Smart inertial particles

Simona Colabrese,[1,*] Kristian Gustavsson,[1,2] Antonio Celani,[3] and Luca Biferale[1]
[1]*Department of Physics and INFN, University of Rome "Tor Vergata," Via della Ricerca Scientifica 1, 00133, Rome, Italy*
[2]*Department of Physics, University of Gothenburg, Origovägen 6 B, 41296, Göteborg, Sweden*
[3]*Quantitative Life Sciences, Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34151, Trieste, Italy*

(Received 15 December 2017; published 6 August 2018)

We performed a numerical study to train smart inertial particles to target specific flow regions with high vorticity through the use of reinforcement learning algorithms. The particles are able to actively change their size to modify their inertia and density. In short, using local measurements of the flow vorticity, the smart particle explores the interplay between its choices of size and its dynamical behavior in the flow environment. This allows it to accumulate experience and learn approximately optimal strategies of how to modulate its size in order to reach the target high-vorticity regions. We consider flows with different complexities: a two-dimensional stationary Taylor-Green-like configuration, a two-dimensional time-dependent flow, and finally a three-dimensional flow given by the stationary Arnold-Beltrami-Childress (ABC) helical flow. We show that smart particles are able to learn how to reach extremely intense vortical structures in all the tackled cases.

DOI: 10.1103/PhysRevFluids.3.084301

I. INTRODUCTION

Controlling and predicting the dynamical evolution and spatial distribution of small particles suspended in complex flows is a fundamental problem in many applied disciplines such as in spray combustion, drug delivery, dispersion of pollutants or contaminants in the environment, spray formation, and rain formation in clouds, to cite just a few cases [1–6]. The dynamics of small particles is also used in turbulence to study the Lagrangian statistics of the flow and/or the instantaneous Eulerian velocity distribution in particle image velocimetry (PIV) techniques [7,8]. Small particles have been instrumented to perform local measurements of flow properties [9]. In this paper, we present a numerical study to show how one can implement a suitable learning algorithm to train microparticles to actively control their dynamical trajectories in order to achieve some predetermined goal, for example, reaching a very intense vorticity region in the flow, escaping from turbulent fluctuations, preferentially tracking specific topological structures, etc. The number of potential applications is high, from engineering of small particles that are able to perform simple tasks by measurements of specific turbulent fluctuations to small active engines able to modify, actuate, and react to the flow structure both at micro- and macroscales [10–15]. On a macroscopic level, there are key applications in marine and atmospheric research, where buoys and small balloons are instrumented to measure local field properties [16–18]. It is crucial to optimize their trajectories during the diurnal cycles, but this is often a complicated task due to large fluctuations in the surrounding flow. Another application lies in understanding, predicting, and taking advantage of the behavior of biological organisms [19], with one example being diatom algae that are known to be able to adjust their densities in the stratified turbulent ocean in order to perform vertical migration [20–23].

*simona.colabrese@roma2.infn.it

It is well known that particles that are heavier or lighter than the fluid systematically detach from the flow streamlines [24–28]. As a consequence, correlations between particle positions and structures of the underlying flow appear. Heavy particles are expelled from vortical structures, while light particles tend to concentrate in their cores. This results in the formation of strong inhomogeneities in the spatial distribution of the particles, an effect often referred to as preferential concentration [29–34].

Thanks to these properties, light particles have been used as small probes that preferentially track any high-vorticity structure, highlighting statistical and topological properties of the underlying fluid conditioned on those structures [35,36].

In this paper, we seek inertial particles capable of sampling *ad hoc* specific flow structures. We imagine smart particles to be endowed with the ability to obtain some partial information about the regions of the flow that they are visiting. They can use this knowledge to learn how to adapt their size and consequently their inertia and density in order to preferentially sample only some predetermined flow properties.

The set of questions we want to address are the following: Can these smart particles learn how to track specific targets in an approximately optimal way in complex flows? Is this achievable without any previous knowledge of the flow structures and by using a set of simple behavioral actions the particle may take? Is learning achievable even when tracer particles would evolve in a chaotic and unpredictable way? Is it possible to guess the typical form of the approximately optimal strategies *a priori*? To what extent are the approximately optimal strategies learned in a stationary flow environment also robust by adding time dependence to the flow?

Similar questions have been investigated by training based on reinforcement learning algorithms for the dynamics of smart microswimmers in Taylor-Green flows [37], in Arnold-Beltrami-Childress (ABC) flows [38], and for the case of fish schooling both in still water and vortical flows [39–42]. A similar approach has also been pioneered in Ref. [43] for the case of birds that can exploit warm thermals to soar in a turbulent environment. Reinforcement learning is a framework to find good strategies for achieving given long-term tasks. It is widely used in artificial intelligence and machine learning. It is based on the interaction between a decision maker (in our case, the inertial particle) and the environment. The decision maker can change its behavior in response to inputs from the system. By trial and error, the decision maker progressively learns how to behave optimally [44–46]. We show that this approach provides a way to construct efficient strategies by accumulating experience also for the cases investigated here.

The paper is organized as follows. In Sec. II, we define the model used to describe inertial particles. It will be the groundwork for the design of smart particles. In Sec. III, we provide an overview on the reinforcement learning approach and on the algorithm used. In Sec. IV, we discuss the application of the reinforcement algorithm to particles in a Taylor-Green-like flow. We give space to the algorithm implementation and to the results achieved, both for stationary and time-dependent flows. Similarly, Sec. V deals with the case of ABC flows. Finally, in Sec. VI, we draw the conclusions and present final remarks.

II. INERTIAL PARTICLES

We focus the discussion on the case of small spherical inertial particles that have a density that is different from the surrounding fluid. We consider only the diluted limit, neglecting possible hydrodynamical interactions, collisions, and aggregation among the particles. The particles are small enough to be considered as pointlike. They are characterized by their mass, $m_{\rm p}$, and by their *adjustable* radius, b. A commonly used model for inertial particles with arbitrary density is the following [24,47–49]:

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{V} + \sqrt{2\chi}\,\eta, \\ \dot{\mathbf{V}} &= -\frac{1}{St}[\mathbf{V} - \mathbf{u}(\mathbf{X}, t)] + \beta D_t \mathbf{u}(\mathbf{X}, t). \end{aligned} \tag{1}$$

TABLE I. Set of parameters corresponding to the $N_a = 11$ actions. The second and the last rows of parameters indicate also the action values for the naive strategies. The dynamics in Eqs. (1) is affected only by the adimensional parameters St and β. All values of St and β in the table are determined by fixing $\beta = 0.176$ and $\mathrm{St} = 0.708$ at $b = b_{\min}$.

	$b/b_{\min}$	β	St	
a_1	1	0.176	0.708	Heavy
a_2	1.3	0.362	0.583	Heavy (B)
a_3	1.6	0.611	0.523	Heavy (B)
a_4	1.9	0.900	0.501	Heavy (B)
a_5	2.2	1.20	0.505	Light
a_6	2.5	1.48	0.527	Light
a_7	2.8	1.74	0.565	Light
a_8	3.1	1.95	0.615	Light
a_9	3.4	2.13	0.678	Light
a_{10}	3.7	2.28	0.751	Light
a_{11}	4	2.40	0.833	Light (C)

Here dots denote time derivatives, $\mathbf{X}$ and $\mathbf{V}$ are dimensionless particle position and velocity, $\mathbf{u}$ and $D_t\mathbf{u} = \partial_t\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}$ denote dimensionless flow velocity field and material derivative evaluated at the particle position, and $\eta(t)$ is a Gaussian white noise $\langle \eta_i(t)\eta_j(t')\rangle = \delta_{ij}\delta(t-t')$ with a small prefactor χ added to avoid structurally unstable solutions in the two-dimensional steady flow.

The dimensionless Stokes number

$$\mathrm{St} = \tau_\mathrm{p}/\tau \tag{2}$$

in (1) is defined in terms of the ratio of the characteristic flow time τ and the particle response time,

$$\tau_\mathrm{p} = b^2/(3\nu\beta), \tag{3}$$

where ν is the kinematic viscosity of the carrying flow. The dimensionless quantity

$$\beta = 3\rho_\mathrm{f}/(\rho_\mathrm{f} + 2\rho_\mathrm{p}) \tag{4}$$

accounts for the added mass effect resulting from the contrast between the particle density, $\rho_\mathrm{p} = 3m_\mathrm{p}/(4\pi b^3)$, and the fluid density ρ_f. The value $\beta = 1$ distinguishes particles that are heavier ($\beta < 1$) or lighter ($\beta > 1$) than the fluid. The dimensionless translational diffusivity χ in (1) is taken to be small, $\chi \ll 1$. For Eqs. (1) to be valid, it is assumed that the particle size b is much smaller than the smallest active scale of the flow and that the Reynolds number based on the particle size, $\mathrm{Re}_\mathrm{p} \equiv |\mathbf{u} - \mathbf{V}|b/\nu$, is very small, $\mathrm{Re}_\mathrm{p} \ll 1$. In the limits $\mathrm{St} \to 0$ or $\beta \to 1$, the dynamics determined by Eqs. (1) tends to the evolution of a tracer: Imposing a finite Stokes drag leads to $\mathbf{V} = \mathbf{u} + O[\mathrm{St}(1-\beta)] + O(\chi)$.

Both St and β depend on the radius b. As a consequence, particles that are able to modify their own radius depending on cues from the surrounding environment will indirectly exercise a control over their trajectory and may be able to alter their dynamics to navigate the flow. We consider a liquid-solid system in which particles can adjust their size b by choosing among a set of discrete values to be able to be either lighter or heavier than the fluid (see Table I and Subsec. V A). The equation system (1) is discretized using the Runge-Kutta fourth-order method with $dt = 0.001$ (much smaller than all the other timescales in the system).

III. REINFORCEMENT LEARNING APPROACH

To identify efficient strategies to sample high-vorticity regions in complex flows, we used reinforcement learning to implement the *one-step Q-learning* algorithm [50]. The general

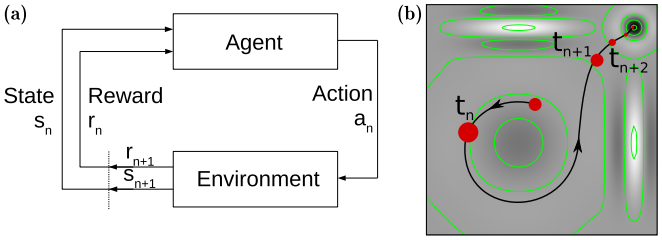

FIG. 1. (a) Sketch of the reinforcement learning agent-environment interactions. (b) Graphical summary of a typical trajectory for a smart particle in a generic two-dimensional flow. State changes occur at times t_n, t_{n+1}, t_{n+2}, ...when the particle enters new vorticity regions that are separated by isolines of the vorticity field Ω_z (green lines). In each state, the particle chooses its size (the action).

reinforcement-learning framework consists of an *agent* that is able to interact with its environment [see Fig. 1(a) for a schematic illustration]. At any given time, the agent has the ability to sense some information about the environment or about itself. This information forms the *state* s, which is an element of the set $\mathcal{S}$ consisting of all the possible distinct states the agent can recognize. Depending on the current state s_n, the agent chooses an *action* a_n from a set $\mathcal{A}$ of possible actions. Which action is chosen affects the interaction between the agent and the environment.

The transition $s_n \to s_{n+1}$ occurs either if the agent encounters a new state ($s_{n+1} \neq s_n$) or if a maximum time, $T_{\max}$, has run out (in the latter case $s_{n+1} = s_n$). The time $T_{\max}$ is chosen to be much larger than the characteristic flow timescale, $T_{\max} \gg \tau$. After the transition, the agent is given a *reward* r_{n+1} quantifying the immediate success of the previously chosen action to reach the target goal.

It is important to remark that in our way of implementing the reinforcement learning algorithm the physical time between consecutive state changes may fluctuate depending on the underlying dynamics. In principle, this implies the possibility for the modeled particle to have a nearly continuous-in-time control on its state if state changes occur frequently enough. An alternative implementation is based on a fixed time interval between two consecutive probes of the ambient state. The latter protocol is suitable for algorithms based on a continuous representation of the state-action space. This protocol is also required to ensure the possibility of reaching one of the optimal policies in the case in which the process is Markovian. In our case, as in most realistic applications, the process is not Markovian. States are vorticity levels and not positions in space; therefore, there is no unique recipe or best practice for modeling transitions. For the sake of this work, we only explored the first protocol explained above, without conditioning state changes on a fixed time interval. We checked *a posteriori* that for the quasioptimal policy the most probable time interval between two consecutive state changes coincides with the upper limit $T_{\max}$ (70% of all cases), so very frequent controls are not the typical ones.

Depending on the reward, the agent updates its *policy* of how to select the action for any given state. Finally, the agent chooses an action a_{n+1} for the new state s_{n+1} using the updated policy and repeats the procedure outlined above [see Fig. 1(a)]. The aim of reinforcement learning is to form a close-to-optimal policy of how to choose the action for a given state to achieve the predetermined goal. This is done by maximizing the long-term accumulated reward.

In our case, the agent is the smart inertial particle. It has as a target to navigate vortex flows to reach regions of high vorticity. One example trajectory in a generic two-dimensional vortex flow is sketched in Fig. 1(b). We assume that the particle can sense the z component Ω_z of the local flow vorticity $\Omega = \nabla \times u$. It can distinguish a discrete number N_s of coarse-grained states corresponding to equally spaced intervals of Ω_z in the full range of Ω_z allowed by the flow. Different states are separated by isolines of the flow vorticity as illustrated in Fig. 1(b). Each time t_n the particle enters a new state s_n, it selects a size (the action a_n) according to the current policy out of a finite discrete set of N_a possible sizes:

$$a_n \in \mathcal{A} = \{b\}_1^{N_a}.$$

As a result, the radius of the particle is a dynamical variable, $b \to b_n$, leading to a change in the particle density $\beta \to \beta(b_n)$ and inertia $\mathrm{St} \to \mathrm{St}(b_n)$. Depending on which sizes the particle chooses, it will in general experience a different dynamical evolution. The particle keeps its size until the time t_{n+1} where it enters the next state s_{n+1} and is given a reward r_{n+1}. In order to train the particle to move into regions of high vorticity, we choose the reward to be proportional to some power of the vorticity.

The core of the learning protocol lies in the policy π of how to chose an action given a state, $\pi(s) : s \to a$. To find an approximately optimal policy, a *training phase* is performed. During this phase, the particle is allowed to explore the effects of taking different actions in the different states. We use the one-step Q-learning algorithm to find an approximately optimal policy, π^*, iteratively by introduction of intermediate policies π_n after the nth state change, such that $\lim_{n\to\infty} \pi_n = \pi^*$. The policy π_n is derived from the Q *value*, $Q_\pi(s_n, a_n)$ (hence the name of the algorithm), according to the ϵ*-greedy* rule: Select the action that maximizes the current Q-value function except for a small probability ϵ of randomly selecting another action independent of the Q value:

$$a_{n+1} = \begin{cases} \arg\max_{a'} Q(s_{n+1}, a') & \text{with probability } 1-\epsilon \\ \text{a random action} & \text{with probability } \epsilon \end{cases}. \tag{5}$$

For every state-action pair (s_n, a_n) at time t_n, the Q value estimates the expected sum of future rewards conditioned on the current status of the system and on the current policy π_n:

$$Q_{\pi_n}(s_n, a_n) = \langle r_{n+1} + \gamma r_{n+2} + \gamma^2 r_{n+3} + \cdots \rangle. \tag{6}$$

The parameter γ in Eq. (6) is the *discount factor*, $0 \leqslant \gamma < 1$. The value of γ changes the resulting strategy: With a myopic evaluation (γ close to 0), the approximately optimal policy greedily maximizes only the immediate reward. As γ gets closer to 1, later rewards contribute significantly (far-sighted evaluation).

Each time a state change occurs, the agent is given a reward r_{n+1}, which is used together with the value of the next state s_{n+1} to update the $Q(s_n, a_n)$ value according to the following rule [44]:

$$Q(s_n, a_n) \leftarrow Q(s_n, a_n) + \alpha[r_{n+1} + \gamma \max_a Q(s_{n+1}, a) - Q(s_n, a_n)], \tag{7}$$

where α is a parameter that tunes the learning rate. The learning estimate is based on the differences between temporally successive predictions, in particular, in one-step Q learning (7), between the current prediction, $Q(s_n, a_n)$, and the one that follows, $r_{n+1} + \gamma \max_a Q(s_{n+1}, a)$. For Markovian systems and if ϵ slowly approaches zero as $n \to \infty$, it is possible to show that the update rule (7) converges to an optimal $Q(s, a)$ with a derived policy $\pi^*(s)$ which assigns to each state the action maximizing the expected long-term accumulated reward [44]. The system of inertial particles considered here is not Markovian, but still we expect approximately optimal policies to be found by the one-step Q-learning algorithm.

Operationally, we broke the training phases into a number N_E of subsequent episodes E, with $E = 1, \ldots, N_E$. The first episode is initialized with an optimistic Q value; i.e., all entries are equal and very large compared to the maximal achievable reward. This has the effect of encouraging exploration and avoiding being trapped around local minima in the search for approximately optimal policies. Each episode ends after a fixed number of total state changes, N, and it is followed by a new episode with a new random initial position of the particle in order to introduce further exploration. The initial velocity of the particle is equal to that of the fluid at the particle position. In order to accumulate experience, the initial Q of each new episode is given by the one obtained at the end of the previous episode. To quantify the learning ability of the smart particle during the training process, we monitor the total amount of reward that the particle gains in each episode:

$$\Sigma(E) = \sum_{n=1}^{N} r_n.$$

Finally, after the training has been performed, for each state s the final policy is to choose the action a with the maximal entry in the Q-value function. To quantify the success of smart particles, we use

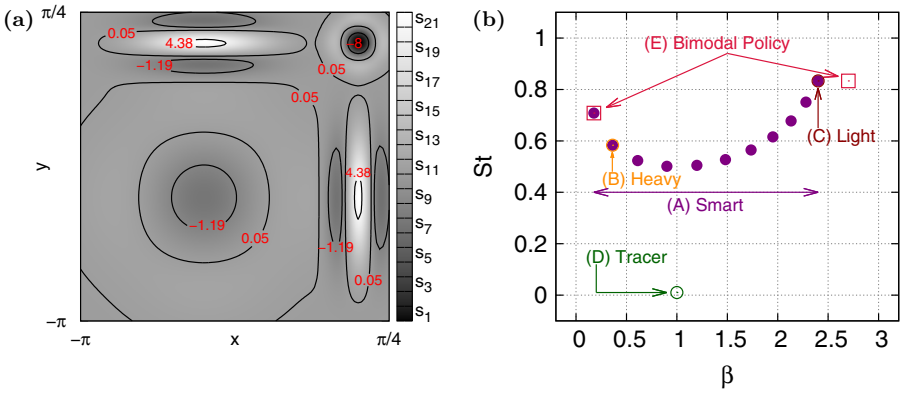

FIG. 2. (a) An illustrative sample of the isolines delimiting the $N_s = 21$ vorticity states. Values corresponding to the underlying vorticity isolines are shown in red. (b) Set of pairs (St, β) corresponding to the selection of the $N_a = 11$ actions. Smart particles are allowed to select each one of the available actions freely (A). We also highlight four different cases corresponding to four possible naive strategies: (B) heavy particle with (St $= 0.58$, $\beta = 0.36$); (C) light particle with (St $= 0.83$, $\beta = 2.4$); (D) tracer with (St $= 0.01$, $\beta = 1$); and (E) simple bimodal policy with only two actions: The particle has one constant low density (St $= 0.83$, $\beta = 2.7$) [light] in strong negative vorticity states or one constant high density (St $= 0.71$, $\beta = 0.18$) [heavy] in all the other states (see also text).

this policy to evaluate the long-term accumulative discounted reward, the *return*:

$$R_{\rm tot} = \left\langle \sum_{n=1}^{N} r_n \gamma^n \right\rangle. \tag{8}$$

Here the sum extends up to the total number of state changes, N, and the average is taken over realizations of the noise and over the initial conditions of Eqs. (1). The return $R_{\rm tot}$ can also be used during training to evaluate the success of the intermediate policies π_n.

IV. APPLICATION TO A TAYLOR-GREEN-LIKE FLOW

As a first example, we studied smart inertial particles in a two-dimensional stationary flow. The flow is differentiable and incompressible everywhere in the considered domain; it consists of four main vortices of different intensities and signs, singled out by appropriate Gaussian functions [see Fig. 2(a) and the Appendix for a detailed description]. This is a simple setup to test the ability of the smart particle to discriminate between vortices of the same sign but with different basins of attraction and intensity.

A. Algorithm implementation

We divide the scalar vorticity field into $N_s = 21$ equally spaced states as sketched in Fig. 2(a).

The only parameters affecting the motion in Eqs. (1) are β and St, given by Eqs. (2)–(4). We assume that the smart particle can increase its size b by a factor of 4 compared to the smallest size, $b_{\rm min}$. We use $N_a = 11$ actions of equally distributed magnification factors compared to $b_{\rm min}$. The values of β and St corresponding to these actions are illustrated in Fig. 2(b) (fixing the values of β and St at $b = b_{\rm min}$ such that the particle is heavy and with St of order unity). In Table I, we report all values for β and St for each one of the $N_a = 11$ different sizes. The window of St is chosen in order to optimize the sensitivity to β of the particle trajectories [25,33].

We contrast the smart particle to naive particles with simple strategies whose actions are illustrated in Fig. 2(b): being a heavy particle (B), being a light particle (C), being a tracer particle (D), and being a bimodal particle that is light in flow regions of large negative vorticity and heavy otherwise (E).

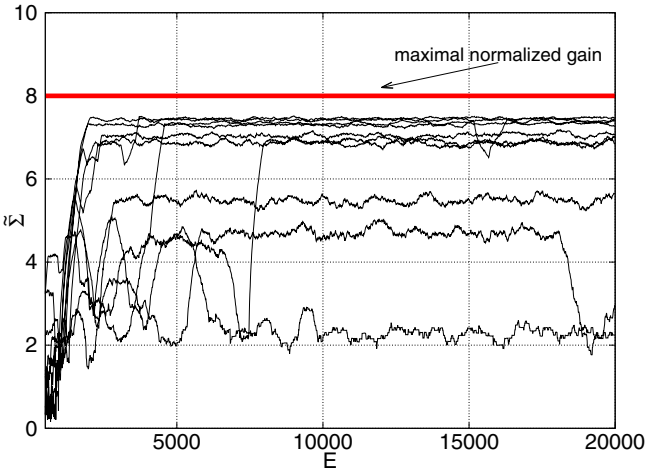

FIG. 3. Dependence of the normalized total gain, $\tilde{\Sigma}$ in Eq. (10), on the episode E for 10 different learning processes (black curves). Every point represents an average over a sliding window of 500 episodes. The red line shows the maximal possible reward.

To probe the efficiency of the algorithm, we choose the difficult task for the particles to reach the small upper-right flow region in Fig. 2(a) independently of the initial condition and within a limited number of state changes, N (here $N = 5000$). The upper-right region has the highest negative vorticity of the flow. We therefore assign a reward proportional to the cube of the negative vorticity experienced by the particle when it crosses the border between two vorticity levels:

$$r_{n+1} = -s_{n+1}^3, \tag{9}$$

where the minus sign is used to target negative vorticity regions. Because of the presence of different peaks in the vorticity distribution, the task is nontrivial. For example, naive light particles with a given fixed density and not too high Stokes number [for example, case (C) in Fig. 2(b)] would simply be attracted by the vortex closest to their initial condition, independently of the sign and intensity of the vortex.

We consider reinforcement learning using the scheme described in Sec. III with mainly fixed learning rate $\alpha = 0.1$ and $\epsilon = 0$ (for a case with time-dependent α and ϵ, see Sec. IV B 1). The other fixed parameters are $\gamma = 0.999$ and $\chi = 0.005$. The parameters are empirically chosen to achieve the convergence to approximately optimal policies. To enhance exploration, the elements of the initial Q value matrix are chosen to be equal to the undiscounted return that a particle would gain if it was in the target region during the entire length of the episode: $Q_{\pi_0}(s, a) = -\Omega_{\min}^3 N$, for all (s, a).

B. Results

We first discuss the training session. In Fig. 3, we show the evolution of the normalized total gain:

$$\tilde{\Sigma}(E) = \sqrt[3]{\frac{\Sigma(E)}{N}}, \tag{10}$$

where the normalization is introduced such that the maximum achievable gain corresponds to the cube root of the maximal reward, or equivalently to the minimal negative vorticity of the flow, $\Omega_{\min} = -8$, found in the upper-right region in Fig. 2(a) (state s_1). For each session, the smart particle increases its performance during the training phase and eventually reaches the smallest vortex for most initial conditions and realizations of the white noise η. Figure 3 shows the evolution of the learning gain as a function of the episode during the training process for 10 different trials. We observe that the

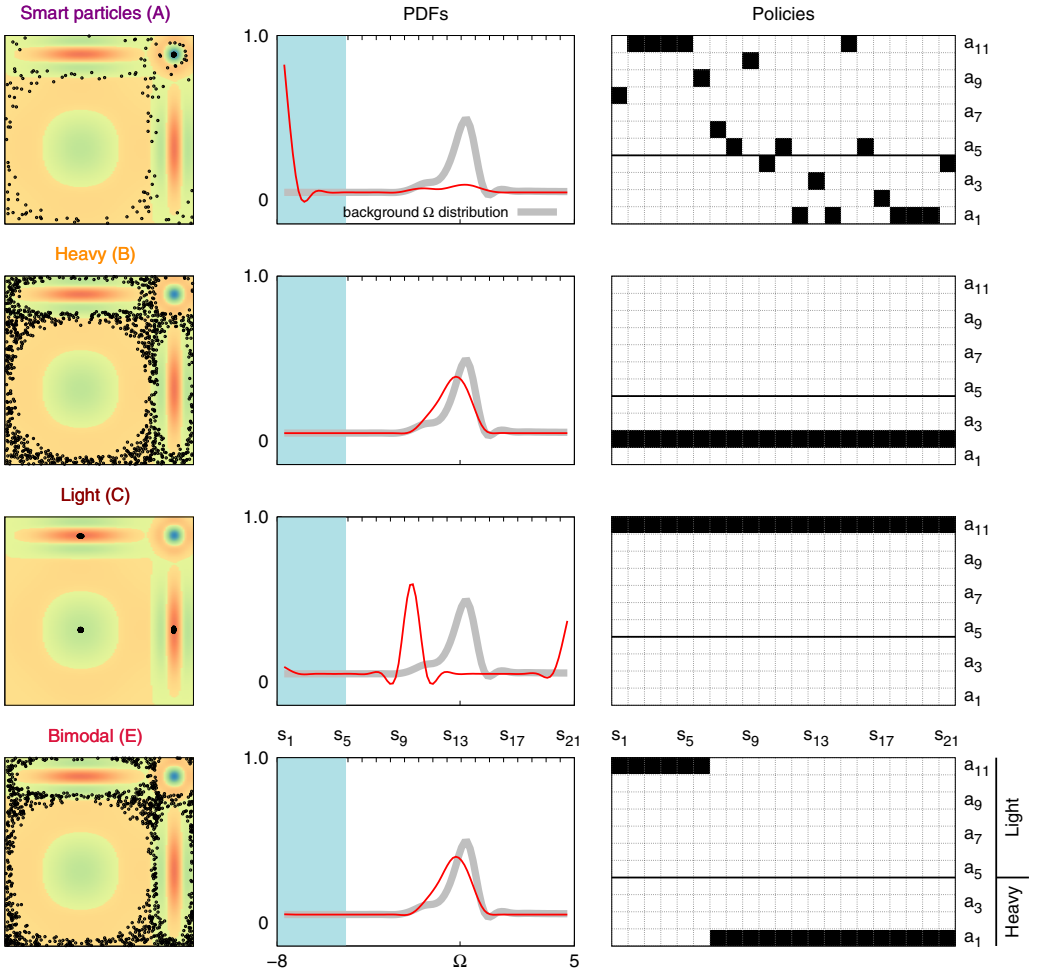

FIG. 4. Left column: Particle positions along 10 representative trajectories plotted after an initial transient for the cases studied in Fig. 2(b) (except for tracer particles which fill space uniformly). From top to bottom: Smart particles (A), heavy (B), light (C), and simple bimodal policy (E). Middle column: probability density function of vorticity sampled along 1000 particle trajectories for each of the cases studied (red curves). The PDFs are compared with the distribution for tracer particles (shaded curve). The blue shaded region marks vorticity levels that are unique to the target region. Right column: the policies (the action a the particle takes given a state s) used in the considered cases.

different trials result in different values of the normalized gain after many episodes. The greedy choice ($\epsilon = 0$) of actions based on Q that we have adopted in this particular numerical experiment allow for little exploration. After an initial transient where much exploration occurs, the evolution of the initially large Q-value matrix almost stagnates. As a result, it might happen that the dynamical relaxation (7) toward the approximately optimal policy gets stuck for many training episodes in a local optimum. The only way to leave the local optimum is given by the exploration due to the choice of initial condition and the realization of the noise η.

After the training, we perform an exam session by taking the policy derived from the final Q obtained from one of the successful trials shown in Fig. 3. In Fig. 4, we show the spatial distribution of smart particles, using the derived policy which gives the highest gain in Fig. 3. This is compared to the spatial distribution for the four naive reference cases discussed above and shown in Fig. 2(b).

TABLE II. The normalized discounted return, $\tilde{R}_{\rm tot}$, for the best training of the smart particles using a greedy policy, compared with the other four cases, (B)–(E). First row: stationary Taylor-Green-like flow (S-TG) given by Eq. (A1). Second row: the same as the first row but for the time-dependent case (T-TG) discussed in Sec. (IV B 2).

		(A)	(B)	(C)	(D)	(E)
S-TG	$\tilde{R}_{\rm tot}$	7.49	0.76	2.4	1.4	1.6
T-TG	$\tilde{R}_{\rm tot}$	5.0	0.0	−2.0	0.0	0.0

We observe that the trajectories of smart particles have high density in the target region, which is not sampled at all in the other instances or just rarely for the case of light particles. Next to the position plots, we show the probability density functions of the vorticity sampled by the particles (middle column), which give a quantitative representation of the frequency at which the particle visits different states. Notice how the smart particles (first row) are able to oversample the target of the training, the intense negative vorticity region, avoiding for most of their time the other lower-reward vortices. It is important to remark that the approximately optimal policy obtained via the reinforcement learning is much better than the one with a bimodal change in density, being heavy for vorticities outside of the target region and light otherwise (fourth row in Fig. 4). We have also tested that the policy obtained by using the RL algorithm is always much better than any other bimodal policy defined with other states when the particles switch from light to heavy (not shown). In other words, to reach the small target it is necessary to take unintuitive actions even in a relatively simple flow as the one considered here. This is summarized by the complex structure of the approximately optimal policy shown in the right column of Fig. 4. In Table II, we report a quantitative comparison between the long-term normalized return

$$\tilde{R}_{\rm tot} = \sqrt[3]{\frac{1-\gamma}{1-\gamma^{N+1}}}\sqrt[3]{R_{\rm tot}} \tag{11}$$

for smart particles and the other reference cases. The normalization used for $R_{\rm tot}$ in (11) is given by the sum of the first N terms of the geometric series with common ratio γ.

1. Additional exploration during the training

In this section, we describe how the performance of the reinforcement learning can be improved by adding additional exploration during the training session using a nongreedy action ($\epsilon > 0$). The training phase starts with an initial positive value of ϵ which is then slowly reduced to zero. This prevents trapping in local minima for a transient phase and then slowly the greedy policy is recovered. A particularly simple and efficient scheme is to decrease both the learning rate α and the exploration rate ϵ as a function of the episode, E, during the training,

$$\alpha_E = \alpha_0/(1+\sigma\, E), \quad \epsilon_E = \epsilon_0/(1+\delta\, E), \tag{12}$$

where σ and δ are positive constants. In Fig. 5, we show the results for a particular choice, $\epsilon_0 = 1/1000$, $\alpha_0 = 1/10$, $\sigma = 1/800$, and $\delta = 1/10\,000$. These values have been derived by trial and error, rather than by doing a full optimization that goes beyond the scope of the present study. With this selection of parameters, the particle have the order of 5000 more exploratory choices during the whole training phase compared to the case of $\epsilon = 0$. We observed that by adding additional exploration, the smart particles are able to find, in a more systematic way, approximately optimal policies that are on the same level or better than the best performing policies found in Fig. 3. The found policies are then stabilized by the adiabatic switch off in Eq. (12). It is important to notice that when using the ϵ-greedy method, one might need to fine-tune the parameters of the learning protocol, ϵ_o, α_0, σ, δ, and that in most cases there is not *a priori* obvious choice. This is because there exists a trade-off between the advantage of taking locally suboptimal actions and the risk to drift

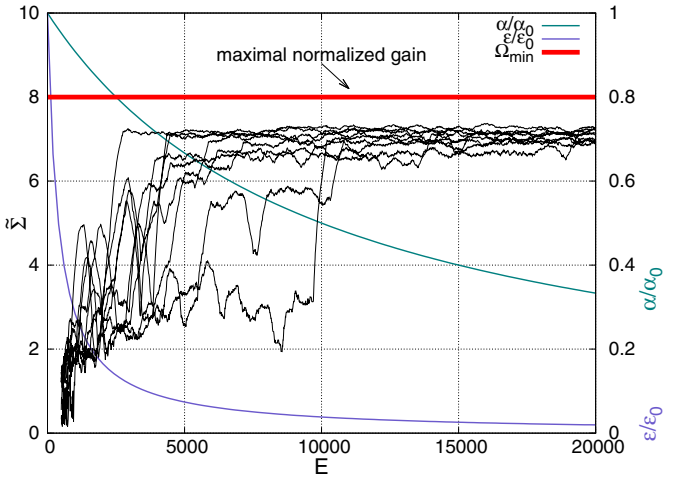

FIG. 5. The normalized learning gain, $\tilde{\Sigma}(E)$ in Eq. (10) against the episodes E for the ϵ-greedy reinforcement learning algorithm. The representation is comparable to Fig. 3 (scale on left vertical axis). The two continuous curves represent the adiabatic decreasing of both the exploration parameter ϵ and the learning rate α (scale on right vertical axis).

in the space of possible solutions due to the excess of randomness introduced by the exploration. Similarly, the performance of the algorithm might depend on the sets of allowed states and actions. If there is a too limited number of options, the particle might not have enough information and/or enough freedom in maneuvering, leading to a failure to reach the target. On the other hand, an excess of inputs and control might lead to a slow convergence to the approximately optimal strategy due to the need to explore a vast number of state-action entries of the Q-value matrix. In Fig. 6, we show the sensitivity of our results upon changing the discount factor γ but keeping all other training parameters constant. As one can see, the normalized gain remains of the same order when γ is varied one order of magnitude. Only if the discount is too myopic ($\gamma \ll 1$) or too far-sighted ($\gamma \sim 1$) does the algorithm fail. This indicates that the application of the reinforcement learning protocol to the problem of inertial particles studied here is robust to variations in the learning parameters. Typical combinations of the learning parameters consistently give good rewards and there is no need for fine-tuning of the learning parameters for the application of finding an acceptably good solution. If the application, on the other hand, is to really find a policy that is arbitrarily close to the approximately

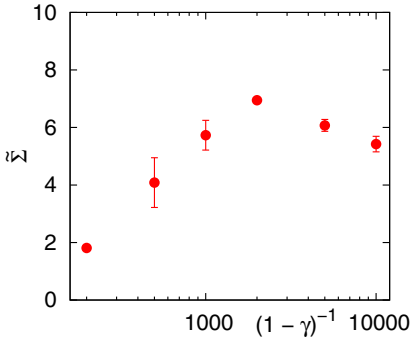

FIG. 6. The normalized learning gain $\tilde{\Sigma}(E)$ as a function of the time horizon of the learning process $(1-\gamma)^{-1}$ averaged over samples of 10 different learning trials. Error bars are estimated on the basis of the scatter inside each sample.

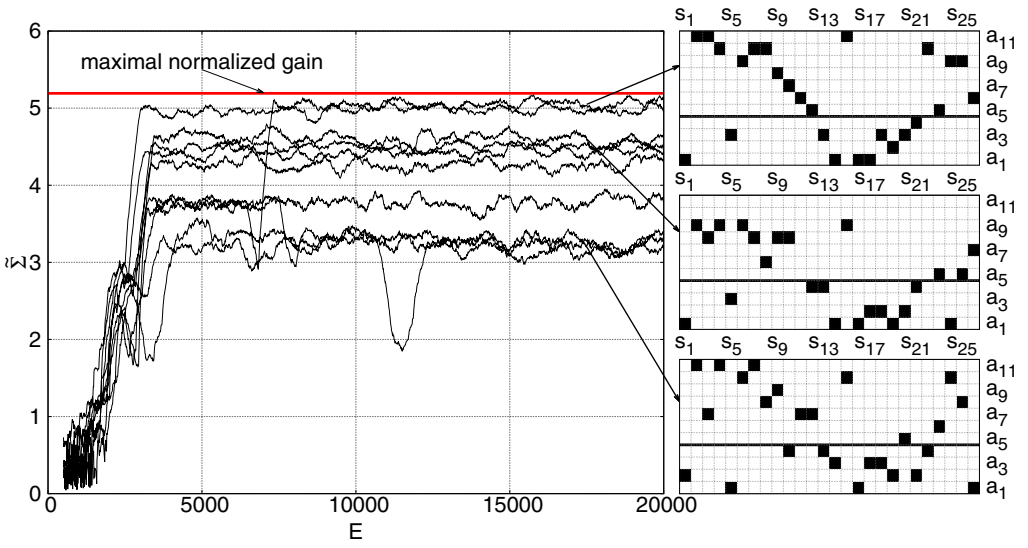

FIG. 7. Dependence of the normalized learning gain, $\tilde{\Sigma}$, for 10 different learning processes (black curves) on the two-dimensional time-dependent flow. Every point represents an average over a sliding window of 500 episodes. The three policies shown to the right are for the cases with best, middle, and worst final gain.

optimal behavior, more care is required in the choice of learning parameters and in the design of the space of allowed states and actions.

2. Time-dependent flows

It is natural to ask how the approximately optimal policy will perform under perturbations of the underlying flow or if the algorithm is robust when applied to time dependent and more complex flows. We have extended the two-dimensional flow in such a way that the four coefficients b_1,...,b_4 building up the four-vortex flow [see Eq. (A1) in the Appendix] acquires an out-of-phase oscillating behavior $b_i(t) = \cos(\omega_0 t + \phi_i)$, where ω_0 is a constant angular frequency and ϕ_i are four different constant phases. The system has been trained for the case $\omega_0 = 0.001$ with constant learning rate $\alpha = 0.1$ and greedy selection of actions ($\epsilon = 0$). In Fig. 7, we show the corresponding results of the total normalized gain during the training phase, $\tilde{\Sigma}(E)$. Notice that the maximal value of $\tilde{\Sigma}$ now depends on time, as the instantaneous maximal negative vorticity evolves in time. For our choice of parameters, it turns out that the cube root of the time average of the cubed minimal negative vorticity over one oscillation is $(\overline{-\Omega_{\min}^3})^{1/3} \approx 5.2$. Even in the presence of time variations, the smart particle learns how to move in the flow to follow the most intense negative vortices, which now oscillates in a nontrivial manner around the four regions of the flow. In Fig. 7, we also show three approximately optimal policies corresponding to three typical learning events. As one can see, there are some systematic patterns that are common for all cases. A visual inspection of the spatial distribution of the particles at four different times during one oscillation of the basic flow can be found in Fig. 8. Here it is possible to see how the particles are indeed trying to follow the moving target with high percentage of success. The top panel of Fig. 8 shows the time dependence of the average vorticity sampled along trajectories of 100 smart particles that started from random initial positions. The average vorticity is compared with the four reference cases (B)–(E). As one can see, smart particles do not remain oscillating in a confined region but exploit the flow to reach, in average, more profitable areas. In the last column on the right of the same figure, we show the instantaneous distribution of vorticity sampled by the smart particles (top) and by the light case (bottom). In the figure, we also show the instantaneous minimum of vorticity in the two-dimensional spatial configuration. We see

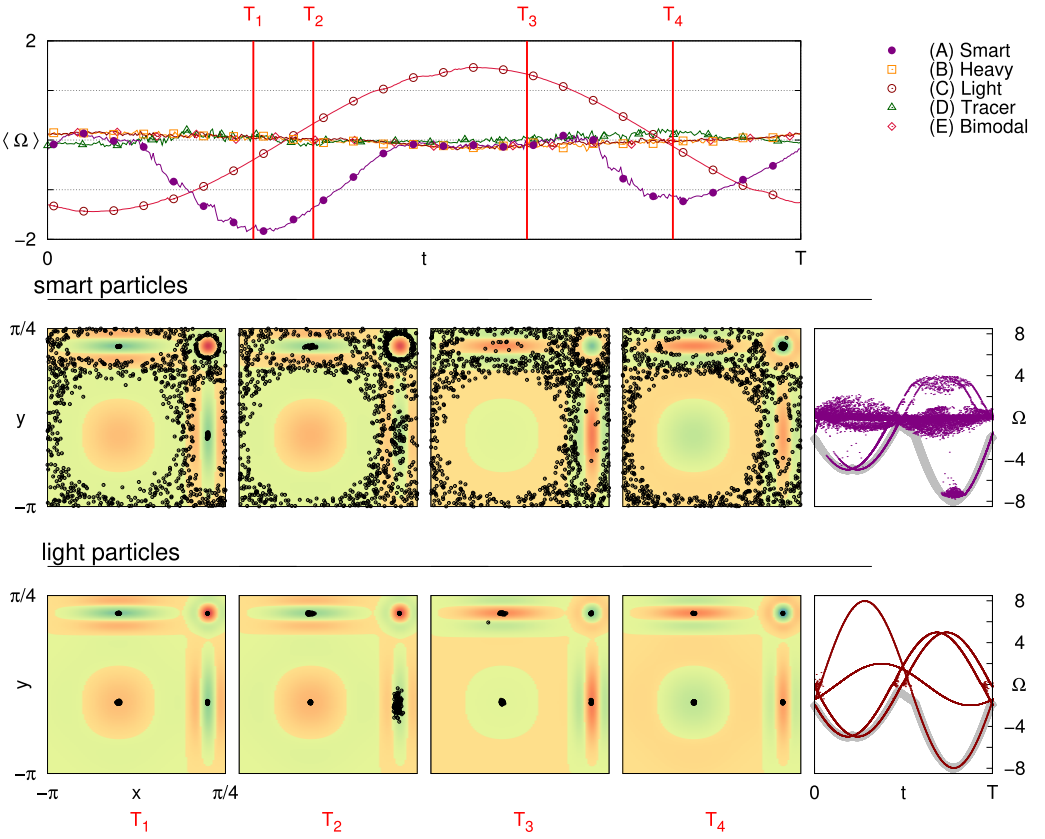

FIG. 8. Upper row: average Ω sampled along the time evolution of 100 particles for each analyzed type: A, smart (purple); B, heavy (orange); C, light (red); D, tracer (green); and E, particle with bimodal policy (magenta). The average vorticity is periodic with the period $T = 2\pi/\omega_0$ of the underlying flow. Middle row: spatial distribution of 100 smart particles at the four different times T_1, T_2, T_3, and T_4 highlighted in the top panel. Lower row: the same but for light particles. The last column on the right shows scatter plots of Ω for smart particles (top) and light ones (bottom) throughout the entire period. The shadowed gray curve represents the instantaneous maximum negative vorticity throughout the flow.

that while light particles are forced to follow the local extrema of vorticity (positive or negative), the smart particles tend to avoid positive vorticity and to be accumulated in regions of intense negative vorticity, as requested by the reward.

V. APPLICATION TO ABC FLOWS

A stationary three-dimensional flow is intriguing because the motion of tracers can be chaotic and very irregular. The Arnold-Beltrami-Childress (ABC) flow has been the subject of many studies in turbulence theory. Its Eulerian velocity field is (in dimensionless coordinates):

$$u_x = C\cos y + A\sin z, \quad u_y = A\cos z + B\sin x, \quad u_z = B\cos x + C\sin y. \tag{13}$$

The flow is characterized by three parameters A, B, and C.

Numerical simulations and theoretical arguments show that the ABC flow has tubelike regions in space within which the streamlines of the flow are confined and the velocity is essentially one-dimensional [51]. Since vorticity $\boldsymbol{\Omega} \equiv \nabla \wedge \boldsymbol{u}$ is parallel to the velocity in an ABC flow, $\boldsymbol{\Omega} = \boldsymbol{u}/2$, these tubes are referred to as principal vortices. Because of symmetries, an ABC flow with $A = B = C$

has three pairs of principal vortices that are mainly aligned with the three directions $\hat{x}$, $\hat{y}$, and $\hat{z}$. Each pair consists of two vortices of opposite sign of velocity and vorticity. Within the principal vortices of an ABC flow, the dynamics is regular, while outside it may become chaotic. Similar to the two-dimensional flow in Sec. IV, we impose here as a target for the smart particle to maximize the magnitude of its vorticity $\Omega \equiv |\mathbf{\Omega}|$. In order to achieve this goal, the particle needs to navigate a complex flow landscape to target the principal vortices with maximal vorticity.

A. Algorithm implementation

To show how general the success of the reinforcement learning is, we adopt a slightly modified version of the learning framework implemented in Sec. IV. We keep $\mathrm{St} = 0.2$ fixed and use as an action to change the value of β. Allowed values are equally distributed in N_a levels between 0 and 3. The state of the particle is given by either $|\mathbf{\Omega}|$ or one of the components of $\mathbf{\Omega}$, equally partitioned in N_s levels between the minimal and maximal value that can be obtained in the ABC flow. We use as reward $|\mathbf{\Omega}|^3$ averaged over the time the particle spend between state changes. This is in contrast to Sec. IV, where the cubed vorticity was evaluated at the position of the state change. As in Sec. IV, we use optimistic learning, where the entries of the initial Q-value matrix is N times the maximal reward and $N = 1000$ is the number of state changes per episode. We keep the learning rate fixed, $\alpha = 0.1$, use a greedy policy, $\epsilon = 0$, use a discount factor, $\gamma = 0.97$, and use no noise in Eq. (1), $\chi = 0$.

B. Flow parameters

For the ABC flow, we use light particles ($\beta = 3$) as naive reference particles. Principal vortices are traps for light particles: Depending on the initial condition, a light particle ends up in either one of the principal vortices. In a symmetric ABC flow ($A = B = C$), the average magnitude of vorticity along trajectories in each of the principal vortices is identical and the smart particle only marginally manages to outperform a light particle (not shown). We therefore consider ABC flows with weakly broken symmetry, $2A = B = C = 1$, and with strongly broken symmetry, $4A = 2B = C = 1$. In the weakly broken case, the dynamics in one direction is distinct from the other two, and in the strongly broken case all three directions have different dynamics. For the case of an asymmetric ABC flow, the situation is similar to the four-vortex flow studied above. If the particle is constantly light, it will be attracted to a vortex region depending on its initial condition. However, since different principal vortices have different intensity of vorticity in the asymmetric ABC flow, not all light particles will end up in the region of strongest vorticity (similar to the four-vortex flow above, where light particles end up in either of the vortices depending on its initial condition). Giving the smart particle appropriate information about the flow, we expect it to be able to learn to go to the principal vortices of highest vorticity and consequently beat the light particle.

C. Results

Figure 9(a) shows the evolution of the normalized total gain $\tilde{\Sigma}$ for training of particles in an ABC flow with weakly broken symmetry. The training has been performed using $|\mathbf{\Omega}|$ as the state and repeated 10 times. The resulting curves (solid purple) are compared to the maximal vorticity in the flow (black dashed) and the steady-state average $\langle|\mathbf{\Omega}|^3\rangle_\infty^{1/3}$ for naive particles with constant $\beta = 3$ (red dashed). We remark that the value of the purple curves is lower than the steady-state average due to the initial transient before the steady state is reached. We therefore also plot as solid blue the steady-state average $\langle|\mathbf{\Omega}|^3\rangle_\infty^{1/3}$ for the best policy (highest averaged reward at the last episode) for the data in Fig. 9(a). Figure 9(b) shows the steady-state distribution of the magnitude of vorticity for the best policy in Fig. 9(a) (blue peak), for tracer particles (green solid line), and for naive light particles (red line and peak). The distribution for the tracer particles shows that over the entire flow, vorticity is more or less uniformly distributed with some reduced probability at small values. The distribution for the smart particles instead shows a sharp peak. A similar peak is also found for the

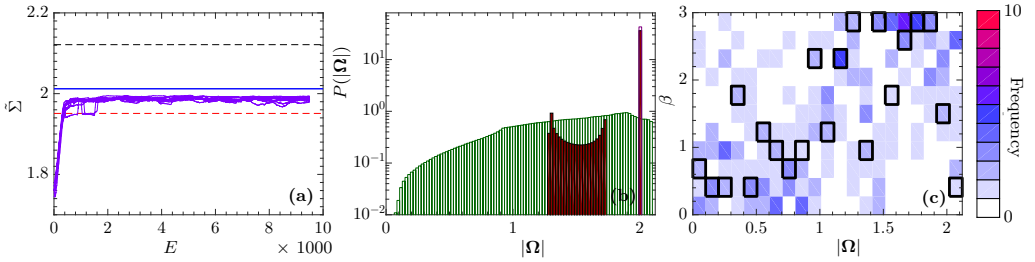

FIG. 9. Result from training of particles in ABC flow with weakly broken symmetry ($2A = B = C = 1$). (a) Normalized total gain $\tilde{\Sigma} = \langle|\mathbf{\Omega}|^3\rangle^{1/3}$ [Eq. (10)] as a function of episode E. The maximal vorticity in the flow is shown as black dashed line. Red dashed line shows the steady-state average $\langle|\mathbf{\Omega}|^3\rangle_\infty^{1/3}$ along the trajectories of many naive light particles. Solid blue line shows the corresponding average $\langle|\mathbf{\Omega}|^3\rangle_\infty^{1/3}$ using the best policy obtained. Black dashed line shows maximal vorticity. (b) Distribution of magnitude of vorticity $|\mathbf{\Omega}|$ for smart particles (blue peak), naive particles with $\beta = 1$ (green), and naive particles with $\beta = 3$ (red region and peak). (c) Frequency of optimal action for each state using the 10 final policies for the cases displayed in panel (a). Black boxes highlight the best policy in panel (a).

light particles, but these also show a band of vorticities around $|\mathbf{\Omega}| = 1.5$. Figure 9(c) shows, for the cases in Fig. 9(a), the best policy found and the frequency at which the 10 final policies select actions for each state.

For nonsmall values of St (St $= 0.5$ and St $= 2$) the reinforced learning scheme basically finds the naive solution, i.e., to be heavy if $|\mathbf{\Omega}|$ is small and light if $|\mathbf{\Omega}|$ larger than some threshold value (not shown). These solutions perform at the same level as the naive solution of being light with constant $\beta = 3$. For the case St $= 0.2$ considered here, the smart particle find a qualitatively different solution that beats the naive solution, although the final gain is only approximately 5% better than the naive solution. As shown in Fig. 9(b), the distribution of vorticity for the smart particle and the naive light particle are similar but with one important difference. The sharp peak close to $|\mathbf{\Omega}| = 2$ corresponds to the principal vortices in the z direction. While all initial conditions end up in these vortices for the smart particles, some initial conditions for naive particles ends up in the subdominant principal vortices orthogonal to the z direction, leading to the band of vorticities around $|\mathbf{\Omega}| = 1.5$, which explains why the naive particles have a lower gain. As shown in Fig. 9(c), the trend in the policy is basically to be light when vorticity is high and heavy when vorticity is low, but there also seems to be some structure needed in the intermediate vorticity states that enables the smart particles to outperform the naive one.

For the ABC flow with strongly broken symmetry, light particles distribute on two pairs of principal vortices, with roughly 80% of the particles in the strong principal vortices in the $\hat{z}$ direction and the rest on the weaker principal vortices in the $\hat{x}$ direction. One such trajectory is shown in Fig. 10(a). Training of the smart particle, on the other hand, using Ω_z as the state, allows it to find strategies that target the dominant principal vortices in the $\hat{z}$ direction for all tested initial conditions. One such example is shown in Fig. 10(b): Starting from the same initial condition, the smart particle reaches the optimal vortex, while the naive particle ends up in a subdominant vortex.

In Fig. 10, we observed that the smart particles using Ω_z as state recognize the two principal vortices of highest vorticity and therefore get a higher reward than the light particles with constant $\beta = 3$. The training progress and resulting policy are shown in Figs. 12(c) and 12(f). The structure of the Q-value matrix suggests that the policy used is quite simple: Be heavy ($\beta = 0$) when $|\Omega_z|$ is smaller than some threshold and light ($\beta = 3$) otherwise. This observation is supported by the data in Table III. The normalized returns quoted in Table III show that if the threshold is chosen around $N^* = 7$–9, where N^* is the number of states where the particle is heavy, the bimodal policy in Table III performs roughly at the same level as the trained solution. This value of the threshold

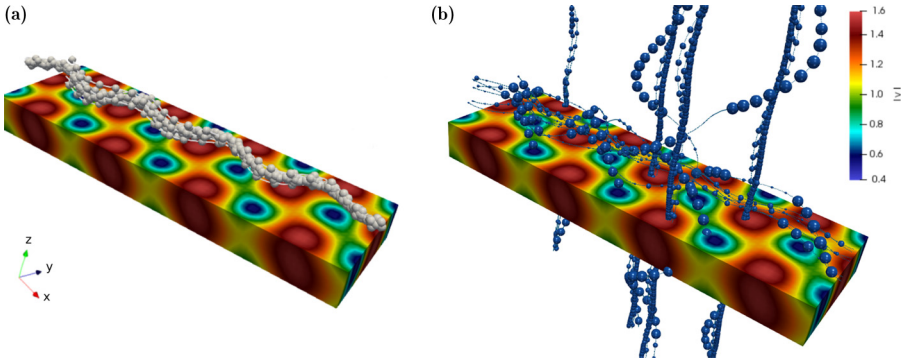

FIG. 10. Particle trajectories in an ABC flow ($4A = 2B = C = 1$) starting from an identical initial condition of a naive light particle (a) and a smart particle (b). The trajectories show that a smart particle manages to find the optimal vertical principal vortices independent of the initial condition, while the naive light particle sometimes get stuck in suboptimal horizontal principal vortices. The plotted particle sizes are proportional to the value of β. The vorticity field is color coded. To allow the visualization of particle trajectories, we have represented only a slice of it.

(also found by the smart particle) does not follow immediately from the distribution of vorticity, shown in Fig. 11.

Figures 12(a) and 12(d) and Figs. 12(b) and 12(e) show results from training with Ω_x and Ω_y as states. We find that if Ω_x is used as the state, the smart particle learns to outperform the naive light particle, using a nontrivial strategy [Fig. 12(d)]. If Ω_y is used as the state, on the other hand, the smart particle is not able to find a strategy that outperforms a light particle. This shows that it is important that the smart particle must measure appropriate information about the flow to be able to find a good strategy. Figure 11 shows the distribution of the components in the underlying ABC flow. The distribution of Ω_y is narrower than the distributions of Ω_x and Ω_z. This explains why it is hard to use Ω_y as the state in order to find regions with large $|\Omega|$.

D. Evaluation of the algorithm

In general, it is hard to evaluate the success of the reinforcement learning algorithm because the global optimal policy is not known. Because of the vast size of possible Q-value matrices, it is in general not possible to do a brute force approach, by testing all possible Q-value matrices. However, in the current problem it turns out that the algorithm is able to find nontrivial solutions also if the number of actions and number of states are reduced. We consider training using Ω_x as a state [the case in Figs. 12(a) and 12(d)], with a reduced number of actions $N_a = 2$ (β can take the values 0 or 3) and states $N_s = 11$. The results for 100 training sessions are displayed in Fig. 13(a). The general trend is nontrivial; the particle should be light for large $|\Omega_x|$, mainly heavy in a range of intermediate $|\Omega_x|$, and light again for small $|\Omega_x|$.

TABLE III. Normalized return $\tilde{R}_{\text{tot}}$ defined in (11) for the examination phase with Q-value matrix such that the optimal action $a^*(s)$ is 0 (heavy) for a number N^* of centered states and 3 (light) for the remaining states. As an example, $N^* = 9$ and $N_s = 21$ states used here gives $a^*(s) = \{3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3\}$.

N^*	19	17	15	13	11	9	7	5	3	1	0
$\tilde{R}_{\text{tot}}$	1.01	1.04	1.13	1.19	1.49	1.50	1.50	1.49	1.45	1.44	1.45

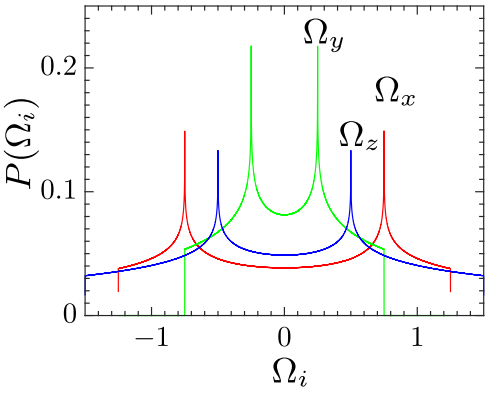

FIG. 11. Flow distribution of different components of $\mathbf{\Omega}$: Ω_x (red), Ω_y (green), and Ω_z (blue) for an ABC flow with strongly broken symmetry, $4A = 2B = C = 1$.

There are $2^{11} = 2048$ possible Q-value matrices. We evaluate the normalized total return $\tilde{R}_{\text{tot}}$ for each Q-value matrix averaging over 1000 episodes starting from 1000 predetermined initial conditions that are identical for each tested Q-value matrix. Figure 13(b) shows the distribution of $\tilde{R}_{\text{tot}}$ for all the 2048 possible Q-value matrices (green). Also displayed is the distribution of $\tilde{R}_{\text{tot}}$ obtained from the 100 policies underlying Fig. 13(a) (purple). We find that the policies obtained by reinforcement learning in general are close to the global optimal solution, and that in some instances the true global optimum is reached. The optimal normalized return is $\tilde{R}_{\text{tot}} = 1.49$, which is comparable to the best gain $\tilde{\Sigma} = 1.50$ for the case $N_a = 11$ and $N_s = 21$ [the data in Figs. 12(a) and 12(d)], but the case with fewer states and actions is more likely to get stuck at poor solutions with a lower return.

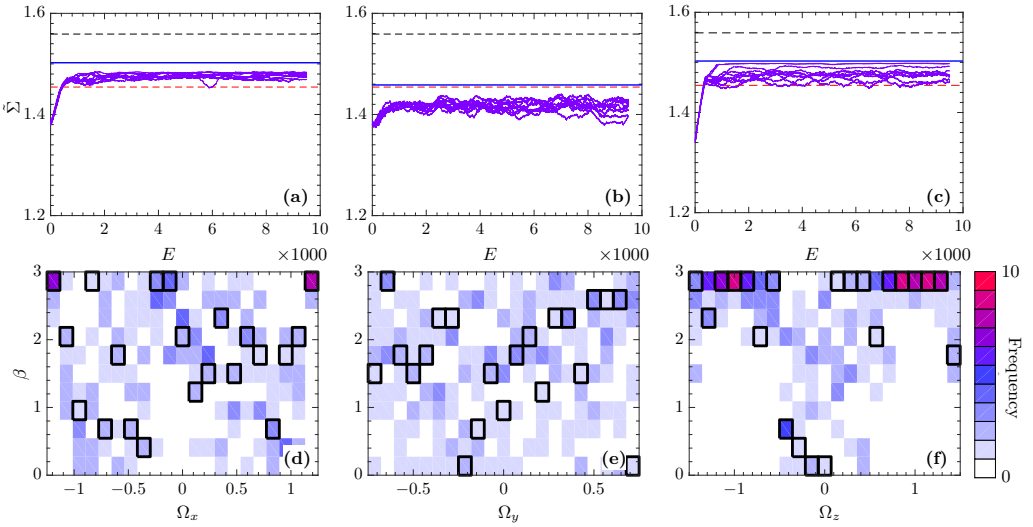

FIG. 12. Result from training of particles in an ABC flow with strongly broken symmetry ($4A = 2B = C = 1$). Data are displayed in the same manner as Fig. 9. [(a)–(c)] Normalized total gain $\tilde{\Sigma}(E)$ as a function of episode during training using the components Ω_x (a), Ω_y (b), and Ω_z (c) as state. [(d)–(f)] Corresponding frequency of optimal action for the final policies in panels (a)–(c).

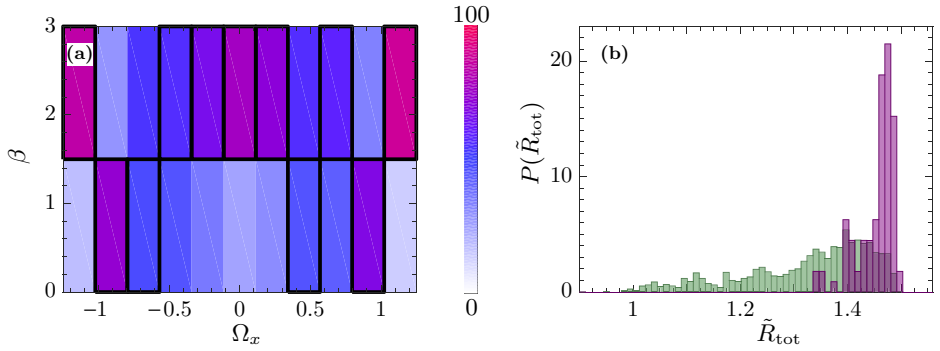

FIG. 13. Evaluation of the performance of smart particles in an ABC flow ($4A = 2B = C = 1$) with $N_a = 2$ actions ($\beta = 0$ or $\beta = 3$) and Ω_x divided into $N_s = 11$ states. (a) Solid black lines indicate the best policy obtained by the smart particles. This policy coincides with the global optimal policy for the set of states and actions used. Colors indicate frequency of the optimal action for each state based on 100 training sessions. (b) Green bars show distribution of average normalized return $\tilde{R}_{\rm tot}$ for all $N_s^{N_a} = 2048$ possible Q-value matrices. Purple bars show distribution of $\tilde{R}_{\rm tot}$ for the 100 training sessions.

VI. CONCLUSIONS

In this paper, we have shown how smart inertial particles can learn to sample the most intense vortical structures in fluid flows of different complexity: a two-dimensional stationary Taylor-Green like configuration, a two-dimensional time-dependent Taylor-Green-like flow, and three-dimensional ABC flows. We achieve this goal by defining the problem within the reinforcement learning framework and using the so-called one-step Q-learning algorithm to obtain approximately optimal policies in an iterative way. We evaluate the learning performance by comparing the acquired ability of smart particles to reach the target region with respect to the analogous skill of particles with fixed size. For the 2D examined flows, it is found that the learned policies allow outperforming of particles that cannot modulate their size or that can only do it rudimentarily. While the trajectories of smart particles have high density in the target region, this region is never sampled, or just rarely, in all the other instances. Even better strategies are obtained by adopting an ϵ-greedy algorithm, i.e., allowing additional exploration during the training session and letting the learning parameter α decrease in time to stabilize the found policies. Smart particles tend to elude positive vorticity and to accumulate in regions of intense negative vorticity even in the time-dependent flow. For the investigated three-dimensional flows, smart particles mainly outperform fixed-size particles except for the case of particles with nonsmall Stokes numbers in the slightly asymmetric ABC flow, for which the smart particles perform on the same level as naive light particles. In this context, it emerges more clearly how general the success of the reinforcement learning is, but also that achieving the predetermined goal depends on the properties particles can measure and on how much the target can be discriminated from noninteresting regions. Despite that a fully realistic description of the particle dynamics and the actual complexity of real flows is far from the goal of this paper, we provide a proof of concept for the possibility to engineer smart inertial particles and to make a case for the use of reinforcement learning algorithms for this purpose. Similar control strategies might also be implemented with heavy particles subjected only to the Stokes drag, as would be the case for diatom algae [20]. It is important to stress that the point particle approximation is only valid if the particle is small enough and that the approximation of one way coupling is only valid if the particle does not change state to frequently in physical time.

There is room for improvement in many directions. For instance, other ways to control the dynamics of engineered particles could be by changing their chirality or ellipsoidal structures. Moreover, other sensory inputs could be explored, one example being the temperature in convection.

ACKNOWLEDGMENTS

L.B., K.G., and S.C. acknowledge funding from the European Research Council under the European Union's Seventh Framework Programme, ERC Grant Agreement No. 339032. K.G. acknowledges funding from the Knut and Alice Wallenberg Foundation, Dnar. KAW 2014.0048.

APPENDIX

The flow domain is a square of size $L = 5\pi/4$ consisting of four quadrilaterals (two squares and two rectangles) of sides $L_1 = \pi$ and $L_2 = \pi/4$, $(L = L_1 + L_2)$. The velocity and vorticity fields are built from the following stream function, made out of the superposition of four different vorticity blobs placed at the center of each subdomain:

$$\begin{aligned}\psi(x,y) = {} & b_1 G_1(x) G_1(y) \sin\left(\frac{x\pi}{L_1}\right) \sin\left(\frac{y\pi}{L_1}\right) P(x) P(y) \\ & + b_2 G_2(x) G_1(y) \sin\left(\frac{x\pi}{L_2}\right) \sin\left(\frac{y\pi}{L_1}\right) P(y) \\ & + b_3 G_2(x) G_2(y) \sin\left(\frac{x\pi}{L_2}\right) \sin\left(\frac{y\pi}{L_2}\right) \\ & + b_4 G_1(x) G_2(y) \sin\left(\frac{x\pi}{L_1}\right) \sin\left(\frac{y\pi}{L_2}\right) P(x), \end{aligned} \tag{A1}$$

where

$$G_1(x) = \exp\left[-(x-\bar{x}_1)^2/\left(2\Delta_1^2\right)\right],$$
$$G_2(x) = \exp\left[-(x-\bar{x}_2)^2/\left(2\Delta_2^2\right)\right],$$

are the Gaussian functions that modulate the vortical structures with widths $\Delta_1 = L_1/4$, $\Delta_2 = L_2/4$ and centers in $\bar{x}_1 = -L_1/2$, $\bar{x}_2 = L_2/2$, and

$$P(x) = [x - \bar{x}_1 - (L - L_1/2)][x - \bar{x}_1 + (L - L_1/2)]$$

is a polynomial of degree 2 such that the orthogonal velocity component vanishes at the boundaries of the domain. The coefficients b_i are fixed as $(b_1, b_2, b_3, b_4) = (-0.1, 0.02, -0.10, 0-02)$ and determine the intensity of the vortical structures to be (approximately) $(5, -8, 5, -2)$ (see Fig. 2). Reflecting boundary conditions are used to confine the particles inside the volume.

[1] A. A. Mostafa and H. C. Mongia, On the modeling of turbulent evaporating sprays: Eulerian versus Lagrangian approach, Int. J. Heat Mass Transf. **30**, 2583 (1987).

[2] G. M. Faeth, Spray combustion phenomena, in *Symposium (International) on Combustion* (Elsevier, Amsterdam, 1996), pp. 1593–1612.

[3] P. Jenny, D. Roekaerts, and N. Beishuizen, Modeling of turbulent dilute spray combustion, Prog. Energy Combust. Sci. **38**, 846 (2012).

[4] A. Kovetz and B. Olund, The effect of coalescence and condensation on rain formation in a cloud of finite vertical extent, J. Atmos. Sci. **26**, 1060 (1969).

[5] R. Langer, New methods of drug delivery, Science **249**, 1527 (1990).

[6] E. Boeker and R. V. Grondelle, Dispersion of pollutants, in *Environmental Physics: Sustainable Energy and Climate Change* (John Wiley & Sons, New Jersey, 2011).

[7] F. Toschi and E. Bodenschatz, Lagrangian properties of particles in turbulence, Annu. Rev. Fluid Mech. **41**, 375 (2009).

[8] R. J. Adrian and J. Westerweel, *Particle Image Velocimetry* (Cambridge University Press, Cambridge, UK, 2011), Vol. 30.
[9] W. L. Shew, Y. Gasteuil, M. Gibert, P. Metz, and J.-F. Pinton, Instrumented tracer for Lagrangian measurements in Rayleigh-Bénard convection, Rev. Sci. Instrum. **78**, 065105 (2007).
[10] X. Ma, S. Jang, M. N. Popescu, W. E. Uspal, A. Miguel-López, K. Hahn, D.-P. Kim, and S. Sánchez, Reversed Janus micro/nanomotors with internal chemical engine, ACS Nano **10**, 8751 (2016).
[11] L. Baraban, D. Makarov, R. Streubel, I. Monch, D. Grimm, S. Sanchez, and O. G. Schmidt, Catalytic Janus motors on microfluidic chip: Deterministic motion for targeted cargo delivery, ACS Nano **6**, 3383 (2012).
[12] A. A. Solovev, E. J. Smith, C. C. B. Bufon, S. Sanchez, and O. G. Schmidt, Light-controlled propulsion of catalytic microengine, Angew. Chem. **123**, 11067 (2011).
[13] D. Walker, Micro autonomous underwater vehicle concept for distributed data collection, in *Proceedings of OCEANS* (2006), pp. 1–4.
[14] R. B. Wynn, V. A. I. Huvenne, T. P. Le Bas, B. J. Murton, D. P. Connelly, B. J. Bett, H. A. Ruhl, K. J. Morris, J. Peakall, and D. R. Parsons, Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience, Marine Geol. **352**, 451 (2014).
[15] E. Calzavarini, Y. X. Huang, F. G. Schmitt, and L. P. Wang, Propelled micro-probes in turbulence, Phys. Rev. Fluids **3**, 054604 (2018).
[16] T. C. Basso, M. Iovieno, S. Bertoldo, G. Perotto, A. Athanassiou, F. Canavero, G. Perona, and D. Tordella, Disposable radiosondes for tracking Lagrangian fluctuations inside warm clouds, *Antennas and Propagation in Wireless Communications (APWC), 2017 IEEE-APS Topical Conference* (2017), pp. 189–192.
[17] D. Roemmich, G. C. Johnson, S. Riser, R. Davis, J. Gilson, W. B. Owens, S. L. Garzoli, C. Schmid, and M. Ignaszewski, The Argo Program: Observing the global ocean with profiling floats, Oceanogr. **22**, 34 (2009).
[18] I. Durre, R. S. Vose, and D. B. Wuertz, Overview of the integrated global radiosonde archive, J. Clim. **19**, 53 (2006).
[19] S. Muiños-Landin, K. Ghazi-Zahedi, and F. Cichos, Reinforcement learning of artificial microswimmers, arXiv:1803.06425.
[20] G. Sarthou, K. R. Timmermans, S. Blain, and P. Tréguer, Growth physiology and fate of diatoms in the ocean: A review, J. Sea Res. **53**, 25 (2005).
[21] B. J. Gemmell, G. Oh, E. J. Buskey, and T. A. Villareal, Dynamic sinking behavior in marine phytoplankton: Rapid changes in buoyancy may aid in nutrient uptake, Proc. R. Soc. London B **283**, 20161126 (2016).
[22] A. E. Walsby, Gas vesicles, Microbiol. Rev. **58**, 94 (1994).
[23] J. Arrieta, A. Barreira, and I. Tuval, Microscale Patches of Nonmotile Phytoplankton, Phys. Rev. Lett. **114**, 128102 (2015).
[24] M. R. Maxey and J. J. Riley, Equation of motion for a small rigid sphere in a nonuniform flow, Phys. Fluids **26**, 883 (1983).
[25] J. Bec, Multifractal concentrations of inertial particles in smooth random flows, J. Fluid Mech. **528**, 255 (2005).
[26] E. Balkovsky, G. Falkovich, and A. Fouxon, Intermittent Distribution of Inertial Particles in Turbulent Flows, Phys. Rev. Lett. **86**, 2790 (2001).
[27] J. Bec, L. Biferale, A. S. Lanotte, A. Scagliarini, and F. Toschi, Turbulent pair dispersion of inertial particles, J. Fluid Mech. **645**, 497 (2010).
[28] L. Biferale, G. Boffetta, A. Celani, A. Lanotte, and F. Toschi, Particle trapping in three-dimensional fully developed turbulence, Phys. Fluids **17**, 021701 (2005).
[29] S. Douady, Y. Couder, and M. E. Brachet, Direct Observation of the Intermittency of Intense Vorticity Filaments in Turbulence, Phys. Rev. Lett. **67**, 983 (1991).
[30] K. D. Squires and J. K. Eaton, Preferential concentration of particles by turbulence, Phys. Fluids A **3**, 1169 (1991).
[31] J. K. Eaton and J. R. Fessler, Preferential concentration of particles by turbulence, Int. J. Multiphase Flow **20**, 169 (1994).

[32] R. Monchaux, M. Bourgoin, and A. Cartellier, Analyzing preferential concentration and clustering of inertial particles in turbulence, Int. J. Multiphase Flow **40**, 1 (2012).
[33] J. Bec, L. Biferale, M. Cencini, A. Lanotte, S. Musacchio, and F. Toschi, Heavy Particle Concentration in Turbulence at Dissipative and Inertial Scales, Phys. Rev. Lett. **98**, 084502 (2007).
[34] G. Boffetta, F. De Lillo, and A. Gamba, Large scale inhomogeneity of inertial particles in turbulent flows, Phys. Fluids **16**, L20 (2004).
[35] L. Biferale, A. Scagliarini, and F. Toschi, On the measurement of vortex filament lifetime statistics in turbulence, Phys. Fluids **22**, 065101 (2010).
[36] R. Ž Milenković, B. Sigg, and G. Yadigaroglu, Bubble clustering and trapping in large vortices, part 2: Time-dependent trapping conditions, Int. J. Multiphase Flow **33**, 1111 (2007).
[37] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Flow Navigation by Smart Microswimmers Via Reinforcement Learning, Phys. Rev. Lett. **118**, 158004 (2017).
[38] 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).
[39] 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).
[40] M. Gazzola, B. Hejazialhosseini, and P. Koumoutsakos, Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers, SIAM J. Sci. Comput. **36**, B622 (2014).
[41] G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. van Rees, and P. Koumoutsakos, Synchronisation through learning for two self-propelled swimmers, Bioinspir. Biomim. **12**, 036001 (2017).
[42] S. Verma, G. Novati, and P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, arXiv:1802.02674.
[43] G. Reddy, A. Celani, T. J. Sejnowski, and M. Vergassola, Learning to soar in turbulent environments, Proc. Natl. Acad. Sci. USA **113**, E4877 (2016).
[44] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 2017).
[45] G. Tesauro, Temporal difference learning and TD-Gammon, Commun. ACM **38**, 58 (1995).
[46] L. Kaelbling, L. Pack, L. Michael, and A. W. Moore, Reinforcement learning: A survey, J. Artif. Intell. Res. **4**, 237 (1996).
[47] T. R. Auton, J. C. R. Hunt, and M. Prud'Homme, The force exerted on a body in inviscid unsteady non-uniform rotational flow, J. Fluid Mech. **197**, 241 (1988).
[48] A. Babiano, J. H. E. Cartwright, O. Piro, and A. Provenzale, Dynamics of a Small Neutrally Buoyant Sphere in a Fluid and Targeting in Hamiltonian Systems, Phys. Rev. Lett. **84**, 5764 (2000).
[49] R. Gatignol, The Faxén formulas for a rigid particle in an unsteady non-uniform Stokes flow, J. Mec. Theor. Appl. **2**, 143 (1983).
[50] C. J. C. H. Watkins and P. Dayan, Q-learning, Mach. Learn. **8**, 279 (1992).
[51] T. Dombre, U. Frisch, J. M. Greene, M. Hénon, A. Mehr, and A. M. Soward, Chaotic streamlines in the abc flows, J. Fluid Mech. **167**, 353 (1986).