Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning

Hongwei Tang ⓘ, Jean Rabault ⓘ, Alexander Kuhnle ⓘ, Yan Wang ⓘ, and Tongguang Wang ⓘ

View Online   Export Citation   CrossMark

32, 053605

Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning

View Online   Export Citation   CrossMark

Hongwei Tang,[1,a] (ID) Jean Rabault,[2,b] (ID) Alexander Kuhnle,[3] (ID) Yan Wang,[1,a] (ID) and Tongguang Wang[1,a,c] (ID)

AFFILIATIONS

[1] College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, 29 Yudao St., Nanjing 210016, China
[2] Department of Mathematics, University of Oslo, 0316 Oslo, Norway
[3] Blue Prism AI Labs, London WC2B 6NH, United Kingdom

[a] **Also at:** Jiangsu Key Laboratory of Hi-Tech Research for Wind Turbine Design, Nanjing, China.
[b] **Also at:** MINES Paristech, PSL Research University, CEMEF, Paris, France.
[c] **Author to whom correspondence should be addressed:** tgwang@nuaa.edu.cn

ABSTRACT

This paper focuses on the active flow control of a computational fluid dynamics simulation over a range of Reynolds numbers using deep reinforcement learning (DRL). More precisely, the proximal policy optimization (PPO) method is used to control the mass flow rate of four synthetic jets symmetrically located on the upper and lower sides of a cylinder immersed in a two-dimensional flow domain. The learning environment supports four flow configurations with Reynolds numbers 100, 200, 300, and 400, respectively. A new smoothing interpolation function is proposed to help the PPO algorithm learn to set continuous actions, which is of great importance to effectively suppress problematic jumps in lift and allow a better convergence for the training process. It is shown that the DRL controller is able to significantly reduce the lift and drag fluctuations and actively reduce the drag by ~5.7%, 21.6%, 32.7%, and 38.7%, at Re = 100, 200, 300, and 400, respectively. More importantly, it can also effectively reduce drag for any previously unseen value of the Reynolds number between 60 and 400. This highlights the generalization ability of deep neural networks and is an important milestone toward the development of practical applications of DRL to active flow control.

Published under license by AIP Publishing. https://doi.org/10.1063/5.0006492

I. INTRODUCTION

Actively controlling a flow to change its characteristics is attractive for many applications in the field of fluid mechanics and could bring large industrial benefits.[1] Since the pioneering work of Prandtl about the use of active flow control (AFC) for delaying boundary layer separation,[2] AFC has witnessed a fast growth and has become an increasingly important technology for the pursuit of industrial and sustainable solutions.[3] Prospective applications of AFC to problems of industrial and environmental importance include, to name a few, reducing the aerodynamic drag on aircrafts,[4,5] manipulating the vortex in the wake of bluff bodies,[6–10] and

optimizing the design and performance of wind turbines[11–13] and gas turbines.[14]

Nevertheless, finding efficient strategies for performing AFC remains a challenge.[1,15] This difficulty is deeply rooted in the nature of the Navier–Stokes equations and their underlying high nonlinearity, as well as in the high dimensionality of possible control parameter spaces. Additionally, considerable challenges exist for applying AFC to engineering situations, such as disturbances inherent to the physical environment and imperfections in the manufacturing or installing of the actuators, which impose hard requirements on the ability of control algorithms to adapt robustly to external conditions. This makes the design of control strategies

a complex endeavor. Therefore, the main issue of AFC is currently the lack of robust, efficient algorithms that can leverage the physical devices available for performing effective control.

In practice, AFC can be open-loop (no feedback mechanism) or closed-loop (when a feedback mechanism is present, i.e., some measurements of the flow are provided to the AFC system to decide the next actuation).[16] Compared with the open-loop control, the closed-loop control possesses more potential to take full advantage of active devices to alter the flow. At present, many implementations of AFC are based on mathematical models of the flow system. For example, Flinois and Colonius[17] developed an adjoint-based optimal control framework to help stabilize the vortex shedding efficiently. Leclercq et al.[18] proposed a feedback-loop strategy using iteratively linearized models to suppress oscillations of resonating flows. Bergmann et al.[19] deduced an optimal control approach for the flow past a circular cylinder using proper orthogonal decomposition reduced-order models. Brackston et al.[20] used a stochastic modeling approach to design a feedback controller and validated it in experiments, effectively suppressing the asymmetric large-scale structure behind a bluff body wake with active flaps. These model-based control strategies are usually based on either harmonic or constant forcing,[21,22] making it, however, challenging for real-world AFC where complex non-linear systems are present in combination with stochastic disturbances.[23]

By contrast, model-free approaches, where the control strategy is found through a data-driven and learning-based approach, are quite suitable for complex, high-dimensional, nonlinear systems.[15,21] Such techniques mainly include genetic algorithms (GAs) and artificial neural networks (ANNs). While GAs have been extensively used for AFC,[23–26] ANNs are receiving growing attention recently due to the fast development of artificial intelligence/machine learning that has taken place in recent years. Furthermore, ANNs have been found so far to surpass GAs in terms of the complexity of the tasks learned and their learning speed.[27,28] Among other methods within the field of machine learning, ANNs used together with reinforcement learning algorithms have attracted great attention.[29,30] The resulting deep reinforcement learning (DRL) paradigm has been successfully deployed to resolve several high-profile, complex problems, such as playing a wide range of Atari game without hard-coding strategies,[31] generating realistic dialogs,[32] or controlling the dynamics of complex robots.[33] Compared with data-driven and supervised learning approaches, which have also found some applications in fluid mechanics within particle image velocimetry (PIV) measurement,[34–36] reduced-order modeling,[37,38] or predictions of flow features,[39–41] DRL allows us to find a solution through trial-and-error, even when no solution is known a priori. One can observe that challenging systems successfully controlled by DRL have remarkably similar properties of nonlinearity and high-dimension, similar to the features of flow phenomena that make AFC challenging. Consequently, DRL is seen as a promising avenue for performing AFC.[15]

Therefore, in recent years, DRL has became a new tool to discover AFC strategies,[15] and it has been shown to outperform previous techniques in several cases.[42] In addition, increases in the computational power available for numerical simulations make it possible to study increasingly complex systems using DRL and simulations. Such applications include optimizing the motion for individual[43] or collective fishes,[44,45] training a glider to autonomously navigate atmospheric thermal current,[46] and controlling the adaptive behavior of microswimmers.[47] Although the computational costs of the simulations needed to train the DRL algorithms still limit their application, they already have helped shed light on several complex problems.

The present work is an extension of the results initially presented by Rabault et al.,[48,49] but with four synthetic jets that are located symmetrically on a cylinder immersed in a two-dimensional domain. Moreover, the ability of DRL to design robust active control strategies for the flow over a range of conditions is further investigated. The proximal policy optimization (PPO) agent together with a two-layer fully connected neural network is used to control the mass flow rates of these four jets to reduce the magnitude and oscillation of the drag. In addition, a new interpolation equation is developed to make the control values change smoothly with time, so that problematical lift oscillations, which are caused by the interpolation function proposed in previous works,[48,49] are almost completely eliminated. In addition, the robustness and feasibility of the obtained control strategy, which shows the best performance in different flow conditions, are discussed. The paper is organized as follows. First, a brief introduction to the numerical method used for performing the simulations and the general theory underlying the DRL algorithm used are provided in Sec. II. The training using the DRL algorithm over a range of Reynolds numbers is then detailed in Sec. III, together with the results which underline the robustness and generalization ability of the control strategy obtained. Finally, a brief summary of the contribution and its significance for the use of DRL within AFC are demonstrated in Sec. IV.

II. PROBLEM SETUP AND METHODOLOGY

A. Problem description

The configuration of the simulation is adapted from the classical benchmark computations carried out by Schäfer et al.[50] (also known as the Turek benchmarks), in which a cylinder of diameter D is immersed in a two-dimensional domain with a size $22D \times 4.1D$, as depicted in Fig. 1. The center of the cylinder is located at a transversal distance of $0.05D$ from the horizontal centerline of the flow domain. This geometric asymmetry helps trigger the vortex shedding if the Reynolds number is greater than the critical value.

For performing AFC, four jets, for which the mass flow rates are controlled by the ANN, are symmetrically located on the upper and lower sides of the cylinder. The angular positions of these four jets are 75° (corresponding to θ_0, as shown in Fig. 2), 105°, 255°, and 285°, respectively. The jets are chosen as synthetic jets, i.e., the sum of the mass flow rates of all jets is enforced to be zero, and the jet directions are set to be perpendicular to the cylinder wall. The injection velocity can be positive or negative, corresponding to blowing or suction, respectively. With such configurations, there could be an extra injected momentum that could act as the propulsion, as discussed in Appendix B. However, the propulsion is small in any case, thanks to the net mass flow rate being kept equal to zero, and it amounts for no more than 5% of the momentum intercepting the cylinder once a pseudo-periodic regime with active control has been achieved. Therefore, this small propulsion effect will be neglected in the following discussion.
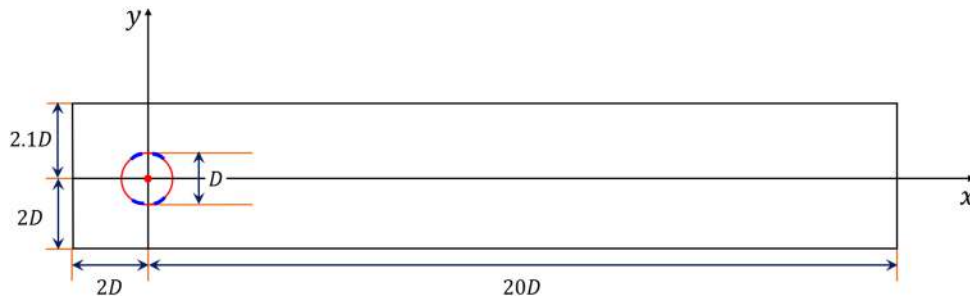
FIG. 1. Geometrical description of the configuration used for simulating the flow past a circular cylinder immersed in a two-dimensional channel, adapted from the work of Schäfer *et al.*[50] The center of the cylinder and the synthetic jets are marked by a red dot and blue arcs, respectively. The cylinder is slightly off the horizontal centerline of the channel (by 0.05D). This geometric asymmetry helps trigger the vortex shedding.

B. Numerical method

In the present study, the flow is assumed to be viscous and incompressible. The governing equations are the two-dimensional, time-dependent Navier–Stokes equations and the continuity equation, which can be expressed in a non-dimensional form as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot (\nabla \boldsymbol{u}) = -\nabla p + \frac{1}{Re}\Delta \boldsymbol{u}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where $\boldsymbol{u}$ is the non-dimensional velocity, t is the non-dimensional time, and p is the non-dimensional pressure. The characteristic length, velocity, density, and time for non-dimensionalizing the problem are D, $\overline{U}$, ρ, and $D/\overline{U}$, respectively, where $\overline{U}$ is the bulk velocity, which will be shown later, and ρ is the density of the fluid. The Reynolds number is defined as $Re = \overline{U}D/\nu$, where ν is the kinematic viscosity of the fluid.
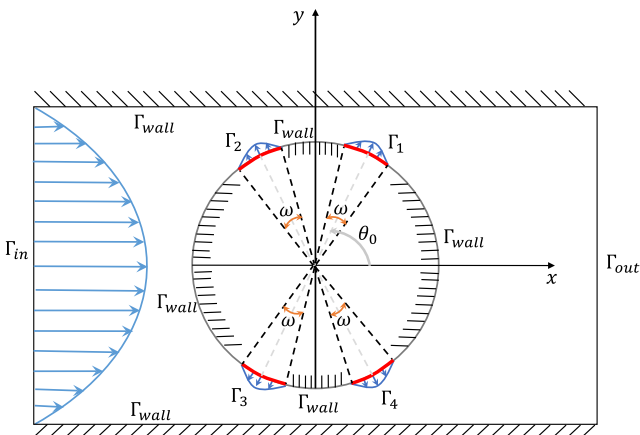

FIG. 2. Flow domain (not at scale) and boundary conditions for the simulation. The jet velocity profiles, determined by the output of ANNs, are prescribed (red arcs) by Γ_j (j = 1, 2, 3, 4). Γ_{wall} means no-slip boundary conditions implemented for solid walls. Γ_{in} is the inflow part, while Γ_{out} represents the outflow. ω is the width of the jets.

Figure 2 shows a schematic of the boundary conditions (for illustration purpose, the geometrical domain is out of scale). The inflow velocity profile in the streamwise direction (Γ_i) is specified as follows (cf. 2D-2 test case reported by Schäfer *et al.*[50]):

$$u_{inlet}(y) = -4U_m(y - 2.1D)(y + 2D)/H^2, \tag{3}$$

where $H = 4.1D$ is the width (along the Y-axis, as depicted in Fig. 1) of the domain and U_m is the horizontal velocity component at the midpoint of the inlet, i.e., the maximum of the inflow velocity. As a consequence, the bulk velocity can be calculated as follows:

$$\overline{U} = \frac{1}{H}\int_{-2D}^{2.1D} u_{inlet}(y)dy = \frac{2}{3}U_m. \tag{4}$$

No-slip boundary conditions (Γ_{wall}), i.e., the velocity of fluid is zero, are applied on the top and bottom walls and on the solid walls of the cylinder. The boundary condition corresponding to an outflow boundary (Γ_{out}) is imposed based on the assumption that the derivative of the velocity along the X-axis is zero at the outlet, which implies that the flow is fully developed or does not change significantly. More strictly, it is set as

$$-p\boldsymbol{n} + \frac{1}{Re}(\nabla \boldsymbol{u} \cdot \boldsymbol{n}) = 0, \tag{5}$$

where $\boldsymbol{n}$ is the unit vector normal to the outlet.

To avoid velocity discontinuity between the boundary of the jets and the no-slip surfaces of the cylinder, the radial velocity profiles (Γ_j) of the four synthetic jets are prescribed as follows:

$$u_{jet}(\theta, Q_i) = \frac{\pi}{\omega D}Q_i\cos\left(\frac{\pi}{\omega}(\theta - \theta_0)\right), \tag{6}$$

where $Q_i(i = 1, 2, 3, 4)$ is the mass flow rate of the four jets centered at $\theta_0 = 75°$, $105°$, $255°$, and $285°$, respectively. $\omega = 10°$ is the width of each jet.

For solving Eqs. (1) and (2) numerically, the incremental pressure correction scheme (IPCS) method[51] is used with explicit linearization of the nonlinear convective term by using the known velocity $\boldsymbol{u}^n$ at the time step $t = n\delta t$, where δt is the numerical time step and n is the number of the time steps considered. This method

is applied as a two-step fractional step method. First, an auxiliary velocity $\hat{u}$ is calculated by

$$\frac{1}{\delta t}(\hat{u} - u^n) = -u^n \cdot (\nabla u^n) - \nabla p^n + \frac{1}{Re}\Delta\frac{\hat{u} + u^n}{2}, \qquad (7)$$

and then the pressure p^{n+1} at $t = (n + 1)\delta t$ is obtained by solving a Poisson equation,

$$\Delta(p^{n+1} - p^n) = \frac{1}{\delta t}\nabla \cdot \hat{u}. \qquad (8)$$

This second step is usually referred to as the projection step.

Finally, the velocity u^{n+1} at $t = (n + 1)\delta t$ is obtained as follows:

$$\frac{1}{\delta t}(u^{n+1} - \hat{u}) = -\nabla(p^{n+1} - p^n). \qquad (9)$$

The computational domain is discretized by an unstructured mesh (triangular cells), and it is much refined around the surface of the cylinder (as shown in Fig. 3) so that the influence of synthetic jets on the flow simulation can be fully considered. The IPCS method is implemented using the finite element method within the FEniCS framework.[52] More precisely, the linear and quadratic basis functions of the continuous Galerkin family of elements are utilized to discretize the pressure and velocity fields, respectively. The resulting system of equations is solved using lower-upper (LU) decomposition, a sparse direct solver from the UMFPACK library.[53] The numerical solution is obtained at each time step, and then the drag F_D and lift F_L are integrated over the whole wall (including the jet surfaces) of the cylinder by

$$F_D = \int (\sigma \cdot n_c) \cdot e_x dS \qquad (10)$$

and

$$F_L = \int (\sigma \cdot n_c) \cdot e_y dS, \qquad (11)$$

where σ is the Cauchy stress tensor, n_c is the unit vector normal to the outer cylinder surface, and $e_x = (1, 0)$ and $e_y = (0, 1)$.

In order to study the mesh convergence and validate the numerical method, the quantities of interest are calculated from simulating the flow at $Re = 100$ and compared with the benchmark data.[50] The drag F_D and lift F_L are normalized as follows:

$$C_D = \frac{2F_D}{\rho \overline{U}^2 D} \qquad (12)$$

and

$$C_L = \frac{2F_L}{\rho \overline{U}^2 D}. \qquad (13)$$

The Strouhal number (St), which is used to describe the characteristic frequency of oscillating flow phenomena, is defined as follows:

$$St = f_s \cdot D/\overline{U}, \qquad (14)$$

where f_s is the shedding frequency computed from the periodic evolution of the lift coefficient C_L.

The simulation results using meshes of three different resolutions are listed in Table I, together with comparison to the bounds suggested by Schäfer et al.[50] The C_D^{max} and C_L^{max} correspond to the maximum of the drag coefficient C_D and lift coefficient C_L, respectively. As can be seen, the resolution of the main mesh, which is used in the present work, is fine enough for the simulation to agree well with the benchmark data. The discrepancies are less than 0.04% in all listed quantities when compared with the fine mesh. Although the maximum of C_L with the main mesh is slightly larger than the suggested upper bound by ~2.2%, the discrepancy is small. Moreover, the maximum of C_D and St are strictly within the suggested interval, which is of great importance as reducing drag is the main
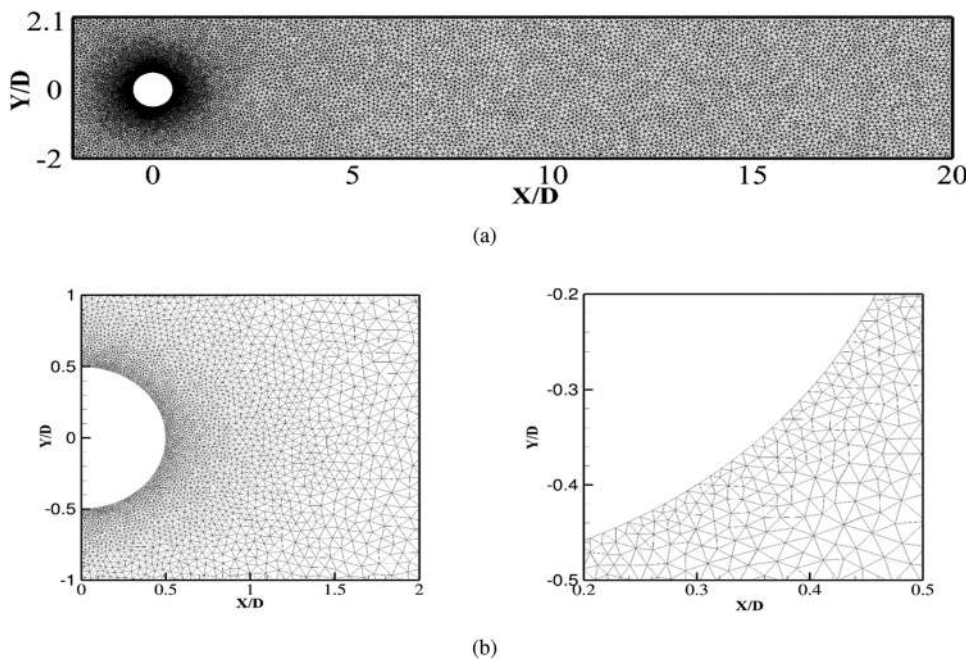

(a)

(b)

FIG. 3. Numerical discretization of the full (a) and partial (b) computational domain. The mesh is much refined around the cylinder to fully consider the influence of actuations on the flow simulation.

TABLE I. Mesh convergence and flow parameters for the 2D flow around a circular cylinder at $Re = 100$, in a configuration corresponding to the benchmark.[50]

Case	Mesh resolution		C_D^{max}	C_L^{max}	St
	Coarse	9374	3.2416	1.0758	0.3025
Present	Main	25 865	3.2299	1.0323	0.3020
	Fine	174 520	3.2311	1.0324	0.3020
Schäfer et al.[50]			3.2200–3.2400	0.9900–1.0100	0.2950–0.3050

focus. Hence, the main mesh depicted in Fig. 3 is deemed sufficiently refined and is used thereafter.

C. DRL control algorithm

Advances in machine learning have promised a renaissance in understanding intrinsic features of many complex systems and gaining unprecedented attention not only in computer science but also in many other disciplines, such as fluid mechanics,[54–56] partial differential equations,[57,58] or design optimization.[59,60] Reinforcement learning is one of the main branches of machine learning and recently attracted a lot of interest following Google DeepMind, defeating top human professionals at the game of Go.[61] Unlike other machine learning methods such as supervised learning, which consists in learning to map an input to its corresponding output based on labeled examples provided by a knowledgeable external supervisor, or unsupervised learning, which is typically interested in finding transformations and clustering properties hidden in data, reinforcement learning is concerned with how to interact with an environment so as to maximize a numerical reward signal.

A simplified overview of the DRL framework used in the present study is schematically depicted in Fig. 4. The framework can be divided into two main parts: the environment and the learning agent. In the present work, the former is the direct numerical simulation (DNS) for the flow past a circular cylinder at low Reynolds numbers, as previously described. The latter corresponds to a concrete deep reinforcement learning algorithm, proximal policy optimization, which is described in detail later in this section. As illustrated in Fig. 4, the learning agent interacts with the environment through three channels: the state of the environment, the action chosen by the agent to influence the environment, and the reward signal that defines the goal of the reinforcement learning problem. Specifically, the state is a partial observation of the flow field. More concretely, the local value of the flow field sampled at 236 probes located around the cylinder and in its wake (black points in Fig. 5) acts as the input based on which the agent can infer the different flow features. These probes do not influence the flow field since the extraction of local physical quantities of flow variables is carried out after the numerical simulation ends at each time step. The ANN used by the agent to parameterize the decision policy distribution is a two-layer fully connected network with 512 neurons in each layer. The resulting action value provided by the agent is then connected to the mass flow rate applied to each jet. The reward function is the time-averaged drag of a training action penalized by the absolute magnitude of the time-averaged lift, which can be expressed as follows:

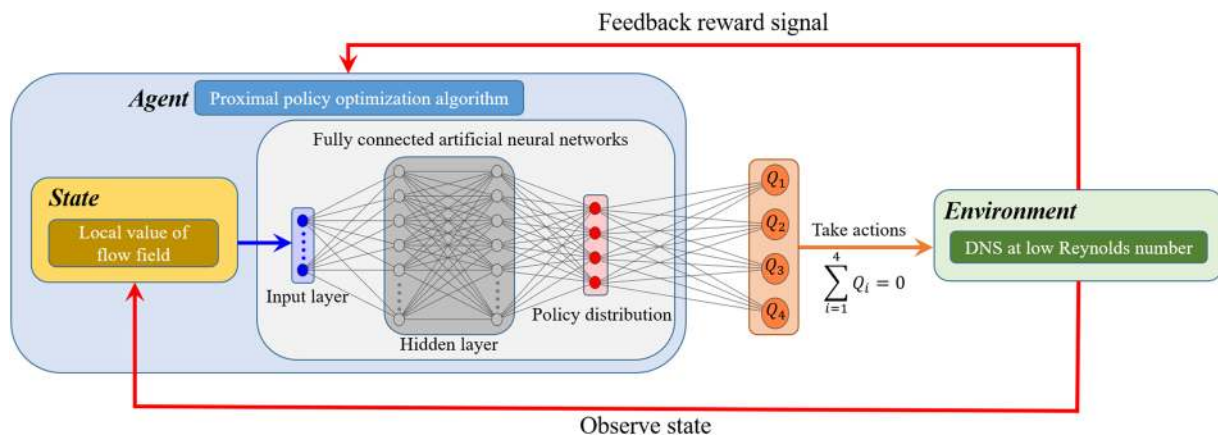$$R_T = |F_D|_T - \beta|F_L|_T, \qquad (15)$$

FIG. 4. Illustration of the DRL framework utilized in the present work for performing AFC. The environment, i.e., a numerical simulation of the flow past a cylinder, is coupled in a closed-loop fashion with the learning agent. Iteratively, the mass flow rate of the jets [$Q_i(i = 1, 2, 3, 4)$] is controlled by the agent according to the observed flow state. In response, the simulation produces the updated flow field as the next state, and a reward signal is used to guide the control strategy toward controlling the flow so as to reduce the drag. Through such coupled interaction, the agent eventually learns to perform effective AFC of the simulated flow.
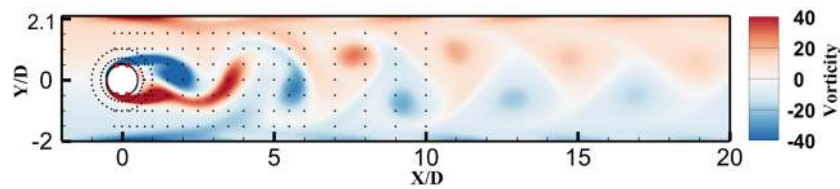
FIG. 5. Unsteady non-dimensional vorticity wake behind the cylinder after flow initialization without active control. The location of the probes is indicated by the black dots. The location of the control jets is indicated by the red dots. This illustrates the configuration used to perform AFC for flow control past the circular cylinder.

where $|\cdot|_T$ indicates the average over an action time step $T = 100\delta t$ (see later) and β is a parameter set to 0.2 in the present work. The lift penalization is used to avoid a "cheating" strategy in which the jets could blow consistently in the same direction with maximum strength after a given point in time. More details on the motivation for the penalization term are provided in the work of Rabault and Kuhnle.[49] In general, a learning agent is able to use the state of the environment it controls to take actions so as to optimize the cumulative value of the reward function, which corresponds to the lowest drag.

The reinforcement learning algorithm used for training the ANN, known as proximal policy optimization (PPO), is one of the state-of-the-art reinforcement learning approaches and has been widely applied to control tasks.[48,62] Compared with other DRL algorithms, PPO is simpler to implement and tune while obtaining comparably good performance. As the PPO algorithm has already been used in a variety of fluid mechanics works, the reader interested in more details on the PPO algorithm itself is invited to consult the previous work on the topic.[48] The PPO method is episode-based, which means that the interactions between the agent and the environment are broken into a number of training interaction sequences.[63] The initial states for the training episodes at each Re are first obtained by performing the simulation without active control until a fully developed unsteady wake, i.e., the Kármán vortex street, is observed. The corresponding solution is stored and used as a starting point for subsequent learning episodes. For the environment with four flow configurations, the initial state is selected randomly from the initialized fields corresponding to Re 100, 200, 300, and 400.

One possible discussion could be whether 236 probes are enough for the ANN to have detailed information about the flow features and perform good or even optimal control of the system. More generally, assessing the efficiency of the decision made by partial observability of the system is a well-known difficulty in reinforcement learning and remains an active and increasingly important research challenge.[64] Based on the previous work on the topic[48] and our experience following preliminary tests during the present study, 236 probes are found to be enough for the ANN to perform adequate training and to attain satisfactory control performance. Much fewer probes (less than 10) could also help the agent to learn a valid strategy, but it will impair the control effects,[48] i.e., lesser drag reduction will be obtained compared to the results using more probes. With the 236 probes used in the present study, the agent is able to gain extensive information about the flow configuration around the cylinder and its far-wake, which is important for taking optimal actions. These probes are purely passive and simply report the local

properties of the flow to the PPO algorithm, without influencing the flow.

In order to use the PPO algorithm on the present problem, two techniques are implemented for structuring the interactions between the agent and the flow environment. First, during the simulation, the action provided by the PPO agent is updated only 200 times per episode and is kept constant for a duration of 100 numerical simulation time steps [this defines the length of one action time step, i.e., the T in Eq. (15)], corresponding to ~3.3% of the vortex shedding period. This limitation is added following the suggestion of Rabault and Kuhnle,[49] and the necessity for such tuning of the action frequency update has also been observed by Braylan et al.[65] As a consequence, in the following, the difference will be distinguished between the numerical time step and the period at which the action update is applied. Second, the instantaneous mass flow rates obtained from the actions are made continuous at the time scale of the numerical simulation dt in order to avoid invalid physical jumps on pressure or velocity distribution around the cylinder wall. Thus, the control value effectively applied changes smoothly with time.

It should be emphasized that a balance needs to be found to avoid a too long update interval, which makes it impossible for the learning agent to respond to the system fast enough, or a too short update interval, which means that the time over which the action is applied is too short to observe a measurable effect on the system, therefore making learning impossible during the first stage of the training.[66] Furthermore, a constraint, $|Q_i^*| = Q_i/Q_{ref}| \leq 0.05$, is imposed for preventing non-physically large actuations, where Q_i is the mass flow rate of the i-th jet and Q_{ref} is the reference mass flow rate intercepting the cylinder. This allows us to avoid divergence of the numerical simulation.

III. RESULTS AND DISCUSSION

A. Active control for flow at higher Reynolds number

Previous works[48,49] have shown that ANNs trained by DRL are capable of finding a good control strategy for controlling the flow obtained in the present configuration at $Re = 100$. However, it is known that the Reynolds number has a strong influence on the complexity of such flows, irrespective of the chaoticity of the cylinder wake, and, ultimately, laminar-to-turbulent transition of the flow past a circular cylinder. For the present flow configuration, the wake becomes more irregular at larger Re.

On the other hand, Protas and Wesfreid[67] have proposed that two parts contribute to the mean drag coefficient C_D observed in such flows: one is the drag C_D^{base} of the steady and symmetric flow,

and the other is the drag C_D^0 resulting from the effect of vortex shedding,

$$C_D = C_D^{base} + C_D^0. \qquad (16)$$

In other words, the averaged drag consists of contributions of steady and unsteady parts, respectively. According to the argument of Bergmann et al.,[19] only the second part (due to oscillatory flow) can be altered by AFC. Therefore, this provides an estimate of the optimal AFC drag reduction attainable.

Since it has been demonstrated that the contribution of C_D^0 increases with Re,[19] it is natural to investigate the control performance of ANNs trained through DRL for flow for increasing Re. Consequently, two individual ANNs are trained to obtain control strategies for flow with $Re = 200$ and 400, respectively. Here, the control configurations same as those of previous works[48] are used, i.e., two jets located at the top and bottom extremities of the cylinder. The drag coefficients when control is applied by the ANNs after training are shown in Figs. 6 and 7, with the results of baseline flow (i.e., without control) being shown as a reference. The drag
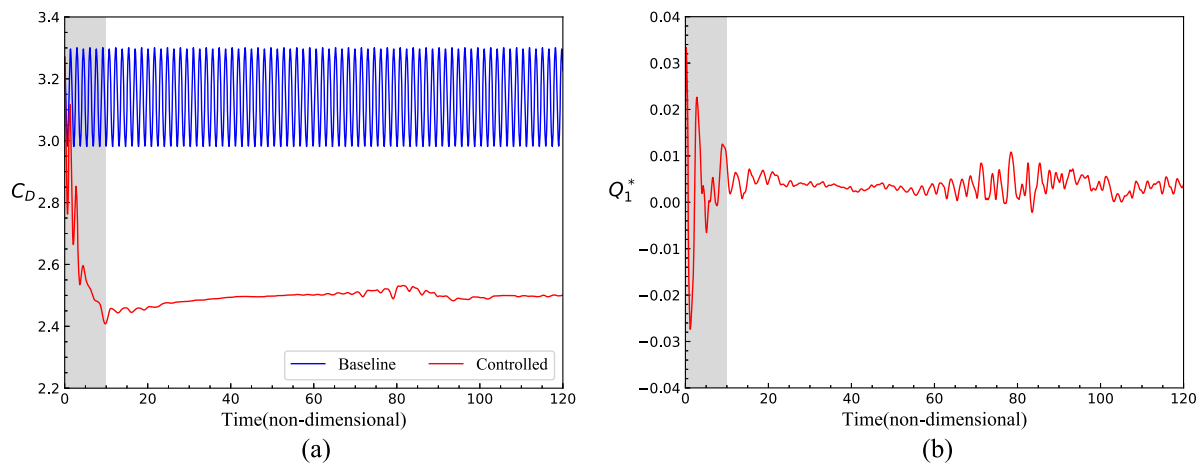

FIG. 6. Active control for flow at $Re = 200$. (a) Time-resolved value of the drag coefficient C_D with (controlled curve) and without (baseline curve) active flow control. (b) Time-resolved value of the normalized mass flow rate of one jet. It can be seen that the PPO agent found a good control strategy to attain a drag reduction of ~20.4%. Two successive phases can be observed with control: in the first, relatively large actuations are performed to greatly reduce the drag, followed by a pseudo-periodic regime in which only small control actuations are needed.
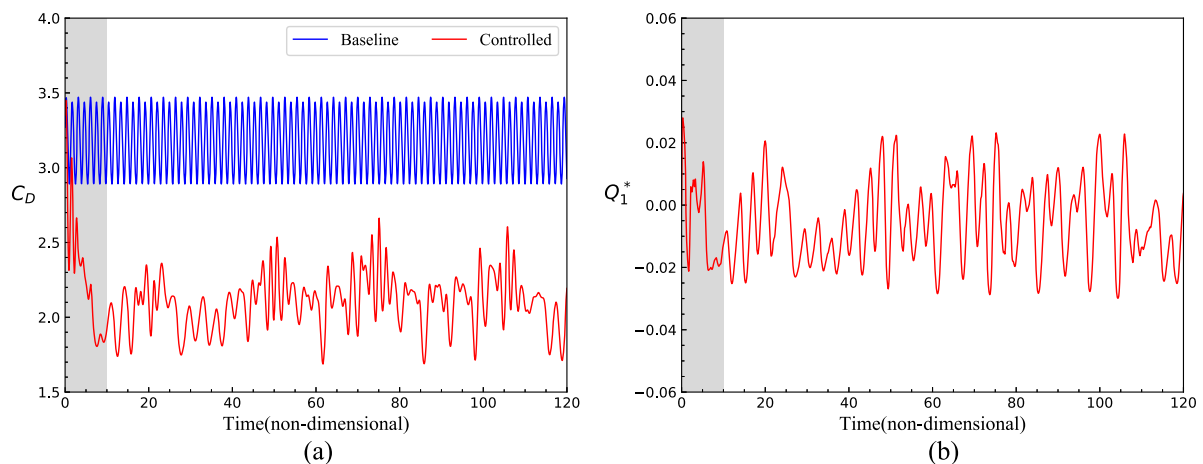

FIG. 7. Active control for flow at $Re = 400$. (a) Time-resolved value of the drag coefficient C_D with (controlled curve) and without (baseline curve) active flow control. (b) Time-resolved value of the normalized mass flow rate of one jet. Two successive phases can be distinguished. Similar to the flow with control at $Re = 100$ (cf. the work of Rabault et al.[48]) and 200, a clear reduction of a drag of ~33.1% is obtained in the first phase. However, in contrast to the Re of 100 and 200, no large decrease of actuations is observed in the second phase. This is due to the inherent instability of the flow at larger Reynolds numbers and illustrates the ability of the PPO algorithm to control systems with pseudo-chaotic properties.

reduction is calculated as $(|C_D|_{base} - |C_D|_{control})/|C_D|_{base}$, where $|C_D|_{base}$ and $|C_D|_{control}$ are the mean value for drag coefficients C_D in the case without and with active control, respectively. A drag reduction of ~20.4% is observed at $Re = 200$, and the final control result is satisfactory, though small oscillations still exist. By contrast, at $Re = 400$, although a reduction of ~33.1% for the averaged drag was achieved, the nonlinear essence of the transitional flow makes it hard for the DRL agent to find a fully stabilized control strategy and to completely suppress oscillations in the drag coefficient. However, the amplitude of the drag oscillations, as well as their frequency, is still decreased, implying that the DRL agent indeed learns some strategy that allows effective control. Similar to what has been observed for the same configuration at $Re = 100$,[48] the active flow control consists of two successive phases. In the first phase (nondimensional time ranging from 0 to ~10), a clear drag reduction is achieved by performing relatively large actuations. The flow is then modified into a pseudo-periodic regime where smaller actuations are used at $Re = 200$. For the flow at $Re = 400$, however, in the second phase, there is less attenuation of the actuations, resulting in big oscillations of drag coefficients even with control. Therefore, it appears that the flow in transitional regimes is quite unstable, which easily leads to a collapse of the modified flow configuration and, in turn, calls for large actuations to regain control of the system. This illustrates the ability of the PPO algorithm to perform control of pseudo-chaotic systems such as obtained from the simulation of flows at moderate to high Reynolds numbers, in good agreement with previously published results.[48,49]

B. Effect of smoothing interpolation functions

As explained in Sec. II C, it is of great importance to use suitable methods to interpolate the intrinsically time-discretized output of the ANN to continuous systems. This is still a topic of ongoing research with no clear optimal solution.[63] The present work chooses to directly interpolate between action updates to generate the control value at each simulation time step. This is simple to implement while maintaining a good performance for policy training and action selection. The interpolation must follow some principles such as smoothness and continuity to avoid numerical instability caused by non-physical phenomenon such as pressure jump in the fluid flow.

The interpolation can be performed in several fashions by considering the different relationships among action updates. Rabault and Kuhnle[49] proposed an exponential decay law based on the control value from the previous action. More precisely, they use the following equation with $\alpha = 0.1$ to calculate a new control value:

$$c_{i+1} = c_i + \alpha(a_j - c_i), \qquad (17)$$

where c_i is the control value at the previous numerical time step, c_{i+1} is the new control, and a_j is the action updated by the ANN. Note that the subscript i means the i-th numerical time step, which is connected to the time step dt of the simulation. By contrast, the subscript j indicates the j-th action update interval, which corresponds to the number of the action update during an episode, and takes place at a period $T = 100\delta$.

The strategy obtained using Eq. (17) is able to stabilize the vortex alley and to reduce drag by ~8% at $Re = 100$. The exponential decay law performs well for the convergence of the control values; however, there are distinct problematic jumps in lift, indicating that the flow state with control is still not perfectly stable. This is visible in Fig. 8. Some other schemes for interpolation also show similar problems. For example, one can consider a more previous control value for performing an update or use a nonlinear interpolation, which can be implemented, respectively, as follows:

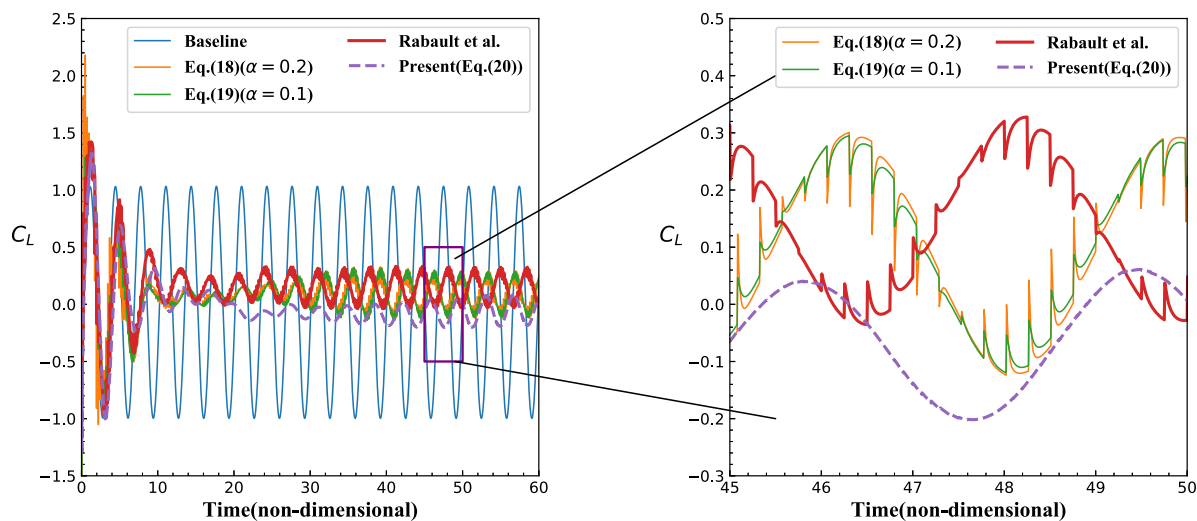$$c_{i+1} = c_i + \alpha(a_j - c_i) + \alpha(a_{j-1} - c_{i-1}), \qquad (18)$$

FIG. 8. Comparison of the time-resolved value of lift coefficients C_L at $Re = 100$ with active control trained using Eq. (17) (used by Rabault *et al.*[48] with $\alpha = 0.1$), Eq. (18), Eq. (19), and Eq. (20) (used in the present work), respectively. The linear smooth law, i.e., Eq. (20), shows the best performance as jumps in lift are almost completely suppressed.

$$c_{i+1} = c_i + \alpha(a_j - c_i) + \alpha(a_j - c_i)^2. \qquad (19)$$

After extensive trial-and-error, it is finally found that linear interpolation between two actions [corresponding to Eq. (20), see under] shows a comparable performance to exponential decay law while effectively eliminating the oscillations of the lift coefficient. Figure 8 shows a comparison of the control effects with the different smoothing laws discussed above. Obviously, Eq. (20) shows the best control performance. The corresponding interpolation law is defined as follows:

$$c_i = a_{j-1} + \frac{a_j - a_{j-1}}{N_e}, \qquad (20)$$

where N_e is the number of numerical time steps between two consecutive updates of actions and $n = 1, 2, \ldots, N_e$ is the current control step.

C. Training a model over a range of Reynolds numbers

To validate the versatility of an artificial neural network trained by deep reinforcement learning to control a flow across different Reynolds numbers, a learning environment supporting four flow configurations with Re varying within the discrete set 100, 200, 300, and 400 is used to train a single ANN. Therefore, the aim here is to train one ANN to perform effective control over a range of flow parameters, in a robust fashion. In this case, four jets are located on the upper and lower sides of the cylinder, as described in Fig. 2. Due to the learning process being treated on an episode base, each flow simulation is first run without active control until a fully developed unsteady wake, i.e., the Kármán vortex street, is observed, and the corresponding state is dumped and selected randomly as an initial start state for subsequent learning episodes. Here, the multi-environment approach proposed by Rabault and Kuhnle[49] is adapted, and the probability for every flow state to be selected as the

initial state of an episode is equal. Since every environment is independent of the others, that is, episodes do not influence each other due to the use of distinct initialization fields at distinct Reynolds numbers,[63] the agent has to remember features for different flow configurations, so that the knowledge learned by the ANN for one flow will not be altered by training on others.

The time series for the drag coefficients obtained using the global control strategy after 800 episodes when Re = 100, 200, 300, 400 is compared with baseline flow (without active flow control), as shown in Fig. 9. Compared with the results presented by Rabault et al.[48] where the control strategy is discovered through training in an environment consisting of one single flow configuration (Re = 100), the global control strategy becomes slightly less effective at Re = 100, but the overall control strategy is significantly more robust since the obtained ANN is able to adapt the actuation to perform near-optimal control (see later in the text) at all Res within the range 60–400. A drag reduction of ~5.7%, 21.6%, 32.7%, and 38.7% is obtained when Re = 100, 200, 300, and 400, respectively. Similar to the results presented in Figs. 6 and 7, the process of active flow control is composed of two phases. The main difference is that it takes a longer time (up to a non-dimensional time of ~20) for attaining the typical value of the drag reduction (i.e., the first phase of the control strategy takes a longer time to complete). In addition, slightly larger fluctuations can be observed during this phase, especially for higher Re.

One interesting result of this experiment is that the active control strategy trained over a range of Reynolds numbers shows comparatively good performance compared with the results shown in Fig. 7. While the average reduction of drag is close, the oscillations in drag are greatly suppressed with the global control strategy. This may be due to two factors. First, only two synthetic jets with angular coordinates 90° and 270° are used for the results in Fig. 7, while the global control strategy is allowed to control four jets, therefore
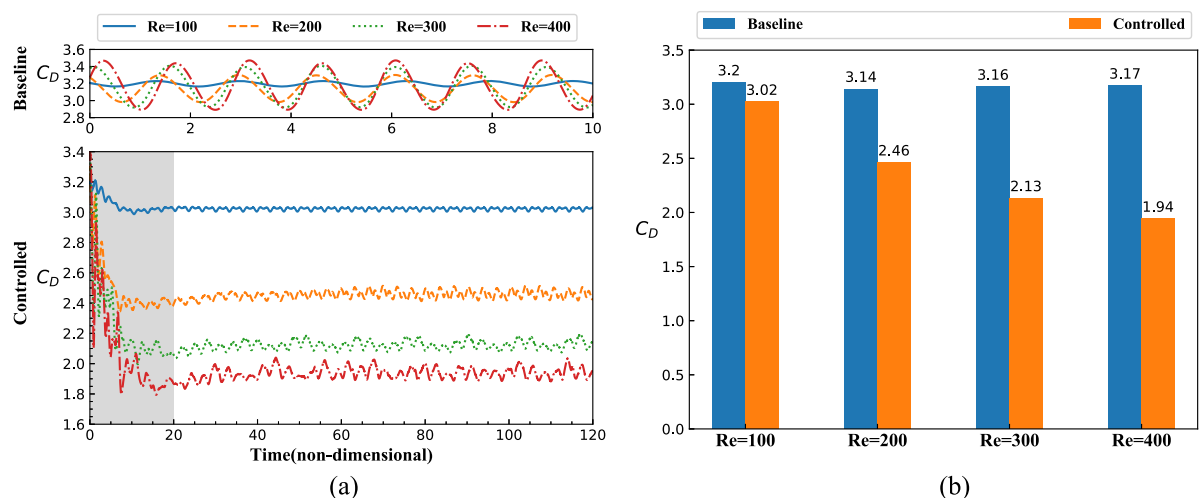

(a)   (b)

FIG. 9. Illustration of the control performance of the global agent (controlled curves) for flows corresponding to Re = 100, 200, 300, and 400 compared with the case without control (baseline). (a) Time series of the drag coefficients C_D. (b) The average of the drag coefficients C_D. The drag is reduced by ~5.7%, 21.6%, 32.7%, and 38.7% when Re = 100, 200, 300, and 400, respectively. Similarly to what can be observed in Figs. 6 and 7, the active flow control consists of two successive stages. However, in comparison, the first stage of control takes a longer time (up to a non-dimensional time of ~20) compared with the case when the controlled strategy is tuned to a single Re value.

allowing a more fine-grained control. Second, the training of the global control strategy is performed over a range of Reynolds numbers, therefore presenting more variability during training. For further exploration of this question, several independent training runs are launched using same control configurations as described in Fig. 5, i.e., four jets, for the flow with $Re = 400$ (the learning environment is then composed of one flow configuration, i.e., no global strategy is used). In this case, the drag coefficients with control show no big difference with what is presented in Fig. 7 (the average of drag with control is similar and still exhibits large oscillations). Such results prove the robustness and good performance of the ANN obtained with global training and point to the utility of training the ANN over a range of conditions. On the other hand, it also indicates that for much more complex systems, an efficient way to obtain good control strategies may be to embed a number of similar but slightly different systems inside the learning environment. This is in good agreement with the commonly accepted concept of transfer learning (TL), the core idea of which is that knowledge gained from one task can help the learning performance in a similar but slightly different task and improve the overall performance.[68]

As expected, the control strategy is more effective at reducing drag for larger Re, due to the relative increase of the controllable contribution of the drag previously discussed, i.e., C_D^0 is relatively bigger at higher Re. To further analyze the results, the fast Fourier transformation (FFT) is applied for investigating the frequency of drag and lift time series with and without active control (60 000 numerical time steps are used for calculating the FFT). For making the results more easily visible, the drag and lift coefficients are subtracted by their average value before FFT analysis is applied so that the purely oscillatory properties of the coefficients in question are revealed. As visible in Fig. 10, there is an obvious reduction on the amplitude of drag fluctuations. Moreover, the characteristic frequency of the flow system actively controlled by the ANN is also modified. These results are similar to what were described by Rabault *et al.*[48]

To study the effect of the control on the flow field in more detail, a visual comparison of the flow undergoing control against the mean pressure and vorticity of the uncontrolled flow is presented in Fig. 11. As can be observed, the area of separated wake increases when the active control is applied. Moreover, the vortex shedding from the cylinder has been substantially enlarged and expanded by the synthetic jets, which causes the observed reduced fluctuations. The resulting flow approaches the state featuring symmetric characteristic as will be discussed next. As a consequence, the pressure drop in the wake of the cylinder becomes lower, causing the reduction of drag.

In order to evaluate the efficiency of the control strategy obtained by the PPO agent, the average values of the drag coefficient with active control are further compared with the drag coefficient values obtained in the case where there is no vortex shedding. Such a flow state still exists in supercritical regime, but it is too unstable to be observed in experiments.[67] However, it is easy to obtain in numerical simulation by using a symmetric boundary condition at the equatorial plane of the flow domain, similar to what is performed in Ref. 48. As can be seen in Fig. 12, with the Reynolds number increasing, the drag obtained at the steady-state decreases (symmetric flow curve). Relatively, the contribution from the unsteady part to the drag becomes increasingly significant. It is promising to see that the drag with active flow control is even smaller than the drag obtained without vortex shedding if $Re \geq 200$, indicating that the control strategy is close to the theoretical optimum.[19]

It is worth emphasizing that only four values of the Reynolds numbers, i.e., $Re = 100, 200, 300$, and 400 (highlighted by red dots in Fig. 12) were used during the training process, while the control is successful for any Re within that range (all markers on the figure correspond to individual simulations where the PPO agent trained on only the four reference Re values was used). In addition, the strategy is still effective for active control even beyond the scope of Re used for training, for instance, at $Re = 80$. This, again, highlights the
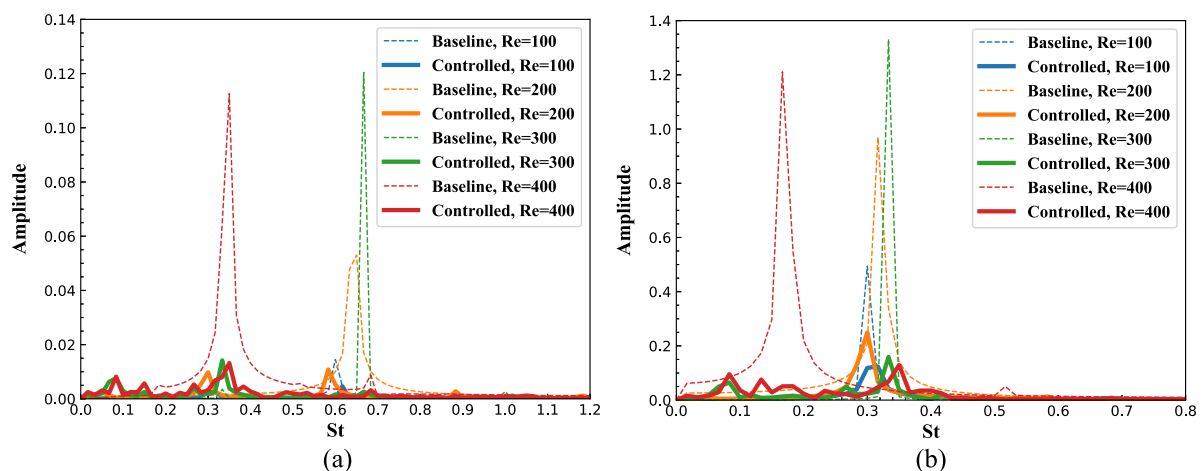

FIG. 10. FFT analysis of drag coefficients C_D (a) and lift coefficients C_L (b) subtracted by their mean values. The baseline curve corresponds to the flow without control, while the controlled curves mean that the flow is controlled by the ANN. The control effects are clearly visible: the amplitudes corresponding to fluctuations of both drag and lift are greatly reduced, and the characteristic frequencies of the flow fields are modified.
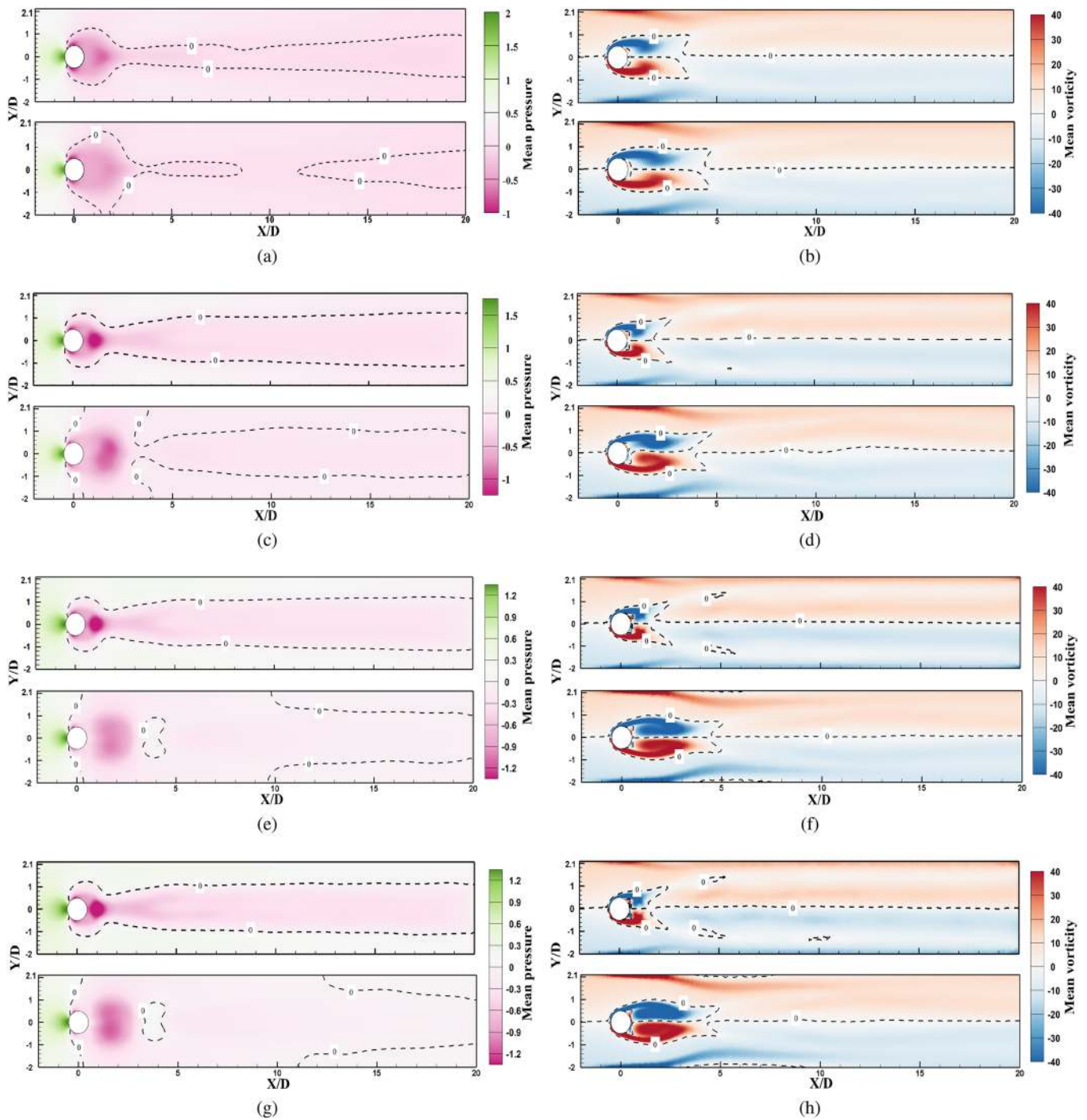
FIG. 11. Comparison of mean pressure (left) and vorticity (right) without (top part of each double panel) and with (bottom part of each double panel) active control. The Reynolds numbers for the four rows of double panels from top to bottom are 100 [(a) and (b)], 200 [(c) and (d)], 300 [(e) and (f)], and 400 [(g) and (h)], respectively. The color bar is common to both parts of each double panel. When the active control is applied, the area of separated wake increases and the vortex shedding from the cylinder is substantially enlarged. The former flow morphology is associated with the reduction of drag and lift, while the later is connected to the lower oscillations in these two forces. For blunt bodies acting in the flow regime considered, the largest contribution to the drag coefficient is due to the pressure fall in the wake, and it is clearly visible at all Res that the control strategy found allows us to mitigate this pressure drop immediately behind the cylinder.
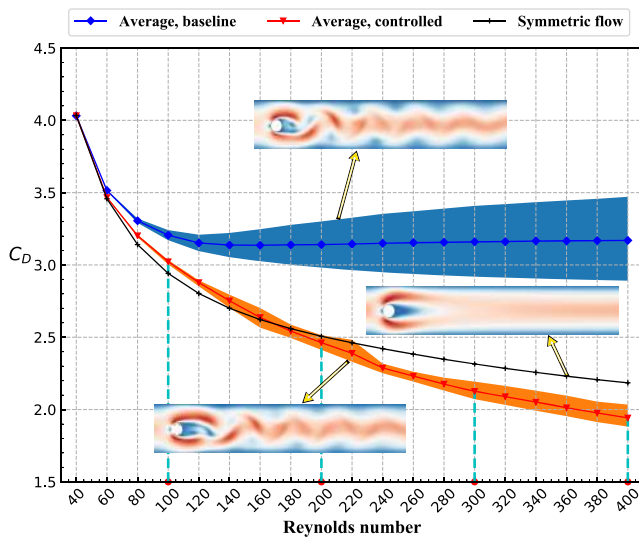
FIG. 12. Average of the drag coefficient for flow with and without control at different Reynolds numbers, and corresponding drag coefficient using a symmetric boundary condition at the equatorial plane of flow domain. The shaded areas indicate the range of oscillation in each respective case when the flow appears to be pseudo-periodic with active flow control. The general active flow control strategy is discovered through training at four Reynolds numbers (highlighted by red dots): 100, 200, 300, and 400. The insets with velocity magnitude as contour represent the structure of the corresponding flow state at a given Re. It is remarkable that control can be successfully applied at any Re between 60 and 400. In addition, the values of the drag coefficient C_D obtained with control compared with the symmetric case suggest that the global control strategy is close to being optimal on the range of Reynolds numbers considered.[19]

IV. CONCLUSIONS

In this study, the framework initially presented in the work of Rabault *et al.*[48,49] is extended by demonstrating the robustness and generalization ability of the PPO algorithm for machine-learning-based AFC. This state-of-the-art DRL method can allow ANNs to discover global active control strategies for flows over a range of Reynolds numbers. An alternative smoothing law performing linear interpolation between two successive actions is proposed to make the control values, i.e., the mass flow rates of the synthetic jets, change smoothly with time. With this method, the lift coefficient is made continuous to avoid non-physical jumps potentially occurring at action updates. The learning environment used for training supports four flow configurations with Reynolds numbers 100, 200, 300, and 400, respectively. After training, the ANN is able to actively control the flow and to reduce the drag by ~5.7%, 21.6%, 32.7%, and 38.7%, when Re = 100, 200, 300, and 400, respectively. More importantly, the ANN can also effectively reduce drag for any previously unseen Re in the range from 60 to 400. By observing the flow field through its mean pressure and vorticity, one can observe that the size of the separated wake and the vortex shedding area behind the cylinder is enlarged, resulting in a reduction in the pressure drop behind

the cylinder and the oscillation frequency induced by the vortices. It should be emphasized that only four values of Re were used during the training process, while the control is successful for any Re in the range 60–400, which highlights the generalization ability of the ANN and is of great importance for practical applications. The averaged drag with control is further compared with the drag value when using a symmetric boundary condition at the equatorial plane of the flow domain. It is promising that the drag of the controlled flow is even smaller than this symmetric baseline value if $Re \geq 200$, suggesting that the control strategy is close to the theoretical optimum.[19] Moreover, the results indicate that, in order to obtain better control performance for more complex systems, such as the flow at higher Re in the present case, embedding within the environment a number of systems with relatively simple but similar properties seems to be an efficient strategy. This is similar to the idea of transfer learning.

It should be noted that due to exploration noise and randomness involved in the training process, the strategy discovered through ANNs together with the PPO method may show a slightly different control performance in different training runs. However, the qualitative strategies found are relatively similar from one run to another.

Despite the relative simplicity of the selected problem, the experience and insights gained from this work are of great importance for progressing toward the application of DRL to more practical engineering problems in fluid mechanics. Although the computational cost remains a challenge to the wide application of DRL within fluid mechanics, this challenge can be progressively solved owing to the rapid advancement of high-performance computing architectures. Therefore, it is anticipated that significantly more complex problems, such as instabilities in boundary layers,[69,70] can be tackled using methodologies based on the present work, possibly in combination with other results and technical improvements such as the encoding of physical invariance of the system to control within the ANN architecture,[71,72] or the identification of reduced-order, hidden features of these systems.[73,74] In order to support the further development of DRL applications in the fluid mechanics community, all codes used are released as an open source (see Appendix A).

DATA AVAILABILITY

The data that support the findings of this study will be openly available at https://github.com/thw1021/Cylinder2DFlowControl-General upon publication in the peer-reviewed journal.

APPENDIX A: OPEN SOURCE CODE

The source code of this project together with all needed packages will released at https://github.com/thw1021/Cylinder2DFlow-ControlGeneral upon publication in the peer-reviewed journal. The computational fluid dynamics (CFD) solver is built on the open source finite element package FEniCS.[52] The DRL agent is based on the open source framework Tensorforce.[75] The present work is based on the multi-environment approach proposed by Rabault and Kuhnle,[49] and the reader can also refer to the open source code at https://github.com/jerabaul29/Cylinder2DFlowControlDRLParallel.

APPENDIX B: EVALUATION OF MOMENTUM INJECTED INTO THE FLOW FIELD USING FOUR JETS

When using four jets as schematically presented in Fig. 2, some extra momentum may be injected into the flow field. In this appendix, a mathematical formulation is derived to evaluate the injected momentum.

The momentum injected into the flow field per unit time by the i-th ($i = 1, 2, 3, 4$) jet in the horizontal direction can be evaluated as follows:

$$M_x^i = \int_{\theta_0^i - \omega/2}^{\theta_0^i + \omega/2} \rho u_{jet}(\theta; Q_i) u_{jet}(\theta; Q_i) \cos\theta \cdot \frac{D}{2} d\theta$$

$$= \frac{\pi^4}{\omega^2(4\pi^2 - \omega^2)} \cdot \frac{2\rho \sin\frac{\omega}{2}}{D} \cdot Q_i^2 \cos\theta_0^i, \quad (B1)$$

where θ_0^i is the position of the i-th ($i = 1, 2, 3, 4$) jet.

Consequently, the total momentum injected by the four jets in the horizontal direction is

$$M_x = \sum_{i=1}^{4} M_x^i = \frac{2\rho \pi^4 \sin\frac{\omega}{2}}{\omega^2(4\pi^2 - \omega^2)D} \sum_{i=1}^{4} Q_i^2 \cos\theta_0^i$$

$$= \frac{2\rho \pi^4 \sin\frac{\omega}{2}}{\omega^2(4\pi^2 - \omega^2)D}(Q_1^2 - Q_2^2 - Q_3^2 + Q_4^2)\cos 75°. \quad (B2)$$
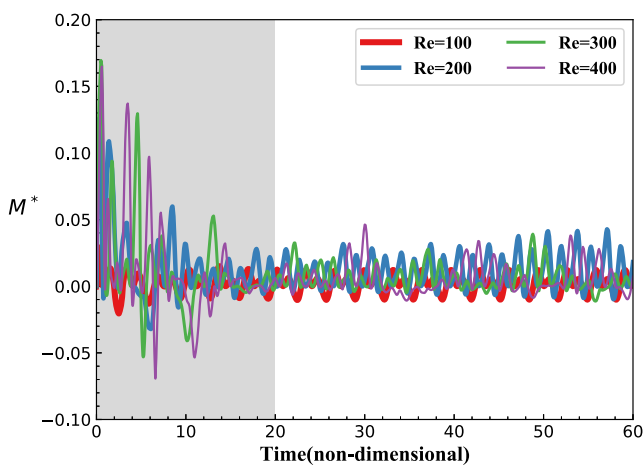

FIG. 13. Time series of normalized momentum horizontally injected into the flow field by the four control jets at Re = 100, 200, 300, and 400.

In the following, the injected momentum in the horizontal direction is normalized as follows:

$$M^* = M_x/M_{ref}, \quad (B3)$$

where $M_{ref} = \int_{-D/2}^{D/2} \rho u_{inlet}(y) u_{inlet}(y) dy$ is the reference momentum intercepting the cylinder.

The time-resolved value of the normalized momentum added by the four jets when applying the active flow control strategy to typical flow environments is shown in Fig. 13. Obviously, more momentum will be injected for flow at a higher Reynolds number.

REFERENCES

[1] M. Gad-el Hak, *Flow Control: Passive, Active, and Reactive Flow Management* (Cambridge University Press, 2000).

[2] P. Ludwig, "Über flüssigkeitsbewegung bei sehr kleiner reibung," in *Proceedings of the 3rd International Mathematics Congress* (Springer, Heidelberg, Germany, 1904), pp. 484–491.

[3] R. Vinuesa, H. Azizpour, I. Leite, M. Balaam, V. Dignum, S. Domisch, F. Anna, S. Langhans, M. Tegmark, and F. Nerini, "The role of artificial intelligence in achieving the sustainable development goals," Nat. Commun. **11**, 233 (2020).

[4] A. F. Shahrabi, "The control of flow separation: Study of optimal open loop parameters," Phys. Fluids **31**(3), 035104 (2019).

[5] D. Dolgopyat and A. Seifert, "Active flow control virtual maneuvering system applied to conventional airfoil," AIAA J. **57**(1), 72–89 (2019).

[6] H. Zhu, T. Tang, H. Zhao, and Y. Gao, "Control of vortex-induced vibration of a circular cylinder using a pair of air jets at low Reynolds number," Phys. Fluids **31**(4), 043603 (2019).

[7] C. Wang, H. Tang, S. C. M. Yu, and F. Duan, "Control of vortex-induced vibration using a pair of synthetic jets: Influence of active lock-on," Phys. Fluids **29**(8), 083602 (2017).

[8] B. Plumejeau, S. Delprat, L. Keirsbulck, M. Lippert, and W. Abassi, "Ultra-local model-based control of the square-back Ahmed body wake flow," Phys. Fluids **31**(8), 085103 (2019).

[9] C. Wang, H. Tang, F. Duan, and S. C. M. Yu, "Control of wakes and vortex-induced vibrations of a single circular cylinder using synthetic jets," J. Fluids Struct. **60**, 160–179 (2016).

[10] C. Raibaudo, P. Zhong, B. R. Noack, and R. J. Martinuzzi, "Machine learning strategies applied to the control of a fluidic pinball," Phys. Fluids **32**(1), 015108 (2020).

[11] S. Aubrun, A. Leroy, and P. Devinant, "A review of wind turbine-oriented active flow control strategies," Exp. Fluids **58**(10), 134 (2017).

[12] R. Pereira, W. A. Timmer, G. de Oliveira, and G. J. W. van Bussel, "Design of HAWT airfoils tailored for active flow control," Wind Energy **20**(9), 1569–1583 (2017).

[13] A. Wolf, T. Lutz, W. Würz, E. Krämer, O. Stalnov, and A. Seifert, "Trailing edge noise reduction of wind turbine blades by active flow control," Wind Energy **18**(5), 909–923 (2015).

[14] J. Bons, S. Benton, C. Bernardini, and M. Bloxham, "Active flow control for low-pressure turbines," AIAA J. **56**(7), 2687–2698 (2018).

[15] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**(1), 477–508 (2020).

[16] S. Collis, R. D. Joslin, A. Seifert, and V. Theofilis, "Issues in active flow control: Theory, control, simulation, and experiment," Prog. Aerosp. Sci. **40**(4), 237–289 (2004).

[17] T. L. B. Flinois and T. Colonius, "Optimal control of circular cylinder wakes using long control horizons," Phys. Fluids **27**(8), 087105 (2015).

[18] C. Leclercq, F. Demourant, C. Poussot-Vassal, and D. Sipp, "Linear iterative method for closed-loop control of quasiperiodic flows," J. Fluid Mech. **868**, 26–65 (2019).

[19] M. Bergmann, L. Cordier, and J.-P. Brancher, "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model," Phys. Fluids **17**(9), 097101 (2005).

[20] R. D. Brackston, J. M. García de la Cruz, A. Wynn, G. Rigas, and J. F. Morrison, "Stochastic modelling and feedback control of bistability in a turbulent bluff body wake," J. Fluid Mech. **802**, 726–749 (2016).

[21] S. L. Brunton and B. R. Noack, "Closed-loop turbulence control: Progress and challenges," Appl. Mech. Rev. **67**(5), 050801 (2015).

[22] W. Schoppa and F. Hussain, "A large-scale control strategy for drag reduction in turbulent boundary layers," Phys. Fluids **10**(5), 1049–1051 (1998).

[23] N. Gautier, J.-L. Aider, T. Duriez, B. R. Noack, M. Segond, and M. Abel, "Closed-loop separation control using machine learning," J. Fluid Mech. **770**, 442–457 (2015).

[24] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence* (Springer, 2017).

[25] F. Ren, C. Wang, and H. Tang, "Active control of vortex-induced vibration of a circular cylinder using machine learning," Phys. Fluids **31**(9), 093601 (2019).

[26] D. Antoine, K. A. F. F. Von Krbek, N. Mazellier, T. Duriez, L. Cordier, B. R. Noack, M. W. Abel, and A. Kourta, "Closed-loop separation control over a sharp edge ramp using genetic programming," Exp. Fluids **57**(3), 40 (2016).

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, A. C. Beattie, I. Sadik, I. Antonoglou, D. H. King, D. Kumaran, D. Wierstra, D. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature **518**(7540), 529–533 (2015).

[28] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning* (ICML, 2016), pp. 1329–1338.

[29] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," in *International Conference on Machine Learning* (ICML, 2016), pp. 2829–2838.

[30] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[31] M. Volodymyr, K. Koray, D. Silver, A. Graves, A. Ioannis, W. Daan, and R. Martin, "Playing Atari with deep reinforcement learning," in NIPS Deep Learning Workshop, 2013.

[32] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2016), pp. 1192–1202.

[33] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 3389–3396.

[34] J. Rabault, J. Kolaas, and A. Jensen, "Performing particle image velocimetry using artificial neural networks: A proof-of-concept," Meas. Sci. Technol. **28**(12), 125301 (2017).

[35] M. A. Mendez, M. T. Scelzo, and J.-M. Buchlin, "Multiscale modal analysis of an oscillating impinging gas jet," Exp. Therm. Fluid Sci. **91**, 256–276 (2018).

[36] M. A. Mendez, A. Gosset, and J.-M. Buchlin, "Experimental analysis of the stability of the jet wiping process, part II: Multiscale modal analysis of the gas jet-liquid film interaction," Exp. Therm. Fluid Sci. **106**, 48–67 (2019).

[37] M. A. Mendez, M. Balabane, and J.-M. Buchlin, "Multi-scale proper orthogonal decomposition of complex fluid flows," J. Fluid Mech. **870**, 988–1036 (2019).

[38] M. Tatar and M. H. Sabour, "Reduced-order modeling of dynamic stall using neuro-fuzzy inference system and orthogonal functions," Phys. Fluids **32**(4), 045101 (2020).

[39] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," Phys. Rev. Fluids **4**, 054603 (2019).

[40] V. Sekar, B. C. Khoo, S. Chang, and B. Cheong Khoo, "Fast flow field prediction over airfoils using deep learning approach," Phys. Fluids **31**(5), 057103 (2019).

[41] S. Z. Islami rad, R. Gholipour Peyvandi, and S. Sadrzadeh, "Determination of the volume fraction in (water-gasoil-air) multiphase flows using a simple and low-cost technique: Artificial neural networks," Phys. Fluids **31**(9), 093301 (2019).

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

[43] 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**(3), B622–B639 (2014).

[44] 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**, 726–749 (2016).

[45] S. Verma, G. Novati, and P. Koumoutsakos, "Efficient collective swimming by harnessing vortices through deep reinforcement learning," Proc. Natl. Acad. Sci. U. S. A. **115**(23), 5849–5854 (2018).

[46] G. Reddy, J. Wong-Ng, A. Celani, T. J. Sejnowski, and M. Vergassola, "Glider soaring via reinforcement learning in the field," Nature **562**(7726), 236 (2018).

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

[48] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," J. Fluid Mech. **865**, 281–302 (2019).

[49] J. Rabault and A. Kuhnle, "Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach," Phys. Fluids **31**(9), 094105 (2019).

[50] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, "Benchmark computations of laminar flow around a cylinder," in *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995* (Vieweg Teubner Verlag, Wiesbaden, 1996), pp. 547–566.

[51] K. Goda, "A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows," J. Comput. Phys. **30**(1), 76–95 (1979).

[52] L. Anders, K.-A. Mardal, and G. Wells, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book* (Springer Science & Business Media, 2012), Vol. 84.

[53] T. A. Davis and I. S. Duff, "An unsymmetric-pattern multifrontal method for sparse LU factorization," SIAM J. Matrix Anal. Appl. **18**(1), 140–158 (1997).

[54] J. Nathan Kutz, "Deep learning in fluid dynamics," J. Fluid Mech. **814**, 1–4 (2017).

[55] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, "Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows," AIAA J. **58**, 25–36 (2019).

[56] A. Beck, D. Flad, and C.-D. Munz, "Deep neural networks for data-driven LES closure models," J. Comput. Phys. **398**, 108910 (2019).

[57] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," J. Comput. Phys. **375**, 1339 (2018).

[58] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[59] X. Yan, J. Zhu, M. Kuang, and X. Wang, "Aerodynamic shape optimization using a novel optimizer based on machine learning techniques," Aerosp. Sci. Technol. **86**, 826–835 (2019).

[60] K. Yonekura and H. Hattori, "Framework for design optimization using deep reinforcement learning," Struct. Multidiscip. Optim. **60**, 1709–1713 (2019).

[61] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature **529**(7587), 484–489 (2016).

[62] E. Bøhn, E. M. Coates, S. Moe, and T. A. Johansen, "Deep reinforcement learning attitude control of fixed-wing UAVs using proximal policy optimization," in *International Conference on Unmanned Aircraft Systems (ICUAS)* (ICUAS, 2019), pp. 523–533.

[63] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).

[64]B. Recht, "A tour of reinforcement learning: The view from continuous control," Annu. Rev. Control, Rob., Auton. Syst. **2**(1), 253–279 (2019).

[65]A. Braylan, M. Hollenbeck, E. Meyerson, and R. Miikkulainen, "Frame skip is a powerful parameter for learning to play Atari," in AAAI-15 Workshop on Learning for General Competency in Video Games, 2015.

[66]A. Neitz, G. Parascandolo, S. Bauer, and B. Schölkopf, "Adaptive skip intervals: Temporal abstraction for recurrent dynamical models," in *Advances in Neural Information Processing Systems* (NeurIPS, 2018), pp. 9816–9826.

[67]B. Protas and J. E. Wesfreid, "Drag force in the open-loop control of the cylinder wake in the laminar regime," Phys. Fluids **14**(2), 810–826 (2002).

[68]M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," J. Mach. Learn. Res. **10**, 1633–1685 (2009).

[69]H. Xu, S. M. Mughal, E. R. Gowree, C. J. Atkin, and S. J. Sherwin, "Destabilisation and modification of Tollmien-Schlichting disturbances by a three-dimensional surface indentation," J. Fluid Mech. **819**, 592–620 (2017).

[70]H. Xu, J.-E. W. Lombard, and S. J. Sherwin, "Influence of localised smooth steps on the instability of a boundary layer," J. Fluid Mech. **817**, 138–170 (2017).

[71]V. Belus, J. Rabault, J. Viquerat, Z. Che, E. Hachem, and U. Reglade, "Exploiting locality and translational invariance to design effective deep reinforcement learning control of the 1-dimensional unstable falling liquid film," AIP Adv. **9**(12), 125014 (2019).

[72]B. Anderson, H. Truong Son, and R. Kondor, "Cormorant: Covariant molecular neural networks," in *Advances in Neural Information Processing Systems* (NeurIPS, 2019), pp. 14510–14519.

[73]S. Pawar, S. E. Ahmed, O. San, and A. Rasheed, "Data-driven recovery of hidden physics in reduced order modeling of fluid flows," Phys. Fluids **32**(3), 036602 (2020).

[74]M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science **367**(6481), 1026–1030 (2020).

[75]A. Kuhnle, M. Schaarschmidt, and K. Fricke, Tensorforce: A tensorflow library for applied reinforcement learning, https://github.com/tensorforce/tensorforce.