

# Deep reinforcement transfer learning of active control for bluff body flows at high Reynolds number

Zhicheng Wang<sup>1,3†</sup>, Dixia Fan<sup>2 †</sup>, Xiaomo Jiang<sup>3,4</sup>, Michael S. Triantafyllou<sup>5,6‡</sup> and George Em Karniadakis<sup>7</sup>

<sup>1</sup>Laboratory of Ocean Energy Utilization of Ministry of Education, Dalian University of Technology, Dalian, 116024, China

<sup>2</sup>School of Engineering, Westlake University, Hangzhou, 310024, China

<sup>3</sup>State Key Lab of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Provincial Key Lab of Digital Twin for Industrial Equipment, Dalian University of Technology, Dalian, 116024, China

<sup>4</sup>School of Energy and Power Engineering, Dalian University of Technology, Dalian, 116024, China

<sup>5</sup>Department of Mechanical Engineering, Massachusetts Institute Technology, Cambridge, MA 02139, USA

<sup>6</sup>MIT Sea Grant College Program, Cambridge, MA, 02139, USA

<sup>7</sup>Division of Applied Mathematics and School of Engineering, Brown University, Providence, RI 02912 USA

(Received xx; revised xx; accepted xx)

We demonstrate how to accelerate the computationally taxing process of deep reinforcement learning (DRL) in numerical simulations for active control of bluff body flows at high Reynolds number ( $Re$ ) using transfer learning. We consider the canonical flow past a circular cylinder whose wake is controlled by two small rotating cylinders. We first pre-train the DRL agent using data from inexpensive simulations at low  $Re$ , and subsequently we train the agent with small data from the simulation at high  $Re$  (up to  $Re = 1.4 \times 10^5$ ). We apply transfer learning (TL) to three different tasks, the results of which show that TL can greatly reduce the training episodes, while the control method selected by TL is more stable compared to training DRL from scratch. We analyze for the first time the wake flow at  $Re = 1.4 \times 10^5$  in detail and discover that the hydrodynamic forces on the two rotating control cylinders are not symmetric.

**Key words:** Deep reinforcement learning, bluff body, drag reduction, high Reynolds number

## 1. Introduction

Deep reinforcement learning (DRL) has been shown to be an effective way of selecting optimal control of flows in diverse applications, including fish bio-locomotion (Gazzola *et al.* 2014; Verma *et al.* 2018), optimization of aerial/aquatic vehicles' path and motion (Reddy *et al.* 2016; Colabrese *et al.* 2017; Novati *et al.* 2019), active flow control for bluff bodies (Ma *et al.* 2018; Bucci *et al.* 2019; Rabault *et al.* 2019; Ren *et al.* 2021), shape optimization (Viquerat *et al.* 2021), and learning turbulent wall models (Bae & Koumoutsakos 2022).

† equal contribution

‡ Email address for correspondence: mistetri@mit.edu

The flow past a smooth circular cylinder has been characterized as a “kaleidoscope” (Morkovin 1964) of interesting fluid mechanics phenomena as the Reynolds number ( $Re_D = \frac{UD}{\nu}$ ) is increases from 20 to  $2 \times 10^5$  (Cheng *et al.* 2017; Dong *et al.* 2006). The flow develops from two-dimensional steady wake to three-dimensional unsteady vortex shedding, followed by wake transition, shear layer instability, and boundary layer transition. Due to the large contrast among flow patterns of different scales, accurate numerical simulations of turbulent flow are usually limited to small and moderate  $Re_D$ . For the same reason, using numerical simulation to study active control of the flow at high  $Re_D$  has rarely been reported.

In our previous study (Fan *et al.* 2020), an efficient DRL algorithm was developed to discover the best strategy to reduce the drag force, by using the DRL to control the rotation of two small cylinders placed symmetrically in the wake of the big cylinder. Specifically, for the flow at  $Re_D = 10^4$ , it has been demonstrated that DRL can discover the same control strategy as experiments in learning from data generated by the high fidelity numerical simulation. Simulated data is noise free, but learning from the simulated data is restricted by the simulation speed. For instance, in (Fan *et al.* 2020), in the case of  $Re_D = 10^4$ , it took 3.3 hours to generate the simulated data for each episode, and one month to finalize the DRL strategy. In contrast, in companion experimental work it only took a few minutes to learn the same control strategy. As  $Re_D$  increases to  $1.4 \times 10^5$ , one can expect that even for the simplest task in (Fan *et al.* 2020), performing DRL from scratch could take months, and hence it might not be practical to apply DRL directly to the more difficult tasks at higher  $Re_D$ .

In this paper, in order to tackle the aforementioned problem, we propose a learning paradigm that first trains the DRL agent using the simulated data at low  $Re_D$  and subsequently transfers the domain knowledge to the learning at higher  $Re_D$ . The rest of the paper is organized as follows: section 2 gives the details of the simulation model, numerical method and DRL algorithm; section 3 presents the DRL and transfer learning results for different tasks at three different  $Re_D$ , namely,  $Re_D = 500$ ,  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ ; section 4 gives the conclusion of the current paper. Finally, appendix A presents the simulation results of the cases that the control cylinders are rotating at constant speed, while appendix B presents the validation of the numerical method for simulations at high Reynolds number.

## 2. Model and Methods

In this paper, the bluff body flow control problem has the same geometry as the one in (Fan *et al.* 2020) but here we focus on demonstrating the feasibility of transferring DRL knowledge from low  $Re_D$  to high  $Re_D$ , solely in the environment of numerical simulation. As shown in Fig. 1, the computational model consists of a main cylinder and two fast rotating smaller control cylinders. This configuration is used to alter the flow pattern around the wake of the big cylinder, with the objective of reducing the effective system drag, or maximizing the system power gain. We note that a similar control strategy has been studied at low Reynolds number and has been shown that when the control cylinders are placed at appropriate locations and rotating at a sufficiently fast speed, they are able to change the boundary layers on the main cylinder, i.e., reattach the boundary layer and form a narrower wake, resulting in notable drag reduction.

### 2.1. Numerical method

The numerical simulation of the unsteady incompressible flow past the cylinders of different diameters is achieved by employing the high-order CFD code *Nektar* that employs spectral element discretization on the  $(x - y)$  plane and Fourier expansion along the cylinder axial direction ( $z$ ) (Karniadakis & Sherwin 2005). In particular, we employ the entropy viscosity method (EVM) based large-eddy simulations (LES), which was originally proposed by (Guermond *et al.* 2011*a,b*)

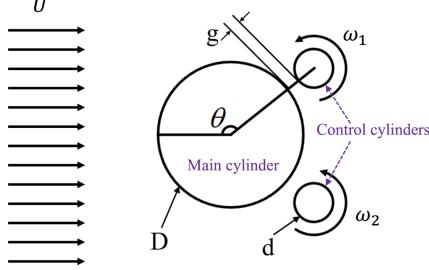


Figure 1: Sketch of the flow control problem.  $U$  is the inflow velocity,  $D$  is the diameter of the main cylinder,  $d = \frac{1}{8}D$  is the diameter of the control cylinder,  $g = \frac{1}{20}D$ ,  $\theta = 120^\circ$ .  $\omega_1, \omega_2$  are the angular velocities of the control cylinder 1 and 2, respectively. Specifically,  $\omega_1 = \epsilon_1 \epsilon_{\max}$ ,  $\omega_2 = \epsilon_2 \epsilon_{\max}$ , where  $\epsilon_1, \epsilon_2 \in [0, 1]$  are given by the DRL agent, and  $\epsilon_{\max}$  is a constant.

and later developed further for complex flows by (Wang *et al.* 2019, 2018; Du *et al.* 2023). In the simulations, the computational domain has a size of  $[-7.5D, 20D] \times [-10D, 10D]$  in  $x, y$  direction, respectively. A uniform inflow boundary condition ( $u = U, v = 0, w = 0$ ) is prescribed at  $x = -7.5D$ , where  $u, v, w$  are the three components of the velocity vector  $\mathbf{u}$ . The outflow boundary ( $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$  and  $p = 0$ ) is imposed on  $x = 20D$ , where  $p$  is the pressure, and  $\mathbf{n}$  is the normal vector. Wall boundary condition is applied on the main cylinder surface, while the velocity on the control cylinders are given by the DRL agent during the simulation. Moreover, a periodic boundary condition is assumed on the lateral boundaries ( $y = \pm 10D$ ). Note that the spanwise length of the computational domain depends on the  $Re_D$ , namely, it is  $6D$  and  $2D$  in the simulation of  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ , respectively.

The computational mesh is similar to the one used in (Fan *et al.* 2020). At  $Re_D = 500$ , the computational domain is partitioned into 2 462 quadrilateral elements, while at  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ , it consists of 2 790 quadrilateral elements. The elements are clustered around the cylinders in order to resolve the boundary layers. Specifically, on the main cylinder wall normal directions, the size of the first layer element ( $\Delta r$ ) is designed carefully so that at  $Re = 10^4$ ,  $\Delta r = 4 \times 10^{-3}$ ; at  $Re_D = 1.4 \times 10^5$ ,  $\Delta r = 1.6 \times 10^{-3} D$ . On this mesh, with the spectral element mode 4,  $y^+ < 1$  can be guaranteed. In the simulation, the time step ( $\Delta t$ ) satisfies the Courant condition  $CFL = \frac{\Delta t |u|}{\Delta x} \leqslant 0.75$ . Note that in all the DRL cases the time duration between two consecutive state queries is fixed at  $0.12 \frac{D}{U}$ , e.g., in the case of  $Re_D = 1.4 \times 10^5$ ,  $\Delta t = 10^{-4}$ , the state data will be collected every 1 200 steps. It is worth noting that the RL guided LES starts from the fully turbulent flow, which is the result of previous simulation of flow in the same geometry configuration with the small cylinders held still.

## 2.2. Deep reinforcement learning and transfer learning

DRL identifies the optimal control strategy by maximizing the expected cumulative reward, using the data generated by the simulation. More details of the DRL can be seen in (Fan *et al.* 2020), but the main principle is summarized as follows,

$$J(\pi) = \mathbb{E}_{(s_i, a_i) \sim p_\pi} \sum_{i=0}^T \gamma^i r_i, \quad (2.1)$$

where  $J$  is the so-called cumulative reward,  $\mathbb{E}$  denotes the calculation of the expected value,  $\gamma \in (0, 1]$  is a discount factor and  $p_\pi$  denotes the state-action marginals of the trajectory distribution induced by the policy  $\pi$ .  $s_i \in \mathcal{S}$  is the observed state,  $a_i \in \mathcal{A}$  is the given actions

with respect to the policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$ , and  $r_i$  is the received reward, at discrete time step  $i$ . As shown in Equation 2.1, the objective of DRL is to discover the policy  $\pi_\phi$  parameterized by  $\phi$ , which maximizes the expected cumulative reward. Specifically, in the current work, the state variable is the concatenation of  $C_L$ ,  $C_D$ ,  $C_{f,1}$  and  $C_{f,2}$ , the action is the concatenation of  $\epsilon_1$  and  $\epsilon_2$ , where  $C_D$  and  $C_L$  are the drag and lift force coefficient,  $C_{f,1}$  and  $C_{f,2}$  are the frictional force coefficient on the control cylinder 1 and 2,  $\epsilon_1$  and  $\epsilon_2$  are the action variable for control cylinder 1 and 2, respectively.

In this paper, the Twin Delayed Deep Deterministic Policy Gradient Algorithm (TD3) (Fujimoto *et al.* 2018; Fan *et al.* 2020) has been employed. TD3 consists of two neural networks, one for the actor and the other one for the critic, and both are feedforward neural networks with two hidden layers and 256 neurons. The discount factor  $\gamma$  is set as 0.99. The standard deviation of the policy exploration noise  $\sigma$  is set as 0.005. The Adam optimizer with learning rate  $10^{-4}$  is used, and the batch size is  $N = 256$ .

Before proceeding to the results, it is worthy discussing the difficulties of training the model-free DRL feeding with simulated data. The first difficulty is how to generate more training data as fast as possible. To this end, this paper will propose a multi-client-mono-server DRL paradigm, in which multiple simulations running simultaneously and independently to provide training data to the single DRL server. In each simulation, once a training data is collected, it will be provided to the DRL server. Note that, the data exchange between the DRL server and simulation clients is achieved by the XML-RPC protocol.

The second difficulty concerns the simulation at high Reynolds number, which makes the multi-client-mono-serve DRL not practical, since a single simulation already requires a significant amount of computing resource. To overcome this difficulty, initially, the DRL agent will be trained using the data collected from the much cheaper simulations at low  $Re_D$ , and then the neural networks will be used in the simulation at high  $Re_D$ , while the network parameters will be re-trained using the new data.

### 3. Results and discussion

In order to demonstrate the feasibility of using TL in DRL for active flow control, we consider three different tasks, in which the states, actions, and reward functions, corresponding to each task, are given as follows,

- **Task 1:** Minimizing  $C_D$ : two states,  $C_D$  and  $C_L$ ; two independent actions,  $\epsilon_1$  and  $\epsilon_2$ ; reward function,  $r = -\text{sign}(C_D)C_D^2 - 0.1C_L^2$ .
- **Task 2:** Maximization of the system power gain efficiency: four states,  $C_D$ ,  $C_L$ ,  $C_{f,1}$  and  $C_{f,2}$ ; one action,  $\epsilon_1 = -\epsilon_2$ ; reward function  $r = -\eta$ , where

$$\eta = |C_D| + \frac{\pi d}{D} (|\epsilon_1|^3 C_{f,1} + |\epsilon_2|^3 C_{f,2}) \epsilon_{\max}^3. \quad (3.1)$$

- **Task 3:** Maximization of the system power gain efficiency: sates and reward functions are the same as those in **Task 2**; two independent actions,  $\epsilon_1$  and  $\epsilon_2$ .

#### 3.1. Learning from scratch at low $Re$

We start DRL for **Task 1** from scratch with  $\epsilon_{\max} = 5$ , in three dimensional simulation of flow at  $Re_D = 500$ . Before presenting the results, we explain why  $\epsilon_{\max} = 5$  is chosen. As shown in our previous study (Fan *et al.* 2020),  $C_D$  decreases with  $\epsilon_{\max}$ , which implies that the generated state variables ( $C_D$ ,  $C_L$ ) will be located in a wider range with increasing  $\epsilon_{\max}$ . In order to enhance the generalization of the neural networks, it is beneficial to experience the training data in a wider range before transferring to a more challenging task.

Fig. 2 shows the evolution of  $\epsilon_1$ ,  $\epsilon_2$ ,  $C_D$  and  $C_L$ , as well as the variation of the vortex shedding

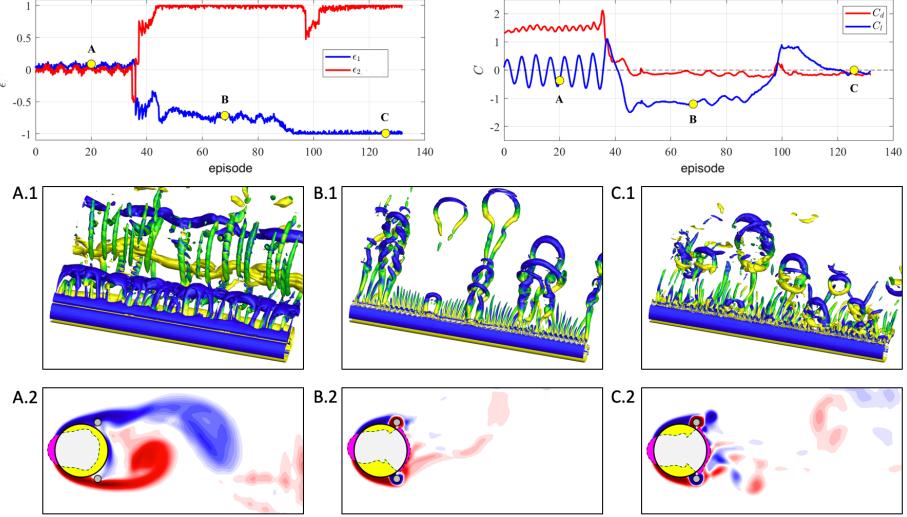


Figure 2: Task 1: Evolution of reinforcement learning and corresponding simulated three-dimensional vortex shedding patterns at  $Re_D = 500$ ,  $\epsilon_{\max} = 5$ . Note that the DRL is starting from scratch, but the simulation is starting from a fully developed flow where the smaller controlling cylinders are stationary.

pattern due to active control by DRL at  $Re_D = 500$ . Initially for (40 episodes approximately), DRL knows very little on how to minimize  $C_D$ ; small values of  $\epsilon_1, \epsilon_2$  are provided by DRL, thus the control cylinders have small effect on the flow, hence regular vortex shedding pattern can be observed, as shown by figure A.1 and A.2. However, around the 40<sup>th</sup> episode, DRL has figured out the correct rotating direction to reduce  $C_D$ , and the wake patterns are now hairpin vortices emanating from the gap between the control and main cylinder. At this stage,  $\epsilon_1 \neq \epsilon_2$ , and the wake behind the main cylinder is not symmetric, which gives rise to a large  $C_L$ , as shown in sub-figures B.1 and B.2. Around 110<sup>th</sup> episode, DRL finally identified the best rotation speed to minimize  $C_D$ , and, correspondingly, the vortex shedding from the main cylinder has been eliminated.

To sum up, when DRL networks are initialized randomly, the DRL agent can gradually manage to learn the optimal policy. The learning process has lasted for more than 100 episodes, and the generated training data,  $C_D$  and  $C_L$ , are roughly in the ranges  $[-0.1, 2.0]$  and  $[-1.5, 1.0]$ , respectively.

### 3.2. Transfer learning from low $Re$ to high $Re$

In the previous subsection, we have demonstrated that for the flow at low Reynolds number, although the training data from the simulation are noise free, it still takes DRL to go through 100 episodes, i.e., 1000 data points, to discover the right policy. Hence, DRL from scratch is impractical to be applied to flow at high  $Re$ , since it will tax the computational resources heavily, as it will take much longer to generate the training data. In order to use the data more efficiently and reduce the overall computing time, we employ a transfer learning approach at high  $Re_D$ .

Fig. 3 presents the time traces of  $\epsilon_1, \epsilon_2$  and  $C_D$  of **Task 1** with  $\epsilon_{\max} = 3.66$  at  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ . In particular, the results of DRL from scratch at  $Re_D = 10^4$  are plotted together. Note that here both cases of TL were initialized using the same DRL network, which was obtained in **Task 1** at  $Re_D = 500$  shown in Fig. 2(a). We observe that TL spent 60 episodes only to discover the optimal policy, while the DRL from scratch went through over 200 episodes to reach a comparable decision. Moreover,  $\epsilon_1$  (pink line, in sub-figure(a)) given by the DRL

183 from scratch keeps oscillating even after 500 episodes, but the value given by the TL shows less  
 184 variation, although the same value of the noise parameter ( $\sigma$ ) is used in both cases.

185 In particular, in the TL at  $Re_D = 10^4$ , as shown in Figure 3, the TL agent manages to find the  
 186 correct rotating direction in less than 10 episodes, and as it reaches 20<sup>th</sup> episodes, the TL starts  
 187 exploring a new policy, during which both  $\epsilon$  and  $C_D$  show notable variations, associated with the  
 188 so-called “catastrophic forgetting” (Kirkpatrick *et al.* 2017). After 55<sup>th</sup> episode, the TL can make  
 189 the correct and stable decision.

190 In **Task 1**, as  $Re_D$  is increased from  $10^4$  to  $1.4 \times 10^5$ , the wake behind the main cylinder  
 191 becomes very complex, although the boundary layer is laminar at both values of  $Re_D$ . The  
 192 learning process at  $Re_D = 1.4 \times 10^5$  is very similar to that of  $Re_D = 10^4$ . In 25 episodes, the TL  
 193 is able to identify the correct rotating directions. At 50<sup>th</sup> episode, the rotating speed on the control  
 194 cylinder 1 begins to show variation and it starts a new exploration, due to the catastrophic  
 195 forgetting. Around 80<sup>th</sup> episode, the rotating speed on the control cylinder 2 also starts to vary,  
 196 but the rotating speeds on both control cylinders quickly return to optimal value, i.e.,  $|\epsilon_{\pm 1}| = 1$ .

197 We would like to emphasize that the high fidelity large-eddy simulation of flow past a cylinder  
 198 accompanied by two rotating cylinders at  $Re_D = 1.4 \times 10^5$  is still a challenge. To the best of the  
 199 authors’ knowledge, the details of the flow pattern have not been studied before. In figure 3 (d), the  
 200 top snapshot exhibits the vortical flow, while the control cylinders remain stationary. The bottom  
 201 snapshot shows the flow structures at 125<sup>th</sup> episode, when the control cylinders are rotating at  
 202 optimal speed. We see that as the control cylinder is rotating, the large scale streamwise braid  
 203 vortices are mostly replaced by the hairpin vortices emanating from the gap between the main  
 204 cylinder and control cylinders, and the wake becomes narrower, which is a sign of a smaller  $C_D$ .

205 Next we turn our attention to Task 2, where the objective is to maximize the system power gain  
 206 efficiency,  $\eta$ , under the condition that  $\epsilon_2 = -\epsilon_1$ . Note that the objective of the current **Task 2** is  
 207 similar to the **Task 2** in our previous study (Fan *et al.* 2020), except that here we have used the  
 208 instantaneous  $C_f$  obtained from the simulation, while  $C_f$  is a pre-defined constant in (Fan *et al.*  
 209 2020). Nonetheless, our previous study has revealed that it is much more difficult for DRL to  
 210 identify the optimal control strategy in **Task 2** than that in **Task 1**. In **Task 2**, we have started from  
 211 the DRL from scratch using two dimensional (2D) simulation at  $Re_D = 500$ . In particular, since  
 212 the 2D simulation is relatively inexpensive and fast, we have run 16 simulations concurrently to  
 213 provide the training data to a single DRL. For the TL at  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ , the  
 214 simulation is very expensive, thus only a single simulation was performed. It is worthy noting  
 215 that  $\eta$  at low  $Re_D$  is very different from that at high  $Re_D$ , as shown in Appendix A, thus it is  
 216 expected that TL in **Task 2** and **Task 3** will be more challenging.

217 Figure 4(a) shows the learning process of DRL from scratch for **Task 2** at  $Re_D = 500$ . In  
 218 particular, the learning process using a single 2D simulation (left part) and 16 2D simulations  
 219 (right) are plotted together. In the left part of figure 4(a), DRL manages to find the correct rotating  
 220 direction after 75 episodes, but it barely makes the optimal decision, although it has been trained  
 221 over 300 episodes. On the other hand, in the case of 16 simulations running concurrently, as  
 222 shown in the right part of figure 4(a), DRL can identify the correct rotating direction in less than  
 223 10 episodes, and is able to make the optimal decision before the 75<sup>th</sup> episode.

224 Subsequently, the DRL networks trained in figure 4 (a) are both applied to the TL at  $Re_D = 10^4$ ,  
 225 and the results are shown in figure 4(b). We observe from the left part of figure 4(b) that although  
 226 the DRL of figure (a) has identified the correct policy it gives a very chaotic  $\epsilon_1$  in the first 60  
 227 episodes in TL. On the other hand, as shown in right part of figure 4(b), the TL is able to give the  
 228 correct rotating direction in less than 10 episodes, and after 30 episodes it has managed to reach  
 229 the optimal decision. When applied to the simulation flow at  $Re = 1.4 \times 10^5$ , the TL manages to  
 230 give stable action values after the 30<sup>th</sup> episode, but in short period just after the 40<sup>th</sup> episode and  
 231 until the end of the simulation, an unstable action is given as shown by the variation of  $\epsilon_1$  and  $\eta$  in

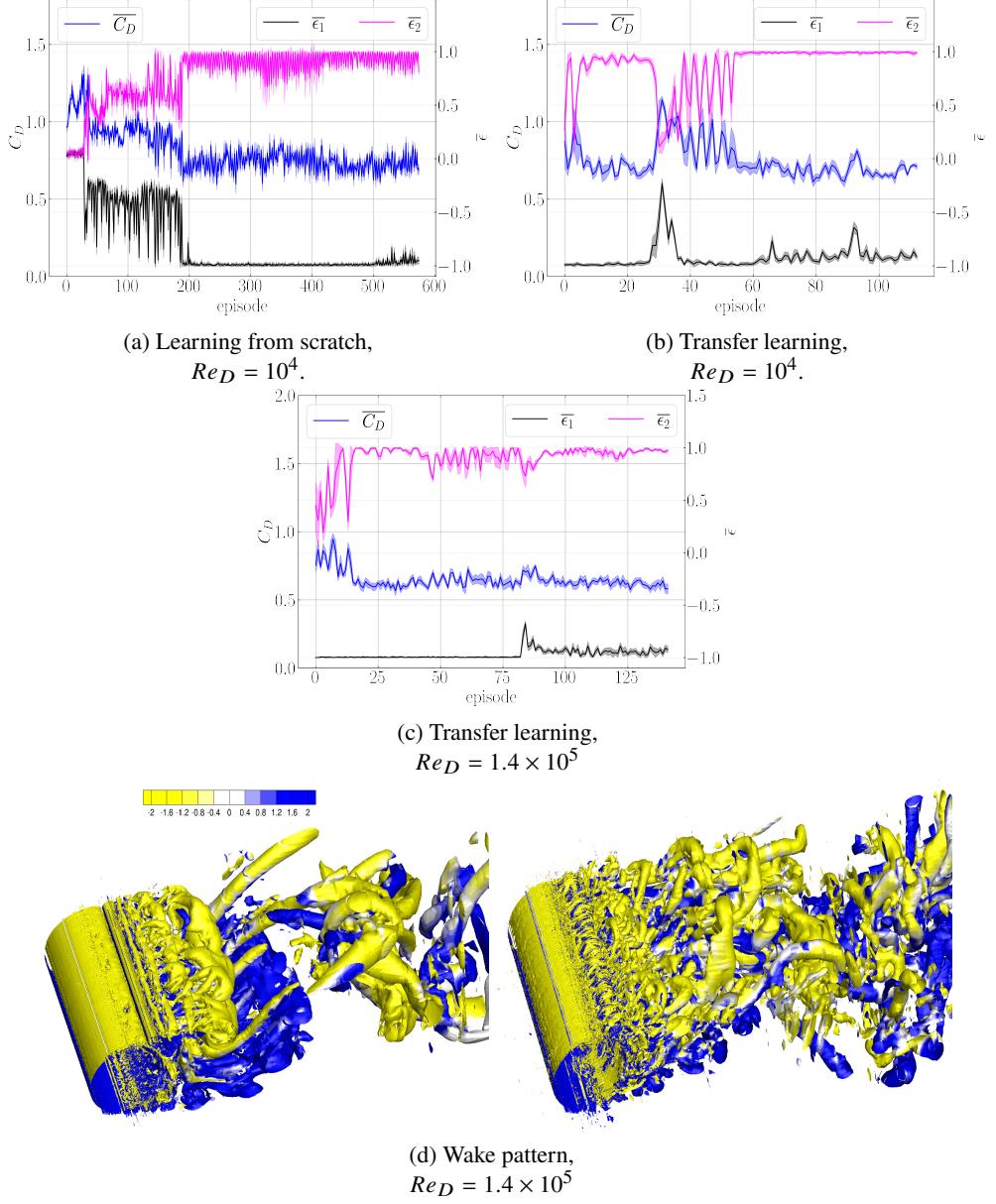


Figure 3: **Task I:** Comparison between DRL from scratch and transfer learning at different  $Re$ . Figure (a) - learning from scratch at  $Re_D = 10^4$ ; figure (b) - transfer learning  $Re_D = 10^4$ ; figure (c) - transfer learning at  $Re_D = 1.4 \times 10^5$ ; figure (d) - vortex shedding pattern, the control cylinders are stationary (upper figure), and the pattern at 125<sup>th</sup> episode (lower figure). Note that in Figures (a),(b) and (c), the pink and black lines are the time traces of actions ( $(\bar{\epsilon}/\epsilon^{max}, \text{ where } \epsilon_{max} = 3.66)$  on control cylinder 1 and 2. Blue line is the time trace of  $C_D$ . The agent of the transfer learning was initialized from the saved agent in the previous case at  $Re_D = 500$ , as shown in figure 2.

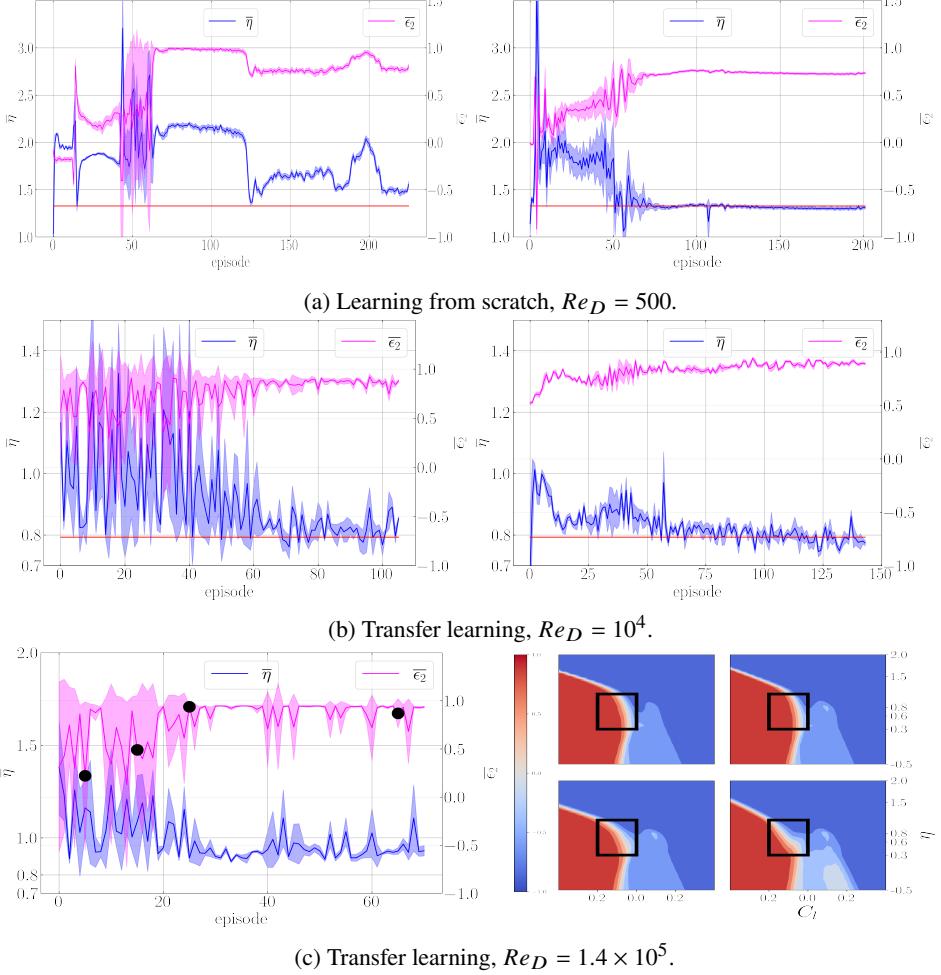


Figure 4: **Task 2:** Transfer learning from 2D low  $Re_D$  to 3D high  $Re_D$ . Figure (a) - learning from scratch,  $Re_D = 500$ : left one - single-client; right one - multi-client. Figure (b) - transfer learning,  $Re_D = 10^4$ : the left and right ones are initialized by the corresponding DRL agents shown in the left and right part of figure (a), respectively. Figure (c) - transfer learning,  $Re_D = 1.4 \times 10^5$ , initialized from the agent corresponding to right part of figure (b). Right panel of figure (c) - visualization of the policy at 5<sup>th</sup>, 15<sup>th</sup>, 25<sup>th</sup> and 65<sup>th</sup> episode. Note in all the simulations here  $\epsilon_{max} = 3.66$ .

the left part of figure 4(c). To explain the variation of the decision given by the agent, the policy at 5<sup>th</sup>, 15<sup>th</sup>, 25<sup>th</sup> and 65<sup>th</sup> episodes are visualized in the right part of figure 4 (c). Specifically, the black boxes  $[-0.25, 0.25] \times [0.3, 0.8]$ , which correspond to the concentrated intervals of  $C_L$  and  $C_D$ , respectively, are highlighted. We observe that the policy in the highlighted regions has not reached the best strategy yet, although the TL has been trained for 70 episodes. Next we consider the hardest problem of this paper, **Task 3**, with the same objective as that of **Task 1**, but  $\epsilon_1$  and  $\epsilon_2$  are independent. Again, we start this task from scratch in 2D simulation at  $Re_D = 500$ , as shown in figure 5(a). Note that here 16 2D simulations were used to provide training data. As shown in figure 5(a), the RL agent was able to give correct rotating directions for both controlling cylinders after 50 episodes. Between 50<sup>th</sup> and 110<sup>th</sup> episodes,  $\epsilon_2$  gradually approaches to -1, while  $\epsilon_1$  oscillates around 0.7, the combination of which gives rise to  $\eta \approx 1.5$ . After 110<sup>th</sup> episode, the

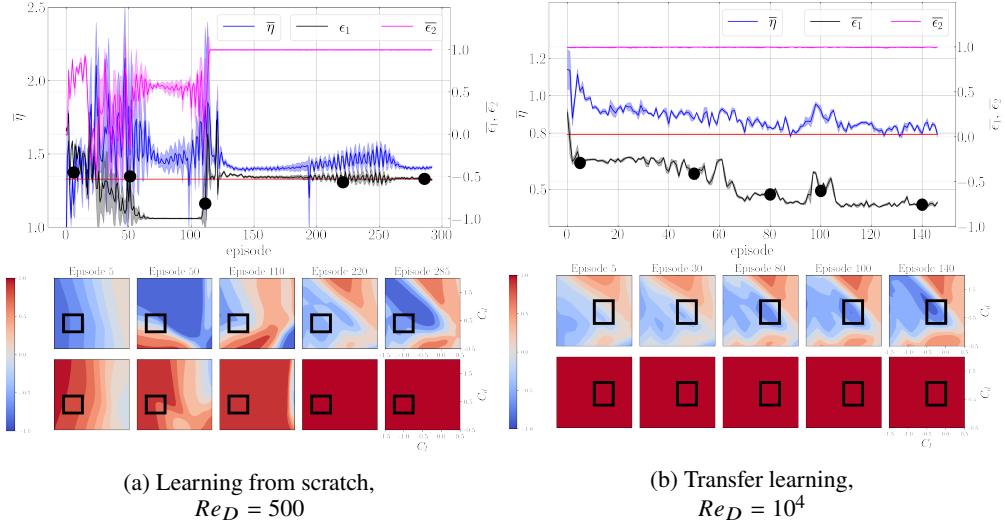


Figure 5: **Task 3:**  $\epsilon_1$  and  $\epsilon_2$  are independent. Figure (a), DRL from scratch using 16 2D simulations at  $Re_D = 500$ ; figure (b), transfer learning at  $Re_D = 10^4$ . Lower panel of the figure, where top row refers to  $\epsilon_1$  and the bottom row refers to  $\epsilon_2$ , shows the policy at the episodes indicated by the black circles in the corresponding figure of upper panel, respectively.

DRL suddenly changes its actions, in a short period of exploration:  $\epsilon_1$  changes to 1,  $\epsilon_2$  is -0.5, and the value of  $\eta$  goes down to 1.37. Meanwhile, the DRL does not stop exploration on  $\epsilon_1$ , as shown by the variation in the black line, starting around 200<sup>th</sup> episode. Around 280<sup>th</sup> episode, the DRL is able to reach a near optimal decision, as demonstrated by the fact that  $\eta$  is close to 1.325, which is the optimal value obtained from **Task 2**.

The transfer learning results of **Task 3** at  $Re_D = 10^4$  are plotted in figure 5(b). We observe that TL is able to reach the correct decision on rotating directions in less than 10 episodes. As more training data is obtained,  $\epsilon_2$  keeps the maximum value 1, and  $\epsilon_1$  is gradually increased to around  $\epsilon_1 = -0.8$ , which gives rise to  $\eta \approx 0.84$  that is slightly greater than the optimal value  $\eta = 0.78$ , obtained in **Task 2**. The policy contour at 5<sup>th</sup>, 50<sup>th</sup>, 110<sup>th</sup>, 220<sup>th</sup> and 285<sup>th</sup> episode for  $Re_D = 500$  and 5<sup>th</sup>, 30<sup>th</sup>, 80<sup>th</sup>, 100<sup>th</sup> and 140<sup>th</sup> episode for  $Re_D = 10^4$  are plotted on the bottom part of figure 5(a) and (b), respectively. During the learning process, at  $Re_D = 500$ ,  $C_D$  and  $C_L$  are mostly in the range [0.1, 0.6] and [-1.25, -0.75], and at  $Re_D = 10^4$ , these two coefficients are mostly in the range [0.3, 0.8] and [-0.5, 0.0], respectively. The corresponding  $C_D$ ,  $C_L$  ranges are highlighted by black boxes in the two figures. We observe that at both  $Re_D$ , the policies for  $\epsilon_1$  and  $\epsilon_2$  both gradually approach the value that results in optimal  $\eta$ , which clearly shows the exploration and exploitation stages at work.

## 4. Summary

We have implemented a deep reinforcement learning (DRL) in the numerical simulation of bluff body flow active control at high Reynolds number ( $Re_D = 1.4 \times 10^5$ ). We demonstrated that by training the DRL using the simulation data at low  $Re_D$ , and then apply transfer learning at high  $Re_D$ , the overall learning process can be accelerated substantially. In addition, the study shows that the transfer learning can result in more stable decision, which is potentially beneficial to the flow control. Moreover, we proposed a multi-clients-single-server DRL paradigm that is able to generate training data much faster to quickly discover an optimal policy. While here we

269 focus on a specific external flow, we believe that similar conclusions are valid for wall-bounded  
 270 flows and different control strategies.

## 271 Appendix A. Simulation results of control cylinders rotating at constant speed

272 In this section, the simulation results for the cases that both control cylinders are rotating at  
 273 constant speed  $|\Omega| = \epsilon_{\max} = 3.66$ . Figure 6 (a) and figure 6(b) show  $C_D$  and  $C_L$  on the three  
 274 cylinders at  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ , respectively. We observe that with rotating control  
 275 cylinders, the  $C_D$  on the main cylinder at both  $Re_D$  is reduced significantly. With  $\epsilon_{\max} = 3.66$ ,  
 276 the control cylinders are not able to cancel the vortex shedding on the main cylinder, thus the  
 277  $C_L$  on the main cylinder at both  $Re_D$  exhibits the frequency of vortex shedding. In particular, at  
 278  $Re_D = 1.4 \times 10^5$ , the magnitude of  $C_L$  on the control cylinder 1 and 2 shows notable discrepancy  
 279 with each other, which leads to symmetry breaking on the average at this  $Re_D$ . In addition, figure  
 280 7 plots the  $C_f$  on the control cylinders at both  $Re_D$ . As shown in the figure, at both  $Re_D$ , on  
 281 average, the magnitude of  $C_f$  on the control cylinders are different.

282 In order to validate the optimal control strategy given by DRL in **Task 2** and **Task 3**, additional  
 283 simulations with  $\epsilon_2 = 0$ ,  $\epsilon_2 = 0.6$ ,  $\epsilon_2 = 0.75$ ,  $\epsilon_2 = 0.9$  and  $\epsilon_2 = 1.0$  for  $Re_D = 500$ , and  $\epsilon_2 = 0$ ,  
 284  $\epsilon_2 = 0.85$ ,  $\epsilon_2 = 0.9$  and  $\epsilon_2 = 1.0$  for  $Re_D = 10^4$  are performed. Figure 8 plots the power gain  
 285 coefficient  $\eta$  as a function of  $\epsilon_2$ . Note that the value of  $\eta$  at  $Re = 10^4$  has been enlarged by  
 286 three times in order to show the variation more clearly. We observe that the minimum of  $\eta$  for  
 287  $Re_D = 500$  is around  $\epsilon_2 = 0.75$ , and it is  $\epsilon_2 = 0.9$  for  $Re_D = 10^4$ .

288

## 289 Appendix B. Validation of simulation of cylinder flow at high $Re_D$

290 In order to validate the numerical method (spectral element plus entropy-viscosity), large-eddy  
 291 simulations of flow past a cylinder at  $Re_D = 1.4 \times 10^5$  have been performed. The computational  
 292 domain has a size of  $[-12D, 16D] \times [-10D, 10D]$  in streamwise ( $x$ ) and crossflow ( $y$ ) directions,  
 293 respectively. The domain is partitioned into 2 044 quadrilateral elements. The size of elements  
 294 around the cylinder in the radial direction is  $0.0016D$  in order to resolve the boundary layer. The  
 295 domain size in the spanwise ( $z$ ) direction is  $3D$ . Uniform inflow velocity is prescribed at the inflow  
 296 boundary, homogeneous Neumann boundary condition for velocity and zero pressure is imposed  
 297 at the outflow boundary, and wall boundary condition is imposed at cylinder surface and periodic  
 298 boundary condition is assumed at all other boundaries. In particular, two simulations on different  
 299 resolution are performed: LES<sup>1</sup>, 3<sup>rd</sup> order spectral-element, 120 Fourier planes,  $\Delta t = 1.5 \times 10^{-4}$ ;  
 300 LES<sup>2</sup>, 4<sup>th</sup> order spectral-element and 160 Fourier planes,  $\Delta t = 1.0 \times 10^{-4}$ .

301 Table 1 presents the statistical values of  $\overline{C_D}$ ,  $\overline{C_L}$ ,  $St$ ,  $L_r$  and  $\phi_s$  from the simulation at  $Re_D =$   
 302  $1.4 \times 10^5$ . We observe that the simulation results agree with the values obtained from literature  
 303 very well, and the results are not sensitive to the mesh resolution. A further validation with the  
 304 experimental measurement is shown in figure 9, which plots the mean local  $\overline{C_p}$  and  $\overline{C_f}$ . Again,  
 305 the simulation results agree with experimental measurement very well. Figure 10 exhibits the  
 306 comparison between present LES and the experimental measurement by Cantwell & Coles (1983).  
 307 It could be observed that the LES results of the mean stremwise velocity  $\frac{\bar{u}}{U_\infty}$  along horizontal line  
 308  $y = 0$  and vertical line  $x/D = 1$  are in good agreement with the experiments. Figure 11 shows the  
 309 streamlines and contours of the mean velocity and Reynolds stress in the near wake. The results  
 310 shown in the figures agree with Fig.5 and Fig. 6 of Braza *et al.* (2006) well.

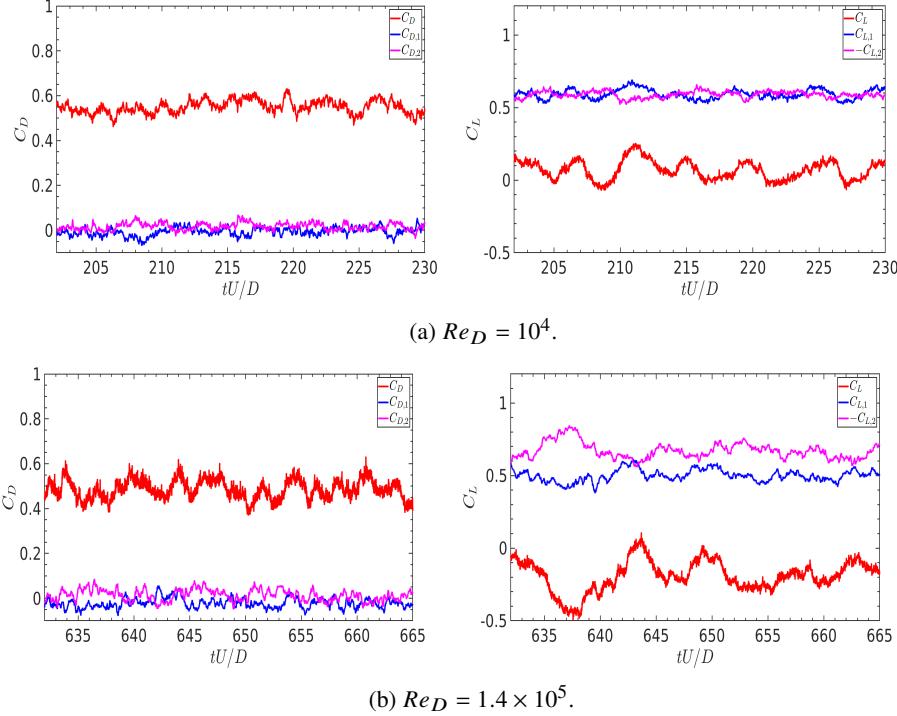


Figure 6: Time series of  $C_D$  and  $C_L$  on the main and control cylinders: figure(a),  $Re_D = 10^4$ ; figure(b),  $Re_D = 1.4 \times 10^5$ . Note that the control cylinders are rotating at constant speed at  $\epsilon_1 = -1$ ,  $\epsilon_2 = 1$ ,  $\epsilon_{\max}=3.66$ . The red lines are the hydrodynamic force coefficients on the main cylinder, while the blue and pink ones are on the control cylinder 1 and 2, respectively.

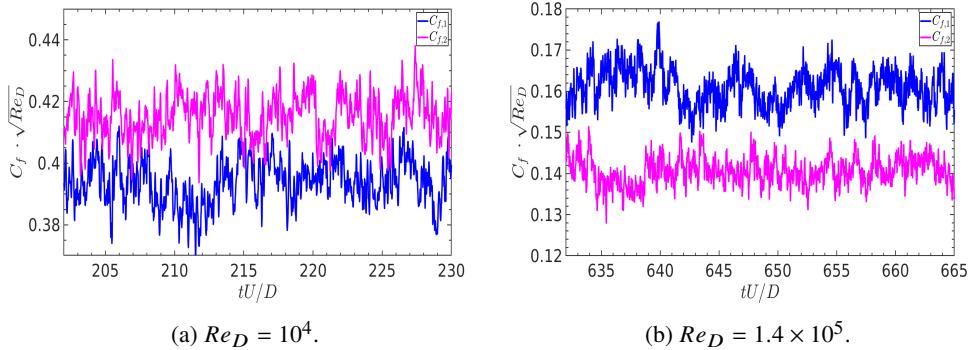


Figure 7: Time series of frictional coefficient ( $C_f$ ) on the control cylinders at  $Re_D = 10^4$  and  $Re_D = 1.4 \times 10^5$ ,  $\epsilon_1 = 1$ ,  $\epsilon_2 = 1$ ,  $\epsilon_{\max}=3.66$ .

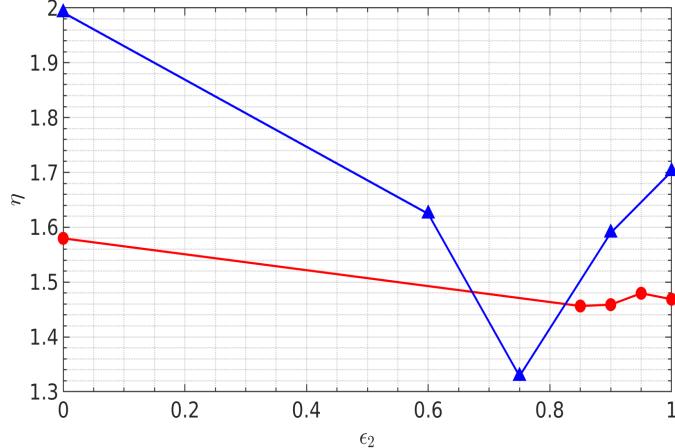


Figure 8: Power gain coefficient ( $\eta$ ) varies with  $\epsilon$ : blue one,  $Re_D = 500$ ; red one,  $Re_D = 10^4$ . Note that at  $Re_D = 10^4$ , the value of  $\eta$  has been enlarged by three times.  $\epsilon_{\max} = 3.66$ .

Table 1: Sensitivity study of the simulation result to the mesh resolution: flow past a single circular cylinder at  $Re_D = 1.4 \times 10^5$ .  $\overline{C_D}$  is the mean drag coefficient,  $\overline{C_L}$  is the r.m.s. value of the lift coefficient, St is the Strouhal number,  $L_r$  is the length of the recirculation bubble and  $\phi_s$  is the separation angle. The (Breuer 2000) LES is case D2. LES<sup>1</sup> and LES<sup>2</sup> corresponds to the mesh resolution 1 and 2, respectively.

Study	$\overline{C_D}$	$\overline{C_L}$	St	$L_r$	$\phi_s$
Present LES <sup>1</sup>	0.95	0.63	0.22	0.69	93
Present LES <sup>2</sup>	1.13	0.49	0.21	0.74	94
(Breuer 2000) LES	1.29	-	0.203	0.46	92.59
(Braza <i>et al.</i> 2006) Experiment	-	-	0.21	0.78	

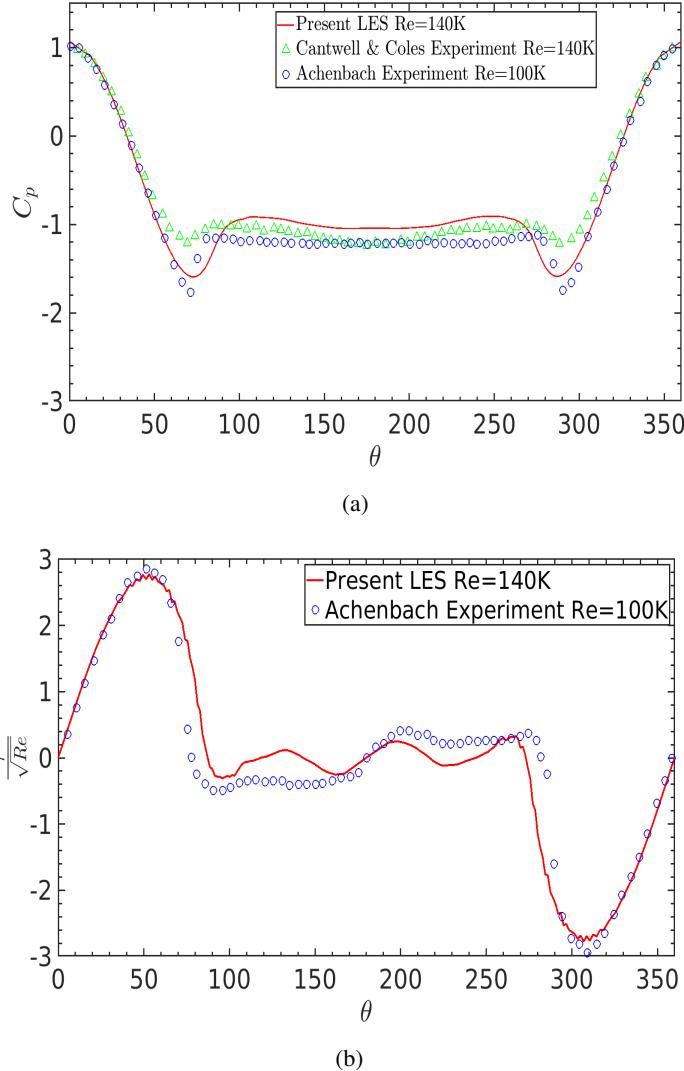


Figure 9: Local pressure and skin friction coefficient at  $Re_D = 1.4 \times 10^5$ , comparison with the literature. Note that “Present LES” is from the simulation using mesh LES<sup>2</sup>, “Cantwell & Coles Experiment” refers to the experiment by Cantwell & Coles (1983), “Achenbach Experiment” refers to the experiment by Achenbach (1968).

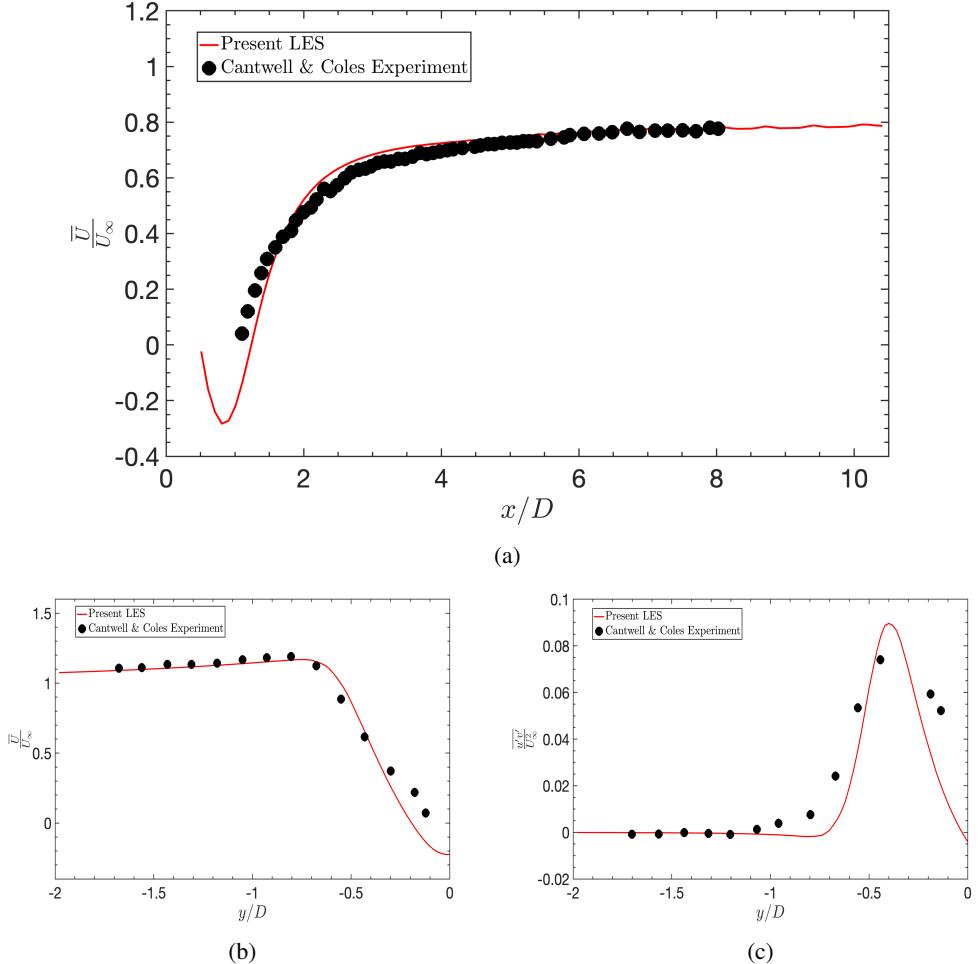


Figure 10: Validation of the LES: (a) Mean center-line velocity  $\bar{U}/U_\infty$ ; (b) mean velocity  $\bar{U}/U_\infty$  at  $\frac{x}{D} = 1$ ; (c) mean turbulent shear stress  $\bar{u}'v'/U_\infty^2$  at  $\frac{x}{D} = 1$ . Note that “Present LES” is from the simulation using mesh LES<sup>2</sup>, “Cantwell & Coles Experiment” refers to the experiment by Cantwell & Coles (1983).

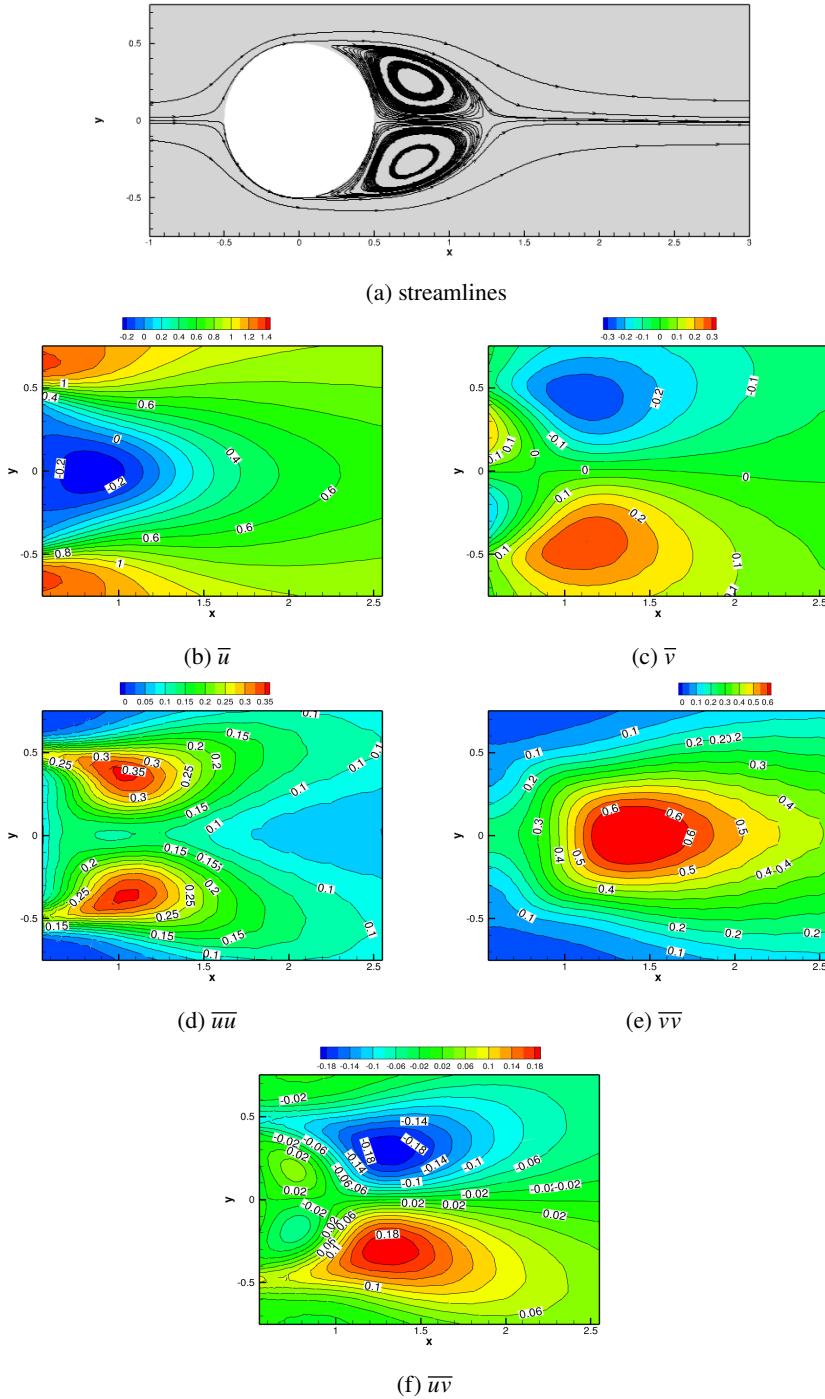


Figure 11: Validation of the LES: mean velocity and Reynolds stress field. Note that the reference results shown in the figures agree with Fig.5 and Fig. 6 of Braza *et al.* (2006).

The present simulation results are from the simulation using mesh resolution LES<sup>2</sup>

## REFERENCES

- ACHENBACH, E. 1968 Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to  $re = 5 \times 10^6$ . *Journal of Fluid Mechanics* **34** (4), 625–639.
- BAE, J. & KOUMOUTSAKOS, P. 2022 Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications* **13**, 1443.
- BRAZA, M., PERRIN, R. & HOARAU, Y. 2006 Turbulence properties in the cylinder wake at high reynolds numbers. *Journal of Fluids and Structures* **22** (6), 757–771, "Bluff Body Wakes and Vortex-Induced Vibrations (BBVIV-4).
- BREUER, MICHAEL 2000 A challenging test case for large eddy simulation: high reynolds number circular cylinder flow. *International Journal of Heat and Fluid Flow* **21** (5), 648–654, turbulence and Shear Flow Phenomena 1.
- BUCCI, M., SEMERARO, O., ALLAUZEN, A., WISNIEWSKI, G., CORDIER, L. & MATHELIN, L. 2019 Control of chaotic systems by deep reinforcement learning. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **475** (2231), 20190351.
- CANTWELL, BRIAN & COLES, DONALD 1983 An experimental study of entrainment and transport in the turbulent near wake of a circular cylinder. *Journal of Fluid Mechanics* **136**, 321–374.
- CHENG, W., PULLIN, D., SAMTANEY, R., ZHANG, W. & GAO, W. 2017 Large-eddy simulation of flow over a cylinder with  $reD$  from  $3.9 \times 10^3$  to  $8.5 \times 10^5$ : a skin-friction perspective. *Journal of Fluid Mechanics* **820**, 121–158.
- COLABRESE, S., GUSTAVSSON, K., CELANI, A. & BIFERALE, L. 2017 Flow navigation by smart microswimmers via reinforcement learning. *Physical Review Letters* **118** (15), 158004.
- DONG, S., KARNIADAKIS, G.E., EKMEKCI, A. & ROCKWELL, D. 2006 A combined direct numerical simulation–particle image velocimetry study of the turbulent near wake. *Journal of Fluid Mechanics* **569**, 185–207.
- DU, QIANG, XIE, YAGUANG, WANG, ZHICHENG, JIANG, XIAOMO & XIE, LEI 2023 An entropy viscosity method for large eddy simulation of turbulent thermal flow in a rotor–stator cavity. *Physics of Fluids* **35** (3), 035126.
- FAN, D., YANG, L., WANG, Z., TRIANTAFYLLOU, M. & KARNIADAKI, S G.E. 2020 Reinforcement learning for bluff body active flow control in experiments and simulations. *Proceedings of the National Academy of Sciences* **117** (42), 26091–26098.
- FUJIMOTO, S., HOOF, V. & MEGER, D. 2018 Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1582–1591.
- GAZZOLA, M., HEJAZIALHOSSEINI, B. & KOUMOUTSAKOS, P. 2014 Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. *SIAM Journal on Scientific Computing* **36** (3), B622–B639.
- GUERMOND, J-L., PASQUETTI, R. & POPOV, B. 2011a Entropy viscosity method for nonlinear conservation law. *Journal of Computational Physics* **230** (11), 4248–4267.
- GUERMOND, J-L., PASQUETTI, R. & POPOV, B. 2011b From suitable weak solutions to entropy viscosity. *Journal of Scientific Computing* **49** (1), 35–50.
- KARNIADAKIS, G.E. & SHERWIN, S. 2005 *Spectral/hp Element Methods for Computational Fluid Dynamics, 2nd edition*. Oxford,UK: Oxford University Press.
- KIRKPATRICK, J., PASCANU, R., RABINOWITZ, N., VENESS, J., DESJARDINS, G., RUSU, A., MILAN, K., QUAN, J., RAMALHO, T., GRABSKA-BARWINSKA, A., HASSABIS, D., CLOPATH, C., KUMARAN, D. & HADSELL, R. 2017 Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* **114** (13), 3521–3526.
- MA, P., TIAN, Y., PAN, Z., REN, B. & MANOCHA, D. 2018 Fluid directed rigid body control using deep reinforcement learning. *ACM Transactions on Graphics (TOG)* **37** (4), 96.
- MORKOVIN, M. 1964 Flow around circular cylinders-a kaleidoscope of challenging fluid phenomena. In *Proceedings of ASME Symposium on Fully Separated Flows*, pp. 102–118.
- NOVATI, G., MAHADEVAN, L. & KOUMOUTSAKOS, P. 2019 Controlled gliding and perching through deep-reinforcement-learning. *Physical Review Fluids* **4** (9), 093902.
- RABAULT, J., KUCHTA, M., JENSEN, A., RÉGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics* **865**, 281–302.
- REDDY, G., CELANI, A., SEJNOWSKI, T. & VERGASSOLA, M. 2016 Learning to soar in turbulent environments. *Proceedings of the National Academy of Sciences* **113** (33), E4877–E4884.

- 366 REN, F., RABAULT, J. & TANG, H. 2021 Applying deep reinforcement learning to active flow control in weakly  
367 turbulent conditions. *Physics of Fluids* **33** (3), 037121.
- 368 VERMA, S., NOVATI, G. & KOUMOUTSAKOS, P. 2018 Efficient collective swimming by harnessing vortices  
369 through deep reinforcement learning. *Proceedings of the National Academy of Sciences* **115** (23),  
370 5849–5854.
- 371 VIQUERAT, J., RABAULT, J., KUHNLE, A., GHRAIEB, H., LARCHER, A. & HACHEM, E. 2021 Direct shape  
372 optimization through deep reinforcement learning. *Journal of Computational Physics* **428**, 110080.
- 373 WANG, Z., TRIANTAFYLLOU, M. S., CONSTANTINIDES, Y. & KARNIADAKIS, G.E. 2018 A spectral-  
374 element/fourier smoothed profile method for large-eddy simulations of complex viv problems.  
375 *Computers and Fluids* **172**, 84–96.
- 376 WANG, Z., TRIANTAFYLLOU, M. S., CONSTANTINIDES, Y. & KARNIADAKIS, G.E. 2019 An entropy-viscosity  
377 large eddy simulation study of turbulent flow in a flexible pipe. *Journal of Fluid Mechanics* **859**,  
378 691–730.