory. The advantages of the RL algorithm include agnostic for control and optimization tasks, easy to be re-used to speed-up optimization in a similar system configuration, and robust to the disturbances in the chaotic system [36, 37]. In the RL algorithm, the agent observes the state of the environment and decides to take an action interacting with the environment. If the agent then receives a reward (or a penalty), it is more likely to repeat (or forego) that action in the future. Overall, the agent learns by trial and error, with the long-term goal to maximize the cumulative expected return, and eventually improve its performance. Applying the RL algorithm, we can obtain the optimal policy, which advises the favorable action to take for the agent [38–41]. The model-free RL algorithm can generally be classified into policy-based methods and value-based methods. In the policy-based method, such as the policy gradient method, the parameter of the policy network θ is optimized to maximize the performance objective $J(\pi_\theta)$. Here, π_θ denotes the parameterized stochastic policy. The policy-based methods are inefficient in sampling, thus leading to slow learning, and are not suitable for complex flow problems [42]. In the value-based methods, such as the Q-learning method, the agent takes action a that tried to maximize the optimal action-value function, i.e., $a(s) = \arg\max_a Q_\theta(s, a)$. Here, s denotes the state of the environment and $Q_\theta(s, a)$ approximates the optimal action-value function $Q^*(s, a)$. Using the Q-learning method, Colabrese et al. [29] showed that gravitactic swimmers can reach high altitudes in steady Taylor-Green vortex flow; Muiños-Landin et al. [43] demonstrated the artificial self-thermophoretic micro-swimmers can navigate under the influence of Brownian motion; Monderkamp et al. [44] trained active Brownian particles through complex motility landscapes; Gazzola et al. [45] and Verma et al. [46] found optimal swimming strategies that minimize drag and energy consumption in the school of fish. The above five examples adopt the off-policy learning techniques, which means that each update stochastic samples the data collected at any point during training, namely, $Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$. Earlier, Reddy et al. [18] adopted the state-action-reward-state-action (SARSA) method, which can be regarded as a variation of the Q-learning method. The main difference is that the SARSA method adopts the on-policy learning technique, which uses the action performed by the current policy to learn the Q-value, namely, $Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$. However, the value-based method can only work in discrete state and action spaces; whilst in most real-world scenarios, such as training a vehicle to navigate, the continuous state and action spaces are preferred to develop more versatile motion for complex navigation tasks.

In this work, we adopt the soft actor-critic (SAC) algorithm, which is an interpolation between policy-based methods and value-based methods. In the SAC algorithm, the agent decides the next action via the actor network, whilst that action is further evaluated by the critic network to guide the training process. In addition, the actor aims to maximize the expected reward (i.e., succeed at the task) while also maximizing entropy (i.e., acting as randomly as possible). In entropy regularized reinforcement learning, the optimization problem can be described as

$$\pi^*(\theta) = \arg\max_{\pi} E_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\left(R(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot|s_t))\right)\right] \tag{13}$$

In the above equation, π^* is the optimal policy. The reward function r depends on the current state of the environment s_t, the current action a_t, and the next state of the environment s_{t+1}. α is the trade-off coefficient. The entropy H of τ is computed from its distribution π as $H(\pi(\cdot|s_t)) = E_{\tau\sim\pi}[-\log\pi(\tau)]$. More details on the SAC algorithm can be found by Haarnoja et al. [47].

Key ingredients in the RL framework include the environmental cues that the agent can observe (i.e., the current state s_t of the environment), the actions the agent takes (i.e., a_t), and the response of the agent to its behavior (i.e., the reward r_t). In this work, to migrate in a large-aspect-ratio convection cell, the observation variables for the agent include the carrier flow velocity $\mathbf{u}_{\text{fluid}}$, the agent's spatial coordinate in the vertical direction y_{agent}, and the fluid temperature T. In Section IV.2, we provide a detailed discussion on selecting observation variables. The action variable is the propelling velocity $\mathbf{u}_{\text{propel}}$ generated by the agent. Following the previous work of Xu et al. [21], we assume the rewards received by the agent are simultaneously affected by the current state, energy consumption, and time consumption of the agent

$$r(t) = r_s(t) + r_e(t) + r_h(t) \tag{14}$$

Here, the r_s denotes the reward affected by the current state of the agent. If the agent migrates out of the flow domain through the top or the bottom boundaries, it will receive a penalty of -ϕ; if the agent moves rightward and gets closer to the right-side boundary, it will receive a basic reward e_{basic} with an empirical pre-factor ε. Thus, r_s is written as

$$r_s(t) = \begin{cases} -\phi, & \text{agent is out of the flow domain} \\ \varepsilon e_{\text{basic}}, & x_{\text{agent}}^{t} > x_{\text{agent}}^{t-1} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Here, we adopt $\phi = 10$ and $\varepsilon = 10$. A detailed discussion on the sensitivity of the hyper-parameters in the reward function can be found in Appendix A. In Eq. 14, the r_e denotes the reward affected by the energy consumption of the agent. If the propelling velocity of the agent $\mathbf{u}_{\text{propel}}$ is in alignment with that of the background flow $\mathbf{u}_{\text{fluid}}$, namely, the angle between these two vectors (denoted by θ) is less than 90°, the agent will receive a reward of $\varepsilon[e_{\text{basic}} + (e_{\max} - e)]$, where $e = 0.5||\mathbf{u}_{\text{propel}}||^2$ and $e_{\text{basic}} = e_{\max} = 0.5(||\mathbf{u}_{\text{propel}}||)^2_{\max}$; if the angle between $\mathbf{u}_{\text{propel}}$ and $\mathbf{u}_{\text{fluid}}$ is greater than 90°, the agent will receive a penalty of $-2\varepsilon(e_{\text{basic}} + e)$. The above designs also imply that, when the agent migrates in alignment with the carrier flow direction, it will receive a lower reward if the propelling velocity is higher; when the agent migrates against the carrier flow direction, it will receive a higher penalty if the propelling velocity is higher. Thus, r_e is written as

$$r_e(t) = \begin{cases} \varepsilon[e_{\text{basic}} + (e_{\max} - e)], & 0^\circ \leq \theta \leq 90^\circ \\ -2\varepsilon(e_{\text{basic}} + e), & 90^\circ \leq \theta \leq 180^\circ \end{cases} \tag{16}$$

In Eq. 14, the r_h denotes the reward affected by the time consumption of the agent. If the agent migrates out of the domain via the right-side boundary, we assume the agent completes the task and it will receive a reward that is inversely proportional to the time taken. This design implies that the sooner the agent reaches the destination, the higher the reward it receives. Thus, r_h is written as

$$r_h(t) = \begin{cases} (t_{\max} - t)/\varepsilon, & x^t_{\text{agent}} > L \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

IV. RESULTS AND DISCUSSION

IV.1. General flow patterns in the RB convection with a large aspect ratio

Typical snapshots of the temperature field at $Ra = 10^8$, $Pr = 0.71$, and $2 \leq \Gamma \leq 32$ are shown in Fig. 1. We can distinguish small-scale thermal plumes and large-scale circulation rolls. Thin thermal boundary layers appear near the bottom heating wall and top cooling wall. Plumes that are released from the boundary layers penetrate upwards (or downwards) towards the opposite wall of lower (or higher temperature), and intense mixing occurs in the central region. Thermals are almost periodically released from relatively fixed locations, and neighboring thermals that move in opposite directions entrain the surrounding fluid and self-organize into circulation rolls. Previously, Wang et al. [48] found that depending on the initial conditions, the flow system at a given aspect ratio evolves to different final turbulent states with different roll numbers. In our work, the convection roll number n is 2, 4, 9, 18 and 34, in the $\Gamma = 2$, 4, 8, 16 and 32 convection cells, respectively; their corresponding mean aspect ratios are $\Gamma_r = \Gamma/n = 1$, 1, 0.889, 0.889 and 0.941, which is consistent with that predicted by the elliptical instability theory [48].

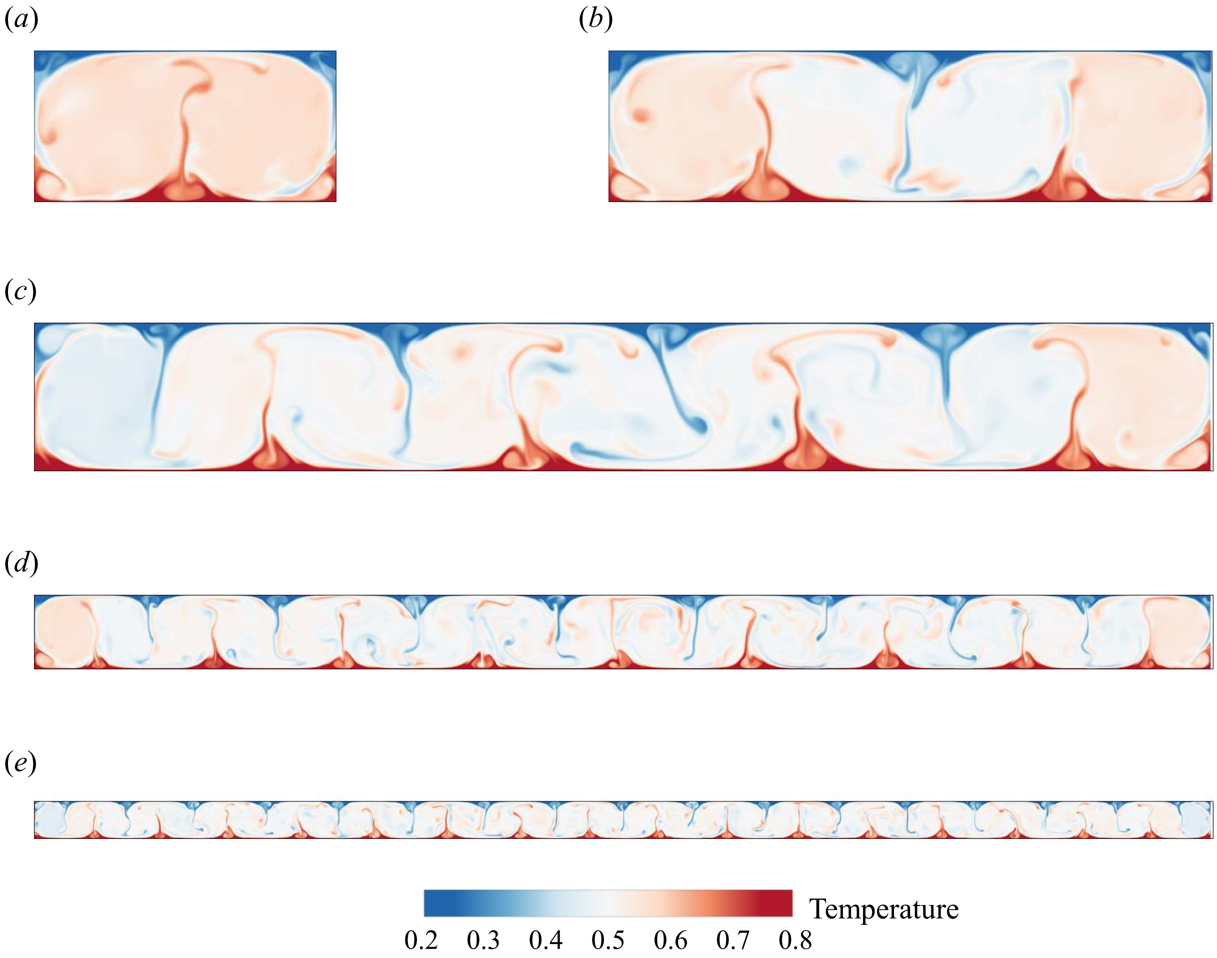

FIG. 1. Typical instantaneous temperature field at $Ra = 10^8$, $Pr = 0.71$, (a) $\Gamma = 2$, (b) $\Gamma = 4$, (c) $\Gamma = 8$, (d) $\Gamma = 16$, (e) $\Gamma = 32$.

We then examine the global response parameter of Reynolds number (Re) on the control parameter Γ. Here, the global flow strength is calculated as $Re = \sqrt{\langle (u^2 + v^2) \rangle_{V,t}} H/\nu$ and $\langle \cdots \rangle_{V,t}$ denotes the spatial and temporal average. The measured Re as a function of Γ is shown in Fig. 2(a), and we can observe enhanced global flow strength with the increase of cell aspect ratio. In addition, with the increase of Γ, the Re gradually reaches an asymptotic value, similar to that in the 3D convection cell [16]. Previously, Xu et al. [21] found the optimized energy-efficient strategy obtained from the reinforcement learning algorithm is not sensitive to small perturbation of the global flow strength, but it changes when the global flow strength increases (or decreases) more than one magnitude of order. Because the self-propelling agent was trained in the $\Gamma = 2$ cell, we further checked that even in the $\Gamma = 32$ cell, the flow strength increases around 22% compared to that in the $\Gamma = 2$ cell, indicating the flow strength in a larger aspect ratio cell does not increase significantly. To

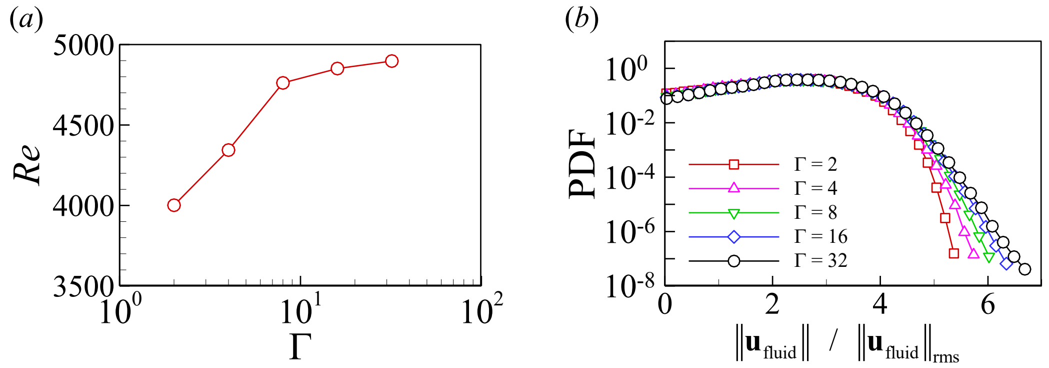

FIG. 2. (*a*) Reynolds number, and (*b*) probability density functions (PDFs) of normalized velocity magnitude $\|\mathbf{u}_{\text{fluid}}\|/\|\mathbf{u}_{\text{fluid}}\|_{\text{rms}}$, for various Γ at $Ra = 10^8$ and $Pr = 0.71$.

quantify the fluctuations of velocity magnitude, we plot the probability density functions (PDFs) of normalized velocity magnitude $\|\mathbf{u}_{\text{fluid}}\|/\|\mathbf{u}_{\text{fluid}}\|_{\text{rms}}$ for various Γ, as shown in Fig. 2(b). We can see the PDF heads collapse for different Γ, whilst the PDF tails become slightly extended with the increase of Γ, which implies an increased degree of fluctuations for the velocity magnitude $\|\mathbf{u}_{\text{fluid}}\|$.

To extract the coherent flow structure from the turbulent database, we adopt the proper orthogonal decomposition (POD) analysis. In the POD, the spatiotemporal vector field $\mathbf{X}(\mathbf{r}, t)$ is decomposed as a superposition of orthogonal eigenfunctions $\phi_i(\mathbf{r})$ and their amplitudes $a_i(t)$ as

$$\mathbf{X}(\mathbf{r}, t) = \sum_{i=1}^{\infty} a_i(t)\phi_i(\mathbf{r}) \tag{18}$$

Here, the vector field $\mathbf{X}(\mathbf{r}, t)$ is chosen as the flow velocity field $\mathbf{X} = (u, v)$. Practically, we can use the singular value decomposition (SVD) on the dataset $\mathbf{X}$ to obtain the flow mode $\phi_i(\mathbf{r})$ and the corresponding mode amplitude $a_i(t)$ [49]. Because the database for turbulent convection in a large-aspect-ratio cell is huge, which consists of fine spatial resolution and long temporal evolution data, the memory consumption to perform standard SVD is intense. Here, we adopt the randomized SVD to reduce the computational load [50]. The shape of the first POD mode $\phi_1(\mathbf{r})$ at various Γ is shown in Fig. 3. The most energetic POD mode consists of horizontally stacked circulation primary rolls rotating in either the clockwise direction or the anti-clockwise direction, and these primary rolls exhibit a periodical pattern. Large values of velocity magnitude appear near the vortex edge, indicating a strong energy barrier for the agent to move across the vortex edge. It is noteworthy that at much higher *Ra* (i.e., $Ra > 10^{10}$), when the large-scale-circulation is weaker and the flow consists of multiple

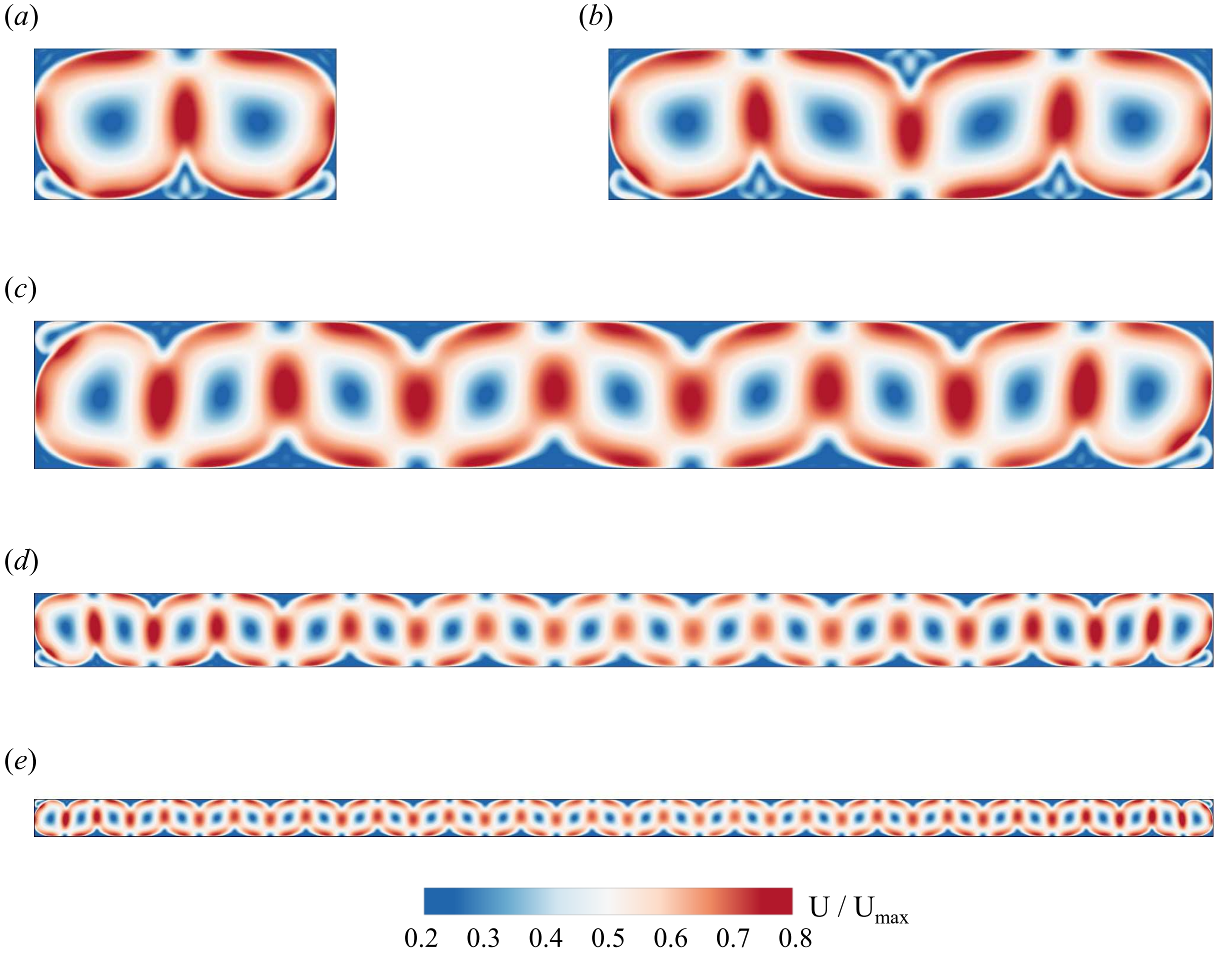

FIG. 3. Contour of the first proper orthogonal decomposition (POD) mode $\phi_1(\mathbf{r})$ at $Ra = 10^8$, $Pr = 0.71$, (*a*) $\Gamma = 2$, (*b*) $\Gamma = 4$, (*c*) $\Gamma = 8$, (*d*) $\Gamma = 16$, (*e*) $\Gamma = 32$. Here, $U = \sqrt{u^2 + v^2}$ is the velocity magnitude, and $U_{\max}$ denotes the maximum value of U for normalization.

mobile and orbital small vortices [51–53], whether the current learning framework still works deserves further study.

We further calculate the stability of the first POD mode as $S_1 = \sqrt{\langle a_1(t)\rangle_t}/\sigma_{a_1}$, such that a larger value of S_1 indicates a more stable pattern of circulation rolls. Similar estimations of the roll stability have been previously used in the Fourier mode decomposition of the turbulent thermal convection [49, 54]. From Fig. 4(a), we can see the stability of the first POD mode decreases with the increase of Γ. We also analyze the energy contained in the first POD mode and calculate the energy percentage as $\lambda_1/\sum_{i=1}^{\infty}\lambda_i$. Here, we have $\lambda_i\delta_{ij} = \langle a_i(t)a_j(t)\rangle_t$ and λ_i denotes the energy of the ith POD mode, δ_{ij} is the Kronecker symbol, and $\langle\cdots\rangle_t$ denotes the temporal average. From Fig. 4(b), we can see the energy percentage in the first POD mode also decreases with the increase of Γ. Although the first

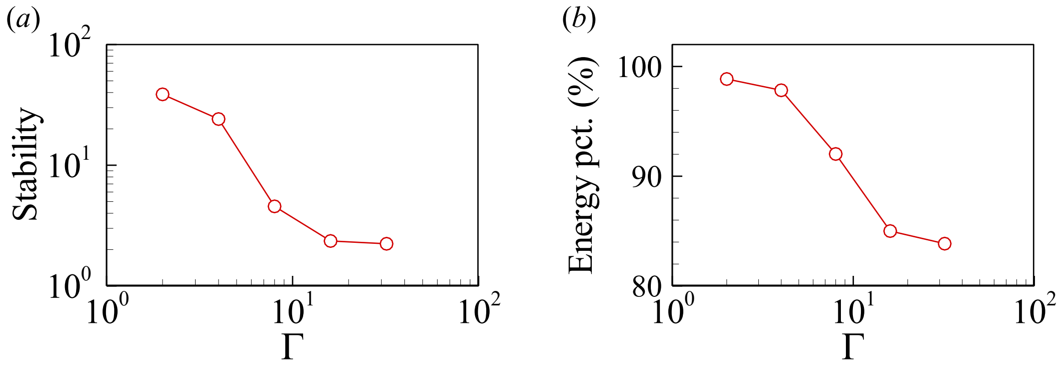

FIG. 4. (a) The stability of the first POD mode, (b) the energy contained in the first POD mode at various Γ.

mode is still the dominant flow structure (e.g., in the $\Gamma = 32$ cell, the energy contained in the first POD mode accounts for more than 83.8% of the total energy), higher-order flow modes become stronger with the increase of Γ. Thus, in a large-aspect-ratio cell, the horizontally stacked circulation rolls that form a periodical pattern are less stable, and those higher-order modes lead to a more irregular flow pattern. Because the agent was trained in the $\Gamma = 2$ cell, the above-mentioned complex flow features bring challenges for the agent to identify an energy-efficient trajectory in a larger Γ cell.

IV.2. Training the agent to migrate in the RB cell with an aspect ratio of 2

We first train the agent to migrate across the turbulent RB cell with $\Gamma = 2$. The agent starts from the point of (0, 0) at the bottom-left corner of the cell, and its goal is to reach the right-side boundary of the cell (i.e., the vertical line of $x = 2$). Here, the simple $\Gamma = 2$ cell consists of the characteristic large-scale coherent structure (i.e., a clockwise rotating primary roll and an anti-clockwise rotating primary roll), thus it serves as a paradigm environment for the learning agent. In Fig. 5, we show the instantaneous trajectories of the smart agent after training, and the corresponding video can be viewed in the supplementary movie. Initially, the agent moves upward driven by the clockwise rotating corner roll at the bottom-left corner [see Fig. 5(a)]. When the agent reaches the edge of the primary roll, which is rotating in the anti-clockwise direction, it moves along with the horizontal currents [see Fig. 5(b)], until it meets the rising thermals. The agent then rises on the thermals and ascends higher [see Fig. 5(c)]. After reaching the top layer of the cell, due to the right-directed propelling velocity, the agent moves rightward and utilizes the horizontal currents [see Fig.

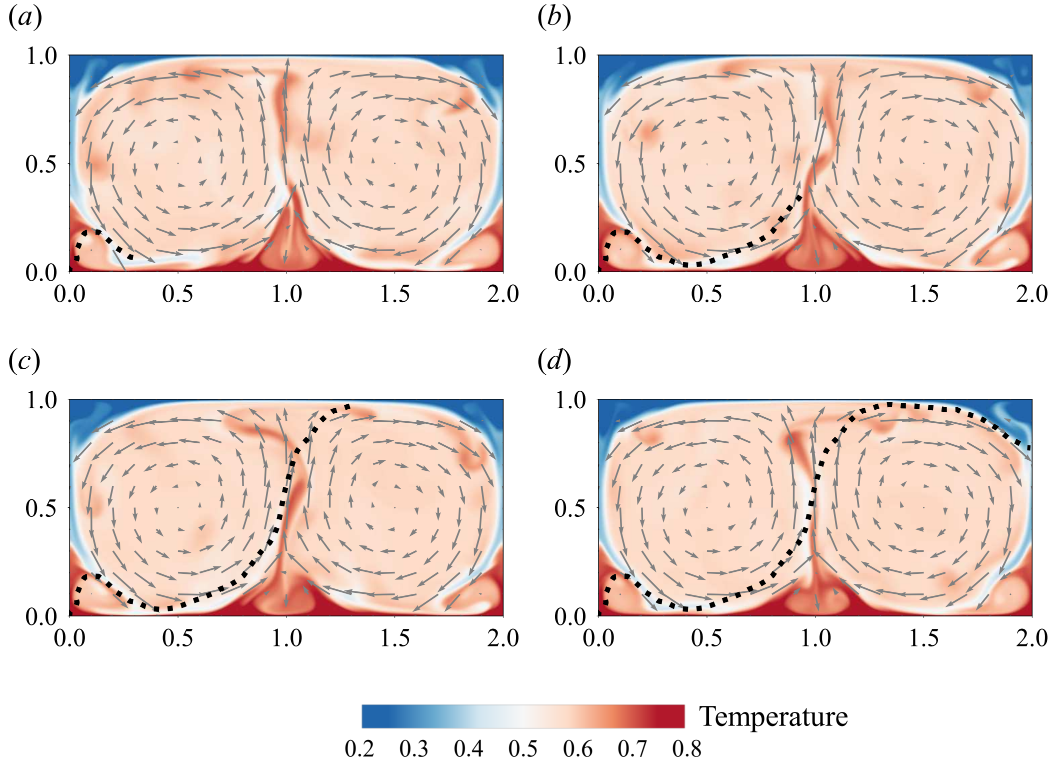

FIG. 5. Trajectory (black dotted line) of the smart agent in the RB convection at (a) $t = 18$, (b) $t = 36$, (c) $t = 54$, and (d) $t = 72$. The contour shows the typical instantaneous temperature field, and the vectors denote the velocity field of the convection.

5(d)]. The migration task is completed when the agent reaches the right-side boundary of the cell. Overall, the smart agent tries to follow the carrier currents as much as possible, and it discovers an effective policy of moving along the edge of the rolls.

A comparison between the smart agent and the naive agent is performed to highlight the differences in propelling behaviors and the savings in energy expenditure. Here, the naive agent refers to the agent that moves straight from the origin to the destination. We set the naive agent to spend the same amount of total time t_{total} as that of the smart agent to complete the migration task, and its destination point is also the same as the point where the smart agent left the right-side boundary. Thus, the velocity of the naive agent keeps a constant direction pointing from the origin to the destination, and it keeps a constant magnitude of $\|\mathbf{u}_{\text{agent}}\| = \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\|/t_{\text{total}}$. In Figs. 6(a) and 6(b), we show trajectories of the smart agent and the naive agent, respectively. From the instantaneous velocity magnitude shown on the color-coded trajectories, we can see that the naive agent generally migrates slower than the smart agent, because the naive agent travels a shorter distance during the same t_{total}. Although the smart agent migrates faster, it does not indicate that the smart agent will consume more energy, since the smart agent can utilize the carrier flow currents to save energy. Along with the trajectories, we also plot the propelling velocity vector (denoted by the red arrows) and the fluid velocity vector (denoted by the blue arrows). We calculate the correlation coefficient between the orientation of the propelling velocity vector (i.e., θ_{propel}) and the orientation of the fluid velocity vector (i.e., θ_{fluid}) as

$$C = \langle [\theta_{\text{propel}}(t) - \langle \theta_{\text{propel}} \rangle] \, [\theta_{\text{fluid}}(t) - \langle \theta_{\text{fluid}} \rangle] \rangle / \left(\sigma_{\theta_{\text{propel}}} \sigma_{\theta_{\text{fluid}}} \right) \tag{19}$$

Here, the orientation is in the range from -180° to 180°. The positive orientation is defined as anti-clockwise rotating the x-axis, and the negative orientation is defined as clockwise rotating the x-axis. For the smart agent, the resulting correlation coefficient of 0.56 suggests positive statistical relevance between them, revealing that the smart agent adjusts its migration direction in response to the changing carrier flow, enabling energy-efficient migration; for the naive agent, the resulting correlation coefficient of -0.74 implies negative statistical relevance. In addition, to quantitatively describe the angles between the propelling velocity vector and the fluid velocity vector, in Figs. 6(c) and 6(d), we plot the histogram of those angles for the smart agent and the naive agent, respectively. We can see for the smart agent, the angles are generally less than 90°, and the frequency exhibits a peak around 20°, which is another evidence that the agent tries to follow the carrier currents. For the naive agent, the frequency of those angles exhibits a peak around 120°, indicating that the naive agent has to generate propelling velocity against the carrier flow currents to keep the shortest migrating path.

We assume that only the propelling velocity $\mathbf{u}_{\text{propel}}$ affects the propelling energy consumption for the agents. Thus, the accumulative energy consumption is calculated as

$$E_{\text{propel}}(t) = \int_0^t \frac{1}{2} \| \mathbf{u}_{\text{propel}}(\tau) \|^2 d\tau \tag{20}$$

In Fig. 7(a), we plot the time series of accumulative energy consumed by the agents. After completing the migration task, the smart agent consumed around 38% of the propelling energy compared to that of the naive agent, meaning migrating in the shortest path does not always save energy. We also calculate the instantaneous energy consumption as

$$e_{\text{propel}}(t) = \frac{1}{2} \| \mathbf{u}_{\text{propel}}(t) \|^2 \tag{21}$$

From Fig. 7(b), we can see the e_{propel} for the smart agent is generally lower than that of the naive agent, and the e_{propel} keeps smaller values for the smart agent during the whole

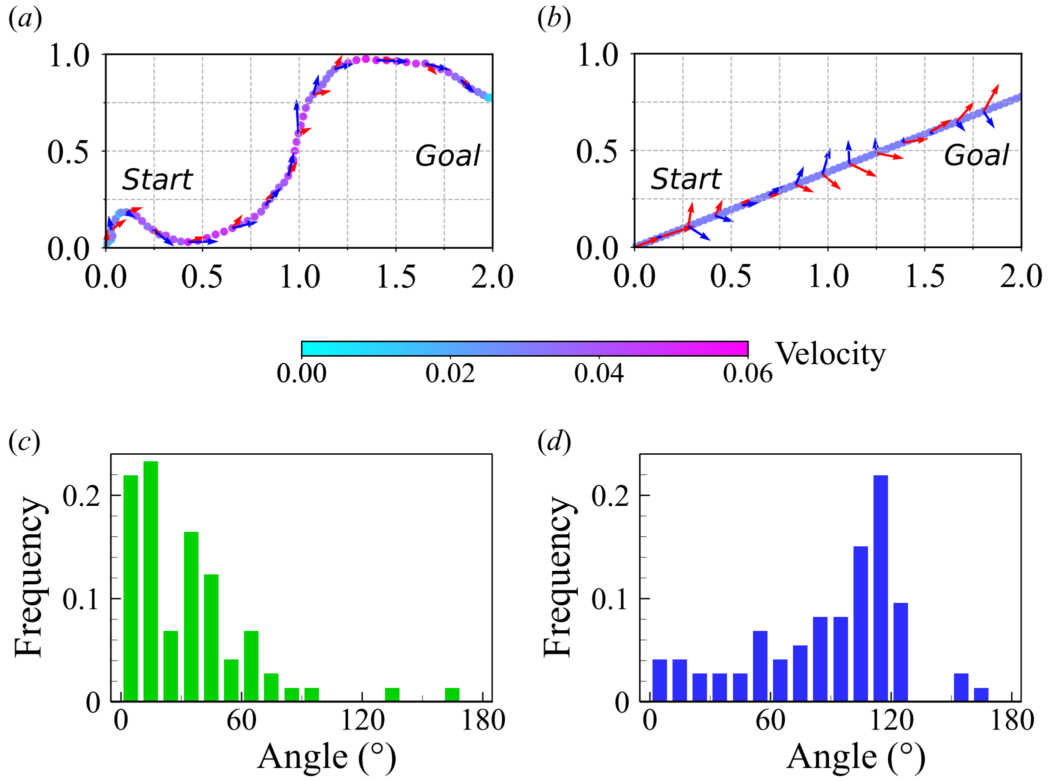

FIG. 6. $(a,\ b)$ Trajectories for the smart agent enabling energy-efficient migration and a naive agent moving straightly, respectively. The red arrows denote the propelling velocity and the blue arrows denote the fluid velocity. The trajectories are color-coded by the instantaneous velocity magnitude of the agent. $(c,\ d)$ Histogram of angles between the propelling velocity vector and the fluid velocity vector for the smart agent and the naive agent, respectively.

migration process. For the naive agent, e_{propel} exhibits a first peak around $t \approx 3$, when it moves across the edges of the left-bottom corner roll that requires a high energy barrier. At around $t \approx 23$, e_{propel} drops to the minimum, because the naive agent reaches the location where flow currents are in alignment with the migration direction. The second peak of e_{propel} appears at around $t \approx 38$, when the naive agent crosses the edge of the primary roll and drifts to the clockwise rotating primary roll. The third peak of e_{propel} appears at around $t \approx 67$, when the naive agent approaches the right-side boundary. We also compare the accumulative total kinetic energy of the agents [see Fig. 7(c)], which is calculated as

$$E_{\text{total}}(t) = \int_0^t \frac{1}{2} \|\mathbf{u}_{\text{agent}}\|^2 d\tau \tag{22}$$

After completing the migration task, the total kinetic energy of the smart agent is almost twice that of the naive agent, mostly contributed by the kinetic energy of the carrier flow.

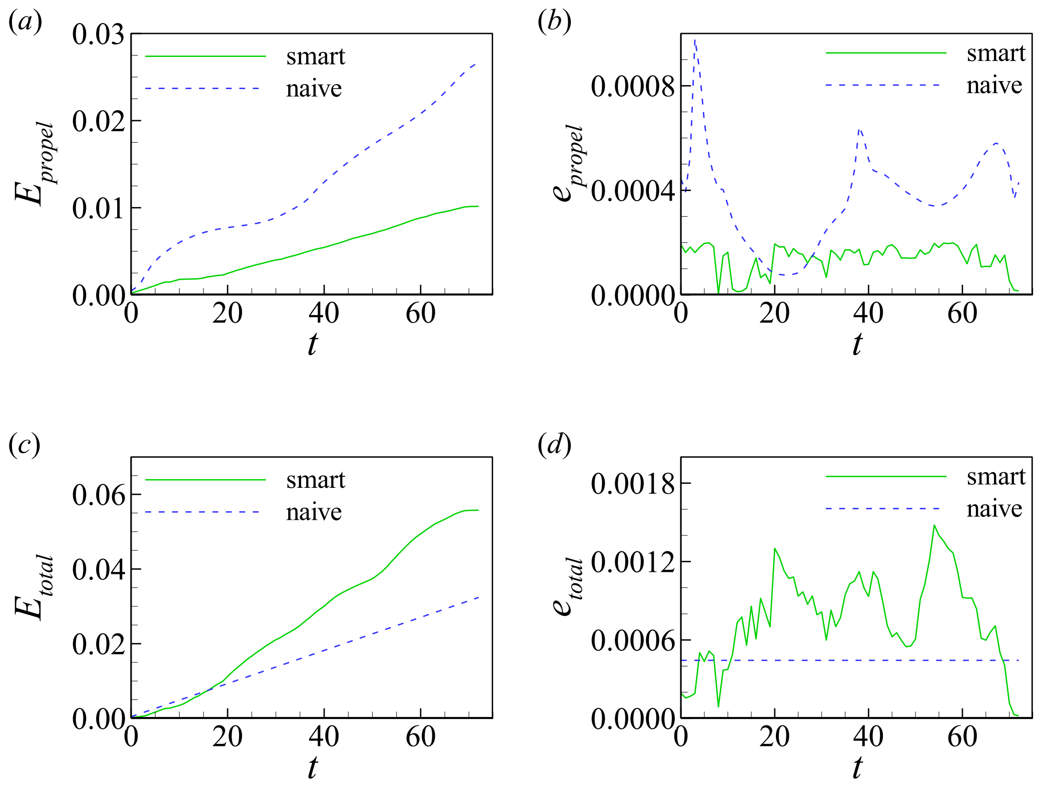

FIG. 7. Comparison of the (a) accumulative energy consumption E_{propel}, (b) instantaneous energy consumption e_{propel}, (c) accumulative total kinetic energy E_{total}, and (d) instantaneous total kinetic energy e_{total} for the smart agent and the naive agent.

Similarly, we calculate the instantaneous total kinetic energy e_{total} as

$$e_{\text{total}}(t) = \frac{1}{2}\|\mathbf{u}_{\text{agent}}(t)\|^2 \tag{23}$$

From Fig. 7(d), we can see the e_{total} of the smart agent is generally higher than that of the naive agent and keeps larger values during the whole migration process. Three peaks appear when the smart agents migrate in alignment with the flow direction, thus utilizing more kinetic energy of the carrier flow. On the other hand, the e_{total} of the naive agent keeps constant due to the constant value of $\|\mathbf{u}_{\text{agent}}(t)\|$ in the simulation settings.

Choosing appropriate environment cues that the agent can observe is crucial in the RL training. Here, we numerically determine the set of observation variables by comparing the evolution of cumulative reward during the training process. We train five different instances of each observation variable set with different random seeds, and each set performs one evaluation rollout every 1000 environment steps. The solid curves correspond to the mean and the shaded region to the minimum and maximum returns over the five trials, and it represents a moving average with a window of 20 timesteps. We first consider the agent is flow-blinded and cannot sense the surrounding flow and temperature information. The agent can have access to its position information, but only the vertical component, i.e., $s = \{y\}$. Here, we do not consider the horizontal position information (i.e., $x \notin \{s\}$); otherwise, the agent trained in the $\Gamma = 2$ cell would fail to find optimal trajectory in a larger Γ cell once its horizontal position is $x > 2$. As shown in Fig. 8(a), the agent performs poorly in the case of flow-blinded, which shows similar behavior as that of navigating through unsteady cylinder flow [55]. We then consider the agent can also sense the carrier flow velocity, i.e., $s = \{y, u, v\}$, and plot the results in Fig. 8(b). We can see that the agent performs much better, and the cumulative reward is higher than that of the flow-blinded agent. In addition, we consider the agent can sense extra vorticity information (i.e., $s = \{y, u, v, \omega\}$) [29, 56], and plot the results in Fig. 8(c). With the consideration of additional flow field information, the cumulative reward converges at an earlier time (i.e., $t \approx 0.5 \times 10^5$ for $s = \{y, u, v, \omega\}$) compared to that of velocity information (i.e., $t \approx 1.0 \times 10^5$ for $s = \{y, u, v\}$). On the other hand, in the turbulent RB convection, the temperature acts as an active scalar that influences the velocity, we next consider the agent can sense extra temperature information (i.e., $s = \{y, u, v, T\}$) [21]. As shown in Fig. 8(d), the agent outperforms previous ones and the cumulative reward converges at an earlier time and almost remains steady, suggesting temperature is an important environment cue for the agent to migrate in thermal convection. In the Appendix B, we evaluate more combinations and then choose $s = \{y, u, v, T\}$ as observation variables. On the other hand, Kubo and Shimizu [57] proposed a framework that can perform fluid flow control with partial observables. We expect the extension of that framework to turbulent flows will simplify the selection of observables.

IV.3. Testing the agent to migrate in the RB cell with a larger aspect ratio

We now apply the obtained policy to test whether the smart agent can migrate in an energy-efficient way in convection cells with larger Γ. The flow mode analysis presented in Section IV.1 indicates that in a larger Γ cell, the dominant flow modes of horizontally stacked rolls are less stable, and the energy contained in higher-order flow modes increases. Despite these challenges brought by the complex flow features, we can still obtain optimized trajectories, as shown in Fig. 9. Starting from the origin of (0, 0) point, the smart agent first escapes the corner roll at the left-bottom cell corner, it then ascends higher and rises on

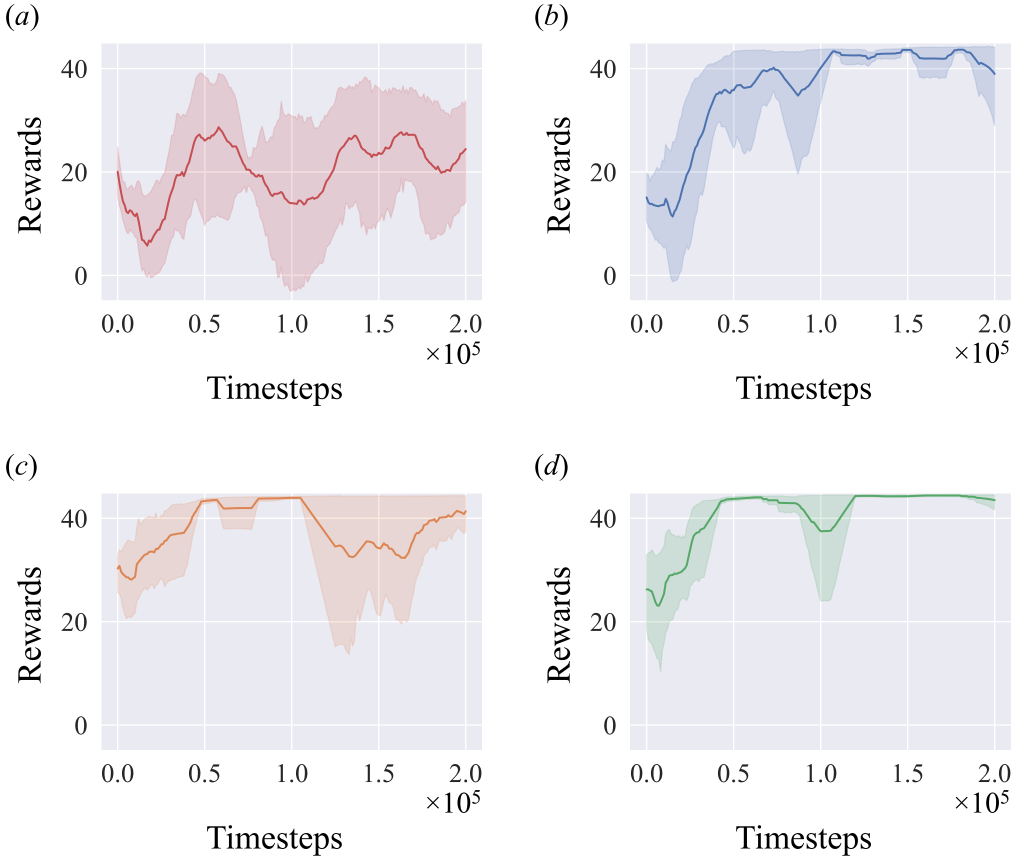

FIG. 8. Evolution of the cumulative reward during training for different combinations of observation variables: (a) $s = \{y\}$; (b) $s = \{y, u, v\}$; (c) $s = \{y, u, v, \omega\}$; (d) $s = \{y, u, v, T\}$.

the thermal (if the first primary roll is clockwise rotating), or follows the horizontal currents (if the first primary roll is anti-clockwise rotating). Afterward, the smart agent always tries to migrate along the edges of the primary rolls, where the carrier fluid flows fast and plenty of kinetic energy from the flow is available.

We then compare the propelling energy consumed by the smart agent and the naive agent in the convection cell with various Γ. In Fig. 10(a), we plot the accumulative propelling energy for the agents when they complete the migration task. Generally, for both agents, the energy consumption increases with the increase of Γ, due to longer migration distance. For the smart agent, it enables an energy-efficient migration strategy via migrating along the edges of horizontally stacked multiple primary rolls, thus we have $E_{\text{propel}} \propto \Gamma$. To quantitatively describe how much propelling energy can be saved, we plot the ratio of energy consumed by the smart agent to that of the naive agent, as shown in Fig. 10(b). We can see that in a larger Γ cell, the ratio of $E_{\text{smart}}/E_{\text{naive}}$ is smaller, meaning more propelling energy

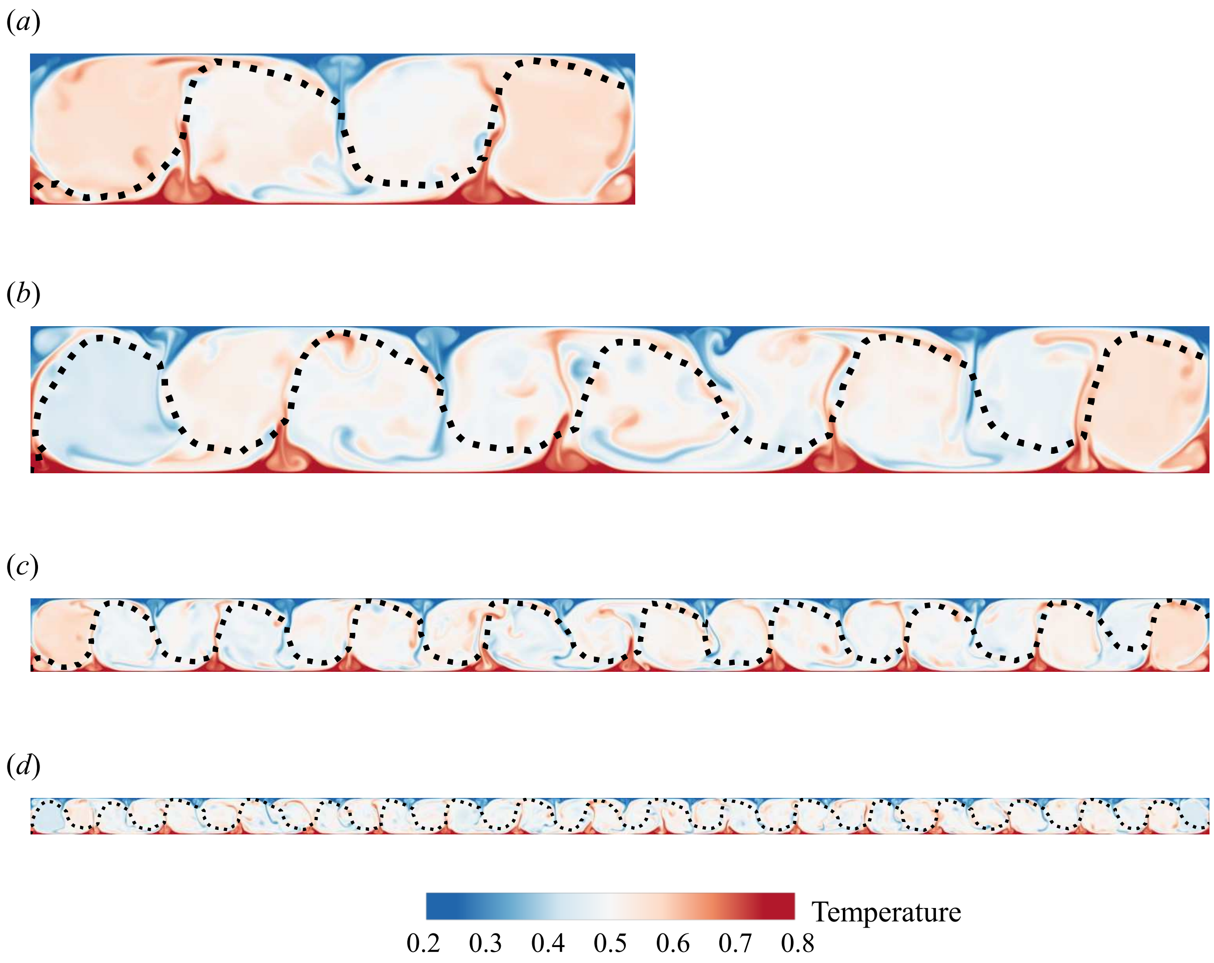

FIG. 9. Trajectory (black dotted line) of the smart agent in the convection cell with (a) $\Gamma = 4$, (b) $\Gamma = 8$, (c) $\Gamma = 16$, and (d) $\Gamma = 32$. The contour shows the typical instantaneous temperature field.

can be saved by the smart agent. The reason is that in a larger Γ cell, the naive agent has to cross more edges of the circulation rolls and overcome higher energy barriers, whilst the smart agent follows the carrier currents in an energy-efficient way.

The above results are obtained with the prescribed and fixed origin position, namely, the location at the (0, 0) point. We further test the robustness of the energy-efficient policy concerning random origin position. We first release the agent in the position where the local flow velocity is weak, i.e., $\|\mathbf{u}_{\text{fluid}}\| < 0.03$, and 100 example trajectories are plotted in Fig. 11. We can see regardless of the random origin position, the successful attempts gradually converge to a similar path line, and the agent migrates along the edges of the primary rolls. It should be noted that we restrict these 'random' positions to be $x < \Gamma/2$, which prevents the agent from being released too close to the outlet. The average success

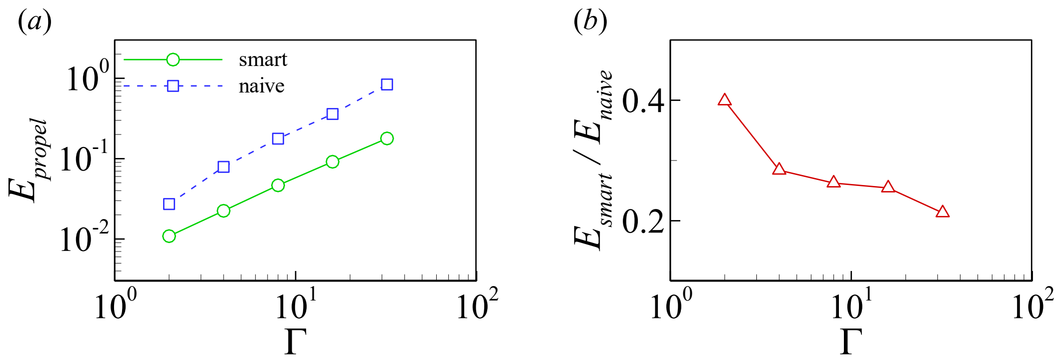

FIG. 10. (a) The propelling energy consumed by the smart agent and the naive agent, (b) the ratio of energy consumed by the smart agent to that of the naive agent as a function of Γ.

rate to complete the migration task in the $\Gamma = 4, 8, 16$, and 32 cell is 74%, 87%, 92%, and 97%, respectively. We then release the agent in the position where the local flow velocity is strong, i.e., $\|\mathbf{u}_{\text{fluid}}\| > 0.03$. The average success rate is much higher, and it is 100%, 100%, 100%, and 99% in the $\Gamma = 4, 8, 16$, and 32 cell, respectively. For the sake of clarity, we do not repeat plotting the example trajectories.

The above results indicate that the success rate to complete the migration task increases with the increase of Γ. On the other hand, in a larger Γ cell, the flow structure is more complex, and we expect it would be more challenging for the agent to complete the migration task and earn a higher success rate. To understand such behaviors, we further consider the following scenarios: (i) the agents being released where carrier flow velocity is $\|\mathbf{u}_{\text{fluid}}\| < 0.01$; (ii) the agents being released where carrier flow velocity is $0.01 < \|\mathbf{u}_{\text{fluid}}\| < 0.02$; (iii) the agents being released where carrier flow velocity is $0.02 < \|\mathbf{u}_{\text{fluid}}\| < 0.03$. We can see from Fig. 12 that in convection cells with the same Γ, the average success rate is higher if the carrier flow velocity at the origin is stronger. Because the global flow strength is stronger in a larger Γ cell (see discussion in Section IV.1), the agents are more likely to be released where carrier flow is strong. Utilizing stronger carrier flow currents, the agents can complete the migration task within a shorter time and earn a higher reward, thus the agent is more likely to repeat that action in the future. The higher sampling frequency for the agent in areas with stronger flow strength leads to higher success rate in larger Γ cell. It should be noted that such a trend is only obvious when the origin of agents possesses weak carrier flow velocity; with strong carrier flow velocity, the success rates are always near 100%.

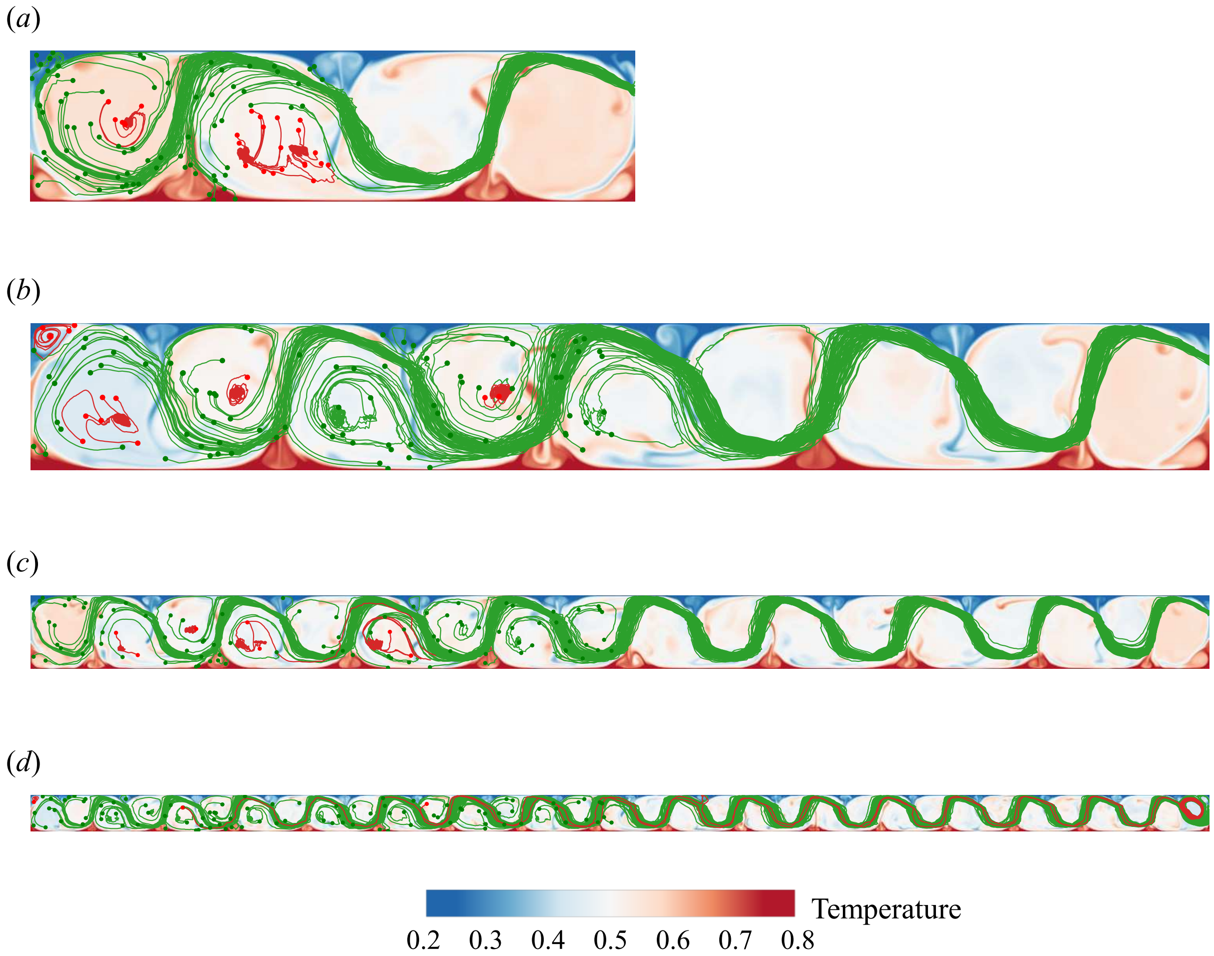

FIG. 11. Example trajectories in the (a) $\Gamma = 4$, (b) $\Gamma = 8$, (c) $\Gamma = 16$, and (d) $\Gamma = 32$ cell (the agents are released where $\|\mathbf{u}_{\text{fluid}}\| < 0.03$). Green lines represent successful attempts to complete the migration, while red lines represent unsuccessful attempts. The contour shows the typical instantaneous temperature field.

V. CONCLUSIONS

In this work, using the reinforcement learning algorithm, we performed numerical training of the self-propelling agent migrating long-distance in a thermal turbulent environment. We choose the paradigmatic turbulent RB convection cell as the flow environment, which can incorporate strong fluctuations of velocity and temperature. To build up the reinforcement learning framework, we designed a reward function that simultaneously considers the current state, energy consumption, and time consumption of the agent. We also compare the evolution of cumulative reward for different combinations of observation variables. We select the position of the agent, as well as the velocity and temperature of the carrier flow

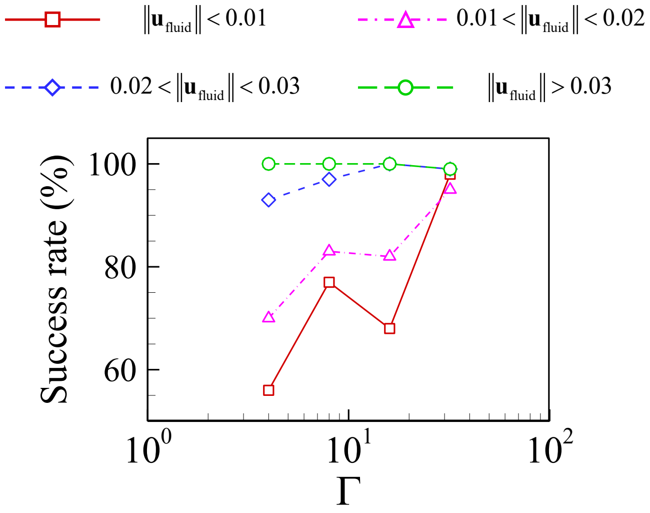

FIG. 12. Average success rate as functions of Γ when the agents are randomly released at the positions with the different local carrier flow velocities.

as appropriate environmental cues. The simulation results in a $\Gamma = 2$ RB cell showed that, compared to a naive agent that moves straight from the origin to the destination, the smart agent can learn to utilize the carrier flow currents to save propelling energy.

We then apply the optimal policy obtained in the $\Gamma = 2$ cell and test the smart agent migrating in convection cells with larger Γ. From flow mode analysis, we found the dominant flow modes in a larger Γ RB cell consist of less stable horizontally stacked rolls, and the energy contained in higher-order flow modes increases with the increase of Γ. Although these complex flow features bring challenges to optimizing the trajectories for the smart agent, we can still obtain energy-efficient migrating trajectories using the policy trained in the $\Gamma = 2$ RB cell. In addition, we found the ratio of propelling energy consumed by the smart agent to that of the naive agent decreases with the increase of Γ, meaning more propelling energy can be saved by the smart agent in a larger Γ cell. The reason is that in a larger Γ cell, the naive agent has to cross more edges of the circulation rolls and overcome higher energy barriers, whilst the smart agent always tries to follow the carrier currents as much as possible.

We also evaluate the optimized policy when the agents are being released from the randomly chosen origin, which aims to test the robustness of the learning framework. We found the success rate increases with the increase of Γ, despite the flow structures being more complex in a larger Γ cell. The main reason is that, in a larger Γ cell, the global flow strength is stronger (evident by the relationship between Re and Γ), and the agent is more likely to be released in positions where the carrier flow velocity is stronger. Utilizing stronger carrier flow velocity, the agent can complete the migration task within a shorter time and receive a higher reward, thus leading to a higher success rate. Our work has implications for long-distance migration problems, for example, the UAVs patrolling in the convective layer of the atmosphere. Migrating in energy-efficient trajectories, the UAVs can increase endurance and cover a wider range.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) through Grant Nos. 12272311 and 12125204, the National Key Project via No. GJXM92579, and the 111 project of China (No. B17037). The authors acknowledge the Beijing Beilong Super Cloud Computing Co., Ltd for providing HPC resources that have contributed to the research results reported within this paper (URL: http://www.blsc.cn/).

Appendix A: Sensitivity of the hyperparameters in the reward function

In our designed reward function (see Eqs. 14-17), we have two hyperparameters: one is ϕ, which represents the penalty when the agent is out of the flow domain; the other is ε, which is the reward scale coefficient. We tuned these two parameters separately to determine the optimal hyperparameters. It should be noted that we compared the value of the normalized reward (i.e., varied between 0 and 1) rather than the absolute value of the reward. We can see from Fig. 13 (a) that, small values of the penalty ϕ (e.g., $\phi = 1$ and 5) results in substantial degradation of performance; large values of the penalty (e.g., $\phi \geq 10$) almost lead to the same performance. As for the reward scale coefficient ε, it is almost insensitive for the investigated value and they can all give optimal policy, as shown in Fig. 13 (b). However, a large value (e.g., $\varepsilon = 100$, not shown here for clarity) would result in $\sum(r_s + r_e) > \sum r_h$ during training, and the agent's failure to explore the successful trajectory within the given time of $t_{\max}$.

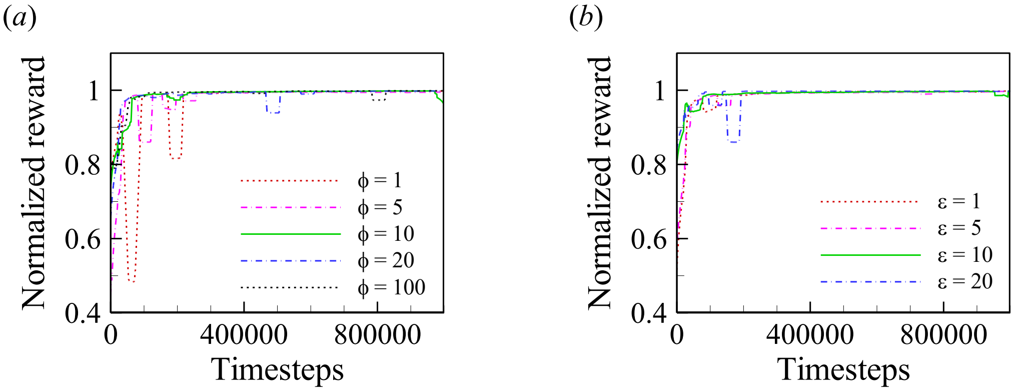

FIG. 13. Sensitivity of the hyperparameters in the reward function: (a) the penalty ϕ when the agent is out of the flow domain, (b) the reward scale coefficient ε.

Appendix B: Evaluation of different combinations of observation variables

In addition to the observation variables described in Section IV.2, we also consider the following different combinations: (i) the agent has access to position and velocity information, and it can sense strain rate [58, 59], i.e., $s = \{y, u, v, s_{xx}\}$ and $s = \{y, u, v, s_{xy}\}$. Here, $s_{xx} = \partial_x u$ and $s_{xy} = (\partial_y u + \partial_x v)/2$. We did not consider the $s_{yy} = \partial_y v$ component of the strain rate tensor, because flow continuity equation gives $s_{xx} + s_{yy} = 0$ in 2D flows, which means s_{xx} and s_{yy} are negatively correlated. (ii) the agent has access to position and velocity information, and it can sense temperature gradient, i.e., $s = \{y, u, v, (\nabla T)_x\}$ and $s = \{y, u, v, (\nabla T)_y\}$. Here, $(\nabla T)_x$ essentially represents the vorticity produced by buoyancy in the 2D convection flow [60]. (iii) the agent has access to position, velocity and temperature information, and it can sense additional vorticity, strain rate, or temperature gradient, i.e., $s = \{y, u, v, T, s_{xx}\}$, $s = \{y, u, v, T, s_{xy}\}$, $s = \{y, u, v, T, \omega\}$, $s = \{y, u, v, T, (\nabla T)_x\}$, and $s = \{y, u, v, T, (\nabla T)_y\}$. In Fig. 14, we plot the evolution of the cumulative reward during training for the above nine combinations of observation variables. We can see these combinations only slightly changes the converging speed of the training, not the asymptotic accumulative reward value. Among them, the $s = \{y, u, v, T, (\nabla T)_x\}$ shown in Fig. 14(h) outperforms other combinations. In practical applications, velocity or temperature sensing could be implemented via a variety of methods, such as pitot tubes, hot wire, and so on; while vorticity, shear strain component, and temperature gradient should be computed from several velocities or temperature sensors, which increases the complexity that the agent has to sense. Thus, as described in Section IV.2, we deliberately keep simple the environmental

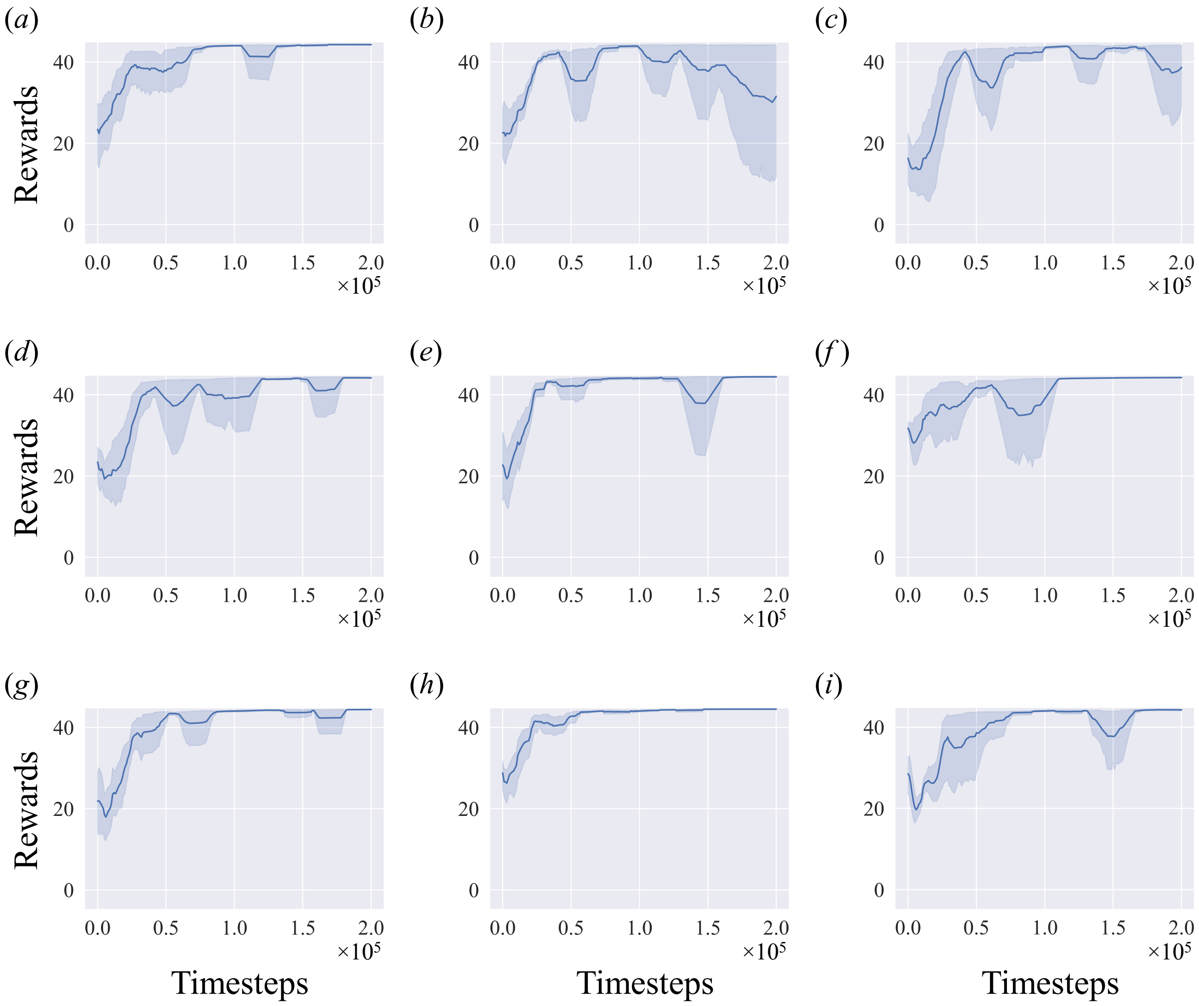

FIG. 14. Evolution of the cumulative reward during training for different combinations of observation variables: (a) $s = \{y, u, v, s_{xx}\}$; (b) $s = \{y, u, v, s_{xy}\}$; (c) $s = \{y, u, v, (\nabla T)_x\}$; (d) $s = \{y, u, v, (\nabla T)_y\}$; (e) $s = \{y, u, v, T, s_{xx}\}$; (f) $s = \{y, u, v, T, s_{xy}\}$; (g) $s = \{y, u, v, T, \omega\}$; (h) $s = \{y, u, v, T, (\nabla T)_x\}$; (i) $s = \{y, u, v, T, (\nabla T)_y\}$.

cues of local information $s = \{y, u, v, T\}$ that the agent can see to guide its migration, such that the amount of data storage by the agent can be reduced in practical applications.

[1] H. Weimerskirch, C. Bishop, T. Jeanniard-du Dot, A. Prudor, and G. Sachs, Frigate birds track atmospheric conditions over months-long transoceanic flights, Science **353**, 74 (2016).

[2] H. J. Williams, E. Shepard, M. D. Holton, P. Alarcón, R. Wilson, and S. Lambertucci, Physical limits of flight performance in the heaviest soaring bird, Proc. Natl. Acad. Sci. **117**, 17884

(2020).

[3] J. P. Croxall, J. R. Silk, R. A. Phillips, V. Afanasyev, and D. R. Briggs, Global circumnavigations: tracking year-round ranges of nonbreeding albatrosses, Science **307**, 249 (2005).

[4] I. Lancaster, The problem of the soaring bird, Am. Nat. **19**, 1055 (1885).

[5] P. B. MacCready, Optimum airspeed selector, Soaring **10**, 10 (1958).

[6] M. Allen, Autonomous soaring for improved endurance of a small uninhabitated air vehicle, in *43rd AIAA Aerospace Sciences Meeting and Exhibit* (2005) p. 1025.

[7] R. Bencatel, J. T. de Sousa, and A. Girard, Atmospheric flow field models applicable for aircraft endurance extension, Prog. Aeosp. Sci. **61**, 1 (2013).

[8] M. Allen, Updraft model for development of autonomous soaring uninhabited air vehicles, in *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006) p. 1510.

[9] N. Lawrance and S. Sukkarieh, Wind energy based path planning for a small gliding unmanned aerial vehicle, in *AIAA Guidance, Navigation, and Control Conference* (2009) p. 6112.

[10] Z. Ákos, M. Nagy, S. Leven, and T. Vicsek, Thermal soaring flight of birds and unmanned aerial vehicles, Bioinspir. Biomim. **5**, 045003 (2010).

[11] K. M. Laurent, B. Fogg, T. Ginsburg, C. Halverson, M. J. Lanzone, T. A. Miller, D. W. Winkler, and G. P. Bewley, Turbulence explains the accelerations of an eagle in natural flight, Proc. Natl. Acad. Sci. **118**, e2102588118 (2021).

[12] G. Ahlers, S. Grossmann, and D. Lohse, Heat transfer and large scale dynamics in turbulent Rayleigh-Bénard convection, Rev. Mod. Phys. **81**, 503 (2009).

[13] F. Chillà and J. Schumacher, New perspectives in turbulent Rayleigh-Bénard convection, Eur. Phys. J. E **35**, 1 (2012).

[14] K.-Q. Xia, Current trends and future directions in turbulent thermal convection, Theor. Appl. Mech. Lett. **3**, 052001 (2013).

[15] B. Atkinson and J. Wu Zhang, Mesoscale shallow convection in the atmosphere, Rev. Geophys. **34**, 403 (1996).

[16] R. J. Stevens, A. Blass, X. Zhu, R. Verzicco, and D. Lohse, Turbulent thermal superstructures in Rayleigh-Bénard convection, Phys. Rev. Fluids **3**, 041501(R) (2018).

[17] A. Pandey, J. D. Scheel, and J. Schumacher, Turbulent superstructures in Rayleigh-Bénard convection, Nat. Commun. **9**, 1 (2018).

[18] G. Reddy, A. Celani, T. J. Sejnowski, and M. Vergassola, Learning to soar in turbulent environments, Proc. Natl. Acad. Sci. **113**, E4877 (2016).

[19] Z. Ákos, M. Nagy, and T. Vicsek, Comparing bird and human soaring strategies, Proc. Natl. Acad. Sci. **105**, 4139 (2008).

[20] G. Reddy, J. Wong-Ng, A. Celani, T. J. Sejnowski, and M. Vergassola, Glider soaring via reinforcement learning in the field, Nature **562**, 236 (2018).

[21] A. Xu, H.-L. Wu, and H.-D. Xi, Migration of self-propelling agent in a turbulent environment with minimal energy consumption, Phys. Fluids **34**, 035117 (2022).

[22] A. Xu, W. Shyy, and T. Zhao, Lattice-Boltzmann modeling of transport phenomena in fuel cells and flow batteries, Acta Mech. Sin. **33**, 555 (2017).

[23] Z. Guo and C. Zheng, Analysis of Lattice-Boltzmann equation for microscale gas flows: relaxation times, boundary conditions and the Knudsen layer, Int. J. Comput. Fluid Dyn. **22**, 465 (2008).

[24] A. Xu, L. Shi, and T. Zhao, Accelerated Lattice-Boltzmann simulation using GPU and OpenACC with data management, Int. J. Heat Mass Transf. **109**, 577 (2017).

[25] A. Xu, L. Shi, and H.-D. Xi, Lattice-Boltzmann simulations of three-dimensional thermal convective flows at high Rayleigh number, Int. J. Heat Mass Transf. **140**, 359 (2019).

[26] A. Xu and B.-T. Li, Multi-GPU thermal lattice Boltzmann simulations using OpenACC and MPI, Int. J. Heat Mass Transf. **201**, 123649 (2023).

[27] B. Castaing, G. Gunaratne, F. Heslot, L. Kadanoff, A. Libchaber, S. Thomae, X.-Z. Wu, S. Zaleski, and G. Zanetti, Scaling of hard thermal turbulence in Rayleigh-Bénard convection, J. Fluid Mech. **204**, 1 (1989).

[28] K. Krishna, Z. Song, and S. L. Brunton, Finite-horizon, energy-efficient trajectories in unsteady flows, Proc. R. Soc. A-Math. Phys. Eng. Sci. **478**, 20210255 (2022).

[29] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Flow navigation by smart microswimmers via reinforcement learning, Phys. Rev. Lett. **118**, 158004 (2017).

[30] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Smart inertial particles, Phys. Rev. Fluids **3**, 084301 (2018).

[31] E. Schneider and H. Stark, Optimal steering of a smart active particle, EPL **127**, 64003 (2019).

[32] J. K. Alageshan, A. K. Verma, J. Bec, and R. Pandit, Machine learning strategies for path-planning microswimmers in turbulent flows, Phys. Rev. E **101**, 043110 (2020).

[33] F. Borra, L. Biferale, M. Cencini, and A. Celani, Reinforcement learning for pursuit and evasion of microswimmers at low Reynolds number, Phys. Rev. Fluids **7**, 023103 (2022).

[34] G. Novati, L. Mahadevan, and P. Koumoutsakos, Controlled gliding and perching through deep-reinforcement-learning, Phys. Rev. Fluids **4**, 093902 (2019).

[35] Z. Zou, Y. Liu, Y.-N. Young, O. S. Pak, and A. C. Tsang, Gait switching and targeted navigation of microswimmers via deep reinforcement learning, Commun. Phys. **5**, 1 (2022).

[36] L. Biferale, F. Bonaccorso, M. Buzzicotti, P. Clark Di Leoni, and K. Gustavsson, Zermelo's problem: optimal point-to-point navigation in 2D turbulent flows using reinforcement learning, Chaos **29**, 103138 (2019).

[37] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem, A review on deep reinforcement learning for fluid mechanics, Comput. Fluids **225**, 104973 (2021).

[38] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT Press, 2018).

[39] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep.-Rev. Sec. Phys. Lett. **810**, 1 (2019).

[40] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. **52**, 477 (2020).

[41] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, Machine learning for active matter, Nat. Mach. Intell. **2**, 94 (2020).

[42] 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).

[43] S. Muiños-Landin, A. Fischer, V. Holubec, and F. Cichos, Reinforcement learning with artificial microswimmers, Sci. Robot. **6**, eabd9285 (2021).

[44] P. A. Monderkamp, F. J. Schwarzendahl, M. A. Klatt, and H. Löwen, Active particles using reinforcement learning to navigate in complex motility landscapes, Mach. Learn.-Sci. Technol. **3**, 045024 (2022).

[45] M. Gazzola, A. A. Tchieu, D. Alexeev, A. de Brauer, and P. Koumoutsakos, Learning to school in the presence of hydrodynamic interactions, J. Fluid Mech. **789**, 726 (2016).

[46] S. Verma, G. Novati, and P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, Proc. Natl. Acad. Sci. **115**, 5849 (2018).

[47] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *International conference on machine learning* (PMLR, 2018) pp. 1861–1870.

[48] Q. Wang, R. Verzicco, D. Lohse, and O. Shishkina, Multiple states in turbulent large-aspect-ratio thermal convection: What determines the number of convection rolls?, Phys. Rev. Lett. **125**, 074501 (2020).

[49] A. Xu, X. Chen, F. Wang, and H.-D. Xi, Correlation of internal flow structure with heat transfer efficiency in turbulent Rayleigh-Bénard convection, Phys. Fluids **32**, 105112 (2020).

[50] N. Halko, P.-G. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. **53**, 217 (2011).

[51] Y. Zhang, Q. Zhou, and C. Sun, Statistics of kinetic and thermal energy dissipation rates in two-dimensional turbulent Rayleigh–Bénard convection, J. Fluid Mech. **814**, 165 (2017).

[52] X. Zhu, V. Mathai, R. J. Stevens, R. Verzicco, and D. Lohse, Transition to the ultimate regime in two-dimensional Rayleigh-Bénard convection, Phys. Rev. Lett. **120**, 144502 (2018).

[53] B.-F. Wang, Q. Zhou, and C. Sun, Vibration-induced boundary-layer destabilization achieves massive heat-transport enhancement, Sci. Adv. **6**, eaaz8239 (2020).

[54] X. Chen, S.-D. Huang, K.-Q. Xia, and H.-D. Xi, Emergence of substructures inside the large-scale circulation induces transition in flow reversals in turbulent thermal convection, J. Fluid Mech. **877** (2019).

[55] P. Gunnarson, I. Mandralis, G. Novati, P. Koumoutsakos, and J. O. Dabiri, Learning efficient navigation in vortical flow fields, Nat. Commun. **12**, 1 (2021).

[56] 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**, 1 (2017).

[57] A. Kubo and M. Shimizu, Efficient reinforcement learning with partial observables for fluid flow control, Phys. Rev. E **105**, 065101 (2022).

[58] J. Qiu, N. Mousavi, K. Gustavsson, C. Xu, B. Mehlig, and L. Zhao, Navigation of microswimmers in steady flow: The importance of symmetries, J. Fluid Mech. **932** (2022).

[59] J. Qiu, N. Mousavi, L. Zhao, and K. Gustavsson, Active gyrotactic stability of microswimmers using hydromechanical signals, Phys. Rev. Fluids **7**, 014311 (2022).

[60] A. Xu, B.-R. Xu, L.-S. Jiang, and H.-D. Xi, Production and transport of vorticity in two-dimensional Rayleigh-Bénard convection cell, Phys. Fluids **34**, 013609 (2022).