Applying deep reinforcement learning to active flow control in turbulent conditions

Feng Ren
Research Center for Fluid Structure Interactions
Department of Mechanical Engineering
The Hong Kong Polytechnic University

Jean Rabault
Department of Mathematics
University of Oslo
Contributed equally to this work

Hui Tang
Research Center for Fluid Structure Interactions
Department of Mechanical Engineering
The Hong Kong Polytechnic University
h.tang@polyu.edu.hk

June 19, 2020

Abstract

Machine learning has recently become a promising technique in fluid mechanics, especially for active flow control (AFC) applications. A recent work [J. Fluid Mech. (2019), vol. 865, pp. 281-302] has demonstrated the feasibility and efficiency of deep reinforcement learning (DRL) in performing AFC for a circular cylinder at a low Reynolds number, i.e., $Re = 100$. As a follow-up study, we investigate the same DRL-based AFC problem at an intermediate Reynolds number ($Re = 1000$), where the flow's strong nonlinearity poses great challenges to the control. The DRL agent interacts with the flow via receiving information from velocity probes and determines the strength of actuation realized by a pair of synthetic jets. A remarkable drag reduction of around 30% is achieved. By analysing turbulent quantities, it is shown that the drag reduction is obtained via elongating the recirculation bubble and reducing turbulent fluctuations in the wake. This study constitutes, to our knowledge, the first successful application of DRL to AFC in turbulent conditions, and it is, therefore, a key milestone in progressing towards the control of fully turbulent, chaotic, multimodal, and strongly nonlinear flow configurations.

1 Introduction

Active flow control (AFC) is a longstanding topic in fluid mechanics. It traditionally involves modifying flow behavior using actuators in order to improve performance metrics, such as reducing the drag on a blunt body, suppressing flow-induced vibrations, or enhancing mixing or thermal convection. AFC algorithms can be divided into two main categories, i.e., open-loop and closed-loop control, depending on whether repeated measurements of the flow are used to adjust the control. Compared with open-loop control, a well-designed closed-loop control is expected to both improve the control performance and be effective over a wider range of flow conditions. However, due to the high dimensionality and strong nonlinearity of AFC problems especially when the flow is turbulent, it is challenging to design closed-loop control laws in an explicit form.

In the past few years, AFC has benefited from advances in the field of machine learning (ML). Genetic Programming (GP) was the first ML technique to be applied to AFC. For example, Gautier *et al.* (2015) applied GP in order to determine explicit control laws for reducing the recirculation zone behind a backwards-facing step. Fan *et al.* (2018) applied linear GP to enhance jet mixing and discovered novel wake patterns. Ren *et al.* (2019) applied GP to suppress vortex-induced vibrations in a series of simulations.

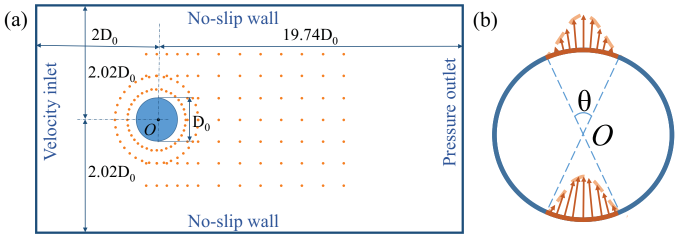

Figure 1: Schematics of (a) the flow domain and layout of velocity sensor array, and (b) the cylinder with jets' velocity profile.

Recently, a novel ML technique, deep reinforcement learning (DRL), has been attracting increasing attention following its many successes in robotics control (Mnih *et al.*, 2015), and at playing sophisticated games such as Go (Silver *et al.*, 2016). Following these successes, DRL is being increasingly applied to fluid mechanics (Brunton *et al.*, 2020; Rabault *et al.*, 2020). Early applications were mainly focused on agile maneuvering and biomimetism. For example, Reddy *et al.* (2016) used DRL to train a glider to fly autonomously by exploiting thermal currents in sunny weathers. Verma *et al.* (2018) studied the locomotion of fish schoolings, and trained rear fishes to harness energy from the wake of leading fishes using DRL. In these studies, owing to the limitations of early DRL algorithms, discretized control was used, i.e., the control space was limited to a few values rather than spanning a continuous range. However, in recent years, ML researchers have developed novel DRL algorithms to overcome such limitations and, in particular, so-called 'policy-based methods' are now well suited to solve continuous-control problems.

Therefore, following the development of these policy methods, including proximal policy optimization (PPO) (Schulman *et al.*, 2017; Heess *et al.*, 2017), which is now regarded as one of the 'state of the art' methods for continuous control, Rabault *et al.* (2019) achieved drag reduction for a circular cylinder in the laminar flow regime using a pair of synthetic jets. They reported a drag reduction of around 8%, resulting from mediated vortex shedding. To speed up the training, Rabault & Kuhnle (2019) also presented a multi-environment approach, which opens the way to performing control at higher Re, when simulations become more expensive. Following this, Tang *et al.* (2020) were able to design a robust DRL controller, which can control any 2D flow behind a cylinder in the range of Reynolds numbers 60 to 400. In addition, Belus *et al.* (2019) used the PPO algorithm to stabilize a thin fluid film using an array of jets, and presented efficient methods to deal with high-dimensionality control spaces. DRL has also been used to control other complex fluid mechanics systems, such as described by the one-dimensional Kuramoto–Sivashinsky (KS) equation (Bucci *et al.*, 2019), as well as Rayleigh-Bénard convection cells (Beintema *et al.*, 2020).

All these works are important contributions towards fully qualifying DRL as an effective control algorithm for fluid mechanics applications, and one can observe a progressive increase in complexity of the flow systems being controlled. We expect that the next milestone, i.e., demonstrating AFC in the turbulent flow regime, will be the final qualification step for applying DRL to control problems in fluid mechanics. In the present work, we present such a turbulent AFC application. For this, we re-use the main setup from Rabault *et al.* (2019), increasing the Reynolds number to 1000, in which case turbulent conditions are reached.

2 Methodology

2.1 Flow configuration

In the present work, we keep the physical system similar to that of the prior study (Rabault *et al.*, 2019), except for the value of the Reynolds number, which is increased to 1000 so as to consider a turbulent flow regime. The general configuration of the flow system is shown in figure 1(a). The 2D cylinder is located in the middle of a relatively narrow channel. The incoming flow is assumed to follow a parabolic profile along the transverse direction. The Reynolds number is defined as $Re = U_0 D_0/\nu$, where U_0 is the mean incoming velocity, D_0 is the diameter of the cylinder, and ν is kinematic viscosity of the fluid. In the following, results will be reported in non-dimensional form, and we define the non-dimensional time reference as $T_0 = D_0/U_0$.

A velocity sensor array is used to perceive the flow environment. Its layout is illustrated by the orange dots in figure 1(a). In total, 151 probes are used, each providing two time-varying signals, i.e., the streamwise and transverse velocity components.

A pair of synthetic jets are used as actuators. These work as a blowing / suction pair, so as to satisfy a zero cumulative mass flow rate constraint, i.e., no mass is added or subtracted to the flow by the pair of jets. The jets extend over an arc

Table 1: Comparisons between DNS and LES results as well as a mesh convergence study.

Configuration	Re	Method	Mesh resolution	Time resolution	$\overline{C_D}$	$\overline{\lvert C_L \rvert}$
I	100	DNS	2048×384	2000	3.204	0.646
II	100	DNS	1024×192	1000	3.200	0.639
III	100	DNS	512×96	500	3.196	0.608
IV	1000	DNS	6144×1152	6000	3.476	2.515
V	1000	DNS	3072×576	3000	3.438	2.463
VI	1000	LES	1536×288	1500	3.293	2.339

of width $\theta = 10^o$, where the jets velocities follow a cosinusoidal profile so as to meet the no-slip boundary condition at their extremities. The permissible velocity range at the center of the jets (normalized by U_0) is $[-1.62, 1.62]$. This value is consistent with the mass flow rate range used by Rabault *et al.* (2019).

The goal of the DRL agent, which is determined through the choice of the reward function, is to reduce drag and lift fluctuations. For this, we use a similar reward function as defined by Rabault & Kuhnle (2019):

$$r = \langle C_D \rangle_S + C \cdot \langle |C_L| \rangle_S, \tag{1}$$

where C_D and C_L are the drag and lift coefficients, respectively. $\langle \cdot \rangle_S$ indicates the average over a duration of one actuation. C is an adjustable positive coefficient that weights the contribution of drag and lift fluctuation in the reward value. It is taken equal to 1 in the following, if not stated otherwise. This value differs from what is used in Ref. (Rabault *et al.*, 2019). The reason for this is the change in the $\overline{|C_L|}/C_D$ ratio in the turbulent case at $Re = 1000$, which is around 3.3 times higher that the corresponding value in the laminar case at $Re = 100$.

2.2 Flow solver

In the prior study by Rabault *et al.* (2019), the solver used failed at Reynolds numbers larger than roughly 500. Here, by contrast, we turn to the lattice Boltzmann method (LBM) to simulate the flow.

In the present code, we use a regular, uniform mesh discretization over the whole domain. The boundary conditions (BCs) are similar to Ref. (Rabault *et al.*, 2019), i.e., the inlet has a constant parabolic velocity profile and the outlet follows a zero pressure condition. Both BCs are implemented using the non-equilibrium extrapolation scheme (Zhao-Li *et al.*, 2002). The half-way bounce-back scheme (He *et al.*, 1997) is used to satisfy no-penetration and no-slip BCs at the top and bottom walls. When considering the cylinder surface with jets, we apply the double linear interpolation method for curved boundary treatment (Yu *et al.*, 2003), and the corrected momentum exchange method to evaluate the drag and lift forces acting on the cylinder (Chen *et al.*, 2013).

In machine learning-based AFC, it is essential to reduce the time taken by the solver to perform each training simulation, as many such simulations may be needed to find an efficient strategy. Thus, instead of applying the LBM solver in a direct numerical simulation (DNS) manner during training, we resort to a large eddy simulation (LES) approach when simulating flows at $Re = 1000$, following the Vreman model (Vreman, 2004). In this case, in addition to the fluid molecular viscosity, the total viscosity involves an eddy viscosity. The eddy viscosity is derived from the local velocity derivatives tensor, and models subgrid scale dissipation. The velocity derivatives are calculated using a second-order finite difference scheme. The Vreman model used here has been implemented and validated in previous works (Ren *et al.*, 2018*a*,*b*).

In order to validate these numerical methods in the current setup, we conduct six test configuration simulations for which the results are summarized in table 1. For configurations at $Re = 100$, even the coarsest mesh (Configuration III) generates $\overline{C_D}$ values comparable to what is obtained with finer mesh resolution (Configurations I and II). For the configurations at $Re = 1000$, LES with the coarsest mesh resolution (Configuration VI) shows good numerical stability and holds relative errors of 5% and 7% (in terms of $\overline{C_D}$ and $\overline{|C_L|}$, respectively), compared with a highly-resolved DNS (Configuration IV). More importantly, Configuration VI only takes approximately 8% of the time of Configuration IV, offering better balance between numerical accuracy and speed at training time.

Following these experiments, mesh setups corresponding to Configuration II and Configuration VI will be used for DRL training at $Re = 100$ and at $Re = 1000$, respectively. By contrast, highly resolved DNS configurations (configurations I and IV) will be used for evaluation of the trained PPO agents. Typical velocity snapshots obtained in both training configurations are presented in figure 2. One can observe that, compared with the laminar simulation at $Re = 100$,

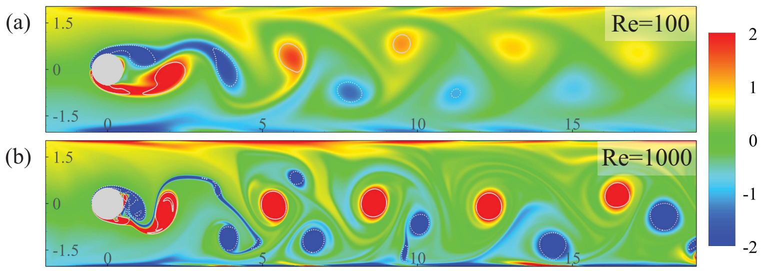

Figure 2: Snapshots of the flow fields obtained at (a) $Re = 100$ and (b) $Re = 1000$. Here, we show the vorticity normalized by U_0/D_0 and scaled to the $[-2, 2]$ interval. Vortices are also identified using the λ_{ci} criterion (Zhou *et al.*, 1999), and drawn with dark grey lines (+) and dark grey dashes (-). The increased non-linearity and chaos associated with the flow at $Re = 1000$, compared with $Re = 100$, is clearly visible.

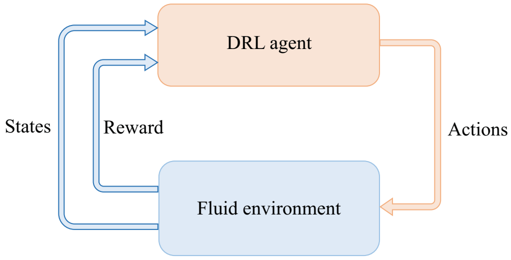

Figure 3: Schematics of the DRL loop

the simulation at $Re = 1000$ reveals more chaotic characteristics. In this case, vortices shed from the cylinder show asymmetrical patterns and strongly interact with the walls.

Using these configurations, each individual training simulation running for a numerical duration of $32T_0$ takes only about 2 and 5 minutes at $Re = 100$ and $Re = 1000$, respectively, using our in-house LBM solver accelerated by a NVIDIA K40c GPU device. This effective implementation is, to a great extent, what makes the present DRL-based AFC feasible.

2.3 Deep reinforcement learning

The DRL setup used for performing AFC is similar to what has been presented in previous work (Rabault *et al.*, 2019). A closed-loop interaction is defined between the DRL agent and the fluid environment, as visible in figure 3. The velocity sensor array gathers information from the CFD simulation, and is used as the state space for the environment. Jet velocities are used as control output, and the performance of the control is measured by the reward function described previously. The now well-established PPO algorithm is used for performing the learning, and the reader curious of implementation details is referred to the details of Appendix A. In this specific work, an in-house implementation of the PPO algorithm is used.

3 Results and discussions

Since the present work resorts on both an in-house CFD solver an in-house DRL code, we start by benchmarking AFC results at $Re = 100$, by comparing our results there with the findings of Rabault *et al.* (2019). We find that both the general strategy found, and the overall performance, are in good agreement. Details are reported in Appendix B. This constitutes an additional validation of the present setup. In all the following, we only discuss the novel training at $Re = 1000$.

We introduce two strategies when attempting to control the more challenging flow at $Re = 1000$. The first learning strategy consists in starting the learning from a randomly initialized policy. By contrast, in the second strategy, the learning at $Re = 1000$ is started from the well-trained policy at $Re = 100$, i.e., transfer learning is used.

During the learning progress, 5 actions are taken during one reference time T_0, and each episode has a duration of $32T_0$. Both the action frequency and the length of each episode are significantly longer than those adopted at $Re = 100$, due

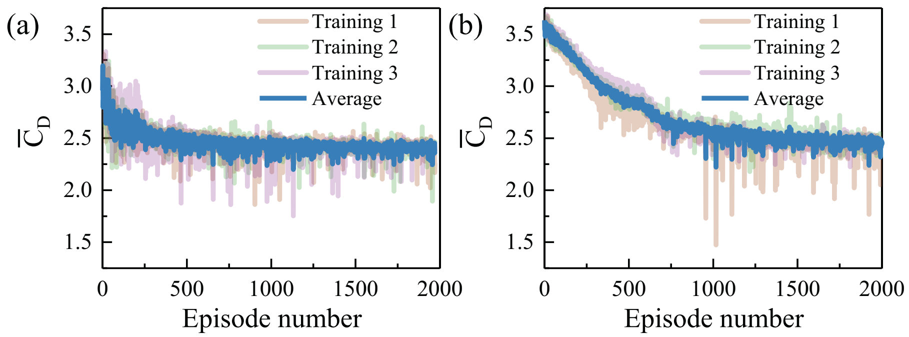

Figure 4: Learning curves of DRL-based AFC at $Re = 1000$ (a) starting from scratch, and (b) starting from the strategy learnt at $Re = 100$.

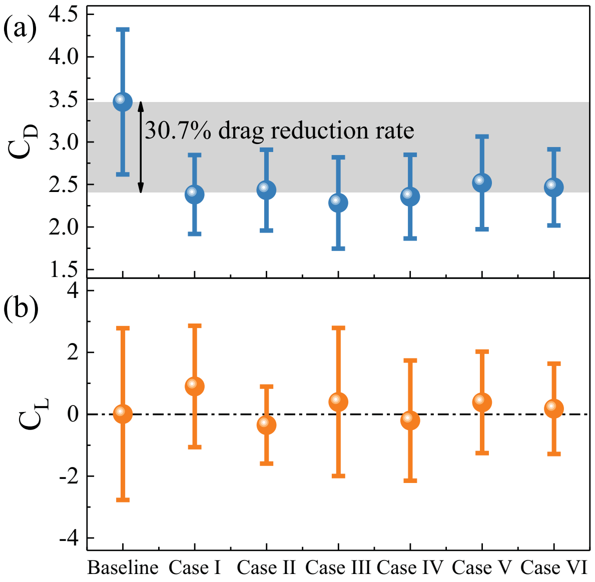

Figure 5: Comparisons between uncontrolled and DRL-controlled cases, using fully trained policies evaluated in deterministic mode. Here we show the time-averaged drag $\overline{C_D}$ (a) and lift $\overline{C_L}$ (b) coefficients obtained in each training case, with their standard deviations denoted as error bars.

to high-frequency fluctuations of the flow that can be observed in turbulent regimes at higher Re. Learning curves using both methods are presented in figure 4, where three learnings are performed in each case. We observe that the learning proceeds along different paths with both methods, but eventually reaches similar performance levels. The early trend shown in figure 4(b), with less reward fluctuations, suggests that although the initial transferred policy is far from a good one, it offers a relatively clear direction to explore, compared with the more chaotic exploration visible in figure 4(a). Compared to the low-Re case, the learning curves take more episodes to converge, illustrating that more complex flow systems are harder to control.

Following the trainings presented in figure 4, each converged policy is evaluated in deterministic mode using highly-resolved DNS configurations. Results are shown in figure 5. In the following descriptions, Cases I-III start from randomly initialized policies, and cases IV-VI start from the strategy learnt at $Re = 100$. Since the LES solver configuration adopted during the learning process has a typical error of 5% (as we preferred a configuration that gave fast computations, rather than high accuracy, as discussed earlier), we prefer to use the DNS configuration (mesh resolution as Configuration IV in Table 1) for all deterministic mode evaluations, for the sake of accuracy. This can also be seen as an additional check of the robustness and validity of the learnt policies: the control is valid and effective, even when changing the resolution of the underlying solver.

The performance of the learnt policies, shown in terms of the time-averaged value of $\overline{C_D}$ and $\overline{C_L}$ when deterministic control is applied, is depicted in figure 5. As visible there, all cases realize a notable drag reduction, ranging from

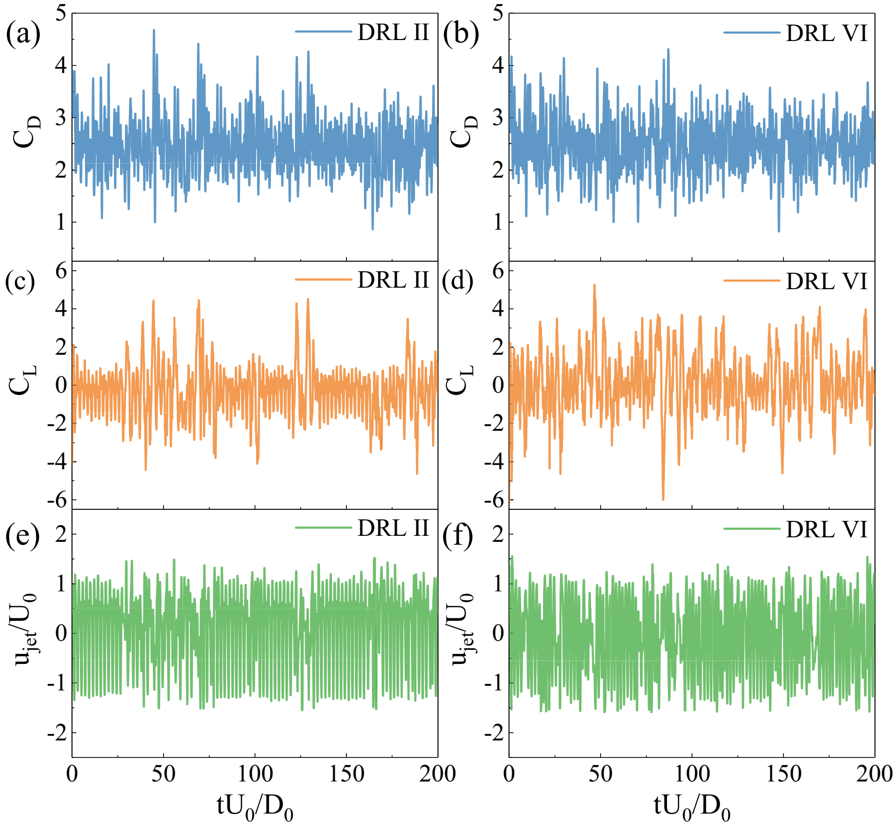

Figure 6: Results obtained by applying deterministic DRL control after the learning is completed. Temporal evolution of (a-b) drag, (c-d) lift, and (e-f) the AFC forcing. The left column corresponds to Case II, while the right column corresponds to Case VI.

27.4% (Case V), to 34.2% (Case III), with an average value of around 30%. Such minor differences between trainings are commonly observed, and arise from the eminently random exploration mechanisms present in the PPO algorithm, together with the strong nonlinearity and chaoticity of the turbulent flow considered. In addition to these effects on the drag, lift fluctuations are also greatly reduced, for example Case II shows a 55.2% reduction of the standard deviation of the lift coefficient C_L compared to baseline. In addition, one can observe that, in the transfer learning cases, the mean value of C_L is closer to zero, suggesting applying transfer learning can help find physically more appropriate control strategies.

To illustrate in more details these control policies, we choose to present two representative cases. Since all trainings achieve very similar drag reduction values, we choose to display the control effect for the two cases with the smallest lift fluctuations, i.e., Case II and Case VI. Temporal variations of C_D, C_L, and the jet velocity corresponding to these two cases are shown in figure 6. At $Re = 1000$, contrarily to what is obtained at $Re = 100$, the turbulent flow cannot be effectively stabilised and both C_D, C_L, and the jet strength u_{jet} remain chaotic.

In order to analyze in more details the control strategies found, we study the statistical properties of the baseline versus controlled flows. All following results are obtained analysing 10k instantaneous flow snapshots collected between non-dimensional times $t/T_0 = 100$ and $t/T_0 = 200$, where the simulations (and control for the actuated cases) start from a fully converged state at $t = 0$.

Results for the first- and second- order turbulent flow quantities are presented in figure 7. One can note from the plots of the time-averaged streamwise velocity that the recirculation bubble is largely elongated when control is active. Measured as the distance between the rear edge of the cylinder and the zero-streamwise velocity point along the midline of the channel, the recirculation bubble is elongated by rates of 211% and 195% in Case II and Case VI, respectively. Similarly to the laminar case (Rabault *et al.*, 2019), this elongated recirculation bubble weakens the Kármán vortex street, spatially delays the vortex shedding, and reduces the drag force. Turbulent fluctuations, as shown from the root-mean-square values of the velocity components, are also significantly reduced in the wake region undergoing control. Moreover, the averaged Reynolds stress is greatly weakened as well, which translates into less

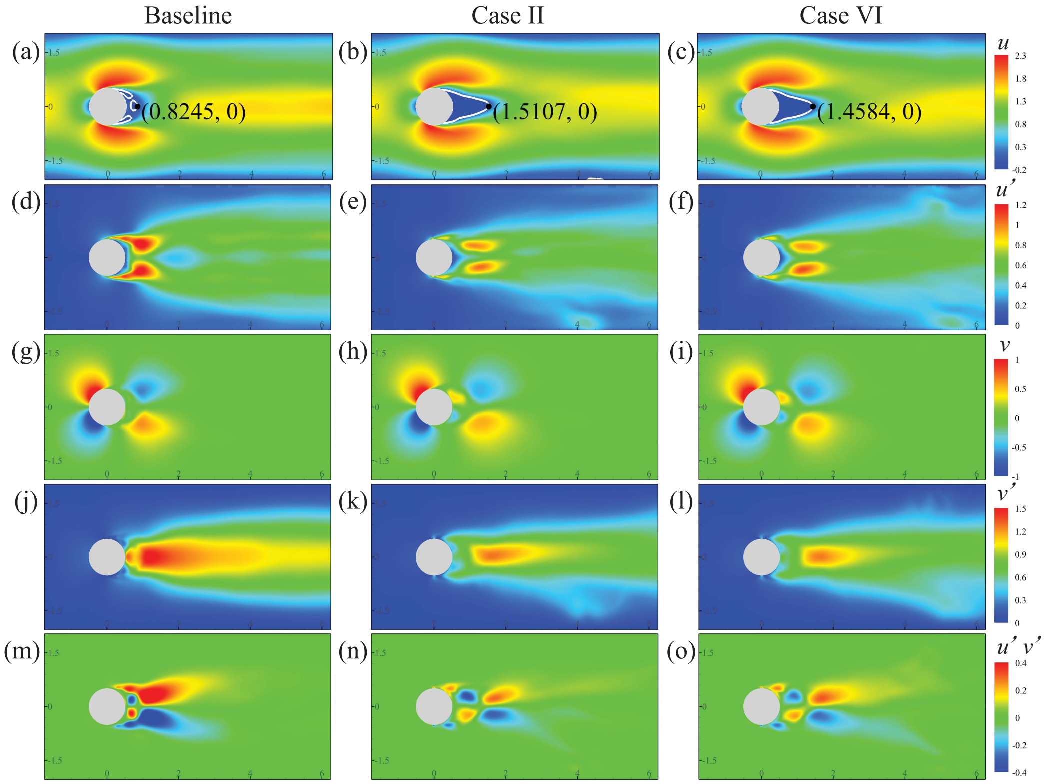

Figure 7: Flow features found through statistical analysis of the flow, in the case without (left column), and with (middle column, Case II, right column, Case VI) actuation. The subfigures (a∼c) present the time-averaged streamwise velocity, (d∼f) present the streamwise velocity rms, (g∼i) present the time-averaged transverse velocity, (j∼l) present the transverse velocity rms, (m∼o) present the averaged Reynolds stress. In (a c), the white line behind the cylinder is identified by the zero-streamwise velocity, and each coordinate corresponds to the edge of the recirculation bubble.

energy dissipation and less drag. Finally, one can observe that the general flow characteristics are similar in cases II and VI. Therefore, despite minor differences in the shape of the wake and the drag and lift fluctuations reductions effectively attained, the different policies found converge to similar control strategies.

An additional way to quantify the effect of the control policy is to study the turbulent kinetic energy (TKE) spectra of fixed points in the wake. This is presented in figure 8. There, the turbulent kinetic energy is calculated as $\sum_i \overline{u_i' u_i'}$, where the index represents each component of the velocity vector, and the superscript $'$ denotes the velocity fluctuation, calculated via subtracting the mean value. In this figure, the frequencies are normalized by the reference frequency $f_0 = U_0/D_0$. As visible in figure 8, the TKE is reduced in the cases undergoing actuation, which corresponds well to the findings of figure 7. In addition, the peak in TKE associated with the natural vortex shedding (which takes place at a value of $St = f/f_0 \approx 0.7$ in the baseline case) is effectively suppressed when control is active, which is one more evidence that the vortex shedding is effectively mitigated by the DRL agent.

In order to investigate how optimal the control policies found are, one can use an approach similar to what is discussed in the work of Bergmann *et al.* (2005), and presented in, among others, Rabault *et al.* (2019); Tang *et al.* (2020). In Ref. (Bergmann *et al.*, 2005), the authors suggest that the drag comes from two contributions, one arising from the 'symmetric base flow' behind the cylinder, which cannot be reduced, and the other one arising due to the vortex shedding in the wake, which, by contrast, can be reduced by AFC. Following the methodology recommended by Bergmann *et al.* (2005), we estimate the symmetric base flow drag by performing a simulation using a symmetrical boundary condition at the midline of the channel. Results are presented in figure 9, where streamlines are shown and the flow field is colored by streamwise velocity. In this case, the drag measured on the half cylinder has a mean value of 0.927. Therefore, according to Bergmann *et al.* (2005), if the vortex shedding is totally suppressed by the AFC, the drag coefficient for a circular cylinder at $Re = 1000$ would be 1.854. In this view, the optimal drag reduction could reach as high as 46%. Therefore, the fact that a drag reduction as large as 34% (case III in figure 5) is obtained with a single pair of small jets,

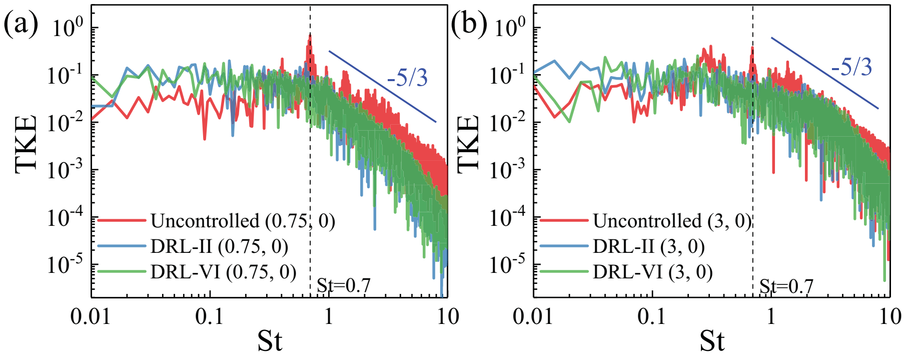

Figure 8: Turbulent kinetic energy spectra calculated using the temporal variations of two fixed points along the midline of the channel, i.e., (a) $0.75D_0$ and (b) $3D_0$ from the cylinder center.

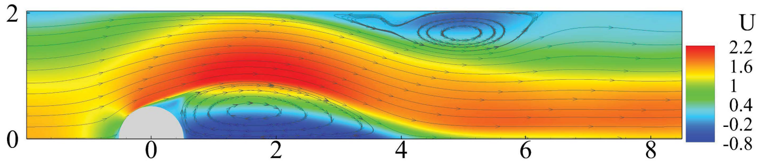

Figure 9: Time-averaged flow field of the half flow domain. The thin lines are the flow streamlines, and the background is colored by the local pressure.

is still remarkable. We anticipate that further improvements can be made by using finer-grained actuation, such as, for example, multiple jet pairs deployed on the cylinder.

4 Summary and conclusion

In the present work, we perform the first DRL-based active flow control in turbulent condition, with the aim to reduce the drag and lift fluctuations experienced by a 2D circular cylinder at $Re = 1000$. Our findings are two-folds:

- At an intermediate $Re = 1000$, where the flow shows turbulent features, DRL can find effective control strategies. Due to the much stronger nonlinear flow features, the learning process involves more episodes to reach convergence than in the laminar regime. Eventually, both randomly-initialized and transfer-learning strategies reach a similar performance level, i.e., a drag reduction of around 30% is obtained in deterministic mode.
- Through analysis of the DRL-controlled flow system, we note that the DRL agent finds effective and valid actuation strategies. The physical mechanism behind the drag reduction is twofold: firstly, the recirculation bubble is greatly enhanced, similar to what was observed in the laminar situation. Secondly, turbulence levels in the wake, and especially in the near-wall region, are significantly reduced, as revealed from the averaged velocity fluctuations and the Reynolds stress.

To the best of our knowledge, this is the first time that DRL, and more specifically the policy-based PPO algorithm, is applied successfully to performing active flow control in the turbulent regime, although the present configuration can only be viewed as weak turbulent flow. Therefore, this work further qualifies DRL as a relevant tool for solving AFC problems, and establishes a new milestone by illustrating the efficiency of DRL on a case significantly more complex than previous studies. We anticipate that further works will continuously increase the level of flow complexity that DRL is able to effectively control, and further progress towards real-world applications of this methodology.

5 Acknowledgement

We gratefully acknowledge financial support from the Research Grants Council of Hong Kong under the General Research Fund (Project No. 15249316 & 15214418), and the Departmental General Research Fund (Project No. G-YBXQ). Jean Rabault acknowledges funding obtained through the Petromaks II project (Grand No. 280625).

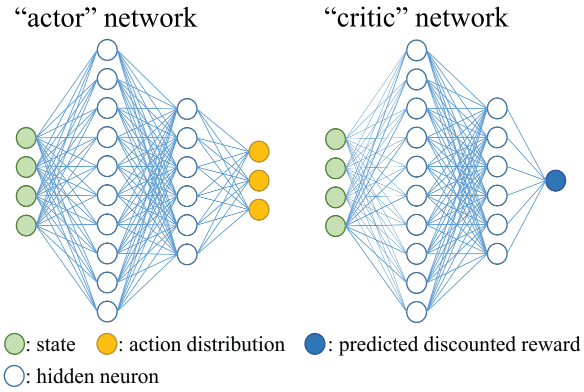

Figure 10: General 'actor-critic' setup used by the the PPO algorithm.

6 Appendix A: Deep reinforcement learning

In the present Appendix, we present a short reminder about the main lines of the PPO algorithm. The reader curious of more details is referred to any of the many discussions on the topic, such as Ref. (Schulman *et al.*, 2017; Rabault *et al.*, 2019, 2020).

The present work relies on the proximal policy optimization (PPO) algorithm (Heess *et al.*, 2017; Schulman *et al.*, 2017). In each episode, the agent applies the control policy N times and collects a sequence of states-actions-reward, i.e.:

$$\tau = (s_1, a_1, r_1), (s_2, a_2, r_2), (s_t, a_t, r_t), ..., (s_N, a_N, r_N). \tag{2}$$

To optimize against a long-term objective, the learning process is driven by the discounted reward, which is defined as:

$$R(t) = \sum_{t'>t} \gamma^{t'-t} r_{t'}, \tag{3}$$

where $0 < \gamma < 1$ is a discount factor usually close to 1, so that later reward values can contribute significantly to the reward goal.

The policy, π_Θ, is modeled by an ANN having the set of weights Θ. As shown in figure 10, the PPO algorithms uses two ANNs: an actor network which input is the state and output the action, and a critic network which input is the state and output the approximation of the discounted reward.

In order to perform training, the right loss function must be defined for each ANN. When training the 'critic' network, which produces a prediction of the discounted reward, an intermediate variable, i.e., the 'advantage', is defined as:

$$\hat{A}_t = \sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\Theta(s_t). \tag{4}$$

Then, the objective of the 'critic' network is to minimize the loss function measuring the discrepancy between the predicted and real values of the discounted reward, i.e.:

$$J_{critic} = \hat{E}_t(-\hat{A}_t^2), \tag{5}$$

where $\hat{E}_t$ denotes the empirical expectation over time.

As learning progresses, the PPO agent attempts to increase its achieved cumulative reward. For this, the 'actor' network is also being trained from the data generated through interaction with the environment. In the PPO implementation we use presently, we follow the work of Schulman *et al.* (2017), where a clipped surrogate objective function is used, i.e.:

$$J_{actor} = \hat{E}_t[min(R_t(\Theta)\hat{A}_t, clip(R_t(\Theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{6}$$

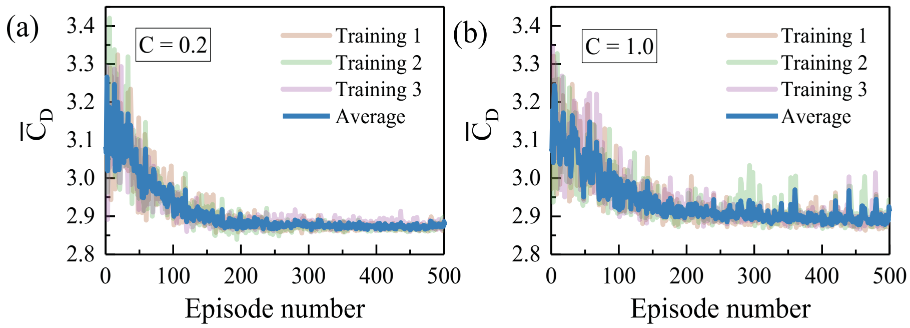

Figure 11: Learning process at $Re = 100$: (a) using lift penalization factor $C = 0.2$, (b) results using $C = 1$.

where ϵ is a hyper-parameter set to be 0.2 as recommended by Schulman *et al.* (2017), and $R_t(\Theta)$ is the probability ratio defined as $R_t(\Theta) = \pi_\Theta(a_t|s_t)/\pi_{old}(a_t|s_t)$. The clip term inside the above equation means that the probability ratio between the new policy and the old policy is constrained to an interval $[1-\epsilon, 1+\epsilon]$. Therefore, excessively large policy updates, which would make the training process unstable, are avoided.

When updating the policy, we use the adaptive moment estimation (Adam) optimizer (Kingma & Ba, 2014), which is a first-order gradient-based optimization for stochastic objective functions. To deal with continuous control, the actor network generates a beta distribution (Chou *et al.*, 2017), from which actions are sampled.

Once the learning has converged and the performance has reached a satisfactory level, a deterministic run is performed. In this case, the DRL agent does not continue learning from the sampled data, but it instead directly generates deterministic actions at each time step. Unlike in the learning stage when actions are sampled from a probability distribution, in the deterministic run, actions are determined by selecting the most likely action provided by the parametric distribution, i.e., no randomness is involved.

7 Appendix B: DRL control of laminar flow

In the present work, we adopt a different flow solver and DRL implementation compared to those used in, for example Ref. (Rabault *et al.*, 2019). Hence, we cross-validate our both parts of our methodology by performing the same AFC optimization as was presented in Ref. (Rabault *et al.*, 2019), but using the present DRL and CFD tools.

For this, we setup the exact same learning at $Re = 100$ as in Ref. (Rabault *et al.*, 2019). The learning curves, performed using two lift penalization factors, i.e., $C = 0.2$ and $C = 1$, are shown in figure 11. During the learning progress, each episode represents an individual case starting from a fully-developed flow without control. Each episode has a duration of $24T_0$ and features 60 actions, corresponding to around 7.3 vortex shedding periods if the flow is uncontrolled. The AFC forcing is smoothly interpolated 80 times within T_0 from the output of the PPO algorithm. The control policy characterized by the PPO agent is updated every 20 episodes.

Deterministic control is then performed using the optimal policies obtained after training. Three typical results are presented in figure 12. Correspondingly, the achieved drag reduction rate is 7.63%, 7.47%, and 7.13%, close to the drag reduction rate reported by Rabault *et al.* (2019). With different penalization coefficients, the outcomes of the trainings eventually fall into different solutions. Qualitatively, if C is relatively small, the agent prefers a larger drag reduction rate, corresponding to the situation shown in figure 12(g h), where the jet on one side keeps blowing while the other jet works in the suction mode. In real applications, this situation is usually not desirable because the subsequent lateral force could bring other disadvantages to the structure (figure 12(d e)). On the other hand, if C is large enough, drag reduction contributes less to the overall reward, thus the final drag reduction rate could be smaller. However, in this situation, the undesirable lateral force is reduced, as in figure 12(f), where we note that the lift force fluctuates around zero. Moreover, compared to the uncontrolled case, the lift fluctuation in figure 12(d f) is reduced to a much smaller level, suggesting that the vortex shedding is better mediated.

Another interesting phenomenon we note from the deterministic control is that, for all cases, the vortex shedding frequency is reduced by around 10%. This frequency shift has been pointed out by Erhard *et al.* (2010) and explained from the theory of flow stability, by analyzing unstable regions with and without control. In a previous study, we also discussed the effect of active control on the flow stability (Wang *et al.*, 2017), and we believe that the control described herein shares a similar mechanism.

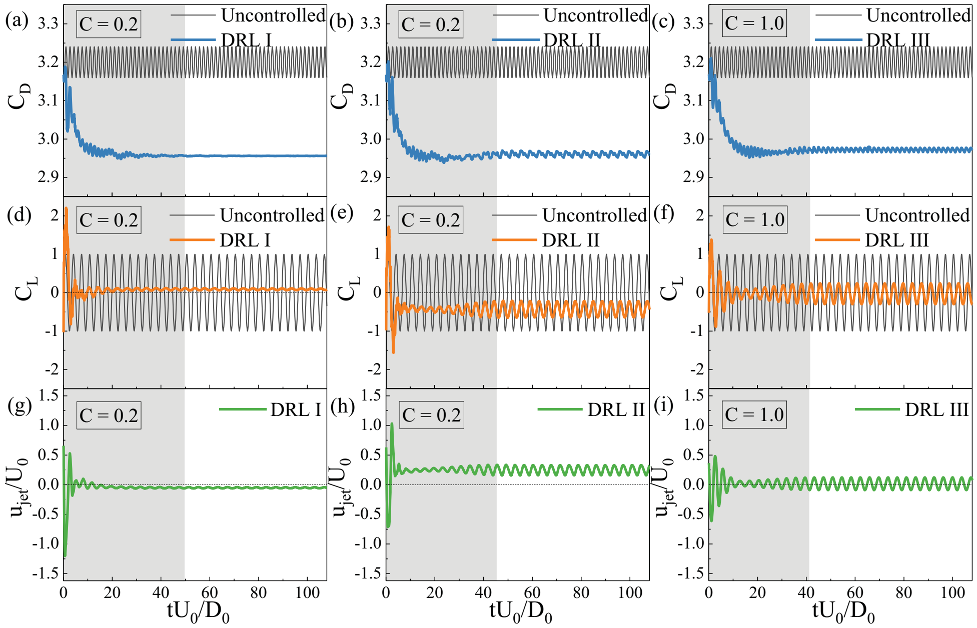

Figure 12: Three typical control strategies observed in the deterministic run at $Re = 100$. Results shown in the left and middle columns use lift penalization factor $C = 0.2$, and results shown in the right column correspond to $C = 1$.

References

Beintema, Gerben, Corbetta, Alessandro, Biferale, Luca & Toschi, Federico 2020 Controlling rayleigh-bénard convection via reinforcement learning. *arXiv preprint arXiv:2003.14358* .

Belus, Vincent, Rabault, Jean, Viquerat, Jonathan, Che, Zhizhao, Hachem, Elie & Reglade, Ulysse 2019 Exploiting locality and translational invariance to design effective deep reinforcement learning control of the 1-dimensional unstable falling liquid film. *AIP Advances* **9** (12), 125014.

Bergmann, Michel, Cordier, Laurent & Brancher, Jean-Pierre 2005 Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of fluids* **17** (9), 097101.

Brunton, Steven L, Noack, Bernd R & Koumoutsakos, Petros 2020 Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics* **52**, 477–508.

Bucci, Michele Alessandro, Semeraro, Onofrio, Allauzen, Alexandre, Wisniewski, Guillaume, Cordier, Laurent & Mathelin, Lionel 2019 Control of chaotic systems by deep reinforcement learning. *Proceedings of the Royal Society A* **475** (2231), 20190351.

Chen, Yu, Cai, Qingdong, Xia, Zhenhua, Wang, Moran & Chen, Shiyi 2013 Momentum-exchange method in lattice boltzmann simulations of particle-fluid interactions. *Physical Review E* **88** (1), 013303.

Chou, Po-Wei, Maturana, Daniel & Scherer, Sebastian 2017 Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 834–843. JMLR. org.

Erhard, P, Etling, Dieter, Muller, U, Riedel, U, Sreenivasan, KR & Warnatz, J 2010 *Prandtl-essentials of fluid mechanics*, , vol. 158. Springer Science & Business Media.

Fan, Dewei, Zhou, Yu & Noack, Bernd R 2018 Artificial intelligence control of a turbulent jet. *Artificial intelligence* **10**, 13.

Gautier, Nicolas, Aider, J-L, Duriez, THOMAS, Noack, BR, Segond, Marc & Abel, Markus 2015 Closed-loop separation control using machine learning. *Journal of Fluid Mechanics* **770**, 442–457.

He, Xiaoyi, Zou, Qisu, Luo, Li-Shi & Dembo, Micah 1997 Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice boltzmann bgk model. *Journal of Statistical Physics* **87** (1-2), 115–136.

HEESS, NICOLAS, TB, DHRUVA, SRIRAM, SRINIVASAN, LEMMON, JAY, MEREL, JOSH, WAYNE, GREG, TASSA, YUVAL, EREZ, TOM, WANG, ZIYU, ESLAMI, SM & OTHERS 2017 Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286* .

KINGMA, DIEDERIK P & BA, JIMMY 2014 Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

MNIH, VOLODYMYR, KAVUKCUOGLU, KORAY, SILVER, DAVID, RUSU, ANDREI A, VENESS, JOEL, BELLEMARE, MARC G, GRAVES, ALEX, RIEDMILLER, MARTIN, FIDJELAND, ANDREAS K, OSTROVSKI, GEORG & OTHERS 2015 Human-level control through deep reinforcement learning. *Nature* **518** (7540), 529–533.

RABAULT, JEAN, KUCHTA, MIROSLAV, JENSEN, ATLE, RÉGLADE, ULYSSE & CERARDI, NICOLAS 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics* **865**, 281–302.

RABAULT, JEAN & KUHNLE, ALEXANDER 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids* **31** (9), 094105.

RABAULT, JEAN, REN, FENG, ZHANG, WEI, TANG, HUI & XU, HUI 2020 Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization. *Journal of Hydrodynamics* **32** (2), 234–246.

REDDY, GAUTAM, CELANI, ANTONIO, SEJNOWSKI, TERRENCE J & VERGASSOLA, MASSIMO 2016 Learning to soar in turbulent environments. *Proceedings of the National Academy of Sciences* **113** (33), E4877–E4884.

REN, FENG, SONG, BAOWEI & HU, HAIBAO 2018*a* Lattice boltzmann simulations of turbulent channel flow and heat transport by incorporating the vreman model. *Applied Thermal Engineering* **129**, 463–471.

REN, FENG, SONG, BAOWEI, ZHANG, YA & HU, HAIBAO 2018*b* A gpu-accelerated solver for turbulent flow and scalar transport based on the lattice boltzmann method. *Computers & Fluids* **173**, 29–36.

REN, FENG, WANG, CHENGLEI & TANG, HUI 2019 Active control of vortex-induced vibration of a circular cylinder using machine learning. *Physics of Fluids* **31** (9), 093601.

SCHULMAN, JOHN, WOLSKI, FILIP, DHARIWAL, PRAFULLA, RADFORD, ALEC & KLIMOV, OLEG 2017 Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

SILVER, DAVID, HUANG, AJA, MADDISON, CHRIS J, GUEZ, ARTHUR, SIFRE, LAURENT, VAN DEN DRIESSCHE, GEORGE, SCHRITTWIESER, JULIAN, ANTONOGLOU, IOANNIS, PANNEERSHELVAM, VEDA, LANCTOT, MARC & OTHERS 2016 Mastering the game of go with deep neural networks and tree search. *nature* **529** (7587), 484.

TANG, HONGWEI, RABAULT, JEAN, KUHNLE, ALEXANDER, WANG, YAN & WANG, TONGGUANG 2020 Robust active flow control over a range of reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Physics of Fluids* **32** (5), 053605.

VERMA, SIDDHARTHA, NOVATI, GUIDO & KOUMOUTSAKOS, PETROS 2018 Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences* **115** (23), 5849–5854.

VREMAN, AW 2004 An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of fluids* **16** (10), 3670–3681.

WANG, CHENGLEI, TANG, HUI, SIMON, CM & DUAN, FEI 2017 Lock-on of vortex shedding to a pair of synthetic jets with phase difference. *Physical Review Fluids* **2** (10), 104701.

YU, DAZHI, MEI, RENWEI, LUO, LI-SHI & SHYY, WEI 2003 Viscous flow computations with the method of lattice boltzmann equation. *Progress in Aerospace Sciences* **39** (5), 329–367.

ZHAO-LI, GUO, CHU-GUANG, ZHENG & BAO-CHANG, SHI 2002 Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice boltzmann method. *Chinese Physics* **11** (4), 366.

ZHOU, JIGEN, ADRIAN, RONALD J, BALACHANDAR, S & KENDALL, TM 1999 Mechanisms for generating coherent packets of hairpin vortices in channel flow. *Journal of fluid mechanics* **387**, 353–396.