# 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, The Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34151, Trieste, Italy.*

(Dated: November 17, 2017)

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 behaviour 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 helical flow. We show that smart particles are able to learn how to reach extremely intense vortical structures in all the tackled cases.



[*] simona.colabrese@roma2.infn.it

## 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 combustion processes, drug delivery, dispersion of pollutants or contaminants in the environment, spray formation and rain formation in clouds, to cite just a few cases [1–5]. 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 Particles Image Velocimetry techniques [6, 7]. Small particles have been instrumented to perform local measurements of flow properties [8]. In this paper we present a numerical study to show how one can implement a suitable learning algorithm to train micro particles 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. There are many advantages to have such smart particles in a flow. One can for example use them to measure specific flow properties, to actively deliver drugs only in particular flow regions or to control the flow by local feedbacks.

It is well known that particles that are heavier or lighter than the fluid systematically detach from the flow streamlines [9–13]. 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 [14–19].

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 [20, 21].

Differently, in this paper we seek for inertial particles capable of sampling ad-hoc specific flow structures. We imagine our smart particles to be endowed with the ability of obtaining 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: 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 behavioural 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 learnt 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 micro-swimmers in Taylor Green flows [22], in ABC flows [23] and for the case of fish schooling in still water [24]. A similar approach has also been pioneered in [25] for the case of birds that can exploit warm thermals to soar in a turbulent environment. We show that reinforcement learning provides a way to construct efficient strategies by accumulating experience also for the cases investigated here.

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

## II. INERTIAL PARTICLES

We focus the discussion to the case of small spherical inertial particles that have a density that is different from the surrounding fluid. In the absence of controls, their dynamics is entirely governed by the flow. By changing their size, the particles may be able to alter their dynamics to navigate the flow. 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 point-like. 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 [9, 26–28]

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

Here dots denote time derivatives, $\mathbf{X}$ and $\mathbf{V}$ are dimensionless particle position and velocity, $\boldsymbol{\eta}(t)$ is a Gaussian white noise introduced to break structurally unstable dynamics, $\langle \eta_i(t)\eta_j(t')\rangle = \delta_{ij}\delta(t-t')$, and $\mathbf{u}$ 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. 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 $\tau$ and the particle response time:

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

where $\nu$ 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 $\rho_\mathrm{f}$. The value $\beta = 1$ distinguishes particles that are heavier ($\beta < 1$) or lighter ($\beta > 1$) than the fluid. The dimensionless translational diffusivity $\chi$ 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 $\mathbf{V} = \mathbf{u} + O(\mathrm{St}(1-\beta)) + O(\chi)$.

## III. REINFORCEMENT LEARNING APPROACH

To identify efficient strategies to sample high-vorticity regions in complex flows, we used reinforcement learning implementing the *one-step Q-learning* algorithm [29]. The general 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. When the state of the system changes to a new state $s_{n+1}$, the agent is given a *reward* $r_{n+1}$. The reward quantifies the immediate success of the previously chosen action to reach the target goal. 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.

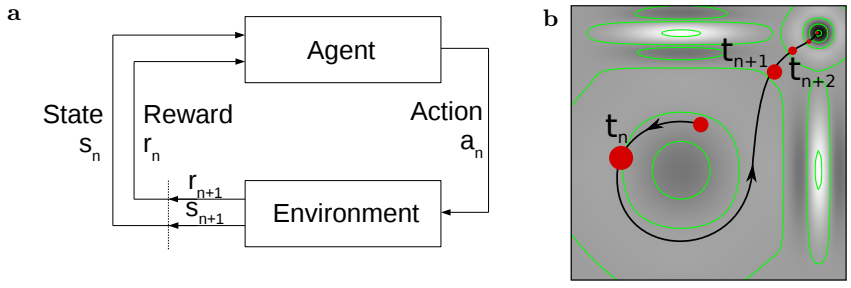


FIG. 1. **a** Sketch of the reinforcement learning agent-environment interactions. **b** Graphical summary of a typical trajectory for a smart particle in a generic two-dimensional flow. State changes occur at times $t_n$, $t_{n+1}$, $t_{n+2}$, ... when the particle enters new vorticity regions that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

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 $\Omega_z$ of the local flow vorticity $\mathbf{\Omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$. It can distinguish a discrete number $N_s$ of coarse grained states corresponding to equally spaced intervals of $\Omega_z$ in the full range of $\Omega_z$ 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:

$$a_n \in \mathcal{A} = \{b_1, b_2, \ldots, b_{N_a}\}\,.$$

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 $\pi$ 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, $\pi^*$, iteratively by introduction of intermediate policies $\pi_n$ during the $n$:th time step, such that $\lim_{n\to\infty} \pi_n = \pi^*$. The policy $\pi_n$ is derived from the *Q-value*, $Q_\pi(s_n, a_n)$, according to the *$\epsilon$-greedy* rule: select the action that maximizes the current $Q$-value function except for a small probability $\epsilon$ of randomly selecting another action independent of the $Q$-value:

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

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 $\pi_n$:

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

The parameter $\gamma$ in Eq. (6) is the *discount factor*, $0 \le \gamma < 1$. The value of $\gamma$ changes the resulting strategy: with a myopic evaluation ($\gamma$ close to 0), the approximately optimal policy greedily maximizes only the immediate reward. As $\gamma$ 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 [29]:

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

where $\alpha$ is a parameter that tunes the learning rate. For Markovian systems and if $\epsilon$ 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 assign to each state the action maximizing the expected long term accumulated reward [29]. 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 $N_\mathrm{E}$ 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 to encourage exploration and to avoid to be 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. For the purpose 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 this policy to evaluate the long-term accumulative discounted reward, the *return*:

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

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 $R_\mathrm{tot}$ can also be used during training to evaluate the success of the intermediate policies $\pi_n$.

## 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 intensity and sign, singled out by appropriate Gaussian functions (see Fig. 2**a** and Appendix I for a detailed description).

### A. Algorithm implementation

We divide the scalar vorticity field into $N_s = 21$ equally spaced states as sketched in Fig. 2**a**. The possible actions

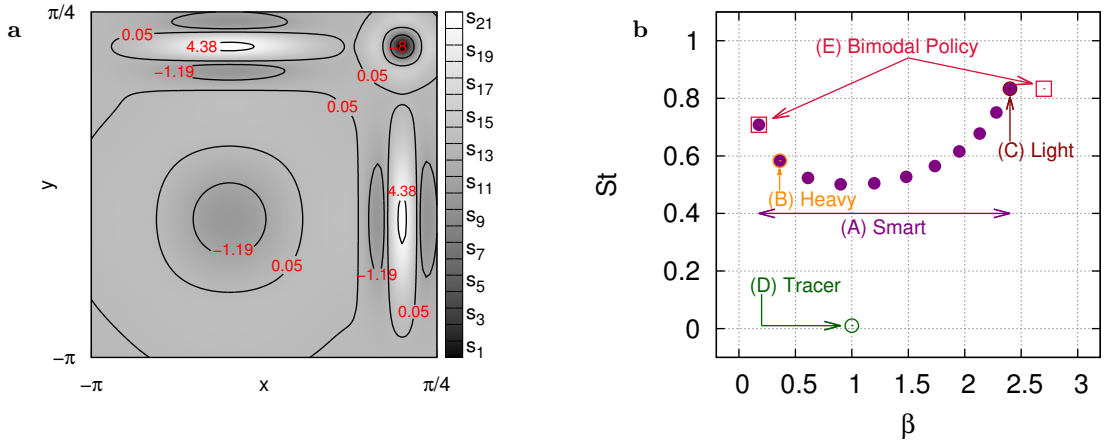


FIG. 2. **a** An illustrative sample of the isolines delimiting the $N_s = 21$ vorticity states. Values corresponding to the underlying vorticity isolines are shown in red. **b** Set of pairs (St,$\beta$) corresponding to the selection of the $N_a = 11$ actions. Smart particles are allowed to select each one of the available actions freely (A). We also highlight four different cases corresponding to four possible naive strategies: (B)-Heavy particle with ($\mathrm{St} = 0.58, \beta = 0.36$) ; (C)-Light particle with ($\mathrm{St} = 0.83, \beta = 2.4$); (D)-Tracer with ($\mathrm{St} = 0.01, \beta = 1$); (E) simple bimodal policy with only two actions: the particle has one constant low density ($\mathrm{St} = 0.83, \beta = 2.7$) [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).

the smart particle can take are chosen among $N_a = 11$ equidistributed sizes in the range $0.5 < b < 2$. The values of St and $\beta$ in Eqs. (2-4) corresponding to these sizes are illustrated in Fig. 2**b**. In Table I we report all values for $\beta$ and St for each one of the $N_a = 11$ different sizes. We contrast the smart particle to naive particles with simple strategies

| | $b$ | $\beta$ | St | |
|---|---|---|---|---|
| $a_1$ | 0.5 | 0.176 | 0.708 | Heavy |
| $a_2$ | 0.65 | 0.362 | 0.583 | ” |
| $a_3$ | 0.8 | 0.611 | 0.523 | ” |
| $a_4$ | 0.95 | 0.900 | 0.501 | ” |
| $a_5$ | 1.1 | 1.20 | 0.505 | Light |
| $a_6$ | 1.25 | 1.48 | 0.527 | ” |
| $a_7$ | 1.4 | 1.74 | 0.565 | ” |
| $a_8$ | 1.55 | 1.95 | 0.615 | ” |
| $a_9$ | 1.7 | 2.13 | 0.678 | ” |
| $a_{10}$ | 1.85 | 2.28 | 0.751 | ” |
| $a_{11}$ | 2 | 2.40 | 0.833 | ” |

TABLE I. Set of parameters corresponding to the $N_a = 11$ actions.

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).

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:

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

where the minus sign is used to target negative vorticity regions. Due to the presence of different peaks in the vorticity distribution the task is non-trivial. 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 Section III with mainly fixed learning rate $\alpha = 0.1$ and $\epsilon = 0$ (for a case with time-dependent $\alpha$ and $\epsilon$ see Section IV B 1). The other fixed parameters are $\gamma = 0.999$ and $\chi = 0.005$. To enhance exploration, the initial $Q$-value matrix coefficients are made equal to the undiscounted return that a particle would gain if it was in the target region during the entire length of the episode: $Q(s_0, a_0) = -\Omega^3_{\min} N$ for all $(s_0, a_0)$.

### 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

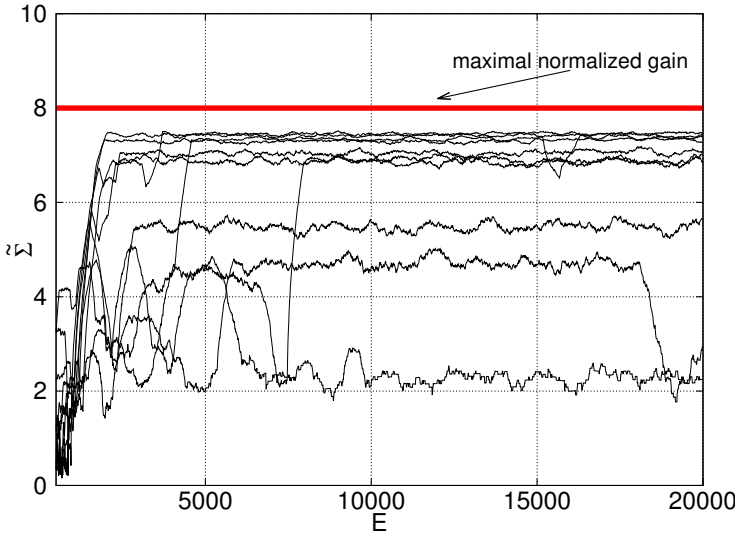


FIG. 3. Dependence of the normalized total gain, $\tilde{\Sigma}$ in Eq. (10) on the episode $E$ for ten different learning processes (black curves). Every point represents an average over a sliding window of 500 episodes. The red line shows the maximal possible reward.

and eventually achieves to reach the smallest vortex for most initial conditions and realisations of the white noise $\boldsymbol{\eta}$. Fig. 3 shows the evolution of the learning gain as a function of the episode during the training process for ten different trials. We observe that the 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 due to the relatively small exploration due to the choice of initial condition and the realisation of the noise $\boldsymbol{\eta}$.

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**. We observe that the trajectories of smart particles have high density in

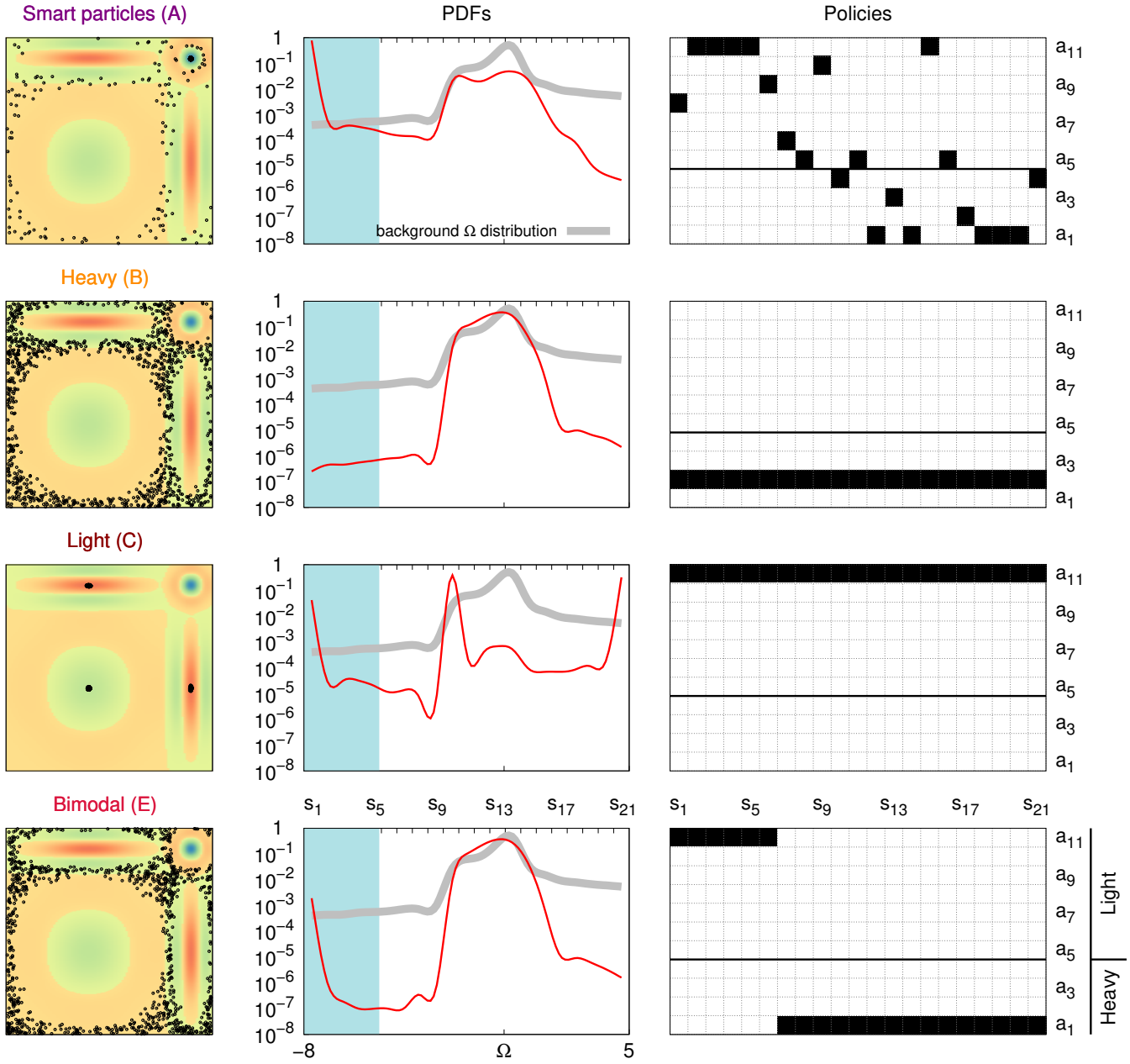


FIG. 4. Left column: Particle positions along 10 representative trajectories plotted after an initial transient for the cases studied in Fig. 2**b** (except for tracer particles which fill space uniformly). From top to bottom: Smart-particles (A), Heavy (B), Light (C), simple bimodal policy (E). Middle 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.

the target region, which isn't 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

| | | (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 |

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. (14). Second row: the same as the first row but for the time dependent case (T-TG) discussed in Sec. (IV B 2).

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). 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

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

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 $\gamma$.

### 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 non-greedy action ($\epsilon > 0$). The training phase starts with an initial positive value of $\epsilon$ 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. This leads to a fixation of the approximately optimal strategy. A particularly simple and efficient scheme is to decrease both the learning rate $\alpha$ and the exploration rate $\epsilon$ as a function of the episode, $E$, during the training:

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

where $\sigma$ and $\delta$ 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 $\delta = 1/10000$. 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 $\epsilon$-greedy method one might need to fine tune the parameters of the learning protocol, $\epsilon_o, \alpha_0, \sigma, \delta$ and that in most cases there is not an priori obvious choice. This is due to the fact that there exists a trade-off between the advantage of taking locally sub-optimal actions and the risk to drift 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 of 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 $\gamma$ but keeping all other training parameters constant. As one can see, the normalized gain remains of the same order when $\gamma$ is varied one order of magnitude. Only if the discount is too myopic ($\gamma \ll 0$) or too far-sighted ($\gamma \sim 1$) the algorithm fails. 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 optimal behaviour, 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

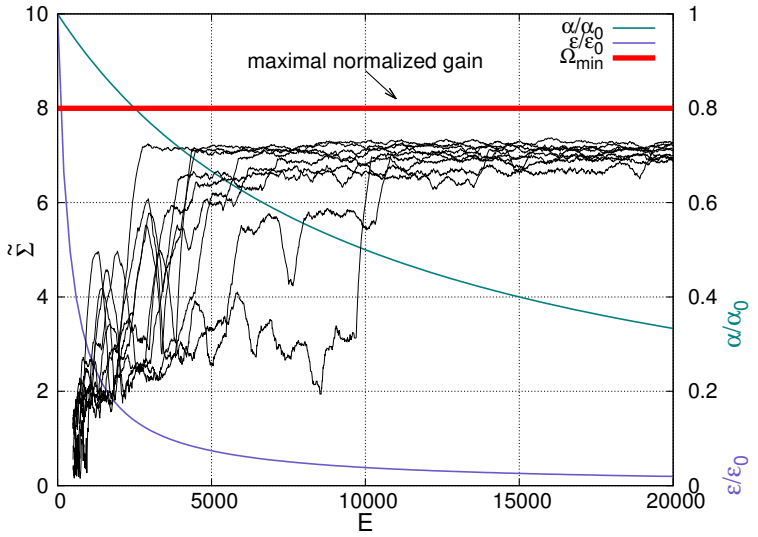


FIG. 5. The normalized learning gain, $\tilde{\Sigma}(E)$ in Eq. (10) against the episodes $E$ for the $\epsilon$-greedy reinforcement learning algorithm. The representation is comparable to Fig. 3 (scale on left vertical axis) . The two continuous curves represent the adiabatic decreasing of both the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

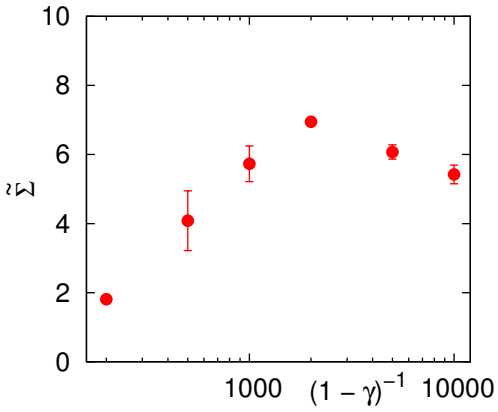


FIG. 6. The normalized learning gain $\tilde{\Sigma}(E)$ as a function of the time horizon of the learning process $(1-\gamma)^{-1}$ averaged over samples of ten different learning trials. Error bars are estimated on the basis of the scatter inside each sample.

two-dimensional flow in such a way that the four coefficients $b_1,\ldots,b_4$ building up the four-vortex flow (see Eq. (14) in Appendix I) acquires an out-of-phase oscillating behavior $b_i(t) = \cos(\omega_0 t + \phi_i)$, where $\omega_0$ is a constant angular frequency and $\phi_i$ 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 $(\overline{-\Omega_{\rm min}^3})^{1/3} \approx 5.2$. Even in 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 non-trivial 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

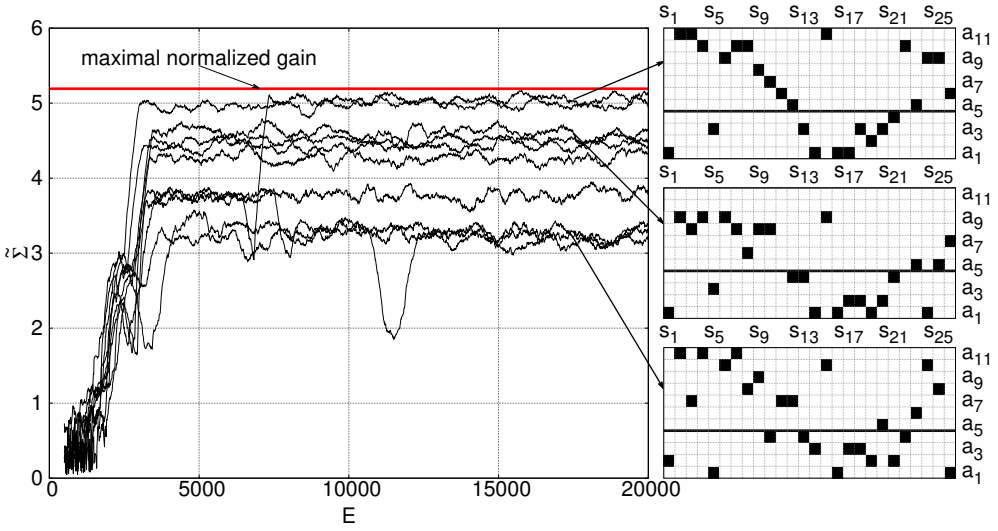


FIG. 7. Dependence of the normalized learning gain, $\tilde{\Sigma}$, for ten different learning processes (black curves) on the two-dimensional time-dependent flow. Every point represents an average over a sliding window of 500 episodes. The three policies shown to the right are for the cases with best,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

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 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 Arnlod-Beltrami-Childress (ABC) flow has been the subject of many studies in turbulence theory. Its Eulerian velocity field is (in dimensionless coordinates):

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

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

Numerical simulations and theoretical arguments show that the ABC flow has tube-like regions in space within which the streamlines of the flow are confined and the velocity is essentially one-dimensional [30]. Since vorticity $\boldsymbol{\Omega} \equiv \boldsymbol{\nabla} \wedge \boldsymbol{u}$ is parallel to the velocity in an ABC flow, $\boldsymbol{\Omega} = \boldsymbol{u}/2$, these tubes are referred to as principal vortices. Due to 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 Section IV, we impose here as a target for the smart particle to maximize the magnitude of its vorticity $\Omega \equiv |\boldsymbol{\Omega}|$. 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 Section IV. We keep $\mathrm{St} = 0.2$ fixed, and use as an action to change the value of $\beta$. Allowed values are equally distributed in $N_a$ levels between 0 and 3. The state of the particle is given by either $|\boldsymbol{\Omega}|$ or one of the components of $\boldsymbol{\Omega}$, equally partitioned in $N_s$ levels between the minimal and maximal value that can be

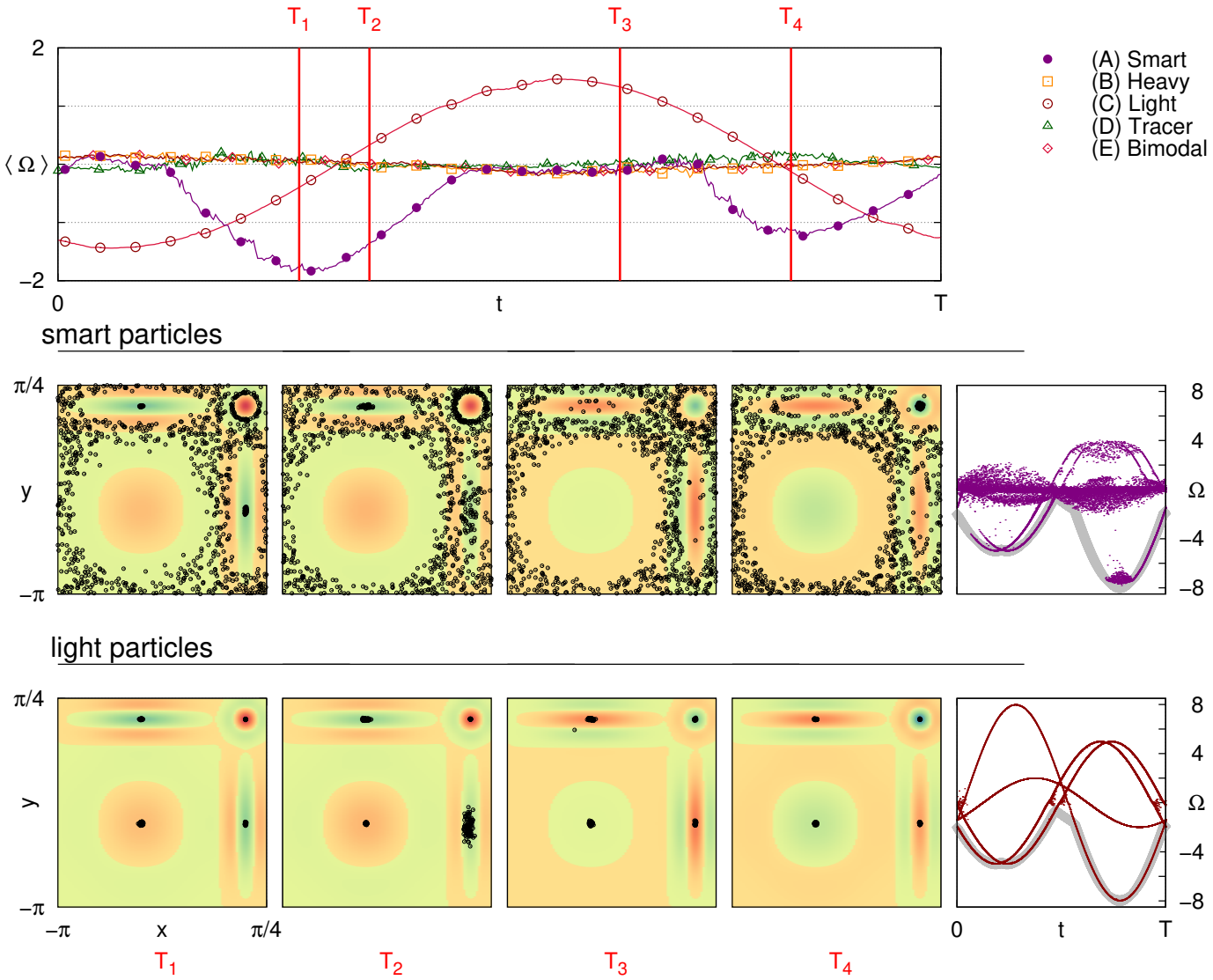


FIG. 8. Upper row: average Ω sampled along the time evolution of 100 particles for each analyzed type: A-smart (purple), B-heavy (orange), C-light (red), D-tracer (green) and E-particle with bimodal policy (magenta). The average vorticity is periodic with the period $T = 2\pi/\omega_0$ of the underlying 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 scattere plots of Ω for smart particles (top) and light ones(bottom) throughout the entire period. The shadowed grey curve represents the instantaneous maximum negative vorticity throughout the flow.

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 Section IV, where the cubed vorticity was evaluated at the position of the state change. As in Section 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

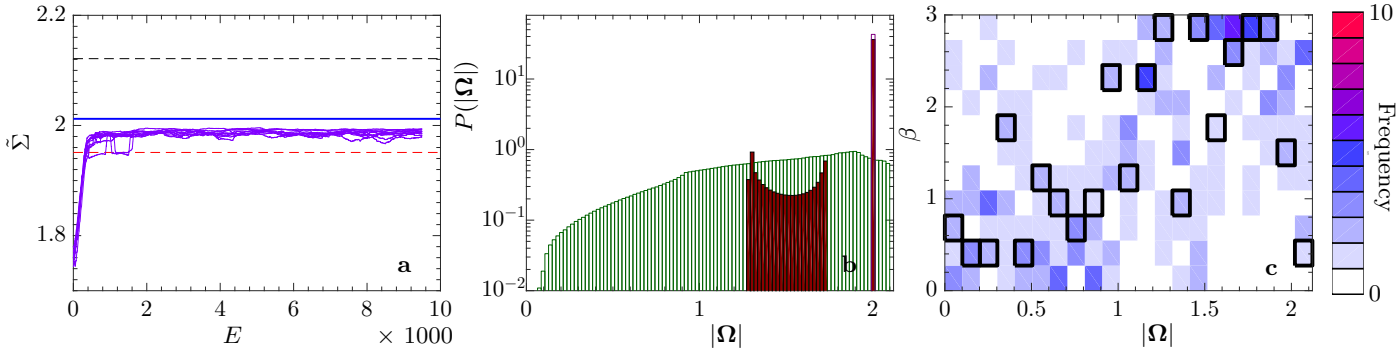


FIG. 9. Result from training of particles in ABC flow with weakly broken symmetry ($2A = B = C = 1$). **a** Normalized total gain $\tilde{\Sigma} = \langle |\mathbf{\Omega}|^3 \rangle^{1/3}$ [Eq. (10)] as a function of episode $E$. The maximal vorticity in the flow is shown as black-dashed. Red-dashed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

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

For non-small 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 $|\Omega| = 1.5$ 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

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

TABLE III. Normalized return $\tilde{R}_{\rm tot}$ definded 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 $a^*(s) = \{3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3\}$.

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 $\Omega_z$ 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.

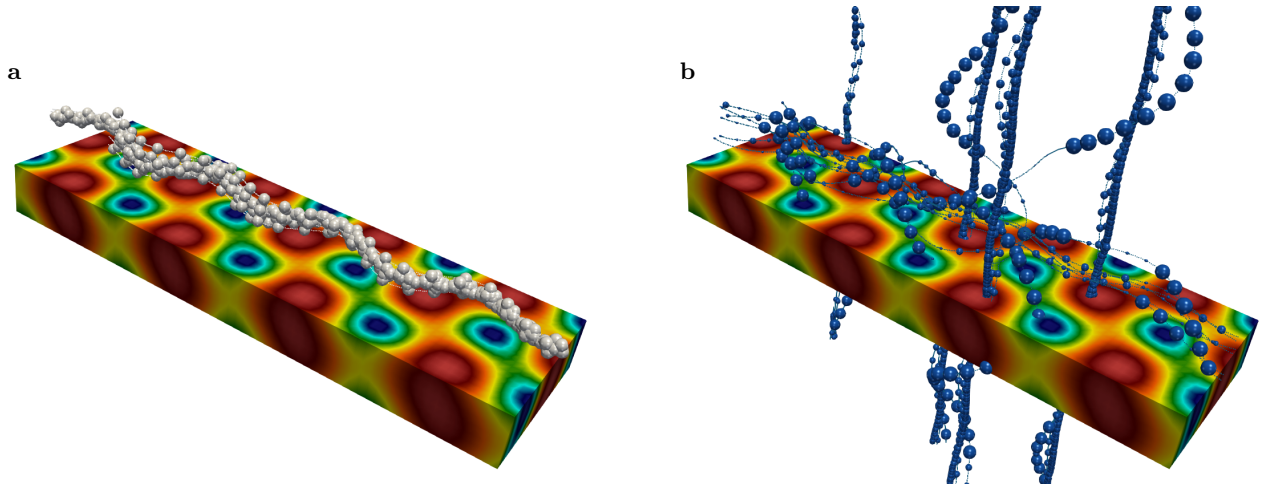


FIG. 10. Particle trajectories in an ABC flow ($4A = 2B = C = 1$) starting from an identical initial condition of a naive light particle (**a**) and a smart particle **b**. The trajectories show that a smart particle manages to find the optimal vertical principal vortices independent of the initial condition, while the naive light particle sometimes get stuck in suboptimal horizontal principal vortices. The plotted particle sizes are proportional to the value of $\beta$

In Fig. 10 we observed that the smart particles using $\Omega_z$ as state recognizes 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 is shown in Fig. 12**c**,**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 $N^* = 7 \sim 9$, 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 (also found by the smart particle) does not follow immediately from the distribution of vorticity, shown in Fig. 11.

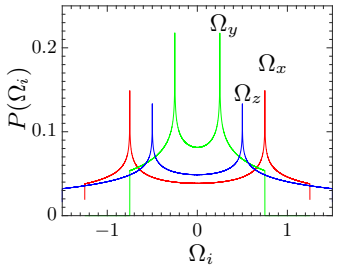


FIG. 11. Flow distribution of different components of $\Omega$: $\Omega_x$ (red), $\Omega_y$ (green), $\Omega_z$ (blue) for an ABC flow with strongly broken symmetry, $4A = 2B = C = 1$.

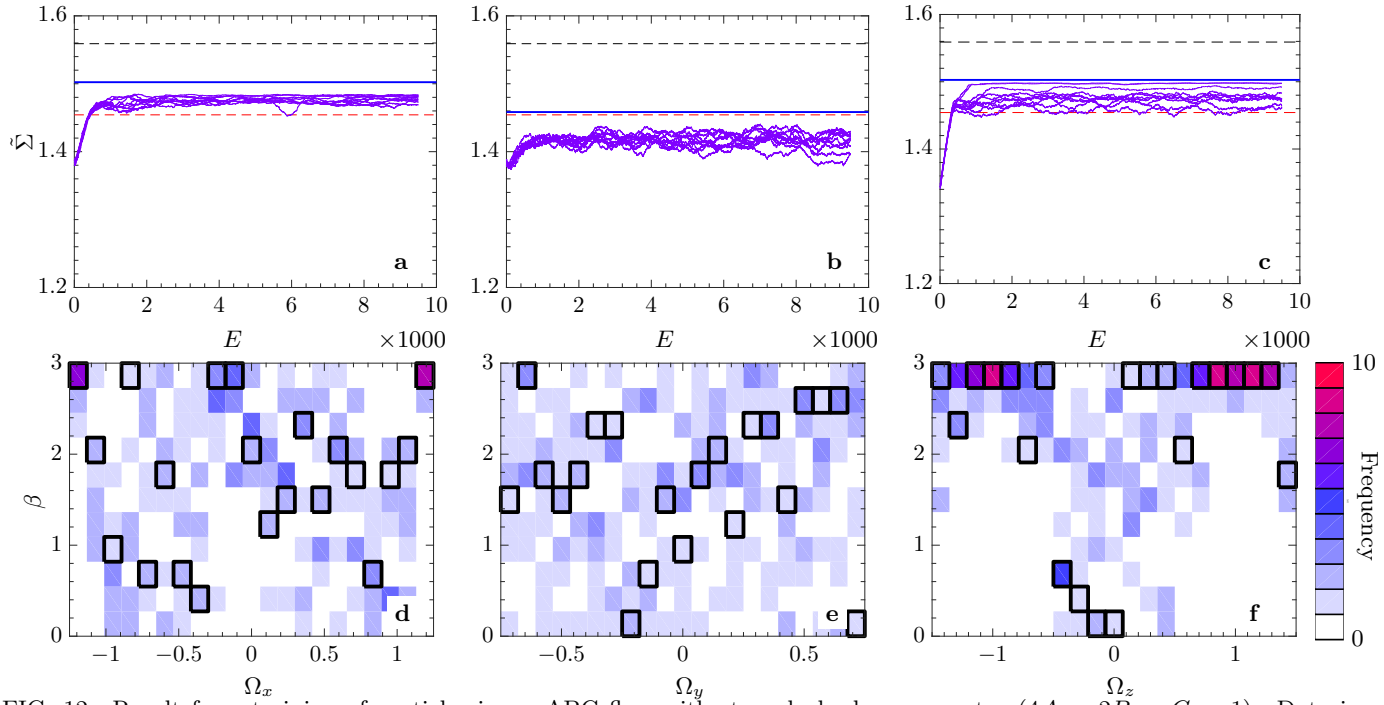


FIG. 12. Result from training of particles in an ABC flow with strongly broken symmetry ($4A = 2B = C = 1$). Data is displayed in the same manner as Fig. 9. **a**–**c** Normalized total gain $\tilde{\Sigma}(E)$ as a function of episode during training using the components $\Omega_x$ (**a**), $\Omega_y$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

Fig. 12**a**,**d** and Fig. 12**b**,**e** show results from training with $\Omega_x$ and $\Omega_y$ as states. We find that if $\Omega_x$ is used as the state, the smart particle learns to outperform the naive light particle, using a non-trivial strategy (Fig. 12**d**). If $\Omega_y$ 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. Fig. 11 shows the distribution of the components in the underlying ABC flow. The distribution of $\Omega_y$ is narrower than the distributions of $\Omega_x$ and $\Omega_z$. This explains why it is hard to use $\Omega_y$ as the state in order to find regions with large $|\mathbf{\Omega}|$.

### 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. Due to 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 non-trivial solutions also if the number of actions and number of states are reduced. We consider training using $\Omega_x$ as a state (the case in Fig. 12**a**,**d**), with a reduced number of actions $N_a = 2$ ($\beta$ 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 non-trivial, the particle should be light for large $|\Omega_x|$, mainly heavy in a range of intermediate $|\Omega_x|$ and light again for small $\Omega_x$.

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

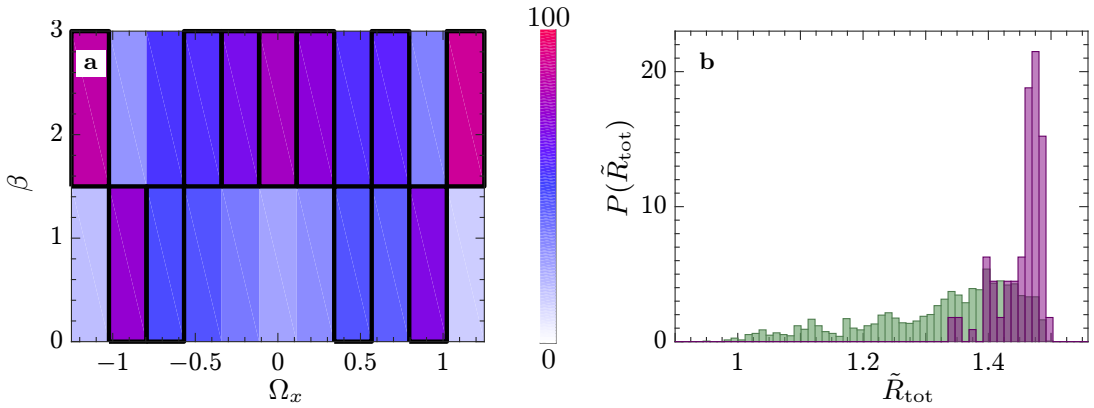


FIG. 13. Evaluation of the performance of smart particles in an ABC flow ($4A = 2B = C = 1$) with $N_a = 2$ actions ($\beta = 0$ or $\beta = 3$) and $\Omega_x$ divided into $N_s = 11$ states. **a** Solid black lines indicates the best policy obtained by the smart particles. This policy coincides with 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 approximally 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 learnt 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 $\epsilon$-greedy algorithm, i.e. allowing additional exploration during the training session and letting the learning parameter $\alpha$ 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 non-small 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 non-interesting 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. 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 LB, KG and SC acknowledge funding from the European Research Council under the European Unions Seventh Framework Programme, ERC Grant Agreement No 339032.

## VII. APPENDIX I

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:

$$\begin{aligned}\psi(x,y) &= b_1 G_1(x) G_1(y) \sin\left(\frac{x\pi}{L_1}\right) \sin\left(\frac{y\pi}{L_1}\right) P(x)P(y) \\ &+ b_2 G_2(x) G_1(y) \sin\left(\frac{x\pi}{L_2}\right) \sin\left(\frac{y\pi}{L_1}\right) P(y) \\ &+ b_3 G_2(x) G_2(y) \sin\left(\frac{x\pi}{L_2}\right) \sin\left(\frac{y\pi}{L_2}\right) \\ &+ b_4 G_1(x) G_2(y) \sin\left(\frac{x\pi}{L_1}\right) \sin\left(\frac{y\pi}{L_2}\right) P(x), \end{aligned} \tag{14}$$

where

$$\begin{aligned} G_1(x) &= \exp(-(x-\bar{x}_1)^2/(2\Delta_1^2)), \\ G_2(x) &= \exp(-(x-\bar{x}_2)^2/(2\Delta_2^2)), \end{aligned}$$

are the Gaussian functions that modulate the vortical structures with widths $\Delta_1 = L_1/4$, $\Delta_2 = L_2/4$ and centers in $\overline{x}_1 = -L_1/2$, $\overline{x}_2 = L_2/2$ 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] AA Mostafa and HC Mongia. On the modeling of turbulent evaporating sprays: Eulerian versus lagrangian approach. *International Journal of Heat and Mass Transfer*, 30(12):2583–2593, 1987.

[2] Chung K Law. *Combustion physics*. Cambridge university press, 2010.

[3] A Kovetz and B Olund. The effect of coalescence and condensation on rain formation in a cloud of finite vertical extent. *Journal of the Atmospheric Sciences*, 26(5):1060–1065, 1969.

[4] Robert Langer. New methods of drug delivery. *Science*, pages 1527–1533, 1990.

[5] Egbert Boeker and Rienk Van Grondelle. Dispersion of pollutants. *Environmental Physics: Sustainable Energy and Climate Change, Third Edition*, pages 261–335.

[6] Federico Toschi and Eberhard Bodenschatz. Lagrangian properties of particles in turbulence. *Annual Review of Fluid Mechanics*, 41:375–404, 2009.

[7] Ronald J Adrian and Jerry Westerweel. *Particle image velocimetry*. Number 30. Cambridge University Press, 2011.

[8] Woodrow L Shew, Yoann Gasteuil, Mathieu Gibert, Pascal Metz, and Jean-François Pinton. Instrumented tracer for lagrangian measurements in rayleigh-bénard convection. *Review of scientific instruments*, 78(6):065105, 2007.

[9] Martin R Maxey and James J Riley. Equation of motion for a small rigid sphere in a nonuniform flow. *The Physics of Fluids*, 26(4):883–889, 1983.

[10] Jérémie Bec. Multifractal concentrations of inertial particles in smooth random flows. *Journal of Fluid Mechanics*, 528:255–277, 2005.

[11] E Balkovsky, Gregory Falkovich, and A Fouxon. Intermittent distribution of inertial particles in turbulent flows. *Physical Review Letters*, 86(13):2790, 2001.

[12] J Bec, L Biferale, AS Lanotte, Andrea Scagliarini, and F Toschi. Turbulent pair dispersion of inertial particles. *Journal of Fluid Mechanics*, 645:497–528, 2010.

[13] Luca Biferale, Guido Boffetta, Antonio Celani, Alessandra Lanotte, and Federico Toschi. Particle trapping in three-dimensional fully developed turbulence. *Physics of Fluids*, 17(2):021701, 2005.

[14] S Douady, Y Couder, and ME Brachet. Direct observation of the intermittency of intense vorticity filaments in turbulence. *Physical Review Letters*, 67(8):983, 1991.

[15] Kyle D Squires and John K Eaton. Preferential concentration of particles by turbulence. *Physics of Fluids A: Fluid Dynamics*, 3(5):1169–1178, 1991.

[16] John K Eaton and JR Fessler. Preferential concentration of particles by turbulence. *International Journal of Multiphase Flow*, 20:169–209, 1994.

[17] Romain Monchaux, Mickael Bourgoin, and Alain Cartellier. Analyzing preferential concentration and clustering of inertial particles in turbulence. *International Journal of Multiphase Flow*, 40:1–18, 2012.

[18] Jeremie Bec, Luca Biferale, Massimo Cencini, Alessandra Lanotte, Stefano Musacchio, and Federico Toschi. Heavy particle concentration in turbulence at dissipative and inertial scales. *Physical review letters*, 98(8):084502, 2007.

[19] G Boffetta, F De Lillo, and A Gamba. Large scale inhomogeneity of inertial particles in turbulent flows. *Physics of Fluids*, 16(4):L20–L23, 2004.
[20] Enrico Calzavarini, Thomas H Van den Berg, Stefan Luther, Federico Toschi, and Detlef Lohse. Microbubble clustering in turbulent flow. Technical report, 2006.
[21] Rade Ž Milenković, Beat Sigg, and George Yadigaroglu. Bubble clustering and trapping in large vortices. part 2: Time-dependent trapping conditions. *International Journal of Multiphase Flow*, 33(10):1111–1125, 2007.
[22] Simona Colabrese, Kristian Gustavsson, Antonio Celani, and Luca Biferale. Flow navigation by smart microswimmers via reinforcement learning. *Phys. Rev. Lett.*, 118:158004, 2017.
[23] Kristian Gustavsson, Luca Biferale, Antonio Celani, and Simona Colabrese. Finding efficient swimming strategies in a three dimensional chaotic flow by reinforcement learning. *Arxiv preprint*, arXiv:1701.08848, 2017.
[24] Mattia Gazzola, Andrew A Tchieu, Dmitry Alexeev, Alexia de Brauer, and Petros Koumoutsakos. Learning to school in the presence of hydrodynamic interactions. *Journal of Fluid Mechanics*, 789:726–749, 2016.
[25] Gautam Reddy, Antonio Celani, Terrence J Sejnowski, and Massimo Vergassola. Learning to soar in turbulent environments. *Proceedings of the National Academy of Sciences*, page 201606075, 2016.
[26] TR Auton, JCR Hunt, and M Prud'Homme. The force exerted on a body in inviscid unsteady non-uniform rotational flow. *Journal of Fluid Mechanics*, 197:241–257, 1988.
[27] Armando Babiano, Julyan HE Cartwright, Oreste Piro, and Antonello Provenzale. Dynamics of a small neutrally buoyant sphere in a fluid and targeting in hamiltonian systems. *Physical Review Letters*, 84(25):5764, 2000.
[28] Renée Gatignol. The faxén formulas for a rigid particle in an unsteady non-uniform stokes-flow. *Journal de Mécanique théorique et appliquée*, 2(2):143–160, 1983.
[29] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. 2017.
[30] Thierry Dombre, Uriel Frisch, John M Greene, Michel Hénon, A Mehr, and Andrew M Soward. Chaotic streamlines in the abc flows. *Journal of Fluid Mechanics*, 167:353–391, 1986.