

RESEARCH ARTICLE | APRIL 19 2023

How to control hydrodynamic force on fluidic pinball via deep reinforcement learning

Feng Haodong (冯浩东); Wang Yue (汪跃); Xiang Hui (向辉); ... et. al



Physics of Fluids 35, 045137 (2023)

<https://doi.org/10.1063/5.0142949>

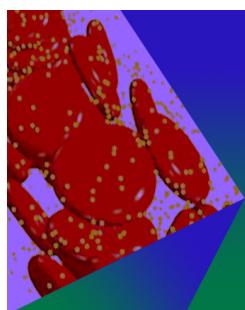


View
Online



Export
Citation

CrossMark



Physics of Fluids

Special Topic: Flow and Forensics

Submit Today!

 AIP
Publishing

How to control hydrodynamic force on fluidic pinball via deep reinforcement learning

Cite as: Phys. Fluids 35, 045137 (2023); doi: 10.1063/5.0142949

Submitted: 18 January 2023 · Accepted: 27 March 2023 ·

Published Online: 19 April 2023



View Online



Export Citation



CrossMark

Haodong Feng (冯浩东),^{1,2,3} Yue Wang (汪跃),³ Hui Xiang (向辉),⁴ Zhiyang Jin (金志扬),^{5,6,a} and Dixia Fan (范迪夏)^{2,7,8,a}

AFFILIATIONS

¹Zhejiang University, Hangzhou, Zhejiang 310027, China

²School of Engineering, Westlake University, Hangzhou, Zhejiang 310030, China

³Microsoft Research AI4Science, Beijing 100080, China

⁴Scien42.tech, Beijing 100101, China

⁵Mechanical and Electrical Engineering College, Hainan University, Haikou, Hainan 570228, China

⁶Hainan Policy and Industrial Research Institute of Low-Carbon Economy, Haikou, Hainan 570228, China

⁷Research Center for Industries of the Future, Westlake University, Hangzhou, Zhejiang 310030, China

⁸Key Laboratory of Coastal Environment and Resources of Zhejiang Province, School of Engineering, Westlake University, Hangzhou, Zhejiang 310030, China

^aAuthors to whom correspondence should be addressed: jinzhiyang94@163.com and fandixia@westlake.edu.cn

ABSTRACT

Deep reinforcement learning (DRL) for *fluidic pinball*, three individually rotating cylinders in the uniform flow arranged in an equilaterally triangular configuration, can learn the efficient flow control strategies due to the validity of self-learning and data-driven state estimation for complex fluid dynamic problems. In this work, we present a DRL-based real-time feedback strategy to control the hydrodynamic force on *fluidic pinball*, i.e., force extremum and tracking, from cylinders' rotation. By adequately designing reward functions and encoding historical observations, and after automatic learning of thousands of iterations, the DRL-based control was shown to make reasonable and valid control decisions in nonparametric control parameter space, which is comparable to and even better than the optimal policy found through lengthy brute-force searching. Subsequently, one of these results was analyzed by a machine learning model that enabled us to shed light on the basis of decision-making and physical mechanisms of the force tracking process. The finding from this work can control hydrodynamic force on the operation of *fluidic pinball* system and potentially pave the way for exploring efficient active flow control strategies in other complex fluid dynamic problems.

Published under a nonexclusive license by AIP Publishing. <https://doi.org/10.1063/5.0142949>

I. INTRODUCTION

Control of fluid flows has been a long-standing challenge for the engineering community.^{1–4} With the novel applications appearing in environment and energy engineering, its importance has increased. Among these challenges, active flow control (AFC) has been an attractive topic in fluid dynamics, where actuators intentionally change the fluid system by applying optimized inputs.^{5–14} Compared with the passive flow control (PFC) strategies usually involving structural optimizations, AFC is adaptive and can achieve more effective control in a broader working range. If the information from the system output is fed back to guide the control, AFC can be divided into open-loop control or closed-loop control. Unlike the open-loop control, the closed-loop control can utilize the feedback signals with the current features,

to adjust the actuator in real-time, thus allowing automatic operation in a broader flow range and higher accuracy.^{15,16}

In recent years, AFC research has started to pay close attention to the *fluidic pinball* problem^{17,18} (shown in Fig. 1) with a simple configuration but rich physics, making it a benchmark testbed for AFC algorithms.^{19–21} With different rotation velocities, this simple arrangement demonstrates a versatile set of flow patterns, categorized into six wake stabilization strategies: phasor control, boat-tailing, base-bleed, and Magnus effect, especially high-frequency forcing and low-frequency forcing.²² The domestication of *fluidic pinball* must consider the richness of these driving mechanisms. In addition, it allows the testing of steady and unsteady control strategies in a larger control parameter space. Therefore, the *fluidic pinball* is a heuristic experiment

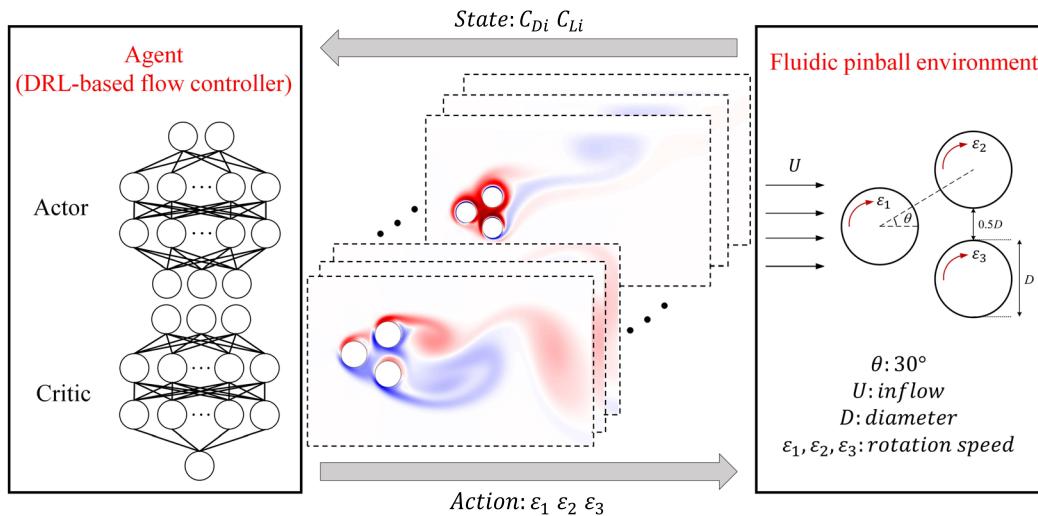


FIG. 1. Framework of the DRL-based flow control for the *fluidic pinball* environments. The neural network in left is the DRL agent, and the system in right is the environment. The vorticity figures located at center intuitively demonstrate that flow can be controlled in real-time, to achieve the expected objective. In each episode, the agent interacts with the environment via the XML-RPC protocol at a fixed frequency. Each interaction consists of the state C_{Di} and C_{Li} feedback from the environment that is determined by the objective of tasks, and action ε_i , $i=1, 2, 3$. After collecting the data from the environment, the agent updates the policy while waiting for the reset of the environment. Along with control, a significant change in vortical wake occurs, shown in the snapshots boxed in the dashed line.

suitable for developing advanced control methods to explore large state and action spaces for various objectives.

Various researchers have applied different AFC algorithms on the *fluidic pinball*, illustrating the effectiveness of the problem setup and the corresponding control strategies. Peitz *et al.*²³ developed the model predictive control (MPC) approach via the Koopman operator to control the lift of all three cylinders by rotating the top and bottom cylinders. Raibaudo *et al.*^{24,25} applied the linear genetic algorithm (GA) to reduce the drag and modify the wake of *fluidic pinball*. Blanchard *et al.*²⁶ presented the performance of Bayesian optimization (BO) to design an open-loop controller for drag reduction. Another open-loop controller was developed by Ghraieb *et al.*²⁷ applying the single-step deep reinforcement learning (DRL) as an optimization method. Maceda *et al.*²⁸ proposed a gradient-enriched machine learning control (gMLC) to stabilize the unstable symmetric wake behind the *fluidic pinball*, which can explore new minima while keeping the convergence efficiency combining the merits of exploitation and exploration. Furthermore, the explorative gradient method (EGM) was proposed by Li *et al.*²⁹ to avoid suboptimal local minima when minimizing the drag of *fluidic pinball*. These control strategies can be employed, not only to optimize large parameter space, but also to reveal unknown solutions or parameter relationships for AFC in *fluidic pinball*.

Apart from work mentioned above, the rapid development of deep learning (DL) methods has paved the way for high-dimensional and complex fluid problems.^{30–32} Lee and You³³ applied DL networks to predict flow field development for the canonical unsteady vortex shedding over a circular cylinder. Kharazmi *et al.*³⁴ developed robust physics-informed neural networks (PINNs) to forecast the motion of a flexible cylinder undergoing vortex-induced vibrations (VIVs). Furthermore, the development of DL also got a foundation and powerful impetus for DRL,³⁵ which simplifies continuous systems into Markov decision processes (MDP).³⁶ The DRL algorithm has been

applied in various domains, including robotics, computer games, and autonomous driving.^{37–40} DRL algorithm explores the control strategy through the interaction with the environment shown in Fig. 1, which is quite convenient and suitable for complex fluid dynamics. In recent years, due to the enormous potential of intensive DRL algorithms, which can deal with strongly nonlinear multi-scale dynamics problems,⁴¹ many pieces of research have been published on using DRL algorithms to solve AFC problems.⁴²

Many DRL applications for AFC can be found in several reviews by Brunton *et al.*,³¹ Rabault *et al.*,⁴³ Viquerat *et al.*,⁴⁴ and Chen *et al.*⁴² In particular, a closed-loop control method for synthetic jets was presented by Rabault *et al.*⁴⁵ to reduce the drag of a cylinder at different Reynolds numbers. Xu *et al.*⁴⁶ used a feasible DRL [proximal policy optimization, (PPO)] algorithm to look for the wake stabilization mechanism of a 2D circular cylinder by tuning the rotations of two small cylinders behind it. A similar setup in the experiment was conducted by Fan *et al.*⁴⁷ Zheng *et al.*⁴⁸ adopted both active learning and DRL based on two jet actuators to eliminate vortex shedding and VIVs at low Reynolds numbers. Wang *et al.*⁴⁹ proposed a python platform for coupling DRL and OpenFOAM in fluid mechanics, then demonstrated its reliability and efficiency using two wake stabilization benchmark problems. Han *et al.*, Amico *et al.*, Mei *et al.*, and Wang *et al.*^{50–53} used DRL to manipulate the wake and hence reduce the drag of bluff bodies. Using expert demonstration from a wake oscillator, Zheng *et al.*⁵⁴ demonstrated VIV reduction using DRL. Ren *et al.*,⁵⁵ and Ren and Tang⁵⁶ developed the DRL-based AFC methods for bluff bodies to hide their hydrodynamic traces using a group of windward-suction-leeward-blowing (WSLB) actuators. Xie *et al.*⁵⁷ employed the DRL method to control a heaving plate breakwater to find the optimal wave dissipation policy. Qin *et al.* introduced the dynamic mode decomposition (DMD) in reward function construction for DRL, so that the agent can learn the AFC policy through the

more global information of the field. Li and Zhang⁵⁸ studied the application of a PPO control algorithm to the flow past a cylinder between two walls in order to suppress vortex shedding successfully. Pino *et al.*⁵⁹ systematically compared and analyzed the algorithms of machine learning and DRL on three typical AFC problems. Paris *et al.*⁶⁰ and Xu and Zhang⁵ also discussed the optimal sensor placement along with the active flow control tasks with DRL. There are also many effective and successful cases of active flow control via DRL, considering various Reynolds numbers and tasks.^{61,62} The above successful applications of DRL in AFC further make us believe DRL algorithms can solve some control problems in fluid dynamics efficiently with the advantages of self-learning, data-driven, dimensional mapping, and generalization capability. They have shown great potential achieving force control for more complex and more challenging fluidic problems.

At the time of submission, most DRL applications were bound to search and solve for lower-dimension and less-mode flow control problems. In this work, we applied the DRL-based control method on a more complex fluid benchmark problem (*fluidic pinball*), with a bigger nonparametric action space, demonstrating its feasibility and future challenges. We present applying DRL-based real-time feedback control to minimize the drag, maximize the lift, and even remain the drag to the expected references in real-time for the *fluidic pinball*. Although some articles have published research on drag reduction, the force tracking as expected freely for *fluidic pinball* still has an exciting gap waiting to be explored, which is focused by seldom previous work. Moreover, this work is also the first attempt to apply the DRL algorithm for the real-time feedback force control on *fluidic pinball* setup for extremum searching and force tracking. The results illustrate that the DRL algorithm is suitable and feasible for challenging multi-input multi-output (MIMO) control problems like the present work.

The paper is organized as follows. In Sec. II, the problem setup, the computational fluid dynamics (CFD) solver methodologies, and the open-loop and DRL-based real-time control approaches are described. An introduction to the DRL algorithm is also presented. In Sec. III, the results of open-loop control with constant rotation speed from brute-force searching are analyzed, focusing first on the introduction of baselines that will compare with the results of DRL-based control. Second, a detailed discussion of force distribution and vortical wake of *fluidic pinball* with overall discrete control rotations is reported. In Sec. IV, the results of drag reduction and drag tracking problems solved via DRL-based real-time control are presented, and the additional comments of drag tracking are discussed. The conclusion follows in Sec. V.

II. MATERIALS AND METHODS

A. Problem setup

In this work, we focus on the *fluidic pinball* problem of a 2D flow past three circular cylinders of diameter D , arranged in an equilaterally triangular configuration⁶³ at the Reynolds number $Re = UD/\nu = 100$, where U is the incoming velocity, and ν is the kinematic viscosity of the fluid. A sketch of the problem setup is shown in Fig. 1. Specifically, the incoming flow from the inlet boundary is uniform. All results will be nondimensionalized by U , D , and time T . Determined by the DRL agent in real-time, the three cylinders can rotate at a speed of $\varepsilon_i \in [-5, 5]$, where $\varepsilon_i = \frac{\omega D}{U}$, ω is the cylinder rotation speed, and $i = 1, 2, 3$ represents the front, top, and bottom cylinders. In addition, the lift

and drag coefficients of both the system and single cylinder are recorded as follows:

$$C_{Di/Li} = \frac{F_{Di/Li}}{0.5\rho U^2 DL}, \quad (1a)$$

$$\bar{C}_{Di/Li} = \frac{1}{T} \int C_{Di/Li}(t) dt, \quad (1b)$$

$$\tilde{C}_{Di/Li} = \sqrt{\frac{1}{T} \int (C_{Di/Li}(t) - \bar{C}_{Di/Li})^2 dt}, \quad (1c)$$

where with the subtext i , $C_{Di/Li}$ represents the drag or lift coefficient of cylinder. Specifically, $C_{D0/L0}$ denotes the system's average drag or lift coefficient, and $i = 1, 2, 3$ represents the front, top, and bottom cylinders. $\bar{C}_{Di/Li}$ is the standard deviation of C_{Di} or C_{Li} . $\bar{C}_{Di/Li}$ is the average value of whole C_{Di} or C_{Li} . $F_{Di/Li}$ is the drag or lift force of corresponding cylinder sharing the same indices with relevant $C_{Di/Li}$. ρ denotes the fluid density and L is the unit length of cylinders.

Before performing the DRL-based flow control, we conducted a large number of simulations for three cylinders rotating at different constant speeds to both serve as the benchmark and have a deep understanding of the problem. The rotation speed of each cylinder is selected from -5 to 5 with an incremental of 1 . Therefore, in total, 1331 combinations of different cylinder rotation speeds are simulated.

B. Numerical method

The numerical solver applied in this work is based on the boundary data immersion method (BDIM).⁶⁴ It solves the viscous time-dependent Navier-Stokes equations and simulates the entire domain by combining the moving body and the ambient fluid through a kernel function. The method has quadratic convergence and has been verified for many numerical simulations for a wide range of fluid problems.^{65,66}

The mesh configuration for this work is a rectangular Cartesian grid with a dense uniform grid near the body and in the near wake, and exponential grid stretching used in the far-field and the numerical domain of $16D \times 8D$ uses a uniform inflow, zero-gradient outflow, and free-slip boundary conditions on the up and low boundaries. Furthermore, no-slip boundary conditions are employed on the circular cylinder. Mesh density is expressed in terms of grid points per chord. A uniform grid of $\delta x = \delta y = D/16$ is used for the results in this work. A detailed description of the method validation is provided in Appendix A.

C. DRL algorithm for real-time feedback control

We employed the twin delayed deep deterministic policy gradient (TD3) algorithm⁶⁷ to perform the DRL-based flow control. The closed-loop interaction between the *fluidic pinball*, i.e., the fluid environment and the DRL agent, is shown in Fig. 1. The fluid environment outputs the C_{Di} and C_{Lb} i.e., the state, from the numerical solver. The *fluidic pinball* uses the rotation actuation, i.e., the action from the DRL, to alter the fluid environment. The performance of the DRL-based flow control is then evaluated using the reward r illustrated as Eqs. (2) and (3) in Sec. IV.

In this work, the TD3 algorithm is implemented using Python code. The Extensible Markup Language-Remote Procedure Call (XML-RPC) protocol is applied for data communication between the cross-language platforms.

Problem formulation. We consider infinite-horizon MDP characterized by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, p_0)$, where $\mathcal{S} \in \mathbb{R}^n$ and $\mathcal{A} \in \mathbb{R}^m$ are continuous state and action spaces, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}_+$ is the transition (dynamics) distribution, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ denotes the reward function, $\gamma \in [0, 1]$ is a discount factor, and p_0 is the initial state distribution. We aim to learn a mapping $\Pi_\theta : \mathcal{S} \mapsto \mathcal{A}$ with parameters θ such that discounted return $\mathbb{E}_{\Gamma \sim \Pi_\theta} [\sum_{t=1}^{\infty} \gamma^t r_t]$, $r_t = \mathcal{R}(\mathbf{s}_t, \mathbf{a}_t)$ is maximized along a trajectory $\Gamma = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots)$ following Π_θ by sampling an action $\mathbf{a}_t \sim \Pi_\theta(\cdot | \mathbf{s}_t)$ and reaching state $\mathbf{s}_{t+1} \sim \mathcal{T}(\cdot | \mathbf{s}_t, \mathbf{a}_t)$ at each decision step t .

TD3 algorithm aims to estimate an optimal state-action value function $Q^* : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ using a value function $Q_\omega(\mathbf{s}, \mathbf{a}) \approx Q^*(\mathbf{s}, \mathbf{a}) = \max_{\mathbf{a}'} \mathbb{E}[\mathcal{R}(\mathbf{s}, \mathbf{a}) + \gamma Q^*(\mathbf{s}', \mathbf{a}')] \forall \mathbf{s} \in \mathcal{S}$, where \mathbf{s}', \mathbf{a}' is the state and action at the following step, and ω parameterizes the function. While Q^* is generally unknown, it can be approximated by repeatedly fitting Q_ω using the update rule

$$\omega^{k+1} \leftarrow \arg \min_{\omega} \mathbb{E}_{(\mathbf{s}, \mathbf{a}, \mathbf{s}') \sim \mathcal{B}} \|Q_\omega(\mathbf{s}, \mathbf{a}) - y\|_2^2,$$

where the Q -target $y = \mathcal{R}(\mathbf{s}, \mathbf{a}) + \gamma \max_{\mathbf{a}'} Q_{\omega^-}(\mathbf{s}', \mathbf{a}')$, \mathcal{B} is a replay buffer that is iteratively grown as new data are collected, and ω^- is a slow-moving average of the online parameters ω updated with the rule $\omega_{k+1}^- \leftarrow (1 - \zeta)\omega_k^- + \zeta\omega_k$ using a constant coefficient $\zeta \in [0, 1)$. In DRL, generally, Π is typically a policy parameterized by a neural network that learns to approximate $\Pi_\theta(\cdot | \mathbf{s}) \approx \operatorname{argmax}_{\mathbf{a}} \mathbb{E}[Q_\omega(\mathbf{s}, \mathbf{a})] \forall \mathbf{s} \in \mathcal{S}$, i.e., the globally optimal policy.

DRL-based flow control with TD3 (Algorithm 1) is a deterministic strategy reinforcement learning algorithm suitable for continuous action space. It greatly improves the learning speed and performance of previous algorithms in numerous challenging tasks in the continuous control setting, which exceeds the performance of numerous state-of-the-art DRL algorithms.⁶⁷ Such merit of TD3 determines its vital role in this work. All the neural networks are feedforward neural networks with two hidden layers, each of width 256 in this work. The discount factor γ is set as 0.99, and the policy exploration Gaussian noise σ is set as 0.1 in the simulation. We use the Adam optimizer with learning rate 10^{-4} to update ω_i and θ . N_j is set to 1000, and the batch size is set to 512. The regularization noise $\tilde{\sigma}$ is set to 0.2, and c is set to 0.5. The target networks are updated every $d = 2$ iterations, and the soft updating rate τ is set to 0.005.

ALGORITHM 1: DRL-based flow control with TD3.

Initialize critic networks Q_{ω_1} , Q_{ω_2} and actor network π_θ with random parameters $\omega_1, \omega_2, \theta$
Initialize target networks $\omega' \leftarrow \omega_1, \omega'_2 \leftarrow \omega_2, \theta' \leftarrow \theta$
Initialize $done = \text{false}$, $k \in \mathbb{N}_+$
Initialize replay buffer \mathcal{D} , observation buffer $\mathcal{O} \in \mathbb{R}^k$

- 1: **for** $e = 1$ **in** N_e **do**
- 2: reset CFD environment;
- 3: $i \leftarrow 0$;
- 4: **while** $done = \text{false}$ **do**
- 5: collect the observation \mathbf{o}_i ;
- 6: store \mathbf{o}_i into \mathcal{O} , state $\mathbf{s}_i = \mathcal{O}[i : i + k]$;
- 7: randomly generate exploration noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$;
- 8: call an action with ϵ_i ;

```

9:    $a_i \leftarrow \operatorname{clip}(\pi_\theta(\mathbf{s}_i) + \epsilon_i, -1, 1)$ ;  

10:  implement  $a_i$  in the CFD environment;  

11:  get next state  $\mathbf{s}_{i+1}$ , calculate reward  $r_i$ ;  

12:   $i = i + 1, \mathbf{s}_i = \mathbf{s}_{i+1}$ ;  

13:  store  $(\mathbf{s}_i, a_i, r_i, \mathbf{s}_{i+1})$  into  $\mathcal{D}$ ;  

14:  if CFD stop then  

15:     $done = \text{true}$ ;  

16:  for  $j = 1$  in  $N_j$  do  

17:    sample  $N(s, a, r, s')$  from  $\mathcal{D}$ ;  

18:     $\epsilon' \sim \operatorname{clip}(\mathcal{N}(0, \tilde{\sigma}^2), -c, c)$ ;  

19:     $\tilde{a} \leftarrow \operatorname{clip}(\pi_{\theta'}(s') + \epsilon', -1, 1)$ ;  

20:     $y \leftarrow r + \gamma \min_{i=1,2} Q_{\omega'_i}(s', \tilde{a})$ ;  

21:     $L_{\omega_i} \leftarrow N^{-1} \sum (y - Q_{\omega_i}(s, a))^2, i = 1, 2$ ;  

22:    update  $\omega_i$  with the loss function  $L_{\omega_i}, i = 1, 2$ ;  

23:  if  $e \bmod t$  then  

24:     $L_\theta \leftarrow -N^{-1} \sum Q_{\omega_1}(s, \pi_\theta(s))$ ;  

25:    update  $\theta$  with the loss function  $L_\theta$ ;  

26:    update the target networks:  

27:     $\omega'_i \leftarrow \tau \omega_i + (1 - \tau) \omega'_i, i = 1, 2$ ;  

28:     $\theta' \leftarrow \tau \theta + (1 - \tau) \theta'$ .

```

Unlike traditional tasks of DRL, flow control is not a standard MDP, as a long-time flow field lasting many steps can impact the next state. Two techniques are presented to solve the problem of employing the DRL on such flow control tasks. First, unlike the general TD3 algorithm, which learns the policy online, the tuned TD3 algorithm in this work transfers the online training into the offline training mode. In each training, the agent can randomly sample data from the buffer filled with the trajectory of the whole flow control process, which means that the agent can access the relative overall flow field feature from start to end. Second, states are encoded by current C_{Di}^t and C_{Li}^t combining with them in many previous steps as $\mathbf{s}_t = (C_{Li}^{t-n}, C_{Di}^{t-n}, C_{Li}^{t-n+1}, C_{Di}^{t-n+1}, \dots, C_{Li}^t, C_{Di}^t)$, where t is the current step and n is the memory steps defined according to the tasks. n is set as 32 for this work.

III. RESULTS OF CYLINDERS WITH CONSTANT ROTATION SPEED

In order to understand better and explore the trend of forces on the *fluidic pinball* with the adjustment of ε_i , we analyzed the 1331 simulations' results of open-loop control with constant rotation speed based on their forces and vortical wakes in detail. Meanwhile, these results from brute-force searching are valuable references for the DRL-based feedback control to verify the performance of the DRL algorithm.

Figures 2–4 are the average of the drag coefficient C_{D0} , the average of the lift coefficient C_{L0} , and the standard deviation of the lift coefficient \bar{C}_{L0} of *fluidic pinball* system with respect to $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ separately. Subfigure (a) is the isosurface of the coefficients in the three-dimensional parametric space corresponding to the rotations of cylinders of 1331 simulations; subfigures (b)–(d) are the contours of sections on three coordinate axes of subfigure (a) at the value equal to 0, 4, -4, where the coefficients $C_{Di/Li} \sim f(\varepsilon_i, \varepsilon_j), i \in [1, 2, 3] \neq j \in [1, 2, 3]$; subfigures (e)–(g) are snapshot vorticities extracted of red spots from subfigures (b)–(d), respectively.

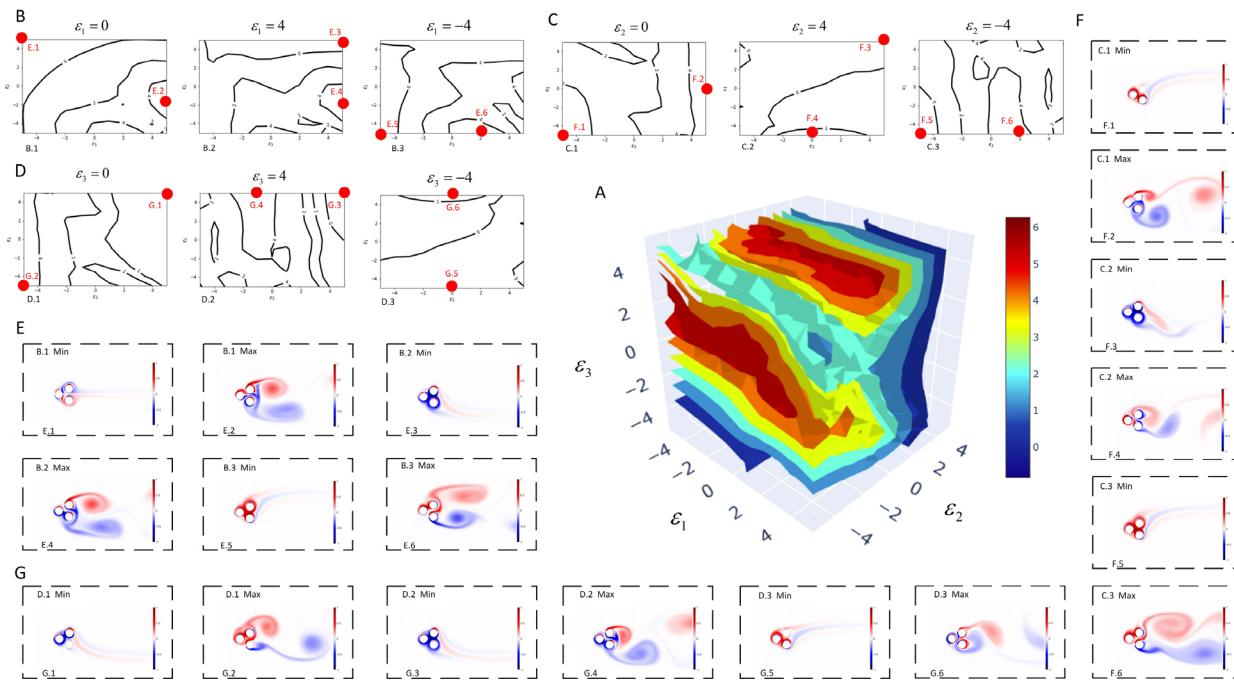


FIG. 2. Average C_{D0} of fluidic pinball system in the three-dimensional parametric space using open-loop control with constant rotation speeds. Isosurface, contours, and vorticities of the C_{D0} : (a) Point at each position is the C_{D0} corresponding to the rotations of cylinders of 1331 simulations ($C_{D0} \sim f(\varepsilon_1, \varepsilon_2, \varepsilon_3)$). Red and light colors mean C_{D0} is higher, and blue and dark colors mean C_{D0} is smaller and negative. (b)–(d) Contours of sections on three coordinate axes of subfigure (a) at the value equal to 0, 4, -4, where $C_{D0} \sim f(\varepsilon_i, \varepsilon_j)$, $i \in [1, 2, 3] \neq j \in [1, 2, 3]$. B.1, B.2, and B.3 are C_{D0} of $\varepsilon_1 = 0, 4, -4$; C.1, C.2, and C.3 are C_{D0} of $\varepsilon_2 = 0, 4, -4$; and D.1, D.2, and D.3 are $\varepsilon_3 = 0, 4, -4$, respectively. One of the fixed rotations are marked on top of contours. (e)–(g) Snapshot vorticities extracted from red spots from subfigures (b)–(d), where rotation speeds of three cylinders are marked for each case. Their corresponding labels can be found near red spots on contours. There are also the simplified tabs on vorticities, with max and min which means its extremum in the contour. Wake patterns of fluidic pinball depend on the patterns of all cylinders' rotation leading to related drag variation, and C_{D0} of constant rotations on rear cylinders show the symmetrical relationship as their symmetric positions along the centerline of the incoming flow.

A. Force distribution

Our main observations can be summarized as follows:

For average C_{D0} :

1. Along with the increasing of ε_3 , C_{D0} is increasing first and then decreasing when $\varepsilon_2 < -2$. When $\varepsilon_2 > -2$, C_{D0} is increasing along with the increasing of ε_3 .
2. Along with the increasing of ε_2 , C_{D0} is increasing first and then decreasing when $\varepsilon_3 > 2$. When $\varepsilon_3 < 2$, C_{D0} is decreasing along with the increasing of ε_3 .
3. The changing of ε_1 does not have much impact on C_{D0} .
4. There are two maximum regions. When $\varepsilon_2 \in (-3, -1)$ and ε_3 is near 5, C_{D0} has the maximum region large more than 6. When $\varepsilon_3 \in (1, 3)$ and ε_2 is near -5, C_{D0} is large more than 6 as well.
5. From above maximum regions to the periphery, C_{D0} decreases gradually, achieving the broad minimum regions less than 0.

For average C_{L0} :

1. C_{L0} is decreasing along with the increasing of ε_2 .
2. When $\varepsilon_2 < 1$, C_{L0} is decreasing along with the increasing of ε_3 . When $\varepsilon_2 > 2$, C_{L0} is always less than 0, shown in Fig. 3(a).
3. The maximum region about 10 is located at $\varepsilon_1 < -2$ and ε_3 near -5.
4. The minimum region about -10 is at $\varepsilon_1 > 2$ and ε_2 near 5.

For average \tilde{C}_{L0} :

1. Most of the region is less than 1 shown in Fig. 4(a).
2. When $\varepsilon_3 > 3$, \tilde{C}_{L0} is increasing along with the increasing of ε_3 , and increasing first and then decreasing along with the increasing of ε_2 .
3. When $\varepsilon_2 < -3$, \tilde{C}_{L0} is increasing along with the decreasing of ε_2 , and increasing first and then decreasing along with the decreasing of ε_3 .
4. There are two maximum regions. $\varepsilon_1 > 4$, $\varepsilon_2 \in (-2, 0)$, ε_3 near 5, and $\varepsilon_3 \in (0, 2)\varepsilon_2$ and ε_1 near -5 have the maximum \tilde{C}_{L0} more than 2.

Moreover, C_{D2} and C_{D3} of the rear cylinders show a symmetrical relationship with each other. In addition, C_{L2} and C_{L3} show a symmetrical and reverse relationship from each other. Moreover, the C_{L2} of the top cylinder and C_{L3} of the bottom cylinder have strong monotonicity along with the variations of speeds of their own cylinders, while the variation patterns of C_{D_i} on them are more complex shown in Fig. 5.

B. Vortical wake analysis

From the force distribution, we observe that different rotation combinations lead to various patterns of vortical wakes and will cause different forces on fluidic pinball. To get an insight into the patterns of

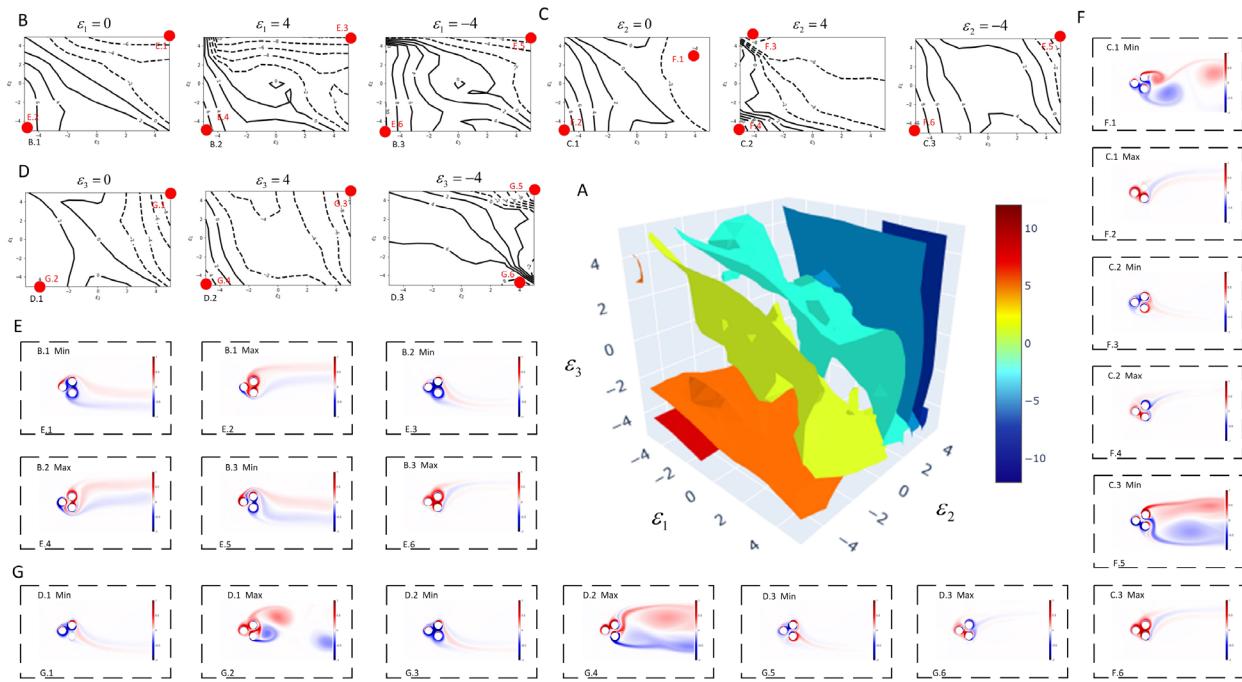


FIG. 3. Average C_{D0} of fluidic pinball system in the three-dimensional parametric space using open-loop control with constant rotation speeds. Isosurface, contours, and vorticities of the C_{D0} : (a) Point at each position is the C_{D0} corresponding to the rotations of cylinders of 1331 simulations ($C_{D0} \sim f(\varepsilon_1, \varepsilon_2, \varepsilon_3)$). Red and light colors mean C_{D0} is higher, and blue and dark colors mean C_{D0} is smaller and negative. (b)–(d) Contours of sections on three coordinate axes of subfigure (a) at the value equal to 0, 4, and -4 , where $C_{D0} \sim f(\varepsilon_i, \varepsilon_j)$, $i \in [1, 2, 3] \neq j \in [1, 2, 3]$. B.1, B.2, and B.3 are C_{D0} of $\varepsilon_1 = 0, 4, -4$; C.1, C.2, and C.3 are C_{D0} of $\varepsilon_2 = 0, 4, -4$; D.1, D.2, and D.3 are $\varepsilon_3 = 0, 4, -4$, respectively. One of the fixed rotations are marked on top of contours. (e)–(g) Snapshot vorticities extracted from red spots from subfigures (b)–(d), where rotation speeds of three cylinders are marked for each case. Their corresponding labels can be found near red spots on contours. There are also the simplified tabs on vorticities, with max and min which means its extremum in the contour. C_{D0} shows a relatively simple and monotonic variation pattern, which has a type of symmetrical and reverse relationship in cases of constant rotations on rear cylinders.

wakes impacts the force distribution, we selected some key vorticities in Fig. 2, Figs. 3, and 4 as representative to demonstrate the complex vortical wakes behind the *fluidic pinball* as the function of ε_1 , ε_2 , ε_3 .

For average C_{D0}

In Fig. 2, E.2, E.4, E.6, F.2, F.4, F.6, G.2, G.4, and G.6 show the complex vortex pattern in the near wake and Karman vortex street in far wake. The rotation speeds of three cylinders in vector form are $(0, -2, 5)$, $(4, -2, 5)$, $(-4, -5, 2)$, $(0, 0, 5)$, $(-5, 4, 0)$, $(-5, -4, 2)$, $(-5, -5, 0)$, $(5, -1, 4)$, and $(5, 0, -4)$, respectively. If the rotation of cylinders is divided into the fix, clockwise, and counterclockwise rotations, two rear cylinders in these 9 cases do not have the same performance leading to this kind of complex vortex pattern. On the other hand, the front cylinders perform all rotation patterns, which means it does not have a decisive influence on this wake. They have the maximum C_{D0} in their contours as the wakes lead to lower pressure behind the *fluidic pinball*. There are complex and chaotic vortices in near wakes improving the velocity after *fluidic pinball* and reducing the pressure as the Bernoulli effect. Then, C_{D0} becomes positive and similar to the difference of pressure between upstream and downstream of the *fluidic pinball* system.

On the other hand, E.1, E.3, E.5, F.1, F.3, F.5, G.1, G.3, and G.5 show a smooth wake after the *fluidic pinball*. The rotation speeds are $(0, 5, -5)$, $(4, 5, 5)$, $(-4, -5, -5)$, $(-5, 0, -5)$, $(5, 4, 5)$, $(-5, -4, -5)$, $(5, 5, 0)$, $(5, 5, 4)$, and $(-5, 0, -4)$, respectively. In these cases, two rear

cylinders do not have the reverse rotations, except for E.1, where rear cylinders rotate in equal and reverse high velocities (a boat-tailing mechanism), leading to a relatively small C_{D0} . The front cylinder, in most cases, rotates in the same direction as the rear except for the fixed front one in E.1, which leads to a slight C_{D0} on the *fluidic pinball*. The smooth and narrow wakes reduce the flow velocities after the *fluidic pinball*. Thus, higher pressure after the system leads to a negative C_{D0} .

With the comparison between E.3 $(-4, 5, 5)$ and E.4 $(-4, -2, 5)$ in Fig. 2, only the rotation of the top cylinder has the change from clockwise to counterclockwise, which transfers the wake from a pattern of wake deflecting to a downward direction to alternating shedding. As the rear cylinders in E.4 have reverse rotations to the inner, their shears bleed from the gap leading to the complex vortex and high C_{D0} . After that, between G.5 $(-5, 0, -4)$ and G.6 $(5, 0, -4)$ in Fig. 2, only the front cylinder's rotation has changed from counterclockwise to clockwise. As the front cylinder in G.5 has a counterclockwise rotation, its shears deflect up and then extend to wake. Some smooth and independent shears in the wake decrease the flow velocity and improve the pressure leading to the negative C_{D0} . As the front and bottom cylinders in G.6 have reverse clockwise rotations, the pressure after the *fluidic pinball* is reduced, leading to the positive C_{D0} . Similarly, between F.5 $(-5, -4, -5)$ and F.6 $(-5, -4, 2)$, only the rotation of the bottom cylinder has changed from counterclockwise to clockwise, which transfers the wake from the pattern of wake deflecting to upward to alternating shedding.

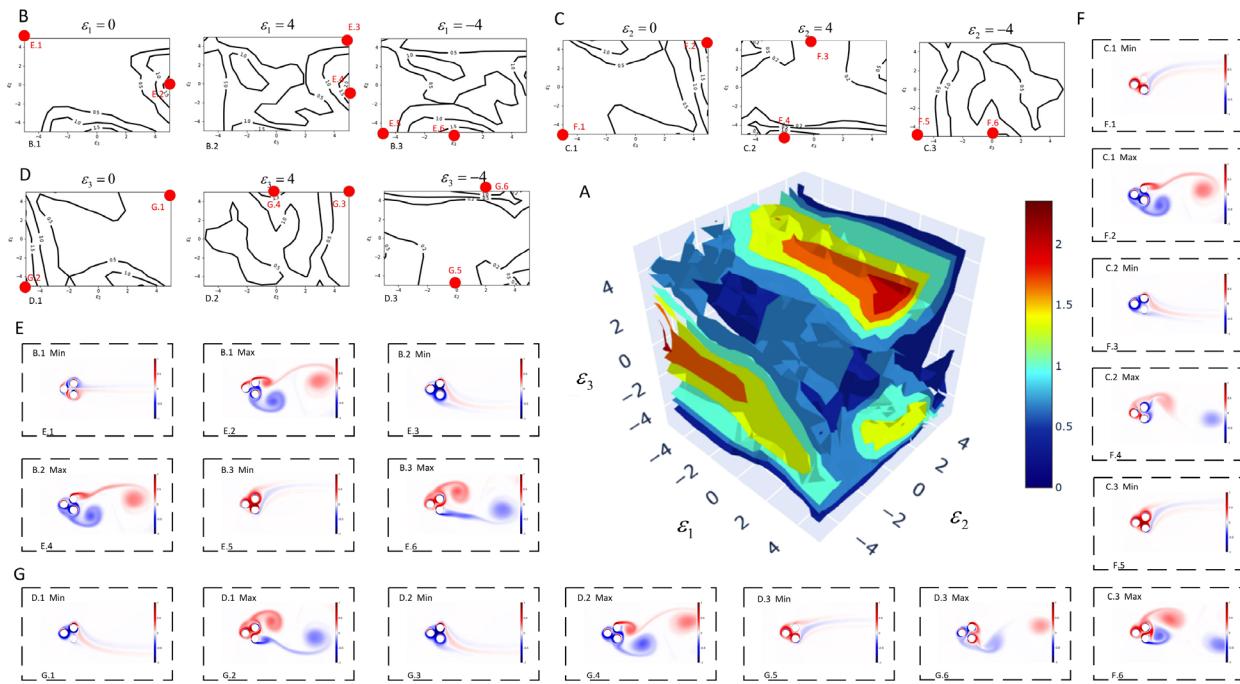


FIG. 4. Average \bar{C}_{L0} of fluidic pinball system in the three-dimensional parametric space using open-loop control with constant rotation speeds. Isosurface, contours, and vorticities of the \bar{C}_{L0} : (a) Point at each position is the \bar{C}_{L0} corresponding to the rotations of cylinders of 1331 simulations ($\bar{C}_{L0} \sim f(\varepsilon_1, \varepsilon_2, \varepsilon_3)$). Red and light colors mean \bar{C}_{L0} is higher, and blue and dark colors mean \bar{C}_{L0} is smaller and negative. (b)–(d) Contours of sections on three coordinate axes of subfigure (a) at the value equal to 0, 4, and -4 , where $\bar{C}_{L0} \sim f(\varepsilon_i, \text{vare}_j), i \in [1, 2, 3] \neq j \in [1, 2, 3]$. B.1, B.2, and B.3 are \bar{C}_{L0} of $\varepsilon_1 = 0, 4, -4$; C.1, C.2, and C.3 are \bar{C}_{L0} of $\varepsilon_2 = 0, 4, -4$; D.1, D.2, and D.3 are $\varepsilon_3 = 0, 4, -4$, respectively. One of the fixed rotations are marked on top of contours. (e)–(g) Snapshot vorticities extracted of red spots from subfigures (b)–(d), where rotation speeds of three cylinders are marked for each case. Their corresponding labels can be found near red spots on contours. There are also the simplified tabs on vorticities, with max and min which means its extremum in the contour. Narrow wakes reduce \bar{C}_{L0} and the cases with minimum \bar{C}_{L0} have narrow wake, while cases with maximum \bar{C}_{L0} have wide wake.

C_{D0} is more significant than 4 in F.6, while C_{D0} is smaller than 0 in F.5 as the change of rotation of the top cylinder.

We can also discover that F.4 ($-5, 4, 0$) and G.6 ($5, 0, -4$) have the same C_{D0} near 1. Meanwhile, F.5 ($-5, -4, -5$) and G.3 ($5, 5, 4$) have the same C_{D0} of less than 0. However, the patterns of their wakes are different. The wake of F.5 deflects to the upside while that of G.3 deflects to the down. The wakes of F.4 and G.6 have a symmetric relationship. Except for the vorticities having a symmetric relationship, the contours like C.2 and D.3 are also symmetric. C.1 and D.1, C.2 and D.3, and C.3 and D.2 in Fig. 2 have 180° rotation symmetry. The drags of inverse rotation speeds on three cylinders with the swap of two rear cylinders are equal.

For average C_{L0}

In Fig. 3, E.2, E.4, E.6, F.2, F.4, F.6, and G.6 show the wakes deflecting to the up position. The rotation speeds of three cylinders in vector form are $(0, -5, -5)$, $(4, -5, -5)$, $(-4, -5, -5)$, $(-5, 0, -5)$, $(-5, 4, -5)$, $(-5, -4, -5)$, and $(-5, 4, -4)$, respectively. There is always a rear cylinder (bottom one or top one) or both of them rotating in the counterclockwise direction so that the shears can be controlled to deflect up. They have the maximum C_{L0} in their contours as the wakes lead to a difference in pressure. As the wake accelerates the flow velocity, the pressure on the upside is lower, caused by higher flow velocity as the Bernoulli effect if the wake deflects into the up direction. Thus, the lift is positive and from down to up in a vertical direction. There are two cases, G.2 and G.4, with rotation $(-5, -4, 0)$

and $(-5, -5, 4)$ also having the maximum C_{L0} in their group, which do not have the up-deflecting wake. Drag in G.2 and G.4 is near 4 with the relative complex wake, but the wake also has a deflecting trend.

On the other hand, E.1, E.3, E.5, F.3, G.1, G.3, and G.5 show the wake after the fluidic pinball deflecting to the down direction. The rotation speeds are $(0, 5, 5)$, $(4, 5, 5)$, $(-4, 5, 5)$, $(5, 4, -4)$, $(5, 5, 0)$, $(5, 5, 4)$, and $(5, 5, -4)$, respectively. There is always a rear cylinder (bottom one or top one) or both of them rotating in the clockwise direction so that the shears can be controlled to deflect down. Thus, the pressure on the downside is lower, caused by higher flow velocity leading to the negative lifts of fluidic pinball. They have the minimum and negative C_{L0} in their contours. There are two cases, F.1 and F.5, with rotation $(3, 0, 4)$ and $(5, -4, 5)$ also having the negative C_{L0} in their groups, which has a relatively complex pattern in near wake after cylinders where the shear of the top cylinder deflects to down and merges with shears of front and bottom cylinders leading the wake to have a down deflecting trend as well.

With the comparison between F.3 ($5, 4, -4$) and G.6 ($-5, 4, -4$) in Fig. 3, only the rotation of the front cylinder has the change from clockwise to counterclockwise, which transfers the wake from down deflection to up deflection. The reverse rotation of the front cylinder also leads to inverse deflection and inverse C_{L0} . After that, between F.4 ($-5, 4, -5$) and G.5 ($5, 5, -4$), except for the reverse rotation on the front cylinder, there are slight tunes for the rear cylinders. The C_{L0} transfers from the positive to negative as the deflective changes of wakes. Similarly, the difference between F.5 ($5, -4, 5$) and

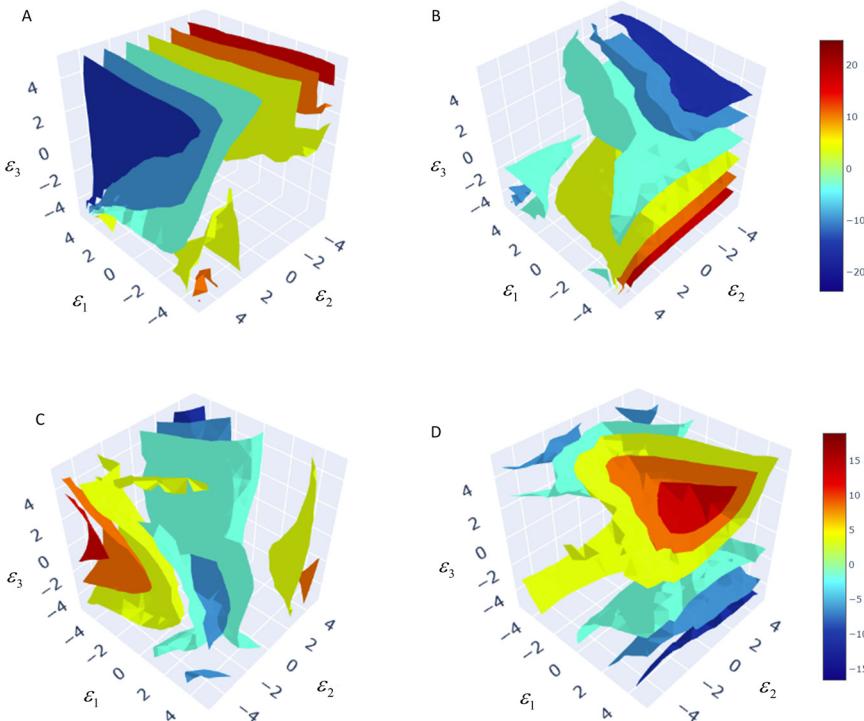


FIG. 5. C_{Li} and C_{Di} of the top and bottom cylinders. C_{Li} and C_{Di} on two rear cylinders in the three-dimensional parametric space. Values in isosurfaces are the coefficients on a single cylinder corresponding to the rotations of three cylinders, which are functions of ε_1 , ε_2 and ε_3 : (a) Points in isosurface are C_{L2} on the top cylinder, where red and light colors mean values are higher, and blue and dark colors mean values are smaller and negative. (b) Points in the isosurface are C_{L3} on the bottom cylinder corresponding to the rotations of cylinders. (c) Points in isosurface are C_{D2} on the top cylinder, where red and light colors mean values are higher, and blue and dark colors mean values are smaller and negative. (d) Points in the isosurface are C_{D3} on the bottom cylinder corresponding to the rotations of cylinders. The C_{Li} on the bottom and top cylinders have strong monotonicity along with the speeds' variations of their own cylinders, while the variation patterns of C_{Di} on them are more complex.

G.4 ($-5, -5, 4$) also sees the same changes of wakes and C_{L0} caused by the reverse rotation of the front cylinder, although its rotation is relatively small in the two cases. Then, the rotation direction of the front cylinder has a decisive influence on the value of C_{L0} on *fluidic pinball*.

From the above three comparisons, we can discover that F.4 ($-5, 4, -5$) and G.5 ($5, 5, -4$) have the same absolute value of C_{L0} of more than 16 and reverse direction. Meanwhile, F.2 ($-5, 0, -5$) and G.1 ($5, 5, 0$) have the same absolute value of C_{L0} of more than 8 in the reverse direction as well. However, the patterns of their wakes are different. The wake of F.4 deflects up while that of G.5 deflects down, the wakes of F.2 deflect up, and G.1 deflects to the down direction, which has a symmetric relationship about the horizontal centerline leading to the inverse directions and the same absolute value of their C_{L0} , respectively. Except for the vorticities having a symmetric relationship, the contours like C.2 and D.3 are also symmetric. C.1 and D.1, C.2 and D.3, and C.3 and D.2 in Fig. 3 have 180° rotation and reverse symmetry different from the C_{D0} . The C_{L0} of inverse rotation speeds on three cylinders with the swap of two rear cylinders is reversed.

For average \tilde{C}_{L0}

In Fig. 4, E.2, E.4, E.6, F.2, F.4, F.6, G.2, G.4, and G.6 show the wide wakes alternating periodic shedding similar to Karman vortex street. The rotation speeds of three cylinders in vector form are $(0, 0, 5)$, $(4, 0, 5)$, $(-4, -5, 0)$, $(5, 0, 5)$, $(-5, 4, -2)$, $(-5, -4, 0)$, $(-5, -5, 0)$, $(5, 0, 4)$, and $(5, 2, -4)$, respectively, which have large \tilde{C}_{L0} . The coordinative rotations of cylinders lead to the vast wakes where the up vortex and down vortex alternating shed. They have the maximum \tilde{C}_{L0} in their contours as the alternating periodic vortices lead to large fluctuations in overall lifts.

On the other hand, E.1, E.3, E.5, F.1, F.3, F.5, G.1, G.3, and G.5 show the smooth and narrow wake after the *fluidic pinball*

deflecting to any direction. The rotation speeds are $(0, 5, -5)$, $(4, 5, 5)$, $(-4, -5, -5)$, $(-5, 0, -5)$, $(5, 4, 0)$, $(-5, -4, -5)$, $(5, 5, 0)$, $(5, 5, 4)$, and $(-5, 0, -4)$, respectively. They have the minimum \tilde{C}_{L0} in their contours. Whether the wakes deflect in up or down directions, the smooth wakes do not cause large fluctuations of lift on *fluidic pinball*, leading a very small \tilde{C}_{L0} (near 0) in general.

In Fig. 4, with the comparison between F.2 ($5, 0, 5$) and G.2 ($-5, -5, 0$), they have the same \tilde{C}_{L0} near 2 but have different wakes, which are symmetric about the horizontal centerline. \tilde{C}_{L0} between F.3 ($5, 4, 0$) and G.5 ($-5, 0, -4$) are equally less than 0.2, where the wakes are also symmetric about the horizontal centerline deflecting to up or down directions. Similarly, for F.6 ($-5, 4, 0$) and G.4 ($5, 0, 4$), their \tilde{C}_{L0} are same near 1.5. Their wakes are symmetric about the horizontal centerline, with alternating shedding vortices. From the above comparisons, we can discover that some cases using different action pairs with the symmetric wakes have the same \tilde{C}_{L0} . Except for the vorticities having a symmetric relationship, the contours like C.2 and D.3 are also symmetric. C.1 and D.1, C.2 and D.3, and C.3 and D.2 in Fig. 4 have 180° rotation symmetry. \tilde{C}_{L0} of inverse rotation speeds on three cylinders with the swap of two rear cylinders are equal.

IV. RESULTS OF DRL-BASED FEEDBACK CONTROL

A. DRL-based force extremum and tracking

In the first task, to demonstrate the validity of the DRL algorithm applied in the current work, we attempt to compare the DRL optimization result with that found in brute-force searching by minimizing the drag on the cylinder while maximizing the lift. The results of the C_{D0} , C_{Li} , and actions are plotted in Fig. 6, while the setup of the reward r in the first task is as follows:

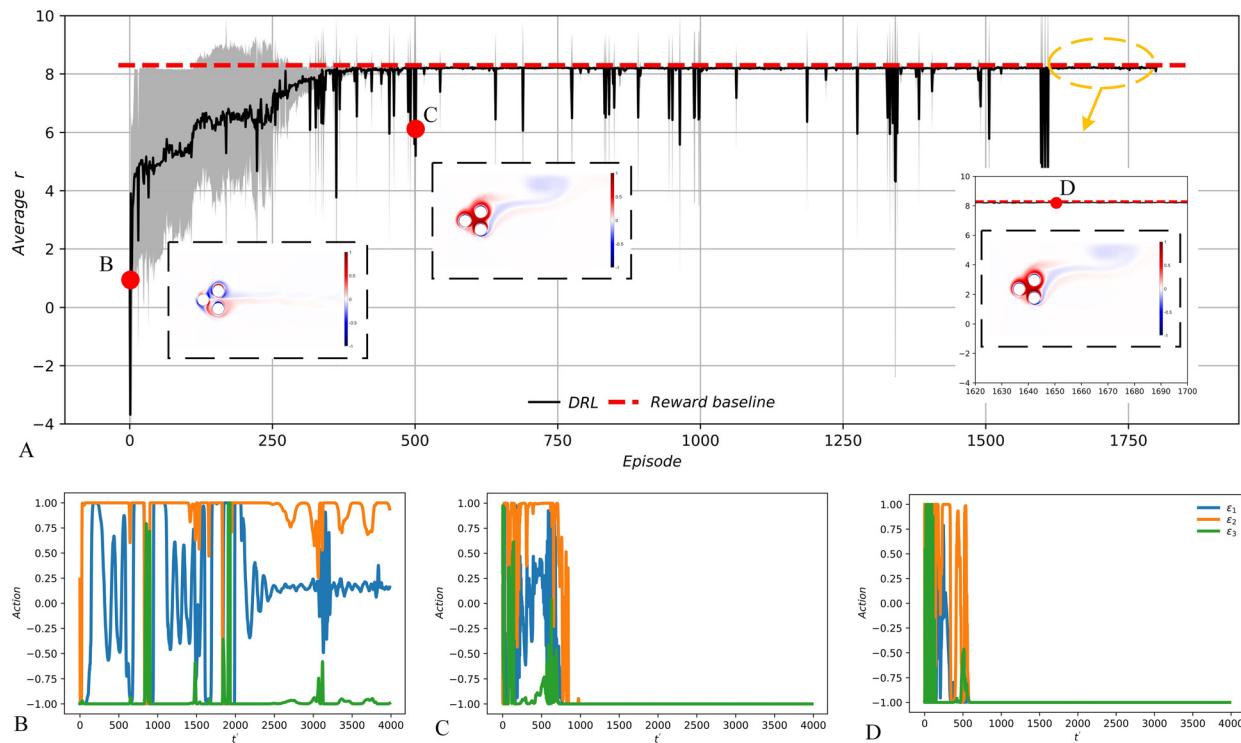


FIG. 6. Training process of DRL-based flow control for forces extremum. The training process represented by the average r against the episode number of four different agents, the corresponding actions evaluated by agents, and vortical wake visualization of various training stages, where $t' = \frac{tD}{U}$. (a) The changing of average r and the vortical wakes of the last time step at three training stages. The shown line here is mean value and the shade area is corresponding variance. (b)–(d) The changing of generated actions in evaluation at three training stages sharing the same legend.

$$r = -\text{sgn}(C_{Di})(|C_{Di}|)^{1/2} + \text{sgn}(C_{Li})(|C_{Li}|)^{1/2}, \quad (2)$$

where the first term denotes the reward for minimizing drag (maximizing thrust) on the front cylinder, and the second term is the reward for maximum and positive direction lift on the bottom cylinder. We reduce the drag of the front cylinder and enhance the lift on the bottom cylinder simultaneously, demonstrating the ability of DRL to find force extremum. The DRL-based method can control the drag and lift to the minimum and maximum values, respectively, like that searched by open-loop control with the same reward values.

DRL-based force control results show that C_{Di} and C_{Li} can achieve steady extremum. The training process is shown in Fig. 6 with 6A, the average reward (black line) that converges is calculated by Eq. (2), and the vortical wakes in the last time step of the three training stages are shown near the curve. 6B, 6C, and 6D are the corresponding actions for three circular cylinders normalized in the range of $(-1, 1)$. Figure 6(a) shows that the reward increases from the initial value to the convergence position near 8.3, comparable with the reward-maximum case calculated from the brute-force searching (red line). Such a result indicates that the DRL-based control can reach as good a result as the exhaustive brute-force searching. The actions in the initial stage in 6B have a different pattern compared with the optimal actions in 6D, where three cylinders rotate in a counterclockwise direction with the fastest normalized speed $(-1.00, -1.00, -1.00)$, which is the case with minimum C_{D0} on the front cylinder and maximum C_{L0} on bottom cylinder among all simulations of open-loop control as well. The average

C_{D1} values are -12.02 and -12.01 for DRL-based control and open-loop control, while the average C_{L3} values are 24.92 and 24.82 , respectively. The wake deflects to the up direction as the counterclockwise rotation of three cylinders. As discussed in the last section, such a smooth, up-deflecting wake can achieve a significant lift and small drag. The above rotation pattern is similar to the Magnus effect according to Ref. 17.

The DRL-based control can achieve the same performance and solution as the open-loop control method in the force extremum problem. It is interesting to explore if C_{D0} on the system can be controlled to track different expected values dynamically. We have tested four cases to demonstrate the better ability of the DRL-based flow control algorithm for drag tracking problems on different C_{D0}^{ref} than the open-loop control method. For DRL control, it is essential to approximately design the reward function r to calculate reward values when the agent inquires about the states. The results of the C_{D0} , actions, and vortical wakes are plotted in Fig. 7, while the setup of the reward in these cases is as follows:

$$r = -(C_{D0} - C_{D0}^{\text{ref}})^2, \quad (3)$$

where C_{D0} is the time varying values at each step and C_{D0}^{ref} is the objective of tracking. To test the ability to drag tracking for arbitrary values, we made $C_{D0}^{\text{ref}} \in [0, 1, 2, 3]$, respectively. The following results demonstrated that C_{D0} always achieves the C_{D0}^{ref} fluctuating in small ranges in steady after a period of adjustment.

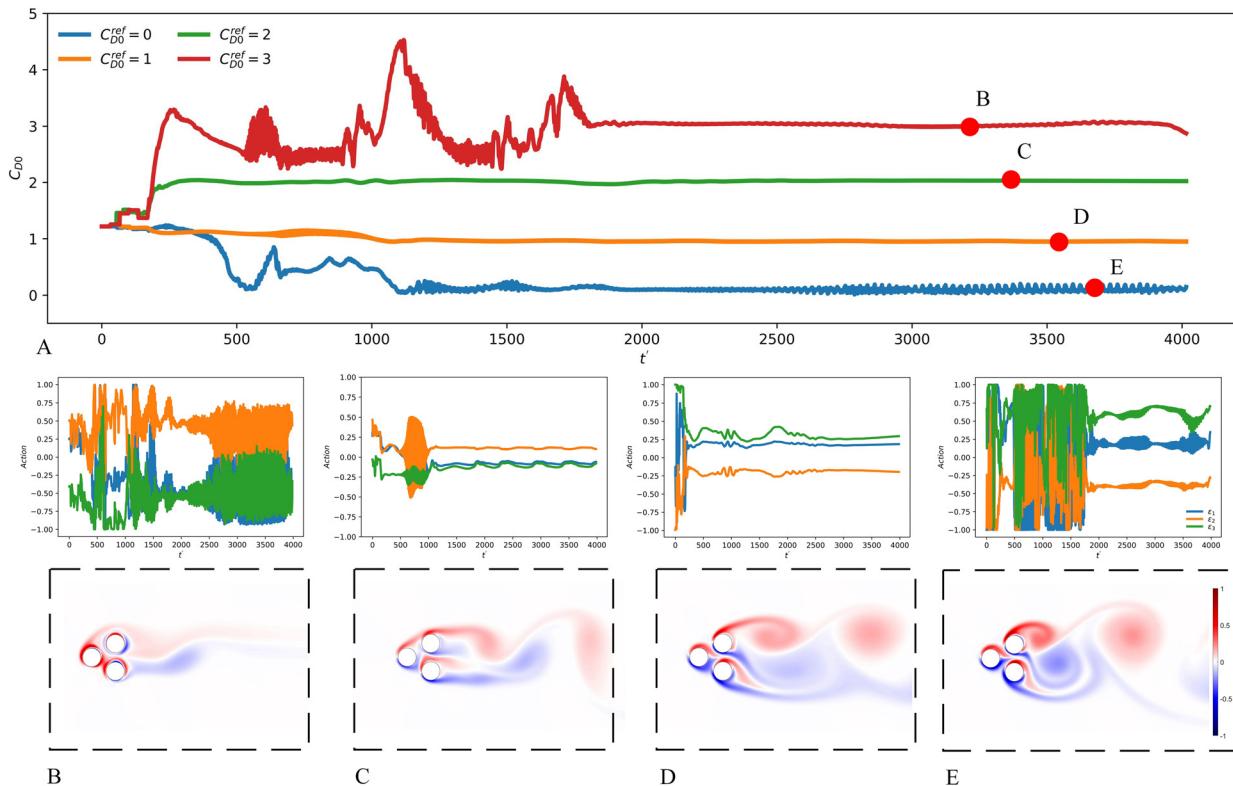


FIG. 7. Result of DRL-based flow control for forces tracking. The C_{D0} in the time domain for four force tracking problems, vortical wake visualization of the last time step after convergence, and the corresponding actions generated by agents. (a) The changing of C_{D0} that retain at 0, 1, 2, and 3 sharing the same legend. (b)–(e) The vortical wake extracted in the last time step from four tests, respectively, sharing the same color bar in subfigure (a) and the corresponding actions.

In Fig. 7(a), the curve of C_{D0} (red line) in the time-domain starts from the initial value to the terminal value fluctuating near 3 with an error less than ± 0.15 after 2000 steps in steady. The average r calculated by Eq. (3) converges to -0.0014 , approximately 0. The curve of C_{D0} (green line) in the time domain starts from the initial value to the terminal value fluctuating near 2 with an error less than ± 0.05 after 500 steps in steady. The average r converges to -0.0009 in the test experiment. The orange curve when $C_{D0}^{ref} = 1$ successfully remains the objective fluctuating about 1 with an error less than ± 0.05 . The average r converges to -0.0019 in the test. Moreover, the blue line when $C_{D0}^{ref} = 0$ also remains the objective fluctuating near 0 with an error less than ± 0.2 , and the average r during the test is -0.0108 .

In Figs. 7(b)–7(e), the vortical wakes are plotted to illustrate their patterns at the final time, and the rotations of three cylinders have periodic fluctuations similar to that of C_{D0} . We can observe that different tracking objectives have different wakes and the DRL algorithm tries to control wakes to unknown patterns with changing action dynamically so that C_{D0} can achieve the expected objectives persistently.

B. Further remarks on learning results

As mentioned earlier, the results of open-loop control with constant rotation speed are considered the reference baselines of the novel DRL-based feedback control. After analyzing the C_{D0} curves, the rotations' patterns, and the changes of vortical wakes generated by the

DRL algorithm, we compared the performances of the above two control methods.

We analyzed the C_{D0} tracking performance of two kinds of methods to verify that the DRL-based control can retain an accurate force with less heave, which shows the accurate control ability of the DRL algorithm for the fluid dynamic setting. We selected the actions in Fig. 7(e) and calculated the average actions for each cylinder as the inputs of the open-loop control. The actions are $(0.0, -0.3, 0.6)$. The changes of C_{D0} from the initial state to the expected value based on two control methods are shown in Fig. 8(a). The r calculated via Eq. (3) are -0.0014 and -0.2373 for DRL-based control and open-loop control, while the \tilde{C}_{D0} are 0.029 and 0.492, respectively, after steady. The system driven by open-loop control cannot achieve the $C_{D0}^{ref} = 3$, which fluctuates near 2.6 and has a gap to the expected C_{D0} , while the result of DRL-based control converges to the objective more accurately. From the wakes generated by two methods in Figs. 8(b) and 8(c), the pattern of wake and vortex shedding by the DRL-based control method is generally different but with many similarities compared with the open-loop method. Thus, the DRL-based control learns the novel policy to track the force reference more precisely with the real-time feedback tune, which is reasonable to achieve better performance. It is also the advantage and importance of DRL-based real-time feedback flow control, as it is able to respond to various instant possible changing transitions of *fluidic pinball* even the small changes.

Although the DRL-based control can achieve better performance, it is still being determined how DRL found the optimal solution and

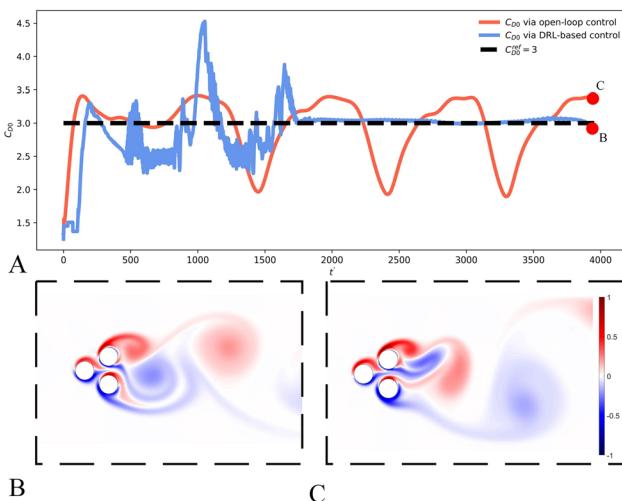


FIG. 8. Comparison for tracking of C_{D0} keeping $C_{D0}^{\text{ref}} = 3$. (a) The curve of C_{D0} changes via DRL-based control (orange line) and open-loop control (blue line). (b) and (c) The vorticities in the last time step via DRL-based control and open-loop control, respectively, sharing the same color bar, which shows DRL can achieve the better policy to retain C_{D0} , and its wake is generally different but with many similarities compared with the open-loop method.

generates such reasonable actions. Thus, we attempted to enlighten why actions are generated reasonably by DRL-based control with the help of open-loop control via brute-force searching results and fluid dynamics knowledge. In the problem that aims to minimize the C_{D1} and maximize the C_{L3} , it was easier to shed light on the decision-making after convergence. After about 800 t' , all actions became -1 , and three cylinders

rapidly rotated counterclockwise. It means the agent has already learned the law of such rotations and can achieve the minimum C_{D1} and maximum C_{L3} by thousands of simulations, in which the patterns of rotations and wake are the same as the result of open-loop control. Then, no matter what feedback C_{D1} and C_{L3} were, rotations did not change and always kept the values. However, for the more complicated force tracking problems, it is much more difficult to comprehend the behavior of decision-making as actions are time-vary and dynamic. We introduced a decision tree (DT) model to comprehend the basis of actions made by DRL.

Decision tree (DT) is a type of machine learning model to realize the prediction function with supervised learning, which has often been used to create easy-to-understand solutions to classification and regression problems.^{68,69} It is a tree that starts at a root node and branches based on conditions. An additional benefit to DT is that they can be represented graphically, which aids in human understanding. Some work has explored to use DT model to show the interpretability of the deep neural network (DNN) and DRL.^{70–74} In this work, we applied a DT model to illuminate which decisions were taken in which situations using the data from a well-trained agent for the $C_{D0}^{\text{ref}} = 1$ tracking problem. In the beginning, the agent was employed to collect the data $(C_{D0}, \bar{s}, \varepsilon_1, \varepsilon_2, \varepsilon_3)$, where \bar{s} is the average value among each state with current feedback and historical C_{D0} . Then, we used the data set to train the DT model with C_{D0} and \bar{s} as input and $\varepsilon_1, \varepsilon_2, \varepsilon_3$ as labels. After training, we can apply the trained DT model to imitate actions decided by input situations, and the mean square error (MSE) was about 0.26, which means the trained DT model fitted the mapping of data well shown in Fig. 9.

From Fig. 9(a), the distribution between $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and \bar{s} formed the clear dividing that $\varepsilon_2 > 0$ and $\varepsilon_3 < 0$ in the right side when $\bar{s} > 1.15$, which made shears around cylinders disperse outside, leading to the smooth and narrow wake reduce C_{D0} (boat-tailing mechanism). However, we cannot observe a clear action trend only via a \bar{s} when \bar{s} is

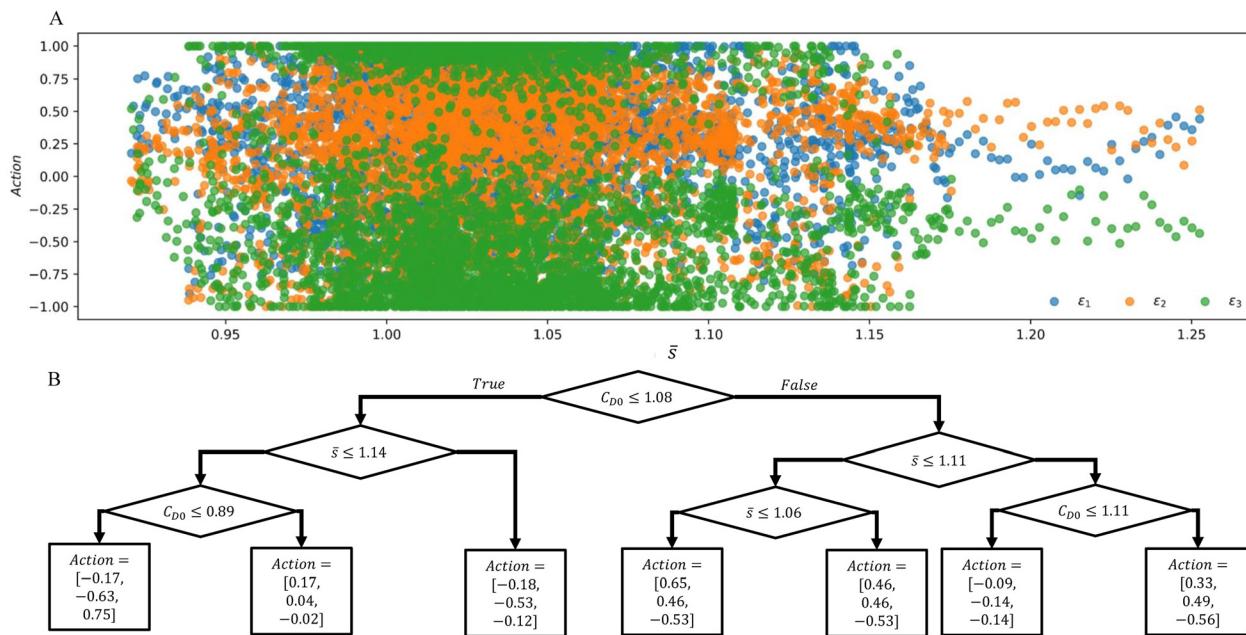


FIG. 9. Distribution of actions and its correspondence with C_{D0} and \bar{s} . (a) The distribution of $\varepsilon_1, \varepsilon_2, \varepsilon_3$ along with changes of \bar{s} . (b) The decision tree model trained via $(C_{D0}, \bar{s}, \varepsilon_1, \varepsilon_2, \varepsilon_3)$ by using the package `tree.DecisionTreeRegressor` in `sklearn` where depth is 3 without random state. The proportion of training and test samples is 8 : 2, and the tested MSE score is 0.26.

small. By using the DT model further integrating the C_{D0} of each step shown in Fig. 9(b), the root node divides the branch into two parts according to whether the C_{D0} is greater than the $C_{D0}^{ref} = 1.08$. When $C_{D0} > 1.11$ and $\bar{s} > 1.11$, that means C_{D0} and \bar{s} are larger than the expected value; the action (0.33, 0.49, and -0.56) was taken similarly as the phenomenon in Fig. 9(a). The base-bleed mechanism (-0.17, -0.63, and 0.75) leading to the complex and wide wake happened when $C_{D0} < 0.89$, although \bar{s} did not show some characteristics. Due to the complexity of DRL, there are thousands of probabilities to generate action. The purpose of our method was not to reveal the one-to-one correspondence between each state and action, but to apply such sample DT model to shed light on the basis of decision-making obeying human understanding.

V. CONCLUSION

We illustrated that the forces of both drag and lift on the system composed of three bluff cylinders (*fluidic pinball*) could be controlled to various objectives by applying DRL-based feedback flow control. The well-trained DRL-based control method finds the optimal policy to achieve the minimum drag and maximum lift, the same as the brute-force searching, and explores the novel control policy and wake pattern to track any expected drags. Furthermore, we systematically analyzed the force distribution and vortical wake pattern of the *fluidic pinball*, which provides the detailed baseline and reference to measure the performance of control approaches. Moreover, we explored to understand the basis of decision-making in DRL-based control for fluid dynamic problems, and the treelike-structure machine learning model can shed light on the decisions generation according to the hydrodynamic forces feedback.

The applications of DRL in fluid dynamics are encouraging while there are also some gaps waiting for exploring and solving. In this work, only the force information was extracted as a state to learn, and we are studying the probability of higher dimensional force and wake control approach based on DRL for *fluidic pinball*, applying the detailed pressure and velocity information from the wake. On the other hand, we have developed a distributed computing framework based on this work, which can solve CFD problems in a significant number of fluid environments at the same time. We are embedding such distributed computing framework in DRL-based flow control. Currently, we are also exploiting more efficient and detailed method to reveal decision-making for DRL based on fluid dynamic problems, and find out better regulation to design the state and reward function.

ACKNOWLEDGMENTS

We would like to acknowledge the funding supported by State Key Laboratory of Ocean Engineering (Shanghai Jiao Tong University) (Grant No. GKZD010081) and the research initiation grant provided by the Westlake University (No. 103110556022101).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Haodong Feng and Yue Wang contributed equally to this paper.
Haodong Feng: Data curation (equal); Investigation (equal); Methodology (equal); Resources (equal); Software (equal); Validation

(equal); Visualization (equal); Writing – original draft (equal). **Yue Wang:** Conceptualization (equal); Data curation (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal). **Hui Xiang:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal). **Zhiyang Jin:** Funding acquisition (equal); Validation (equal); Visualization (equal). **Dixia Fan:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in Github at <https://github.com/HDFengChina/How-to-Control-Hydrodynamic-Force>, Ref. 75.

APPENDIX A: VALIDATION OF NUMERICAL METHOD

Different resolutions in the CFD solver are tested to trade between the accuracy and calculation time. We made resolution

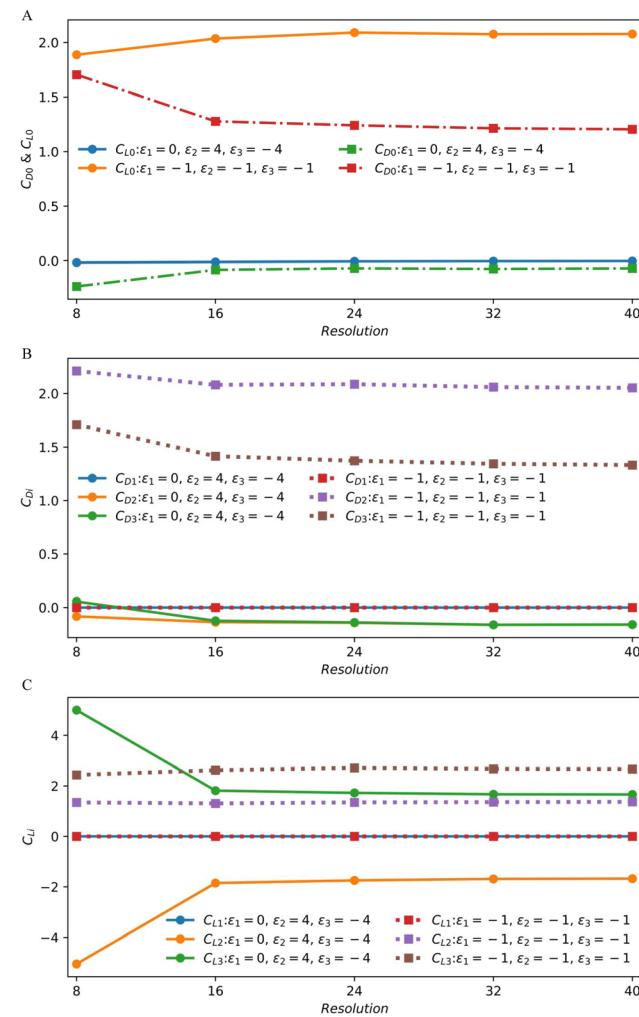


FIG. 10. Test the effect of resolution on C_{D_i} and C_{L_i} in CFD. (a) The C_{D0} and C_{L0} effected by various resolutions. (b) and (c) The C_{D_i} and C_{L_i} , $i = 1, 2, 3$, respectively.

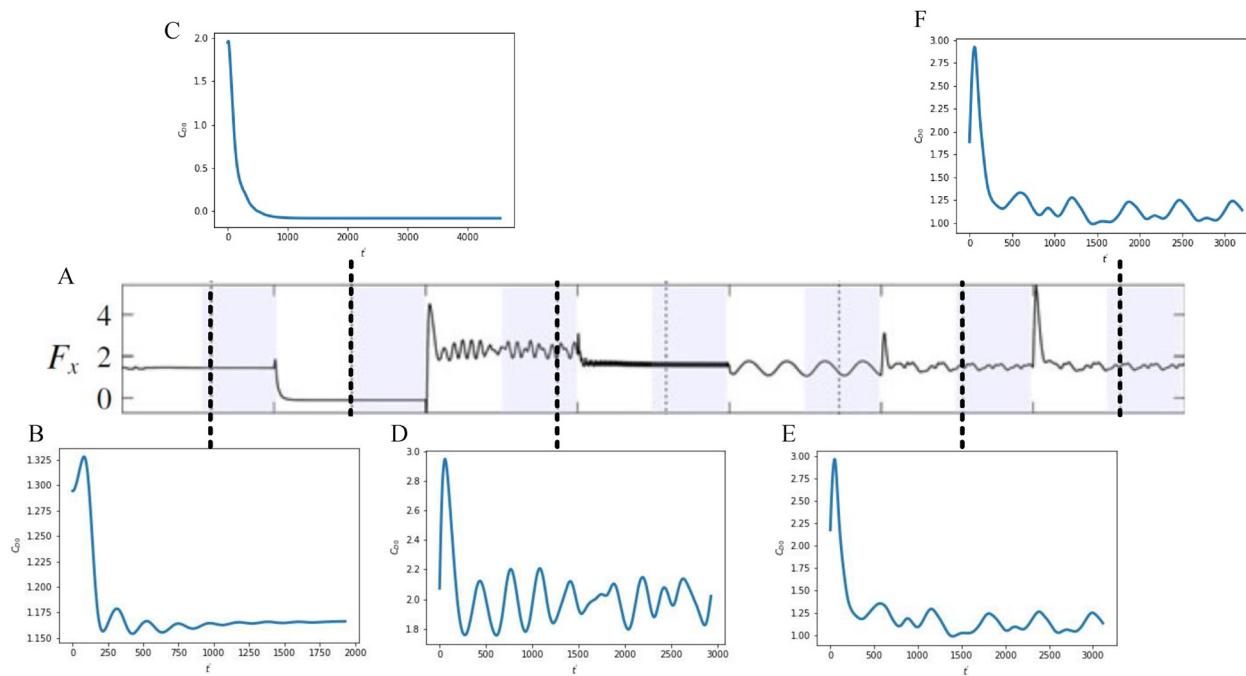


FIG. 11. Verification of the CFD environment with previous work. (a) The screenshot from Ref. 17 as the reference of our CFD environment. (b)–(f) The C_{D0} of the system corresponds to the unforced no rotation, boat tailing, base bleed, and Magnus effect.

TABLE I. List of hyper-parameters used in DRL.

Hidden layers	2 (actor and critic)	State dimension	32 (tracking)/64 (extremum)
Exploration noise	$\mathcal{N}(0, 0.1)$	Number of neurons in hidden layers	256 (actor & critic)
Target update rate	5×10^{-3}	Critic noise	$\mathcal{N}(0, 0.2)$
Buffer size	5×10^5	Optimizer	Adam
Actor learning rate	10^{-4}	Critic learning rate	10^{-4}
Action dimension	3	Discount factor	0.99
Number of environment	1	Batch size	512
Gap of target update	2	Number of critic in target	2

equal to 8, 16, 24, 36, and 40 to calculate C_{Di} and C_{Li} for cases of boat tailing and the Magnus effect, respectively. The calculation time is significantly increasing along with the increase in resolution. We can observe that C_{Di} and C_{Li} keep similar values when $resolution \geq 16$ from Fig. 10, while the values are discrepant when $resolution = 8$. Thus, we selected $resolution = 16$ in this work to speed up the training process with also the relatively accurate calculation.

The numerical method applied in this work is feasible to represent the performance of C_{Di} and C_{Li} when $resolution = 16$. We verified it with previously published paper¹⁷ as a reference in some different control mechanisms, including the boat tailing, base bleed, Magnus effect, and unforced no rotation shown in Fig. 11. By comparing the C_{D0} with corresponding references, the results from our solver can accurately represent the trend of these values. The purpose of this work is not to achieve high-accuracy

values but to show the control ability of DRL in the CFD environment.

APPENDIX B: HYPER-PARAMETERS IN THE DRL-BASED FLOW CONTROL

The detailed hyper-parameters used by DRL-based flow control in our work are listed in Table I.

REFERENCES

- R. Kerswell, “Nonlinear nonmodal stability theory,” *Annu. Rev. Fluid Mech.* **50**, 319–345 (2018).
- C. W. Rowley and S. T. Dawson, “Model reduction for flow analysis and control,” *Annu. Rev. Fluid Mech.* **49**, 387–417 (2017).
- P. Ashill, J. Fulker, and K. Hackett, “A review of recent developments in flow control,” *Aeronaut. J.* **109**, 205–232 (2005).

- ⁴H. Choi, W.-P. Jeon, and J. Kim, "Control of flow over a bluff body," *Annu. Rev. Fluid Mech.* **40**, 113–139 (2008).
- ⁵D. Xu and M. Zhang, "Reinforcement-learning-based control of convectively unstable flows," *J. Fluid Mech.* **954**, A37 (2023).
- ⁶D. Greenblatt and D. R. Williams, "Flow control for unmanned air vehicles," *Annu. Rev. Fluid Mech.* **54**, 383–412 (2022).
- ⁷M. R. Jovanović, "From bypass transition to flow control and data-driven turbulence modeling: An input–output viewpoint," *Annu. Rev. Fluid Mech.* **53**, 311–345 (2021).
- ⁸C.-A. Yeh, M. G. Meena, and K. Taira, "Network broadcast analysis and control of turbulent flows," *J. Fluid Mech.* **910**, A15 (2021).
- ⁹C. S. Greco, G. Paolillo, T. Astarita, and G. Cardone, "The von Kármán street behind a circular cylinder: Flow control through synthetic jet placed at the rear stagnation point," *J. Fluid Mech.* **901**, A39 (2020).
- ¹⁰E. J. Gutmark and F. F. Grinstein, "Flow control with noncircular jets," *Annu. Rev. Fluid Mech.* **31**, 239–272 (1999).
- ¹¹J. Kim and T. R. Bewley, "A linear systems approach to flow control," *Annu. Rev. Fluid Mech.* **39**, 383–417 (2007).
- ¹²L. N. Cattafesta III and M. Sheplak, "Actuators for active flow control," *Annu. Rev. Fluid Mech.* **43**, 247–272 (2011).
- ¹³T. Shaqarin, B. R. Noack, and M. Morzyński, "The need for prediction in feedback control of a mixing layer," *Fluid Dyn. Res.* **50**, 065514 (2018).
- ¹⁴U. K. Kaul, "An active flow control approach for spatially growing mixing layer," *J. Fluids Eng.* **144**, 061110 (2022).
- ¹⁵M. Linkmann, F. Knierim, S. Zammert, and B. Eckhardt, "Linear feedback control of invariant solutions in channel flow," *J. Fluid Mech.* **900**, A10 (2020).
- ¹⁶O. M. Aamo and M. Krstic, *Flow Control by Feedback: Stabilization and Mixing* (Springer Science & Business Media, 2003).
- ¹⁷R. Ishar, E. Kaiser, M. Morzyński, D. Fernex, R. Semaan, M. Albers, P. S. Meysonnat, W. Schröder, and B. R. Noack, "Metric for attractor overlap," *J. Fluid Mech.* **874**, 720–755 (2019).
- ¹⁸N. Deng, B. R. Noack, M. Morzyński, and L. R. Pastur, "Low-order model for successive bifurcations of the fluidic pinball," *J. Fluid Mech.* **884**, A37 (2020).
- ¹⁹S. Li, W. Li, and B. R. Noack, "Machine-learned control-oriented flow estimation for multiactuator multi-sensor systems exemplified for the fluidic pinball," *J. Fluid Mech.* **952**, A36 (2022).
- ²⁰N. Deng, L. R. Pastur, M. Morzyński, and B. R. Noack, "Route to chaos in the fluidic pinball," in *Fluids Engineering Division Summer Meeting* (American Society of Mechanical Engineers, 2018), Vol. 51555, p. V001T01A005.
- ²¹L. R. Pastur, N. Deng, M. Morzyński, and B. R. Noack, "Reduced-order modeling of the fluidic pinball," in *Chaotic Modeling and Simulation International Conference* (Springer, 2018), pp. 205–213.
- ²²G. Y. C. Macea, B. R. Noack, F. Lusseyran, N. Deng, L. Pastur, and M. Morzyński, "Artificial intelligence control applied to drag reduction of the fluidic pinball," in *Proceedings of Applied Mathematics and Mechanics (PAMM)* (Wiley Online Library, 2019), Vol. 19, p. e201900268.
- ²³S. Peitz, S. E. Otto, and C. W. Rowley, "Data-driven model predictive control using interpolated Koopman generators," *SIAM J. Appl. Dyn. Syst.* **19**, 2162–2193 (2020).
- ²⁴C. Raibaudo, P. Zhong, B. R. Noack, and R. J. Martinuzzi, "Machine learning strategies applied to the control of a fluidic pinball," *Phys. Fluids* **32**, 015108 (2020).
- ²⁵C. Raibaudo and R. Martinuzzi, "Unsteady actuation and feedback control of the experimental fluidic pinball using genetic programming," *Exp. Fluids* **62**, 219 (2021).
- ²⁶A. B. Blanchard, G. Y. C. Macea, D. Fan, Y. Li, Y. Zhou, B. R. Noack, and T. P. Sapsis, "Bayesian optimization for active flow control," *Acta Mech. Sin.* **37**, 1786–1798 (2021).
- ²⁷H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, and E. Hachem, "Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows," *Phys. Rev. Fluids* **6**, 053902 (2021).
- ²⁸G. Y. C. Macea, Y. Li, F. Lusseyran, M. Morzyński, and B. R. Noack, "Stabilization of the fluidic pinball with gradient-enriched machine learning control," *J. Fluid Mech.* **917**, A42 (2021).
- ²⁹Y. Li, W. Cui, Q. Jia, Q. Li, Z. Yang, M. Morzyński, and B. R. Noack, "Explorative gradient method for active drag reduction of the fluidic pinball and slanted Ahmed body," *J. Fluid Mech.* **932**, A7 (2022).
- ³⁰M. Brenner, J. Eldredge, and J. Freund, "Perspective on machine learning for advancing fluid mechanics," *Phys. Rev. Fluids* **4**, 100501 (2019).
- ³¹S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020).
- ³²S. L. Brunton, "Applying machine learning to study fluid mechanics," *Acta Mech. Sin.* **37**, 1718–1726 (2022).
- ³³S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *J. Fluid Mech.* **879**, 217–254 (2019).
- ³⁴E. Kharazmi, D. Fan, Z. Wang, and M. S. Triantafyllou, "Inferring vortex induced vibrations of flexible cylinders using physics-informed neural networks," *J. Fluids Struct.* **107**, 103367 (2021).
- ³⁵R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).
- ³⁶R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.* **112**, 181–211 (1999).
- ³⁷D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science* **362**, 1140–1144 (2018).
- ³⁸S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intelligent Transp. Syst.* **23**(3), 740–759 (2020).
- ³⁹J. Wang, J. Cao, S. Wang, Z. Yao, and W. Li, "IRDA: Incremental reinforcement learning for dynamic resource allocation," *IEEE Trans. Big Data* **8**, 770–783 (2020).
- ⁴⁰B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artif. Intell. Rev.* **55**, 945–990 (2022).
- ⁴¹H. Feng, L. Yu, and Y. Chen, "Optimal control and reinforcement learning for robot: A survey," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing* (Springer, 2021), pp. 54–66.
- ⁴²W.-L. Chen, Y. Huang, C. Chen, H. Yu, and D. Gao, "Review of active control of circular cylinder flow," *Ocean Eng.* **258**, 111840 (2022).
- ⁴³J. Rabault, F. Ren, W. Zhang, H. Tang, and H. Xu, "Deep reinforcement learning in fluid mechanics: A promising method for both active flow control and shape optimization," *J. Hydron.* **32**, 234–246 (2020).
- ⁴⁴J. Viquerat, P. Meliga, A. Larcher, and E. Hachem, "A review on deep reinforcement learning for fluid mechanics: An update," *Phys. Fluids* **34**, 111301 (2022).
- ⁴⁵J. 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).
- ⁴⁶H. Xu, W. Zhang, J. Deng, and J. Rabault, "Active flow control with rotating cylinders by an artificial neural network trained by deep reinforcement learning," *J. Hydron.* **32**, 254–258 (2020).
- ⁴⁷D. Fan, L. Yang, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Reinforcement learning for bluff body active flow control in experiments and simulations," *Proc. Natl. Acad. Sci. U.S.A.* **117**, 26091–26098 (2020).
- ⁴⁸C. Zheng, T. Ji, F. Xie, X. Zhang, H. Zheng, and Y. Zheng, "From active learning to deep reinforcement learning: Intelligent active flow control in suppressing vortex-induced vibration," *Phys. Fluids* **33**, 063607 (2021).
- ⁴⁹Q. Wang, L. Yan, G. Hu, C. Li, Y. Xiao, H. Xiong, J. Rabault, and B. R. Noack, "DRLinfluids—An open-source python platform of coupling deep reinforcement learning and openfoam," *Phys. Fluids* **34**, 081801 (2022).
- ⁵⁰B.-Z. Han, W.-X. Huang, and C.-X. Xu, "Deep reinforcement learning for active control of flow over a circular cylinder with rotational oscillations," *Int. J. Heat Fluid Flow* **96**, 109008 (2022).
- ⁵¹E. Amico, G. Cafiero, and G. Iuso, "Deep reinforcement learning for active control of a three-dimensional bluff body wake," *Phys. Fluids* **34**, 105126 (2022).
- ⁵²Y.-F. Mei, C. Zheng, Y. Hua, Q. Zhao, P. Wu, and W.-T. Wu, "Active control for the flow around various geometries through deep reinforcement learning," *Fluid Dyn. Res.* **54**, 015510 (2022).
- ⁵³Y.-Z. Wang, Y. Hua, N. Aubry, Z.-H. Chen, W.-T. Wu, and J. Cui, "Accelerating and improving deep reinforcement learning-based active flow control: Transfer training of policy network," *Phys. Fluids* **34**, 073609 (2022).

- ⁵⁴C. Zheng, F. Xie, T. Ji, X. Zhang, Y. Lu, H. Zhou, and Y. Zheng, “Data-efficient deep reinforcement learning with expert demonstration for active flow control,” *Phys. Fluids* **34**, 113603 (2022).
- ⁵⁵F. Ren, J. Rabault, and H. Tang, “Applying deep reinforcement learning to active flow control in weakly turbulent conditions,” *Phys. Fluids* **33**, 037121 (2021).
- ⁵⁶F. Ren, C. Wang, and H. Tang, “Bluff body uses deep-reinforcement-learning trained active flow control to achieve hydrodynamic stealth,” *Phys. Fluids* **33**, 093602 (2021).
- ⁵⁷Y. Xie, X. Zhao, and M. Luo, “An active-controlled heaving plate breakwater trained by an intelligent framework based on deep reinforcement learning,” *Ocean Eng.* **244**, 110357 (2022).
- ⁵⁸J. Li and M. Zhang, “Reinforcement-learning-based control of confined cylinder wakes with stability analyses,” *J. Fluid Mech.* **932**, A44 (2022).
- ⁵⁹F. Pino, L. Schena, J. Rabault, and M. A. Mendez, “Comparative analysis of machine learning methods for active flow control,” *J. Fluid Mech.* **958**, A39 (2022).
- ⁶⁰R. Paris, S. Beneddine, and J. Dandois, “Robust flow control and optimal sensor placement using deep reinforcement learning,” *J. Fluid Mech.* **913**, A25 (2021).
- ⁶¹P. Varela, P. Suárez, F. Alcántara-Ávila, A. Miró, J. Rabault, B. Font, L. M. García-Cuevas, O. Lehmkühl, and R. Vinuesa, “Deep reinforcement learning for flow control exploits different physics for increasing Reynolds number regimes,” in *Actuators* (Multidisciplinary Digital Publishing Institute, 2022), Vol. 11, p. 359.
- ⁶²L. Guastoni, J. Rabault, P. Schlatter, H. Azizpour, and R. Vinuesa, “Deep reinforcement learning for turbulent drag reduction in channel flows,” *arXiv:2301.09889* (2023).
- ⁶³G. Y. C. Maceda, F. Lusseyran, and B. R. Noack, “xMLC—A toolkit for machine learning control,” *arXiv:2208.13172* (2022).
- ⁶⁴G. D. Weymouth and D. K. Yue, “Boundary data immersion method for Cartesian-grid simulations of fluid-body interaction problems,” *J. Comput. Phys.* **230**, 6233–6247 (2011).
- ⁶⁵S. C. Schlanderer, G. D. Weymouth, and R. D. Sandberg, “The boundary data immersion method for compressible flows with application to aeroacoustics,” *J. Comput. Phys.* **333**, 440–461 (2017).
- ⁶⁶A. Li, S. Shi, and D. Fan, “Fluid forces and vortex patterns of an oscillating cylinder pair in still water with both fixed side-by-side and tandem configurations,” *J. Offshore Mech. Arct. Eng.* **144**, 021903 (2022).
- ⁶⁷S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *International Conference on Machine Learning* (PMLR, 2018), pp. 1587–1596.
- ⁶⁸W.-Y. Loh, “Classification and regression trees,” *Wiley Interdiscip. Rev.: Data Min. Knowl. Discovery* **1**, 14–23 (2011).
- ⁶⁹R. H. Alsagheer, A. F. Alharan, and A. S. Al-Haboobi, “Popular decision tree algorithms of data mining techniques: A review,” *Int. J. Comput. Sci. Mobile Comput.* **6**, 133–142 (2017).
- ⁷⁰D. Gunning, M. Stefk, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, “XAI—Explainable artificial intelligence,” *Sci. Robotics* **4**, eaay7120 (2019).
- ⁷¹B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, “Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model,” *Complexity* **2021**, 6634811.
- ⁷²O. Bastani, Y. Pu, and A. Solar-Lezama, “Verifiable reinforcement learning via policy extraction,” in *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, 2018*.
- ⁷³E. Puiutta and E. Veith, “Explainable reinforcement learning: A survey,” in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (Springer, 2020), pp. 77–95.
- ⁷⁴G. A. Vouros, “Explainable deep reinforcement learning: State of the art and challenges,” *ACM Comput. Surv.* **55**, 1 (2023).
- ⁷⁵Dataset: H. D. Feng, (2023). “How-to-control-hydrodynamic-force-on-fluidic-pinball-via-deep-reinforcement-learning,” Github, <https://github.com/HDFengChina/How-to-Control-Hydrodynamic-Force>.