Machine learning strategies for path-planning microswimmers in turbulent flows

Jaya Kumar Alageshan,[1,*] Akhilesh Kumar Verma,[1,†] Jérémie Bec,[2,‡] and Rahul Pandit[1,§]
[1]*Centre for Condensed Matter Physics, Department of Physics, Indian Institute of Science, Bangalore 560012, India*
[2]*MINES ParisTech, PSL Research University, CNRS, CEMEF, CS 10207, 06904 Sophia-Antipolis, France*

(Received 25 November 2019; accepted 30 March 2020; published 27 April 2020)

We develop an *adversarial-reinforcement* learning scheme for microswimmers in statistically homogeneous and isotropic turbulent fluid flows, in both two and three dimensions. We show that this scheme allows microswimmers to find nontrivial paths, which enable them to reach a target on average in less time than a naïve microswimmer, which tries, at any instant of time and at a given position in space, to swim in the direction of the target. We use pseudospectral direct numerical simulations of the two- and three-dimensional (incompressible) Navier-Stokes equations to obtain the turbulent flows. We then introduce passive microswimmers that try to swim along a given direction in these flows; the microswimmers do not affect the flow, but they are advected by it. Two nondimensional control parameters play important roles in our learning scheme: (a) the ratio $\tilde{V}_s$ of the microswimmer's bare velocity V_s and the root-mean-square (rms) velocity $u_{\rm rms}$ of the turbulent fluid and (b) the product $\tilde{B}$ of the microswimmer-response time B and the rms vorticity $\omega_{\rm rms}$ of the fluid. We show that the average time required for the microswimmers to reach the target, by using our adversarial-reinforcement learning scheme, eventually reduces below the average time taken by microswimmers that follow the naïve strategy.

DOI: 10.1103/PhysRevE.101.043110

I. INTRODUCTION

Machine learning techniques and advances in computational facilities have led to significant improvements in obtaining solutions to optimization problems, e.g., to problems in path planning and optimal transport, referred to in control systems as Zermelo's navigation problem [1]. With vast amounts of data available from experiments and simulations in fluid dynamics, machine learning techniques are being used to extract information that is useful to control and optimize flows [2]. Recent studies include the use of reinforcement learning, in fluid-flow settings, e.g., (a) to optimize the soaring of a glider in thermal currents [3] and (b) in the development of an optimal scheme in two-dimensional (2D) and three-dimensional (3D) fluid flows that are time independent [4,5]. Optimal locomotion, in response to stimuli, is also important in biological systems ranging from cells and microorganisms [6–8] to birds, animals, and fish [9]; such locomotion is often termed *taxis* [10].

It behooves us, therefore, to explore machine learning strategies for optimal path planning by microswimmers in turbulent fluid flows. We initiate such a study for microswimmers in 2D and 3D turbulent flows. In particular, we consider a dynamic-path-planning problem that seeks to minimize the average time taken by microswimmers to reach a given target, while moving in a turbulent fluid flow that is statistically homogeneous and isotropic. We develop a multiswimmer, *adversarial $\mathcal{Q}$-learning* algorithm to optimize the motion of such microswimmers that try to swim towards a specified target (or targets). Our adversarial $\mathcal{Q}$-learning approach ensures that the microswimmers perform at least as well as those that adopt the following naïve strategy: at any instant of time and at a given position in space, a naïve microswimmer tries to point in the direction of the target. We examine the efficacy of this approach as a function of the following two dimensionless control parameters: (a) $\tilde{V}_s = V_s/u_{\rm rms}$, where the microswimmer's bare velocity is V_s and the turbulent fluid has the root-mean-square velocity $u_{\rm rms}$, and (b) $\tilde{B} = B\omega_{\rm rms}$, where B is the microswimmer-response time and $\omega_{\rm rms}$ the rms vorticity of the fluid. We show, by extensive direct numerical simulations (DNSs), that the average time $\langle T \rangle$ required by a microswimmer to reach a target at a fixed distance is lower if it uses our adversarial $\mathcal{Q}$-learning scheme than if it uses the naïve strategy.

II. BACKGROUND FLOW AND MICROSWIMMER DYNAMICS

For the low-Mach-number flows we consider, the fluid-flow velocity $\mathbf{u}$ satisfies the incompressible Navier-Stokes (NS) equation. In two dimensions, we write the NS equations in the conventional vorticity-stream-function form, which accounts for incompressibility in two dimensions [11]:

$$(\partial_t + \mathbf{u}\cdot\nabla)\omega = \nu\nabla^2\omega - \alpha\,\omega + F_\omega. \tag{1}$$

Here, $\mathbf{u} \equiv (u_x, u_y)$ is the fluid velocity, ν is the kinematic viscosity, α is the coefficient of friction (present in two dimensions, e.g., because of air drag or bottom friction), and the vorticity $\omega = (\nabla \times \mathbf{u})$, which is normal to $\mathbf{u}$ in two

*jayaka@iisc.ac.in
†akhilesh@iisc.ac.in
‡jeremie.bec@mines-paristech.fr
§rahul@iisc.ac.in; also at Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore 560064, India.

dimensions. The 3D incompressible NS equations are

$$(\partial_t + \mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p/\rho + \mathbf{f} + \nu\nabla^2\mathbf{u}, \quad \nabla\cdot\mathbf{u} = 0, \quad (2)$$

where p is the pressure, and the density ρ of the incompressible fluid is taken to be 1; the large-scale forcing F_ω (large-scale random forcing in two dimensions) or $\mathbf{f}$ (constant energy injection in three dimensions) maintains the statistically steady, homogeneous, and isotropic turbulence, for which it is natural to use periodic boundary conditions.

We consider a collection of $\mathcal{N}_p$ passive, noninteracting microswimmers in the turbulent flow; $\mathbf{X}_i$ and $\hat{\mathbf{p}}_i$ are the position and swimming direction of the microswimmer. Each microswimmer is assigned a target located at $\mathbf{X}_i^T$. We are interested in minimizing the time $\mathcal{T}$ required by a microswimmer, which is released at a distance $r_0 = |\mathbf{X}_i(0) - \mathbf{X}_i^T|$ from its target, to approach within a small distance $r = |\mathbf{X}_i(\mathcal{T}) - \mathbf{X}_i^T| \ll r_0$ of this target. The microswimmer's position and swimming direction evolve as follows [12]:

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{u}(\mathbf{X}_i, t) + V_s\,\hat{\mathbf{p}}_i, \quad (3)$$

$$\frac{d\hat{\mathbf{p}}_i}{dt} = \frac{1}{2B}[\hat{\mathbf{o}}_i - (\hat{\mathbf{o}}_i.\hat{\mathbf{p}}_i)\,\hat{\mathbf{p}}_i] + \frac{1}{2}\omega\times\hat{\mathbf{p}}_i. \quad (4)$$

Here, we use bilinear (trilinear) interpolation in two (three) dimensions to determine the fluid velocity $\mathbf{u}$ at the microswimmer's position $\mathbf{X}_i$ from Eq. (2); $V_s\hat{\mathbf{p}}_i$ is the swimming velocity, B is the time scale associated with the microswimmer to align with the flow, and $\hat{\mathbf{o}}_i$ is the control direction. Equation (4) implies that $\hat{\mathbf{p}}_i$ tries to align along $\hat{\mathbf{o}}_i$. We define the following nondimensional control parameters: $\tilde{V}_s = V_s/u_{\rm rms}$, where $u_{\rm rms} = \langle|\mathbf{u}|^2\rangle^{1/2}$ is the root-mean-square (rms) fluid flow velocity, and $\tilde{B} = B/\tau_\Omega$, where $\tau_\Omega = \omega_{\rm rms}^{-1}$; $\omega_{\rm rms} = \langle|\omega|^2\rangle^{1/2}$ denotes the root-mean-square vorticity.

III. ADVERSARIAL $\mathcal{Q}$-LEARNING FOR SMART MICROSWIMMERS

Designing a strategy consists in choosing appropriately the control direction $\hat{\mathbf{o}}_i$, as a function of the instantaneous state of the microswimmer, in order to minimize the mean arrival time $\langle\mathcal{T}\rangle$. To develop a *tractable* framework for $\mathcal{Q}$ learning, we use a *finite number of states* by discretizing the fluid vorticity ω at the microswimmer's location into three ranges of values labeled by $\mathcal{S}_\omega$ and the angle θ_i, between $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{T}}_i$, into four ranges $\mathcal{S}_\theta$, as shown in Fig. 1. The choice of $\hat{\mathbf{o}}_i$ is then reduced to a map from $(\mathcal{S}_\omega, \mathcal{S}_\theta)$ to an *action* set, $\mathcal{A}$, which we also discretize into the following four possible actions: $\mathcal{A} := \{\hat{\mathbf{T}}_i, -\hat{\mathbf{T}}_i, \hat{\mathbf{T}}_{i\perp}, -\hat{\mathbf{T}}_{i\perp}\}$, where $\hat{\mathbf{T}}_i = (\mathbf{X}_i^T - \mathbf{X}_i)/|\mathbf{X}_i^T - \mathbf{X}_i|$ is the unit vector pointing from the swimmer to its target and $(\hat{\mathbf{T}}_{i\perp}\cdot\hat{\mathbf{T}}_i) = 0$. Therefore, for the naïve strategy $\hat{\mathbf{o}}_i(s_i) \equiv \hat{\mathbf{T}}_i$, $\forall\, s_i \in (\mathcal{S}_\omega, \mathcal{S}_\theta)$. This strategy is optimal if $\tilde{V}_s \gg 1$: Microswimmers have an almost ballistic dynamics and move swiftly to the target. For $\tilde{V}_s \simeq 1$, vortices affect the microswimmers substantially, so we have to develop a nontrivial $\mathcal{Q}$-learning strategy, in which $\hat{\mathbf{o}}_i$ is a function of $\omega(\mathbf{X}_i, t)$ and θ_i.

In our $\mathcal{Q}$-learning scheme, we assign a *quality* value to each state-action binary relation of microswimmer i as follows: $\mathcal{Q}_i : (s_i, a_i) \to \mathbb{R}$, where $s_i \in (\mathcal{S}_\omega, \mathcal{S}_\theta)$ and $a_i \in \mathcal{A}$; and we use the ϵ-greedy method [13] (with parameter ϵ_g),

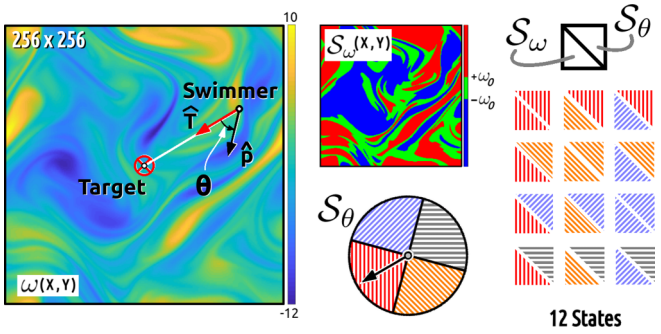

FIG. 1. Left: A pseudocolor plot of the vorticity field, with a microswimmer represented by a small white circle; the black arrow on the microswimmer indicates its swimming direction, $\hat{\mathbf{p}}$, the red arrow represents the direction towards the target, $\hat{\mathbf{T}}$, and θ is the angle between $\hat{\mathbf{p}}$ and $\hat{\mathbf{T}}$. Top center: The discretized vorticity states (red |||, $\omega > \omega_0$; green \\\, $-\omega_0 \leqslant \omega \leqslant \omega_0$; blue ///, $\omega < -\omega_0$). In our approach we use $\omega_0 = \omega_{rms}$. Bottom center: The color code for the discretized θ (red |||, $-\pi/4 \leqslant \theta < \pi/4$; orange \\\, $\pi/4 \leqslant \theta < 3\pi/4$; blue ///, $-3\pi/4 \leqslant \theta < -\pi/4$; gray ≡, $3\pi/4 \leqslant \theta < 5\pi/4$). Right: All possible discrete states of the microswimmers denoted by colored squares where the lower half stands for the vorticity state, $\mathcal{S}_\omega$, and the upper half represents the direction state, $\mathcal{S}_\theta$.

in which the control direction is chosen from the probability distribution $\mathcal{P}[\hat{\mathbf{o}}_i(s_i)] = \epsilon_g/4 + (1-\epsilon_g)\,\delta(\hat{\mathbf{o}}_i(s_i) - \hat{\mathbf{o}}_{\rm max})$, where $\hat{\mathbf{o}}_{\rm max} := {\rm argmax}_{a\in\mathcal{A}}\,\mathcal{Q}_i(s_i, a)$ and $\delta(\cdot)$ is the Dirac delta function. At each iteration, $\hat{\mathbf{o}}_i$ is calculated as above and the microswimmer evolution is performed by using Eqs. (3) and (4). In the canonical $\mathcal{Q}$-learning approach, during the learning process, each of the $\mathcal{Q}_i$'s are evolved by using the Bellman equation [14] below, whenever there is a state change, i.e., $s_i(t) \neq s_i(t+\delta t)$:

$$\mathcal{Q}_i(s_i(t), \hat{\mathbf{o}}_i(s_i(t))) \mapsto (1-\lambda)\,\mathcal{Q}_i(s_i(t), \hat{\mathbf{o}}_i(s_i(t))) + \lambda\,[\mathcal{R}_i(t) + \gamma\,\max_{a\in\mathcal{A}}\mathcal{Q}_i(s_i(t+\delta t), a)], \quad (5)$$

where λ and γ are learning parameters that are set to optimal values after some numerical exploration (see Table I), and $\mathcal{R}_i$ is the reward function. For the path-planning problem we define $\mathcal{R}_i(t) = |\mathbf{X}_i(t - n\,\delta t) - \mathbf{X}_i^T| - |\mathbf{X}_i(t) - \mathbf{X}_i^T|$, where $n = \min_{l\in\mathbb{N}}\{s_i(t - l\,\delta t) \neq s_i(t)\}$. According to Eq. (5), any $\hat{\mathbf{o}}_i$ for which $\mathcal{R}_i$ is positive can be a solution, and there exist many such solutions that are suboptimal compared to the naïve strategy.

To reduce the solution space, we propose an *adversarial* scheme: Each microswimmer, the *master*, is accompanied by a *slave* microswimmer, with position $\mathbf{X}_i^{Sl}(t)$, that shares the same target at $\mathbf{X}_i^T$ and follows the naïve strategy, i.e.,

TABLE I. List of learning parameter values: γ is the earning discount, λ is the learning rate, ϵ_g is the ϵ-greedy algorithm parameter that represents the probability with which the nonoptimal action is chosen, ω_0 is the cutoff used for defining S_ω, and $\omega_{\rm rms}$ is the rms value of ω.

$\gamma = 0.99$	$\lambda = 0.01$
$\epsilon_g = 0.001$	$\omega_0/\omega_{\rm rms} = 1.0$

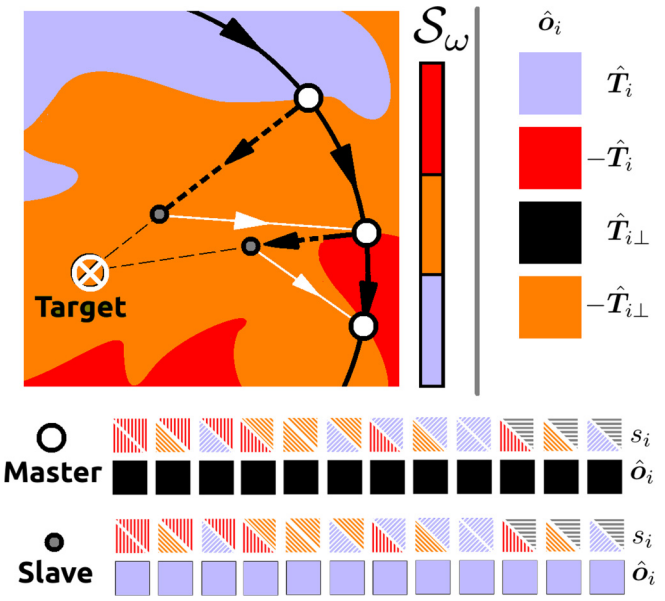

FIG. 2. Topleft: A schematic diagram illustrating the trajectories of master (black line) and slave (dashed black line) microswimmers superimposed on a pseudocolor plot of the two-dimensional (2D) discrete vorticity field $\mathcal{S}_\omega$; the master undergoes a state change at the points shown by solid white circles; white arrows indicate the resetting of the slave's trajectory. Top right: Color code for the control direction $\hat{o}_i$; for the states $s_i \in (\mathcal{S}_\omega, \mathcal{S}_\theta)$ see Fig. 1. Bottom: Control maps for the master and slave; for the purpose of illustration, we use $\hat{\mathbf{o}}_i = \hat{\mathbf{T}}_{i\perp}$ for the master; for $\tilde{V}_s \gg 1$ and $\tilde{B} = 0$, this leads to the circular path shown in our schematic diagram.

$\hat{\mathbf{o}}_i^{Sl}(t) \equiv \hat{\mathbf{T}}_i^{Sl} = (\mathbf{X}_i^{Sl} - \mathbf{X}_i^T)/|\mathbf{X}_i^{Sl} - \mathbf{X}_i^T|$. Now, whenever the master undergoes a state change, the corresponding slave's position and direction are reinitialized to that of the master; i.e., if $s_i(t) \neq s_i(t + \delta t)$, then $\mathbf{X}_i^{Sl}(t + \delta t) = \mathbf{X}_i(t + \delta t)$ and $\hat{\mathbf{p}}_i^{Sl}(t + \delta t) = \hat{\mathbf{p}}_i(t + \delta t)$ (see Fig. 2). Then the reward function for the master microswimmer is given by $\mathcal{R}_i^{AD}(t) = |\mathbf{X}_i^{Sl}(t) - \mathbf{X}_i^T| - |\mathbf{X}_i(t) - \mathbf{X}_i^T|$; i.e., only those changes that improve on the naïve strategy are favored.

In the conventional $\mathcal{Q}$-learning approach [13,15], the matrices $\mathcal{Q}_i$ of each microswimmer evolve independently; this matrix is updated only after a state change, so a large number of iterations are required for the convergence of $\mathcal{Q}_i$. To speed up this learning process, we use the following multiswimmer, parallel-learning scheme: all the microswimmers share a common $\mathcal{Q}$ matrix, i.e., $\mathcal{Q}_i = \mathcal{Q}$, $\forall i$. At each iteration, we choose one microswimmer at random, from the set of microswimmers that have undergone a state change, to update the corresponding element of the $\mathcal{Q}$ matrix (flow chart in Appendix A); this ensures that the $\mathcal{Q}$ matrix is updated at almost every iteration and so it converges rapidly.

IV. NUMERICAL SIMULATION

We use a pseudospectral DNS [16,17], with the 2/3 dealiasing rule to solve Eqs. (1) and (2). For time marching we use a third-order Runge-Kutta scheme in two dimensions and the exponential Adams-Bashforth time-integration scheme in three dimensions; the time step δt is chosen such that the Courant-Friedrichs-Lewy (CFL) condition is satisfied. Table II gives the parameters for our DNSs in two and three dimensions, such as the number N of collocation points and the Taylor-microscale Reynolds numbers $R_\lambda = u_{\rm rms}\lambda/\nu$, where the Taylor microscale $\lambda = [\sum_k k^2 E(k)/\sum_k E(k)]^{-1/2}$.

TABLE II. Parameters: N, the number of collocation points; ν, the kinematic viscosity; α, the coefficient of friction; δt, the time step; and R_λ, the Taylor-microscale Reynolds number.

	Two dimensions	Three dimensions
N	256×256	$128 \times 128 \times 128$
ν	0.002	0.002
α	0.05	0.00
δt	5×10^{-4}	8×10^{-3}
R_λ	130	30

A. Naïve microswimmers

The average time taken by the microswimmers to reach their targets is $\langle\mathcal{T}\rangle$ (see Fig. 3). If $\hat{\mathbf{T}}_i = (\mathbf{X}_i - \mathbf{X}_i^T)/|\mathbf{X}_i - \mathbf{X}_i^T|$ is the unit vector pointing from the microswimmer to the target, then for $\tilde{V}_s \gg 1$ we expect the naïve strategy, i.e.,, $\hat{\mathbf{o}}_i = \hat{\mathbf{T}}_i$, to be the optimal one. For $\tilde{V}_s \simeq 1$, we observe that the naïve strategy leads to the trapping of microswimmers [Fig. 3(b)] and gives rise to exponential tails in the arrival-time ($\mathcal{T}$) probability distribution function (PDF); in Fig. 4 we plot the associated complementary cumulative distribution function (CCDF) $P^>(\mathcal{T}) = \int_{\mathcal{T}}^\infty \wp(\tau)\, d\tau$, where $\wp(\tau)\, d\tau$ is the probability of particle arrival in the time interval $[\tau, \tau + d\tau]$ and τ is the time since initialization of the microswimmer. As a consequence of trapping, $\langle\mathcal{T}\rangle$ is dominated by the exponential tail of the distribution, as can be seen from Fig. 4.

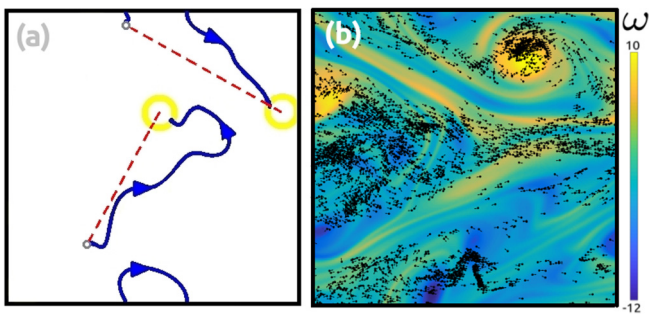

FIG. 3. (a) Illustrative (blue) paths for two microswimmers, with their corresponding (yellow) circular target regions (mapping in red dashed lines) where the microswimmer is eventually absorbed and reinitialized. We consider random positions of targets and initialize a microswimmer at a fixed distance from its corresponding target with randomized $\hat{\mathbf{p}}$; (b) a snapshot of the microswimmer distribution, in a vorticity field (ω), for the naïve strategy, at time $t = 30\tau_\Omega$, with $\tilde{V}_s = 1$. Here, the initial distance of the microswimmers from their respective targets is $L/3$ and the target radius is $L/50$; we use a system size L with periodic boundary conditions in all directions.

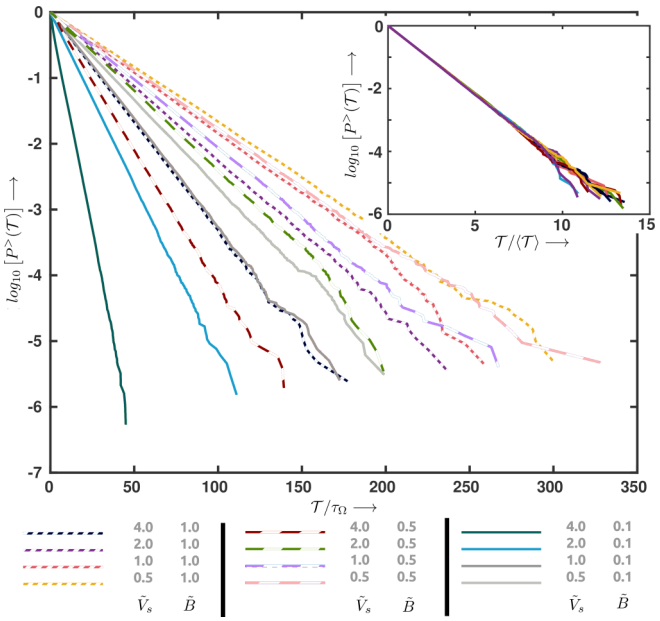

FIG. 4. Plots showing exponential tails in $P^{>}(\mathcal{T})$ for the naïve strategy, with different values of $\tilde{V}_s$ and $\tilde{B}$. The inset shows how these data collapse when $\mathcal{T}$ is normalized, for each curve, by the corresponding $\langle\mathcal{T}\rangle$, which implies $P^{>}(\mathcal{T}) \sim \exp(-\mathcal{T}/\langle\mathcal{T}\rangle)$.

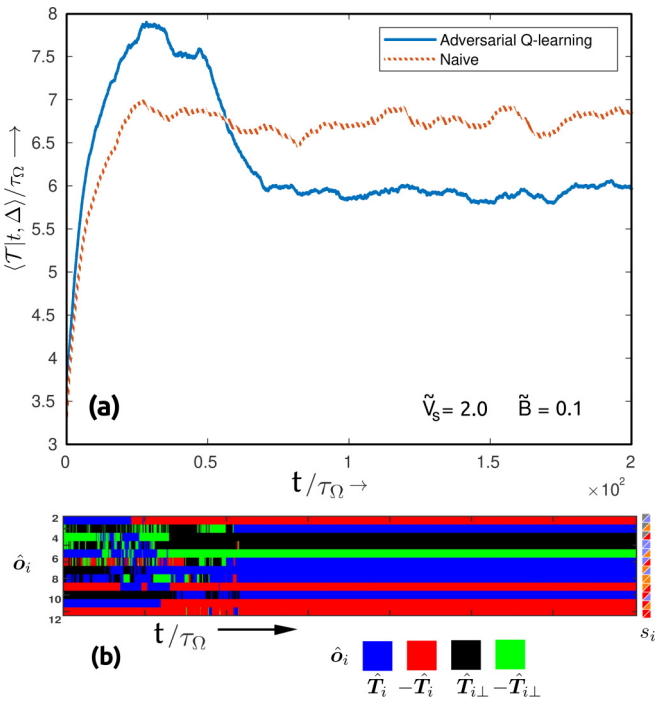

FIG. 5. Learning statistics: (a) Plot of $\langle\mathcal{T}|t,\Delta\rangle$, with $\Delta = 10\ \tau_\Omega$, in two dimensions. Adversarial $\mathcal{Q}$-learning initially shows a transient behavior before settling to a lower value of $\langle\mathcal{T}\rangle$ than that in the naïve strategy. (b) The evolution of the control map, $\hat{\mathbf{o}}_i$, where the color codes represent the actions that are performed for each of the 12 states. Initially, $\mathcal{Q}$-learning explores different strategies and settles down to an $\hat{\mathbf{o}}_i$ that shows, consistently, improved performance relative to the naïve strategy.

B. Smart microswimmers

In our approach, the random initial positions of the microswimmers ensure that they explore different states without reinitialization for each epoch. Hence, we present results with 10 000 microswimmers, for a single epoch. In our single-epoch approach, the control map $\hat{\mathbf{o}}_i$ reaches a steady state once the learning process is complete [Fig. 5(b)]. We would like to clarify here that, in our study, the training is performed in the fully turbulent time-dependent flow; even though this is more difficult than training in a temporally frozen flow, the gains, relative to the naïve strategy, justify this additional level of difficulty.

We use the adversarial $\mathcal{Q}$-learning approach outlined above (parameter values in Table I) to arrive at the optimal scheme for path planning in a 2D turbulent flow. To quantify the performance of the smart microswimmers, we introduce equal numbers of smart (master-slave pairs) and naïve microswimmers into the flow. The scheme presented here pits $\mathcal{Q}$-learning against the naïve strategy and enables the adversarial algorithm to find a strategy that can outperform the naïve one. (Without the adversarial approach, the final strategy that is obtained may end up being suboptimal.)

V. RESULTS

The elements of $\mathcal{Q}$ evolve during the initial-training stage, so $P^{>}(\mathcal{T})$ also evolves in time until the system reaches a statistically steady state (in which the elements of $\mathcal{Q}$ do not change). Hence, $\langle\mathcal{T}\rangle$ also changes during the initial-training stage; to capture this time dependence, we define $\langle\mathcal{T}(t)\rangle := 1/N(t)\ \sum_{i=1}^{N(t)} \mathcal{T}_i$, where $\mathcal{T}_i$ is the time taken by the ith microswimmer, since its initialization, to arrive

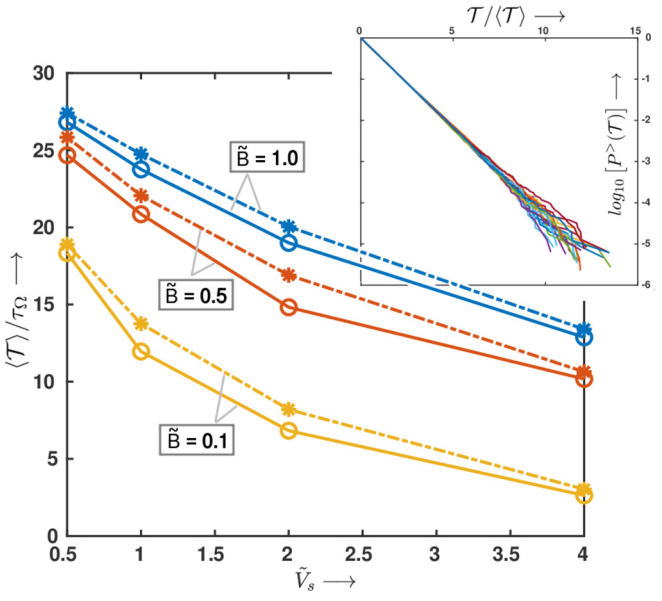

FIG. 6. The dependence of $\langle T\rangle$ on $\tilde{V}_s$, for different values of $\tilde{B}$, shown for the naïve strategy (dotted line) and for adversarial $\mathcal{Q}$-learning (solid line), for our 2D turbulent flow. The plot shows that, in the parameter space that we have explored, our adversarial $\mathcal{Q}$-learning method yields a lower value $\langle T\rangle$ than in the naïve strategy. The plot in the inset shows that the CCDF of T has an exponential tail.

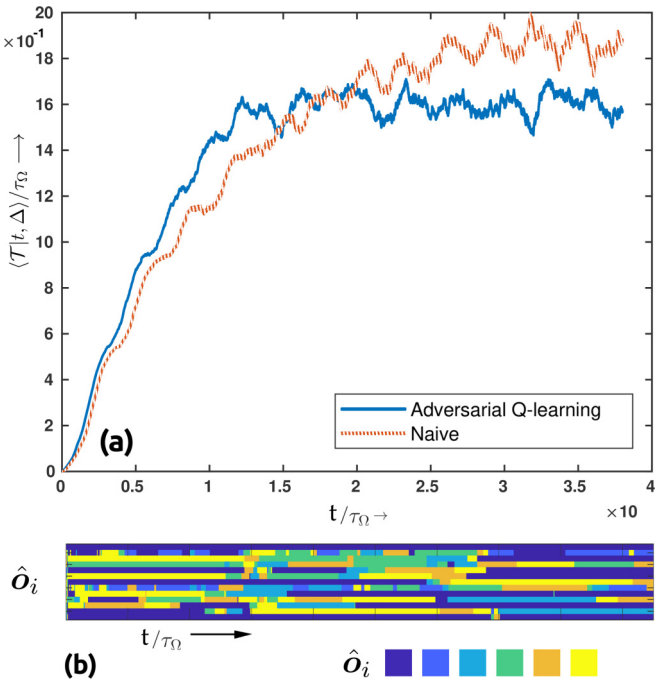

FIG. 7. Learning statistics in three dimensions: (a) The performance trend, $\langle \mathcal{T}|t,\Delta\rangle/\tau_\Omega$, with $\Delta = 10\tau_\Omega$ for adversarial $\mathcal{Q}$-learning (blue line) and naïve strategy (red broken line) for microswimmers in a 3D homogeneous isotropic turbulent flow, for $\tilde{V}_s = 1.5$ and $\tilde{B} = 0.5$. The trend shows a slow rise in performance, similar to that observed in two dimensions. In three dimensions the $\mathcal{Q}$-learning is performed by using 13 states and 6 actions defined in Appendix B. (b) The evolution of $\hat{\mathbf{o}}_i$ in three dimensions shows that learning has not reached a steady state due to lower probability of swimmers reaching the target, compared to the 2D case.

at its target at the time instant t and $N(t)$ is the number of microswimmers that reach their targets at time instant t. We find that $\langle \mathcal{T}(t)\rangle$ shows large fluctuations, so we average it over a time window Δ and define $\langle \mathcal{T}|t,\Delta\rangle := 1/\Delta \int_t^{t+\Delta} \langle \mathcal{T}(\tau)\rangle\, d\tau$. The initial growth in $\langle \mathcal{T}|t,\Delta\rangle$ arises because $\langle \mathcal{T}|t,\Delta\rangle \leqslant t$. The plots in Figs. 5(a) and 7 show the time evolution of $\langle \mathcal{T}|t,\Delta\rangle$ for the smart and naïve microswimmers. Note that $\hat{\mathbf{o}}_i$ becomes a constant, for large t, in Fig. 5(b); this implies that the elements of $\mathcal{Q}$ have settled down to their steady-state values.

Figures 5(a) and 5(b) show the evolution of $\langle \mathcal{T}|t,\Delta\rangle$ and $\hat{\mathbf{o}}$, respectively, for the naïve strategy and our adversarial $\mathcal{Q}$-learning scheme. After the initial learning phase, the $\mathcal{Q}$-learning algorithm explores different $\hat{\mathbf{o}}$, before it settles down to a steady state. It is not obvious, *a priori*, if there exists a stable, nontrivial, optimal strategy for microswimmers in turbulent flows that could outperform the naïve strategy. The plot in Fig. 6 shows the improved performance of our adversarial $\mathcal{Q}$-learning scheme over the naïve strategy, for different values of $\tilde{V}_s$ and $\tilde{B}$; in these plots we use $\langle \mathcal{T}\rangle = \langle \mathcal{T}|t\to\infty,\Delta\rangle$, so that the initial transient behavior in learning is excluded. The inset in Fig. 6 shows that $P^{>}(\mathcal{T})$ has an exponential tail, just like the naïve scheme in Fig. 4, which implies the smart microswimmers also get trapped; but a lower value of $\langle \mathcal{T}\rangle$ implies they are able to escape from the traps faster

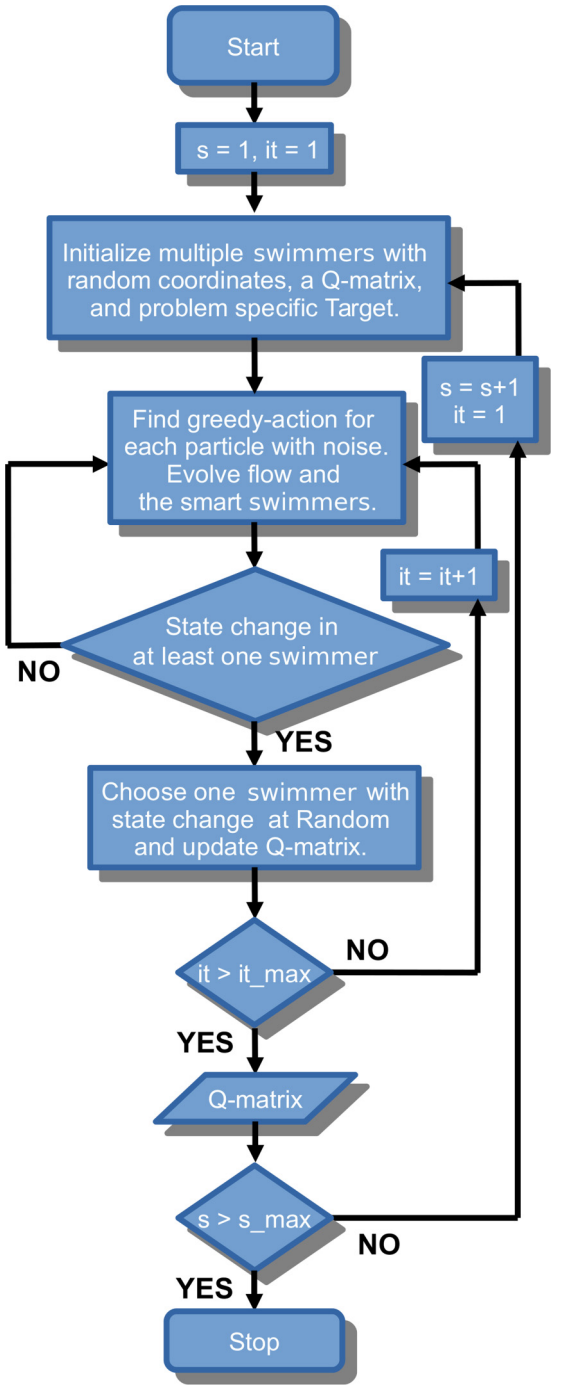

FIG. 8. This flowchart shows the sequence of processes involved in our adversarial $\mathcal{Q}$-learning algorithm.

than microswimmers that employ the naïve strategy. Note that the presence of a possible noise in the measurement of the discrete vorticity $\mathcal{S}_\omega$ should not change our findings because of the coarse discretization we use in defining the states.

In a 3D turbulent flow, we also obtain such an improvement with our adversarial $\mathcal{Q}$-learning approach over the naïve strategy. The details about the 3D flows, parameters, and the definitions of states and actions are given in Appendix B. In Fig. 7 we show a representative plot, for the performance

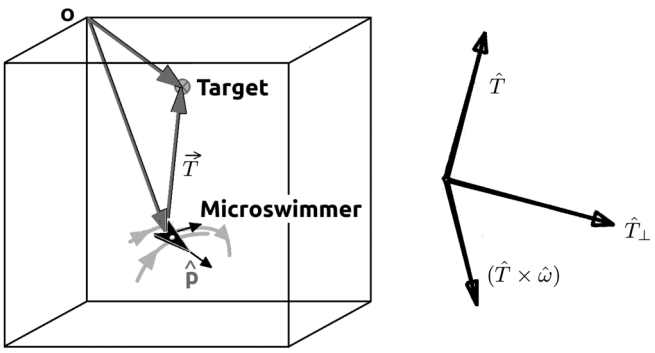

FIG. 9. We define a Cartesian coordinate system by using the orthonormal triad $\{\hat{\mathbf{T}}, (\hat{\mathbf{T}} \times \hat{\boldsymbol{\omega}}), \hat{\mathbf{T}}_\perp\}$; thus, all the vectorial quantities are represented in terms of this observer-independent coordinate system.

measure, which demonstrates this improvement in the 3D case (cf. Fig. 5 for a 2D turbulent flow).

VI. CONCLUSIONS

We have shown that the generic $\mathcal{Q}$-learning approach can be adopted to solve control problems arising in complex dynamical systems. In Ref. [18], global information of the flows has been used for path-planning problems in autonomous-underwater-vehicle navigation to improve their efficiency, based on the Hamilton-Jacobi-Bellman approach. In contrast, we present a scheme that uses only the local flow parameters for the path planning.

The flow parameters (Table II) and the learning parameters (Table I) have a significant impact on the performance of our adversarial $\mathcal{Q}$-learning method. Even the choice of observables that we use to define the states $(\mathcal{S}_\omega, \mathcal{S}_\theta)$ can be changed and experimented with. Furthermore, the discretization process can be eliminated by using deep-learning approaches, which can handle continuous inputs and outputs [19]. Our formulation of the optimal-path-planning problem for microswimmers in a turbulent flow is a natural starting point for detailed studies of control problems in turbulent flows.

Note added. We were recently made aware of Ref. [20], where they tackle the problem using an actor-critic reinforcement learning scheme.

We contrast, below, our reinforcement-learning approach with that of Ref. [20]:

(a) Reference [20] uses 900 discrete states, which are defined based on the approximate location of the microswimmer. By contrast, our scheme uses only the local vorticity $(\mathcal{S}_\omega)$, at the position of the microswimmer, and the orientation $(\mathcal{S}_\theta)$; after discretization, we retain only 12 states. In analogy with navigation parlance, Ref. [20] uses a GPS and our approach uses a lighthouse along with a local-vorticity measurement.

(b) In Ref. [20], the states are sensed periodically and the elements of $\mathcal{Q}$ are updated at every sensing instant. In contrast, we monitor the states continuously and update the elements of $\mathcal{Q}$ only when there is a state change. If the periodicity of sensing is smaller than the rate of change in states of the microswimmer, both schemes should show similar convergence behaviors.

(c) Reference [20] uses a conventional, episode-based training approach, which is sequential, whereas we use multiple microswimmers to perform parallel training.

(d) Reference [20] uses an actor-critic approach, whereas we use an adversarial learning method.

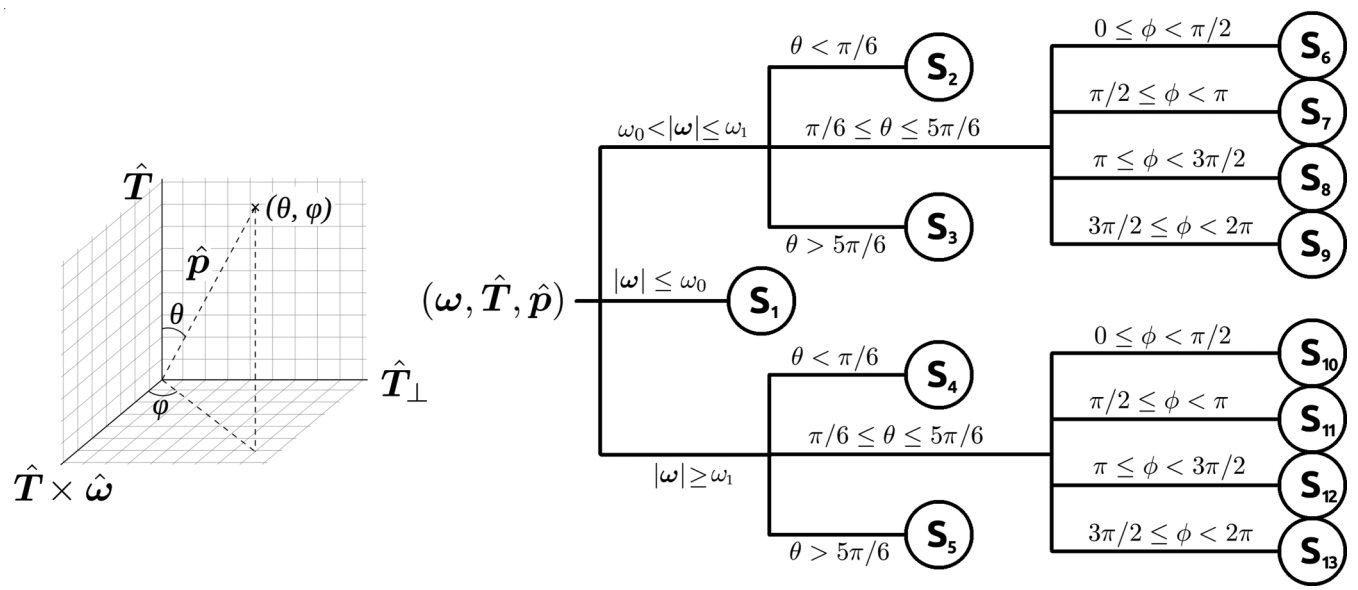

FIG. 10. Discretization of states in three dimensions: We define a spherical-polar coordinate system for each particle with the z axis pointing along the $\hat{\mathbf{T}}$ direction and the x axis along $\hat{\mathbf{T}}_\perp$. We define the canonical angles θ and ϕ, and discretize the states into 13, based on the magnitude of $\vec{\boldsymbol{\omega}}$, where ω_0 and ω_1 are state-definition parameters (we use $\omega_0 = \omega_{\rm rms}/3$ and $\omega_1 = \omega_{\rm rms}$), and the direction of $\hat{\boldsymbol{p}}$, with respect to the triad, is defined in Fig. 9.

ACKNOWLEDGMENTS

We thank DST, CSIR (India), BRNS, and the Indo-French Centre for Applied Mathematics (IFCAM) for support.

APPENDIX A: FLOWCHART

Figure 8 shows the sequence of processes involved in our adversarial $\mathcal{Q}$-learning scheme. Here *it* stands for the iteration number and s is the number of sessions. We use a greedy action in which the action corresponding to the maximum value in the $\mathcal{Q}$ matrix, for the state of the microswimmer, is performed; the ϵ-greedy step ensures with probability ϵ_g that the nonoptimal action is chosen. Furthermore, we find that episodic updating of the values on the $\mathcal{Q}$ matrix lead to a deterioration of performance; therefore, we use continuous updating of $\mathcal{Q}$.

APPENDIX B: STATE AND ACTION DEFINITIONS FOR 3D TURBULENT FLOW

From our DNS of the 3D Navier-Stokes equation we obtain a statistically steady, homogeneous-isotropic turbulent flow in a $128 \times 128 \times 128$ periodic domain. We introduce passive microswimmers into this flow. To define the states, we fix a coordinate triad, defined by $\{\hat{\mathbf{T}}, (\hat{\mathbf{T}} \times \hat{\boldsymbol{\omega}}), \hat{\mathbf{T}}_\perp\}$ as shown in Fig. 9; here, $\hat{\mathbf{T}}$ is the unit vector pointing from the microswimmer to the target, $\hat{\boldsymbol{\omega}}$ is the vorticity pseudovector, and $\hat{\mathbf{T}}_\perp$ is defined by the conditions $\hat{\mathbf{T}}_\perp \cdot \hat{\mathbf{T}} = 0$ and $\hat{\mathbf{T}}_\perp \cdot (\hat{\mathbf{T}} \times \hat{\boldsymbol{\omega}}) = 0$. This coordinate system is ill defined if $\vec{T}$ is parallel to $\vec{\boldsymbol{\omega}}$. To implement our $\mathcal{Q}$-learning in three dimensions, we define 13 states, $\mathcal{S} = (\mathcal{S}_{|\boldsymbol{\omega}|}, \mathcal{S}_\theta, \mathcal{S}_\phi)$ (see Fig. 10), and 6 actions, $\mathcal{A} = \{\hat{\mathbf{T}}, -\hat{\mathbf{T}}, (\hat{\mathbf{T}} \times \hat{\boldsymbol{\omega}}), -(\hat{\mathbf{T}} \times \hat{\boldsymbol{\omega}}), \hat{\mathbf{T}}_\perp, -\hat{\mathbf{T}}_\perp\}$. Consequently, the $\mathcal{Q}$ matrix is an array of size 13×6.

[1] E. Zermelo, Über das navigationsproblem bei ruhender oder veränderlicher windverteilung, Z. Angew. Math. Mech. **11**, 114 (1931).

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

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

[4] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Flow Navigation by Smart Microswimmers via Reinforcement Learning, Phys. Rev. Lett. **118**, 158004 (2017).

[5] 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).

[6] D. Dusenbery, *Living at Micro Scale: The Unexpected Physics of Being Small* (Harvard University Press, Cambridge, MA, 2009).

[7] W. M. Durham *et al.*, Turbulence drives microscale patches of motile phytoplankton, Nat. Commun. **4**, 2148 (2013).

[8] F. G. Michalec, S. Souissi, and M. Holzner, Turbulence triggers vigorous swimming but hinders motion strategy in planktonic copepods, J. R. Soc. Interface **12**, 20150158 (2015).

[9] 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).

[10] E. Barrows, *Animal Behavior Desk Reference: A Dictionary of Animal Behavior, Ecology, and Evolution, Third Edition* (Taylor & Francis, Philadelphia, 2011).

[11] R. Pandit *et al.*, An overview of the statistical properties of two-dimensional turbulence in fluids with particles, conducting fluids, fluids with polymer additives, binary-fluid mixtures, and superfluids, Phys. Fluids **29**, 111112 (2017).

[12] T. J. Pedley and J. O. Kessler, Hydrodynamic phenomena in suspensions of swimming microorganisms, Annu. Rev. Fluid Mech. **24**, 313 (1992).

[13] C. J. Watkins and P. Dayan, Technical note: Q-learning, Mach. Learn. **8**, 279 (1992); C. Watkins, Learning from delayed rewards, Ph.D. thesis, University of Cambridge, Cambridge, England, 1989.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 2011).

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, Reinforcement learning: A survey, J. Artif. Intell. Res. **4**, 237 (1996).

[16] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods* (Springer, Berlin, 2006).

[17] R. Pandit, P. Perlekar, and S. S. Ray, Statistical properties of turbulence: An overview, Pramana **73**, 157 (2009).

[18] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, Optimal path planning in time-varying flows using adaptive discretization, IEEE Rob. Autom. Lett. **3**, 458 (2018).

[19] T. P. Lillicrap *et al.*, Continuous control with deep reinforcement learning, *International Conference on Learning Representations*, San Juan, Puerto Rico (2016).

[20] L. Biferale *et al.*, Zermelo's problem: Optimal point-to-point navigation in 2D turbulent flows using reinforcement learning, Chaos **29**, 103138 (2019).