Physics of Fluids

RESEARCH ARTICLE | NOVEMBER 03 2022

Estimating forces from cross-sectional data in the wake of flows past a plate using theoretical and data-driven models

Wenwen Tong (童文文) ; Shizhao Wang (王士召) ; Yue Yang (杨越) ✉

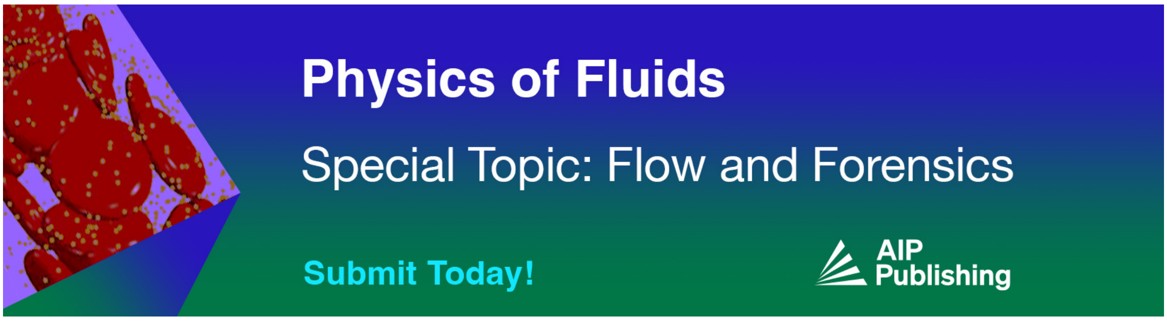
Check for updates

Physics of Fluids 34, 111905 (2022)
https://doi.org/10.1063/5.0125374

CrossMark

View Online

Export Citation



Estimating forces from cross-sectional data in the wake of flows past a plate using theoretical and data-driven models

Cite as: Phys. Fluids **34**, 111905 (2022); doi: 10.1063/5.0125374
Submitted: 11 September 2022 · Accepted: 11 October 2022 ·
Published Online: 3 November 2022

Wenwen Tong (童文文),[1] Shizhao Wang (王士召),[2] and Yue Yang (杨越)[1,3,a)]

AFFILIATIONS

[1] State Key Laboratory for Turbulence and Complex Systems, College of Engineering, Peking University, Beijing 100871, China
[2] LNM, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China and School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[3] HEPDS-CAPT, Peking University, Beijing 100871, China

[a)] Author to whom correspondence should be addressed: yyg@pku.edu.cn

ABSTRACT

We report a comparative study of theoretical and data-driven models for estimating forces from velocity data in the wake of three-dimensional flows past a plate. The datasets with a range of angles of attack are calculated using the immersed boundary method. First, we develop a theoretical model to estimate forces on a flat plate from cross-sectional velocity data in the far wake. This algebraic model incorporates the local momentum deficit and pressure variation. Second, we develop several data-driven models based on the convolutional neural network (CNN) for force estimation by regarding the velocity field on a series of cross sections as images. In particular, we design three CNN architectures for integrating physical information or attention mechanism, and use different training datasets for interpolation and extrapolation tasks. The model performances indicate that the optimized CNN can identify important flow regions and learn empirical physical laws. The theoretical and CNN models are assessed by multiple criteria. In general, both models are accurate (with errors less than 10%), robust, and applicable to complex wake flows. The theoretical model is superior to the CNN model in terms of the completeness, cost, and interpretability, and the CNN model with the appropriate training data and optimized CNN architecture has better description and accuracy.

I. INTRODUCTION

The wake of biological locomotion contains rich physical information of a moving body.[1,2] The velocity field in the far wake, similar to the footprint, can be measured to infer aerodynamic forces exerted on the moving body.[3,4] Moreover, the investigation of the wake structure can guide the design of underwater vehicles, including the shape optimization and performance improvement of the moving body with machine learning.[5,6]

In recent decades, various theoretical models have been developed to estimate forces from the wake, revealing the relationship between the force and local flow field.[7–9] In particular, the rapid development of machine learning and deep learning showed some promising performance in flow analysis,[10–13] providing new insight into data-driven models for the force estimation.[14–16]

Although both theoretical and data-driven models are useful to predict forces of a moving body, they are usually investigated separately. A comprehensive assessment of two types of models is necessary in the practical application.[17,18] The flow past a plate is an effective model problem in experimental and numerical studies.[19–23] Thus, we develop both theoretical and data-driven models based on cross-sectional data in the wake of a plate, and then analyze strengths and weaknesses of these two models via a comparative study.

Estimating forces, such as lift and drag, from wake flow structures is useful for experiments and practical applications by avoiding the near-wall measurement with complex boundary conditions of a moving body. The flow field in the wake can be obtained using the particle image velocimetry (PIV)[24–28] and other measurement techniques. The shedding three-dimensional vortical structure can be utilized to estimate forces of the moving body by aerodynamic models based on local velocity and vorticity fields.[17,29,30] For the wake with discrete vortices, a three-dimensional vortical structure can be simplified as a vortex ring to calculate force from its vortical impulse.[31] The vortex-force theory and impulse theory reveal underlying physics between the force and vortical structure.[8,32] Kang *et al.*[9] proposed the finite-domain impulse theory to estimate the unsteady aerodynamic force of the body. Tong *et al.*[22] developed the impulse theory based a discrete vortex surface in the wake to estimate the thrust of a flapping plate. Nonetheless, it is difficult to develop a simple model for predicting aerodynamic forces based on three-dimensional vortical structures with complex topology in real wake flows,[17] which is an obstacle for applying the structure-based models to practical problems.

In comparison, estimating lift and drag from the flow field measured in a cross section in the wake is much more convenient than extracting the flow field within vortical structures in applications. First, the Kutta–Joukowski (KJ) theorem was applied to estimate lift of a moving body with the circulation calculated in the wake section,[33,34] but the direct application of the two-dimensional KJ theorem can underestimate the lift in three-dimensional flows.[7,34] Wang *et al.*[3] proposed the wake-sectional KJ model to predict the time-averaged lift of a flapping plate in three-dimensional flows. Second, the drag can be modeled via the momentum change in two-dimensional flows. Spedding and Hedenström[35] estimated the drag based on the velocity data in a wake section in the experiment of a two-dimensional steady flow past a flat plate. Olasek and Karczewski[4] proposed a pressure reconstruction based on the velocity data to reduce the error of drag estimation for a two-dimensional airfoil. However, it is not straightforward to extend the two-dimensional models to three-dimensional flows due to complex vortical structures.

Compared to theoretical models, various data-driven models[36,37] emerge recently by applying machine learning to flow analysis.[11] The supervised learning has been applied to a range of flow problems with neural network.[37,38] The unsupervised learning, for example, generative adversarial network,[39] can be applied to generate and reconstruct the flow field.[40] In particular, the convolutional neural network (CNN) shows remarkable performance in image recognition,[41–44] so regarding the flow field as images is a natural choice in data-driven models and flow analysis. The CNN has been applied to flow problems such as the prediction of turbulent heat transfer and velocity fluctuation,[45,46] the super-resolution reconstruction from coarse flow data,[47] and force estimations for two-dimensional flows past a plate, airfoil, or cylinder.[15,48–50]

To improve the performance of data-driven models, physical information can be introduced into the neural network, such as the Mach number and the Reynolds number.[48,49] In addition, the attention mechanism has been proven effective in image recognition by focusing on the effective region of images.[51–53]

On the other hand, there still exists several challenges, for example, the physical interpretability[54,55] and generalization ability,[50] for data-driven modeling in fluid dynamics. Jagodinski *et al.*[56] demonstrated that the CNN is capable of uncovering the dynamically critical region for predicting the intensity of ejections in a turbulent flow, but the underlying physical laws remain unknown. The generalization ability of supervised machine learning is affected by the disparity of training and test data,[18,50,57] causing the questionable performance for data extrapolation tasks. Hasegawa *et al.*[58] investigated the two-dimensional flows around bluff bodies with various shapes using CNN. They found that the neural network can predict unlearned flow data given a proper training dataset, but the model adaption for different flow configurations remains an open problem.

Therefore, although both theoretical and data-driven models have moderate applications in fluid dynamics, significant challenges still exist in practical problems. In the present study, we develop both theoretical and data-driven models to estimate aerodynamic forces of a moving body from the cross-sectional velocity data in the wake. Then, we assess and compare two models with multiple criteria. The comprehensive comparative study can shed light on the development and selection of the most suitable model in flow applications.

The outline of this paper is as follows: In Sec. II, we describe the numerical implementation for obtaining the velocity data in flows past a plate. In Sec. III, we develop theoretical models based on velocity data within an effective region in the wake section. In Sec. IV, we develop several data-driven models based on different CNN architectures. In Sec. V, we compare the theoretical and CNN models under various criteria. Some conclusions are drawn in Sec. VI.

II. SIMULATION OVERVIEW

A. Plate kinematics

We carried out the direct numerical simulation (DNS) of three-dimensional flows past a finite plate with the free-stream velocity U. The plate has the chord length c and the fixed wing span $b = R_{Ac}$ with the aspect ratio R_A. The Reynolds number is defined by $Re = Uc/\nu$ with the kinematic viscosity ν. Figure 1 sketches the plate and the Cartesian coordinate system with the streamwise x direction, spanwise y direction, and vertical z direction. All variables and parameters in the present study are non-dimensionalized by U and c.

We focus on the modeling of flows past a stationary plate below, which is further extended to flows past a flapping plate and a butterfly-like elliptical plate in Appendixes A and B, respectively. Motivated by the investigation of flows past low-aspect-ratio wings,[8,19,20,22,59–61] we set $Re = 300$ and $R_A = 2$ with a range of angles of attack $\alpha = \{5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ\ 40^\circ\}$ as listed in Table I. The flow is steady for small α and has separation for $\alpha \geq 25^\circ$ with periodically shedding leading-edge vortices.

B. Immersed boundary method

The incompressible flow past a plate is governed by the Navier–Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{u} + \boldsymbol{f}, \tag{1}$$

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

where $\boldsymbol{u}$, p, and $\boldsymbol{f}$ denote the nondimensional velocity, pressure, and external body force, respectively. The immersed boundary method with the discrete stream function[62] is applied to solve Eqs. (1) and (2).

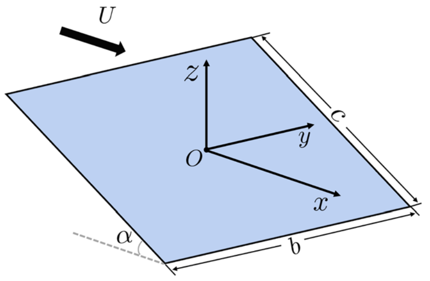


FIG. 1. Schematic of the flow past a plate with the coordinate system *O-xyz*.

TABLE I. Parameters for the DNS of flows past a stationary plate. In the row of "Separation," "N" denotes the steady flow at small α and "Y" the separation flow at large α.

Case	1	2	3	4	5	6	7	8
α (°)	5	10	15	20	25	30	35	40
Separation	N	N	N	N	Y	Y	Y	Y

The plate is described by Lagrangian marker points uniformly distributed on the immersed boundary. The regularized delta function δ_h[63] is used to interpolate and spread forces on Eulerian and Lagrangian points as

$$\sum_{j=1}^{M}\left(\sum_{\boldsymbol{x}}\delta_h(\boldsymbol{x}-\boldsymbol{X}_j)\delta_h(\boldsymbol{x}-\boldsymbol{X}_k)(\Delta s)^2(\Delta x)^3\right)\boldsymbol{F}_L(\boldsymbol{X}_j) = \frac{\boldsymbol{U}_b(\boldsymbol{X}_k)-\boldsymbol{U}^*(\boldsymbol{X}_k)}{\Delta t}, \tag{3}$$

and

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{M}\boldsymbol{F}_L(\boldsymbol{X}_j)\delta_h(\boldsymbol{x}-\boldsymbol{X}_j)(\Delta s)^2, \tag{4}$$

where $\boldsymbol{x}$ and $\boldsymbol{X}$ are Eulerian and Lagrangian points, respectively; Δs and Δx are Lagrangian and Eulerian grid spacings, respectively; $\boldsymbol{f}$ and $\boldsymbol{F}_L$ are the forces on Eulerian and Lagrangian points, respectively; $\boldsymbol{U}_b$ and $\boldsymbol{U}^*$ are specified and predicted velocities at k-th Lagrangian points, respectively; and M is the total number of Lagrangian points on the immersed boundary. The implementation detail and validation of the immersed boundary method can be found in Refs. 60 and 62.

The streamwise force coefficient of the plate is

$$C_d = \frac{F_d}{\frac{1}{2}\rho U^2 R_A c^2}, \tag{5}$$

where F_d is the total streamwise force exerted on the plate, and ρ is the fluid density. From the DNS data, it is calculated by

$$C_d^{DNS} = \frac{\sum_{k=1}^{M} F_{L,x}(\boldsymbol{X}_k)(\Delta s)^2}{\frac{1}{2}\rho U^2 R_A c^2}, \tag{6}$$

where $F_{L,x}$ denotes the x component of the force on Lagrangian points.

The present simulation was conducted in a rectangular domain of $\Omega \in [-14, 22] \times [-18, 18] \times [-18, 18]$. The uniform inflow velocity is applied at the inlet, and the fixed pressure condition is prescribed at the outlet. The slip-wall condition is specified at the other four boundaries. The no-slip condition is used on the plate. The initial condition is $\boldsymbol{u} = (U, 0, 0)$ in the entire domain. To achieve high spatial resolution near the plate, a locally refined mesh with the minimum spacing $\Delta x = 0.0125$ around the immersed boundary and the maximum spacing $\Delta x = 0.4$ is applied in the far field. The total number of grid points is 12×10^6. The effectiveness of the present computational domain and mesh spacing has been validated by the convergence test in our previous studies[22,61].

III. ESTIMATING FORCES FROM THEORETICAL MODELS

A. Modeling based on surface integrals

The total force exerted on the plate is determined by the integral of stresses[32] as

$$\boldsymbol{F} = -\int_{\partial B}(-p\boldsymbol{n}+\boldsymbol{\tau})\mathrm{d}S, \tag{7}$$

where ∂B denotes the plate surface in Fig. 2 and

$$\boldsymbol{\tau} = \mu\boldsymbol{\omega}\times\boldsymbol{n} \tag{8}$$

denotes the shear stress with the dynamic viscosity μ, vorticity $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$, and the unit surface normal $\boldsymbol{n}$ of the plate.

The fluid domain V_f around the plate ∂B is bounded by the outer surface

$$\Sigma = \sum_{i=1}^{6} S_i \tag{9}$$

composed of six planes, where S_1 and S_2 are upstream and downstream surfaces, respectively, and $S_3 \sim S_6$ are the other four surfaces of V_f in Fig. 2. The momentum balance of the fluid can be written as

$$\rho\boldsymbol{a} = -\nabla p + \mu\nabla^2\boldsymbol{u}, \tag{10}$$

where $\boldsymbol{a} = \mathrm{D}\boldsymbol{u}/\mathrm{D}t$ is the acceleration of the fluid, and the balance in V_f has

$$\int_{V_f}\rho\boldsymbol{a}\mathrm{d}V = \int_{V_f}(-\nabla p + \mu\nabla^2\boldsymbol{u})\mathrm{d}V. \tag{11}$$

Applying $\nabla^2\boldsymbol{u} = \nabla(\nabla\cdot\boldsymbol{u}) - \nabla\times\boldsymbol{\omega}$ with Eq. (2) and the divergence theorem to the right-hand side of Eq. (11) yields

$$\int_{V_f}\rho\boldsymbol{a}\mathrm{d}V = \int_{\partial B+\Sigma}(-p\boldsymbol{n} - \mu\boldsymbol{n}\times\boldsymbol{\omega})\mathrm{d}S, \tag{12}$$

where ∂B and Σ are the inner and outer surfaces of V_f, respectively.

Substituting Eq. (12) with Eq. (8) into Eq. (7), we can calculate the force on the plate by

$$\boldsymbol{F} = -\int_{V_f}\rho\boldsymbol{a}\mathrm{d}V + \int_{\Sigma}(-p\boldsymbol{n}+\boldsymbol{\tau})\mathrm{d}S. \tag{13}$$

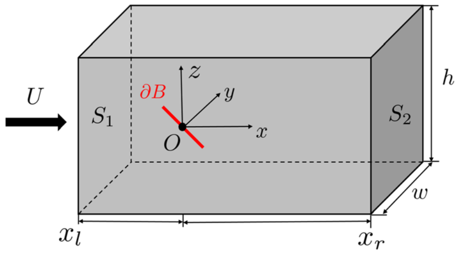


FIG. 2. The rectangular computational domain for the flow past a plate with six surfaces, where S_1 is the upstream surface at $x = x_l$ and S_2 is the downstream surface at $x = x_r$ with the height h and width w. The red line denotes the plate bounded by the surface ∂B.

The viscous term $\boldsymbol{\tau}$ in Eq. (13) can be neglected for moderate and high *Re*. In the present simulation with $Re = 300$, the contribution of $\boldsymbol{\tau}$ on the total force is approximately 0.6%. Substituting the decomposition of the fluid acceleration

$$\boldsymbol{a} = \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) - \boldsymbol{u}(\nabla \cdot \boldsymbol{u}) \tag{14}$$

with Eq. (2) into Eq. (13), we obtain the decomposition of the total force on the plate

$$\boldsymbol{F} = \boldsymbol{F}_u + \boldsymbol{F}_p + \boldsymbol{F}_m + \boldsymbol{F}_B, \tag{15}$$

where

$$\boldsymbol{F}_u = -\frac{\partial}{\partial t}\int_{V_f} \rho \boldsymbol{u} \mathrm{d}V \tag{16}$$

denotes the force contributed by the unsteady term,

$$\boldsymbol{F}_p = \int_{\Sigma} (-p\boldsymbol{n}) \mathrm{d}S \tag{17}$$

denotes the force contributed by the pressure,

$$\boldsymbol{F}_m = \int_{\Sigma} (-\rho \boldsymbol{n} \cdot \boldsymbol{u}\boldsymbol{u}) \mathrm{d}S \tag{18}$$

denotes the force contributed by the change of the fluid momentum, and

$$\boldsymbol{F}_B = \int_{\partial B} (-\rho \boldsymbol{n} \cdot \boldsymbol{u}\boldsymbol{u}) \mathrm{d}S \tag{19}$$

denotes the force contributed by the velocity on the plate.

In the present cases, the unsteady term $\boldsymbol{F}_u$ in Eq. (15) is neglected for a steady flow and for estimating time-averaged forces in flows with periodically shedding vortices in V_f. Moreover, we have $\boldsymbol{F}_B = 0$ for the stationary plate due to $\boldsymbol{u} = 0$ on ∂B and also have $\boldsymbol{F}_B = 0$ for the flapping plate and butterfly-like elliptical plate in Appendixes A and B due to the opposite $\boldsymbol{n}$ on the top and bottom of the plate. Thus, we approximate the streamwise force

$$F_x = \boldsymbol{e}_x \cdot \int_{S_1+S_2} (-p\boldsymbol{n}) \mathrm{d}S + \boldsymbol{e}_x \cdot \int_{\Sigma} (-\rho \boldsymbol{n} \cdot \boldsymbol{u}\boldsymbol{u}) \mathrm{d}S, \tag{20}$$

with the unit vector $\boldsymbol{e}_x$ in the x direction.

B. Modeling of the pressure contribution

In former experimental studies, only the momentum integral $\boldsymbol{F}_m$ was used to calculate drag for a flow past an airfoil,[4,35] because the pressure field is difficult to obtain from the experiential velocity data. Thus, $\boldsymbol{F}_p$ in Eq. (15) is usually neglected for steady flows past a two-dimensional airfoil with small α, for which $\boldsymbol{F}_m$ can approximate drag force well in the far wake.[35] The pressure reconstruction was introduced to further reduce the drag estimation error in the experiment,[4] and the maximum discrepancy is about 20%. For two-dimensional flows past an elliptical cylinder, the pressure was calculated to improve the estimation of the mean drag at large *Re*.[64] However, we find that the pressure contribution cannot be neglected for three-dimensional unsteady flows in the present simulation, and it needs to be calculated by the cross-sectional velocity field in our force modeling.

For the practical interest, it is crucial to choose an appropriate Σ to calculate F_x in Eq. (20). Considering a sufficiently large Σ, the velocity on surfaces $S_3 \sim S_6$ is close to the uniform inflow so that the surface integral on these surfaces are neglected due to $\boldsymbol{n} \cdot \boldsymbol{u} \simeq 0$. The flow is also approximated to be uniform on S_1 with small x_l in Fig. 2; then, Eq. (20) becomes

$$F_x = \int_{S_2} \rho(U^2 - u_x^2)\mathrm{d}S + \int_{S_1} p\mathrm{d}S + \int_{S_2} (-p)\mathrm{d}S, \tag{21}$$

where u_x is the streamwise velocity.

The pressure contribution in Eq. (21) is modeled based on the cross-sectional velocity data via Bernoulli's equation[32]

$$p + \frac{1}{2}\rho|\boldsymbol{u}|^2 = C, \tag{22}$$

where C is a constant for steady and inviscid potential flow. The value of C has no effect on F_x in Eq. (21) for equal areas of S_1 and S_2, so we take $C = 0$ for simplicity. Thus, Eq. (21) is approximated by

$$F_x = \int_{S_2} \rho(U^2 - u_x^2)\mathrm{d}S + \frac{1}{2}\int_{S_2} \rho(|\boldsymbol{u}|^2 - U^2)\mathrm{d}S, \tag{23}$$

which is only determined by the velocity field in a wake cross section S_2. Note that Eq. (22) can have a notable discrepancy in the near wake of the plate with strong vortical flows.

C. Selection of the effective region

Figure 3(a) illustrates the modeling process using the flow past a stationary plate with $Re = 300$, $R_A = 2$, and $\alpha = 30°$. In this flow, vortices, identified by the Q-criterion,[65] periodically shed off from the plate into the wake. In Fig. 3(b), the cross-sectional contour of u_x at $x = 10$ in the far wake is dominated by the velocity defect profile in the central region, which is referred to as the effective region S_e for the major contribution to F_x. In Fig. 3(c), the flow is almost uniform at $|z| > 4$.

According to Eq. (23), the large velocity defect in the effective region determines F_x. Thus, we obtain a theoretical model to estimate F_x based on velocity data in a wake cross section at x as

$$F_x(x) = \int_{S_e} \rho(U^2 - u_x^2)\mathrm{d}S + \frac{1}{2}\int_{S_e} \rho(|\boldsymbol{u}|^2 - U^2)\mathrm{d}S, \tag{24}$$

where we define

$$S_e(x) = \{(y(x), z(x)) | u_x(y(x), z(x)) < u_{th}\}, \tag{25}$$

with a threshold $u_{th} < U$. The estimation of Eq. (24) depends on the choice of S_e controlled by the value of u_{th}, and we set $u_{th} = 0.99$ similar to the definition of the boundary layer thickness with 99% of the free-stream velocity.

The theoretical model in Eq. (24) estimates the streamwise force coefficient as

$$C_d^M(\boldsymbol{u}, u_{th}, x) = C_m(\boldsymbol{u}, u_{th}, x) + C_p(\boldsymbol{u}, u_{th}, x), \tag{26}$$

where the contribution from the momentum deficit is

$$C_m = \frac{\displaystyle\int_{S_e} \rho(U^2 - u_x^2)\mathrm{d}S}{\displaystyle\frac{1}{2}\rho U^2 R_A c^2}, \tag{27}$$

and the contribution from the pressure variation is

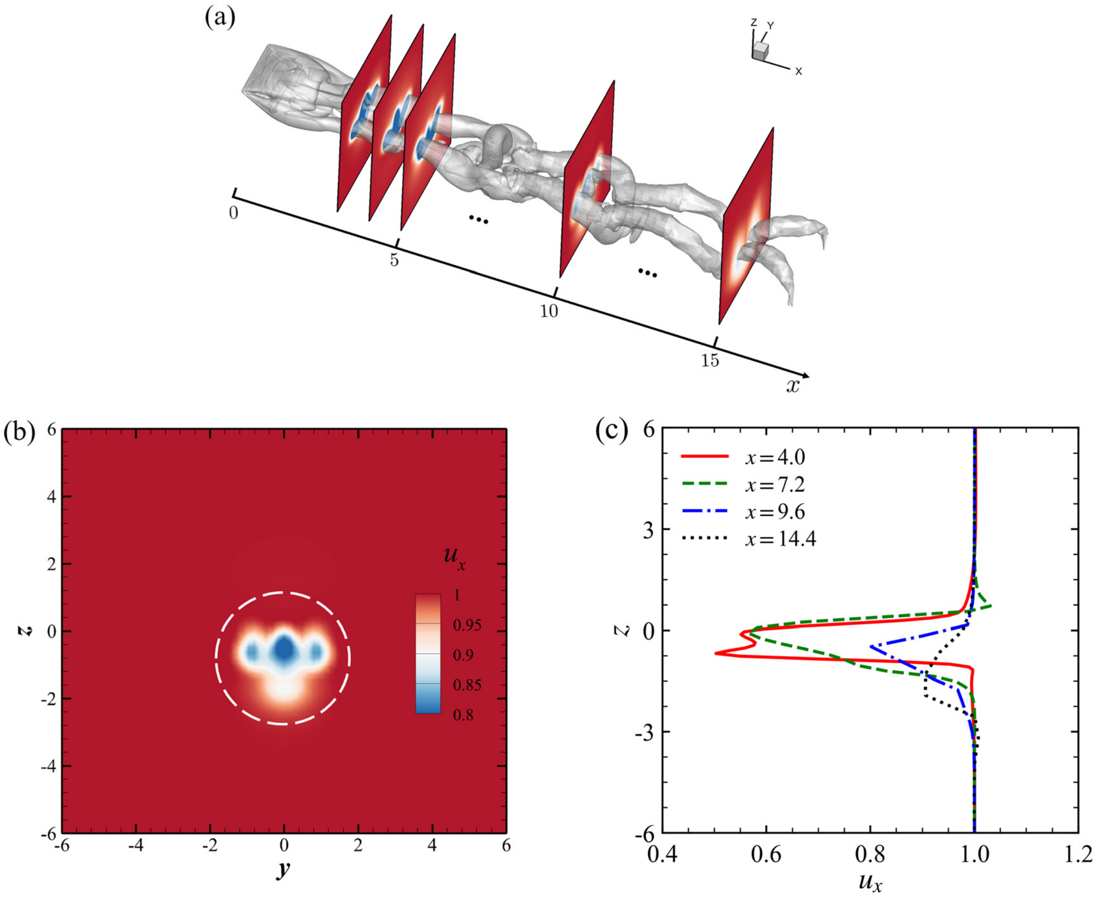


FIG. 3. Wake structures in the flow past a stationary plate with $Re = 300$, $R_A = 2$, and $\alpha = 30°$ at $t = 35$. (a) Isosurface of $Q = 0.02$ and contours of u_x on a series of cross sections along x. (b) Contour of u_x on the cross section at $x = 10$. (c) Profiles of u_x along the z direction at $y = 0$ at different x.

$$C_p = \frac{\frac{1}{2}\int_{S_e}\rho(|\boldsymbol{u}|^2 - U^2)\mathrm{d}S}{\frac{1}{2}\rho U^2 R_A c^2}. \tag{28}$$

For a two-dimensional flow, C_m can contribute over 80% of the drag force, and $|C_p/C_m|$ is usually less than 10%.[4] For a three-dimensional flow, Eq. (28) with $|\boldsymbol{u}|^2 = u_x^2 + u_y^2 + u_z^2$ suggests $C_d^M = C_m/2$ and $C_p = -C_m/2$ for $u_y = u_z = 0$, indicating the significance of C_p in force estimation for three-dimensional wake flows.

D. Assessment of theoretical models

Estimating forces on a moving body from its far wake has significant practical interest in experimental studies and engineering applications.[7,34,35] First, we split the whole wake into the near wake ($x < 5$), far wake ($5 \leq x \leq 15$), and very far wake ($x > 15$) in the present simulation. Figure 3(a) shows that the distribution of cross-sectional velocity data varies with x. Since Bernoulli's equation can break down in the near wake with small x, and the velocity can be largely dissipated in the very far wake with large x, we estimate F_x based on the cross-sectional velocity data using Eq. (26) in the far wake at $5 \leq x \leq 15$.

As depicted in Figs. 3 and 4, the complexity of wake structures grows with α for the flow past a stationary plate. For the present cases with $\alpha \leq 20°$ in Table I, we calculate F_x based on the velocity field at a sufficiently large time $t = 35$ when the flow has reached the steady state. For the cases with $\alpha \geq 25°$, we approximate the time-averaged force coefficient

$$\overline{C_d^M} = \frac{1}{T}\int C_d^M(\boldsymbol{u}(t))dt \simeq C_d^M(\bar{\boldsymbol{u}}) \tag{29}$$

based on instantaneous flow fields during a period T with periodical vortex shedding by a model

$$\bar{\boldsymbol{u}} = \frac{1}{T}\int_t^{t+T}\boldsymbol{u}(s)ds \tag{30}$$

based on the time-averaged velocity. We set $T \simeq 10$ in Eq. (29) based on the statistical stationary period of C_d^{DNS} in the numerical simulation. It is found that the discrepancy of $C_d^M(\bar{\boldsymbol{u}})$ from $\overline{C_d^M}$ is less than 5%, so the approximation Eq. (29) is used for the periodical flows and the overline of $\overline{C_d^M}$ is omitted below for clarity.

The theoretical model in Eq. (26) is assessed by the relative error

$$\varepsilon(C_d^M(x)) = \frac{|C_d^M(x) - C_d^{DNS}|}{C_d^{DNS}} \tag{31}$$

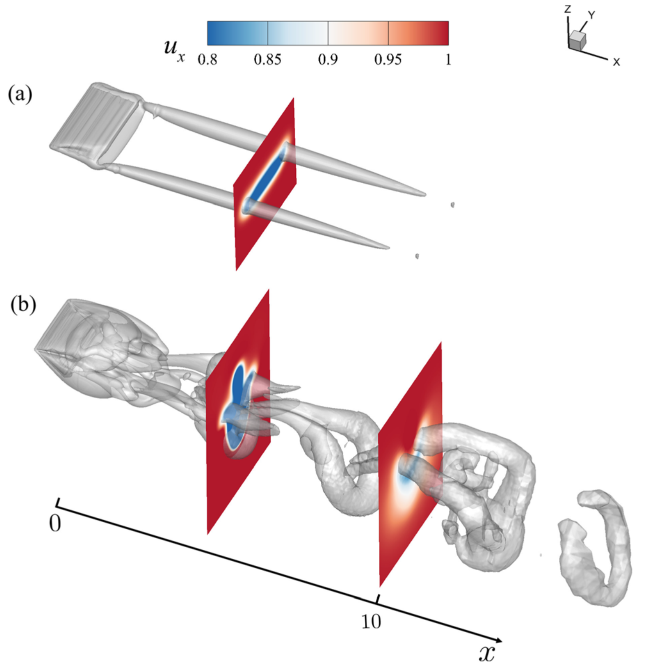


FIG. 4. Isosurfaces of $Q = 0.02$ and contours of u_x on y–z planes in the flow past a stationary plate with $Re = 300$, $R_A = 2$, and (a) $\alpha = 5°$ and (b) $\alpha = 40°$.

between C_d^M from the model prediction and C_d^{DNS} from DNS at x in the wake, and the effect of pressure modification C_p in the model Eq. (26) is evaluated. Figure 5 plots the estimations from Eq. (26) in the far wake for the flow past a stationary plate with $\alpha = 20°$ and $\alpha = 30°$. The model Eq. (26) with the pressure modification predicts C_d well in the far wake at $5 \le x \le 15$ with $\varepsilon(C_d^M(x))$ generally less than 10%. By contrast, the model in Eq. (27) without the pressure contribution significantly overpredicts C_d. Note that Bernoulli's equation in Eq. (22) is not well satisfied in the near wake due to strong vortical flows and in the very far wake due to the large viscous dissipation, so $\varepsilon(C_d^M(x))$ is large for $x < 5$ and it grows with x for $x > 15$ (not shown).

The spatially mean error $\bar{\varepsilon}$ for force estimation in the far wake is calculated by averaging $\varepsilon(C_d^M(x))$ over 50 equally distributed cross sections within $5 \le x \le 15$. Table II lists $\bar{\varepsilon}$ for estimating C_d from Eqs. (27) and (26) in the far wake. The mean error is less than 10% for $\alpha \le 25°$ from the model in Eq. (26), and it generally grows with α due to the increasing wake complexity. By contrast, estimating forces only from the momentum deficit in Eq. (27) has large $\bar{\varepsilon} = 77.4\% \sim 100.5\%$. Table II shows that the maximum error from the model in Eq. (26) can be well controlled in 25% in the wake within $5 \le x \le 15$ for all eight cases. The maximum error is located close to the two ends $x = 5$ and $x = 15$ due to the breakdown of Bernoulli's equation. Hence, the incorporation of the pressure contribution Eq. (28) in the theoretical model is important for three-dimensional wake flows. In addition, we can estimate forces only from u_x in the cross section by setting $u_y = u_z = 0$ in the theoretical model so that our model can be more applicable for practical applications. The relative error for the model dependent on u_x only slightly grows by 2% for the cases of the stationary plate.

IV. ESTIMATING FORCES FROM DATA-DRIVEN MODELS

A. Building blocks of CNN

Compared with the theoretical model based on underlying physics, the data-driven model has more flexibility via automatically extracting important information from input data. The CNN, for example, the AlexNet,[42] VGGNet,[43] and ResNet,[44] is widely used in classification and regression problems. In particular, the CNN shows remarkable performance in image recognition by efficiently processing red-green-blue (RGB) images and learning latent image features. The velocity field in a cross section in the present simulation can be regarded as a RGB image, so the CNN can be effective to predict forces, similar to extracting image features, based on velocity data.

A CNN consists of convolutional layers, pooling layers, fully connected layers, and activation functions. The convolutional layer is the core of CNN for extracting spatial features.[41] The data in the convolutional layer are expressed in the form of (C, H, and W), where C

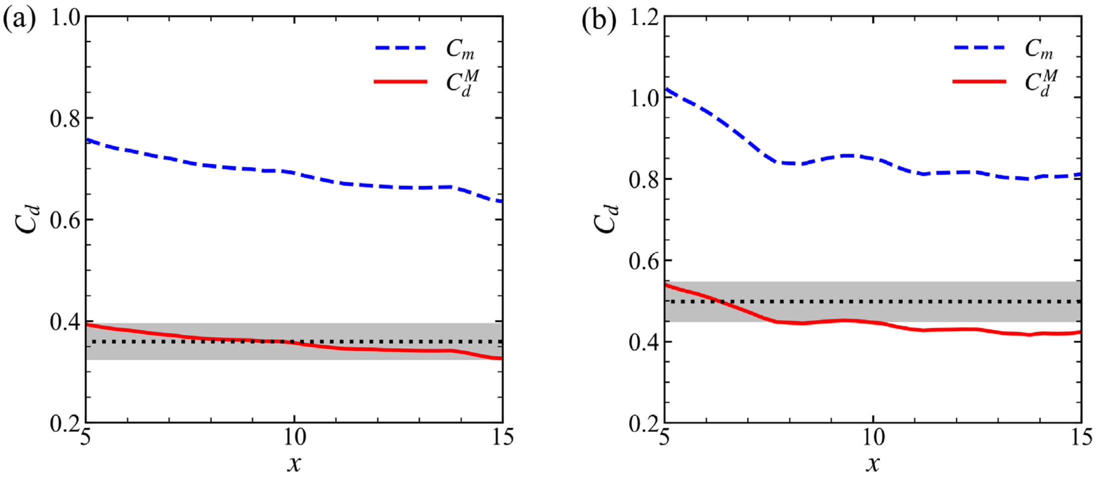


FIG. 5. Comparison of theoretical models of C_d^M in Eq. (26) and C_m in Eq. (27) based on wake cross-sectional velocity field along x with (a) $\alpha = 20°$ and (b) $\alpha = 30°$. The gray shade denotes 10% relative error. The dotted line denotes C_d^{DNS}.

TABLE II. Relative errors of theoretical models of C_d^M in Eq. (26) and C_m in Eq. (27) for estimating forces in the far wake ($5 \leq x \leq 15$) with various α. The average of $\bar{\varepsilon}(C_d^M)$ over all cases is 6.58%.

Case	1	2	3	4	5	6	7	8
α (°)	5	10	15	20	25	30	35	40
C_d^{DNS}	0.234	0.267	0.314	0.360	0.427	0.498	0.584	0.661
$\bar{\varepsilon}(C_d^M)$	2.5%	2.3%	2.6%	4.0%	7.7%	10.5%	11.7%	11.3%
$\bar{\varepsilon}(C_m)$	100.5%	94.2%	92.6%	92.2%	80.3%	72.3%	69.0%	77.4%
$\varepsilon(C_d^M)_{\max}$	6.2%	4.3%	5.5%	8.9%	15.9%	16.3%	20.5%	25.0%
$\varepsilon(C_m)_{\max}$	110.0%	103.8%	103.4%	109.6%	105.1%	103.8%	100.4%	120.8%

denotes the number of channels, and H and W are the height and the width of data, respectively. For the network input, the velocity has three components with $C=3$. To process the input data of a convolutional layer, C kernels are utilized to extract features, generating C feature maps.

As sketched in Fig. 6, in the convolution operation $\boldsymbol{y}_{con} = \phi(\boldsymbol{x}_{con}, \boldsymbol{k}_{con})$, the output $\boldsymbol{y}_{con}$ is generated by a convolutional function ϕ of input $\boldsymbol{x}_{con}$ and kernel $\boldsymbol{k}_{con}$. Here, $\boldsymbol{x}_{con}(C_{in}, H_{in}, W_{in})$ is three-dimensional input data, with the subscript "in" for input quantities. The kernel $\boldsymbol{k}_{con}(C_{in}, K, K)$ with $C_{in} \times K \times K$ learning parameters extracts a class of features from $\boldsymbol{x}_{con}$ and generates one feature map $\boldsymbol{y}_{con}(1, H_{out}, C_{out})$, with the subscript "out" for output quantities and $K=3$ in our convolutional layers.[43,44] The value of each element in the feature map is calculated by

$$\boldsymbol{y}_{con}(1,j,k) = \sum_{c=1}^{C_{in}} \sum_{p=1}^{K} \sum_{q=1}^{K} \boldsymbol{x}_{con}(c, j+p, k+q)\boldsymbol{k}_{con}(c,p,q), \quad (32)$$

with $1 \leq j \leq H_{out}$ and $1 \leq k \leq W_{out}$. In general, the number of channels increases and the feature map size decreases after the convolution[42] with $C_{out} > C_{in}$, $H_{out} < H_{in}$, and $W_{out} < W_{in}$, and the features of input data can be effectively extracted layer by layer.

As sketched in Fig. 7, a typical building block in CNN is composed of the convolutional layer, batch normalization (BN), and activation function. The BN accelerates and stabilizes training by normalizing the layer inputs.[66] The activation function $ReLU(x) = \max(0, x)$ increases nonlinear characteristics of the neural network and avoids gradient vanishing in deep CNN.[67]

Furthermore, the residual building block[44] is adopted in the present CNN architecture. As sketched in Fig. 8, a shortcut connection is added to the simple building block in Fig. 7, with the residual operation $\boldsymbol{y}_{con} = ReLU(\mathcal{F}(\boldsymbol{x}_{con}, \boldsymbol{w}_{con}) + \boldsymbol{x}_{con})$, where $\boldsymbol{x}_{con}$, $\boldsymbol{y}_{con}$, and $\boldsymbol{w}_{con}$ are the input tensor, output tensor, and weight parameter of the block, respectively. The residual operation can facilitate optimizing the network and improving network performance with the depth.

In the CNN, the pooling layer for downsampling reduces the size of feature maps and further extracts information through maximum or average pooling. The fully connected layer in the last layer of CNN obtains the prediction result.[43,44] This layer is constructed by neurons to extract data features with the linear transformation $\boldsymbol{y}_{fc} = \boldsymbol{w}_{fc}^T \boldsymbol{x}_{fc} + \boldsymbol{b}_{fc}$, where $\boldsymbol{x}_{fc}$, $\boldsymbol{y}_{fc}$, $\boldsymbol{w}_{fc}$, and $\boldsymbol{b}_{fc}$ are the input data, output data, weight, and bias, respectively. The weight $\boldsymbol{w}_{fc}$ has shape (N_{in}, N_{out}), and the bias $\boldsymbol{b}_{fc}$ has shape (N_{out}),[43] where N_{in} and N_{out} denote the number of neurons in the input and output data, respectively. Both learning parameters $\boldsymbol{w}_{fc}$ and $\boldsymbol{b}_{fc}$ in the network are updated in network training.

B. CNN architectures

The present main CNN architecture is adapted from the ResNet18, a typical setup of ResNet.[44] Other typical CNN architectures, for example, VGGNet[43] and MobileNet,[68] are also tested. We find that the force prediction is not very sensitive to CNN architectures. For the scale of the current dataset, we reduce the number of convolutional layers in ResNet18 to avoid overfitting. The network architecture is presented in Fig. 9. The network performance is dependent on the input variables,[69,70] and both the velocity and vorticity are natural choices for network input from the physical point of view. In the practical application, it is easier to obtain the velocity than the vorticity, so we choose the velocity as the input. The velocity data $\boldsymbol{u}_{CNN}$ is extracted from a cross section in DNS with the spatial resolution H_0 and W_0 in the z and y directions, respectively.

The CNN has one basic block (see Fig. 7) and four residual blocks (see Fig. 8), including nine convolutional layers. Table III lists that the feature-map size of layers Conv 1 ~ 5 decreases and the number of

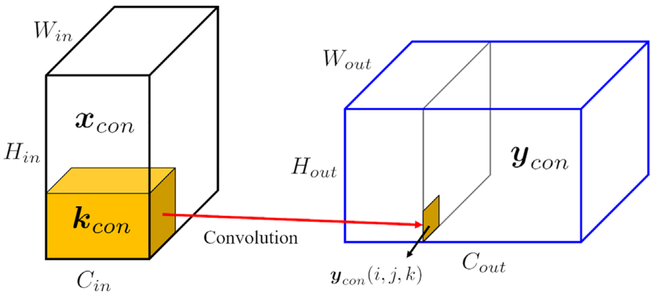


FIG. 6. Schematic of the convolution operation, including the input data $\boldsymbol{x}_{con}$ with shape (C_{in}, H_{in}, W_{in}), the filter $\boldsymbol{k}_{conv}$ with shape (C_{in}, K, K), and the output data $\boldsymbol{y}_{con}$ with shape $(C_{out}, H_{out}, W_{out})$.

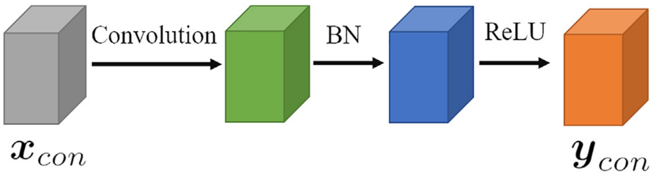


FIG. 7. Schematic of the basic building block of the CNN architecture.

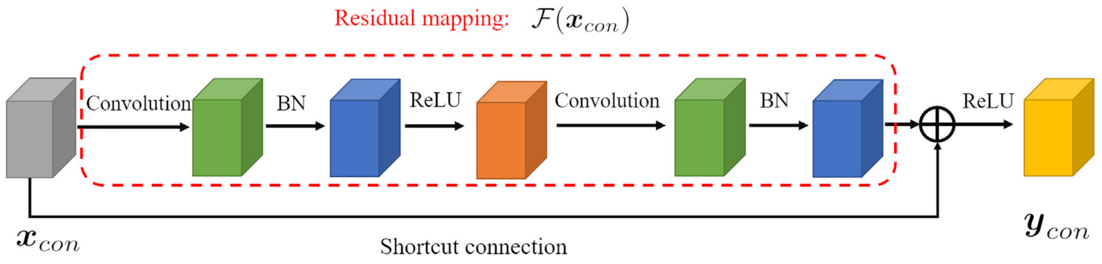


FIG. 8. Schematic of the residual building block of the CNN architecture.

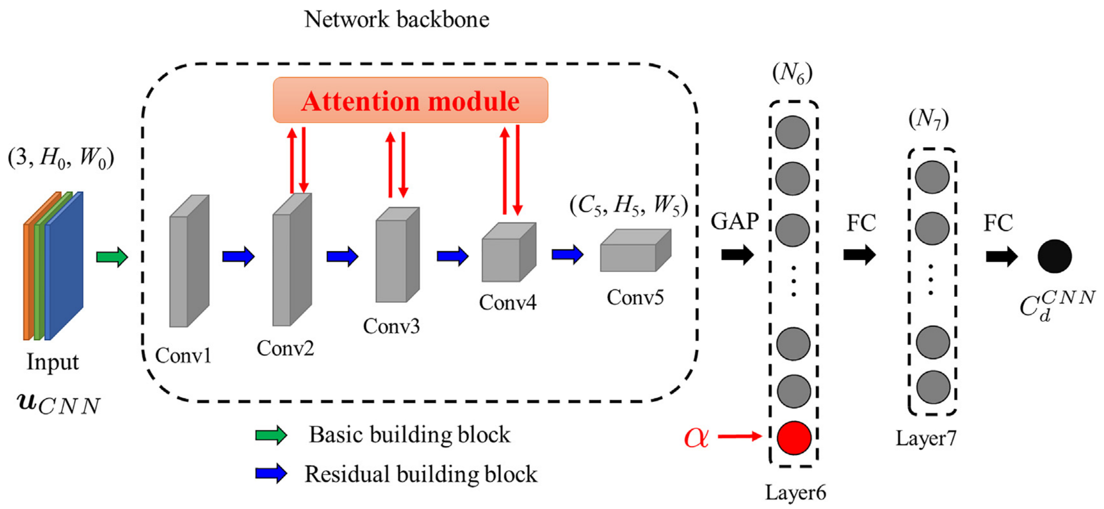


FIG. 9. Schematic of the CNN architecture. The input is three-dimensional velocity data $\boldsymbol{u}_{CNN}$ with shape $(3, H_0, W_0)$, and the output is the estimation of C_d^{CNN}. The CNN consists of one basic building block and four residual building blocks. Conv1 ∼Conv5 denote convolutional layers with the number C_5 of channels of Conv5. Layer6 and Layer7 denote fully connected layers with the numbers N_6 and N_7 of neurons. GAP denotes the global average pooling, and FC the fully connection. The physical information α and attention module (the red part) are added into the network to improve the network performance. In the baseline network, the physical information α and attention module are removed.

TABLE III. Feature-map size of the layers in Fig. 9.

Layer	Conv1	Conv2	Conv3	Conv4	Conv5
Size	$(64, H_0, W_0)$	$(64, H_0, W_0)$	$(128, H_0/2, W_0/2)$	$(256, H_0/4, W_0/4)$	$(512, H_0/8, W_0/8)$

channels increases with the network depth. For the last layer Conv5 with $C_5 = 512$, the global average pooling (GAP) is utilized to convert three-dimensional data with the shape of $(C_5,\ H_5,$ and $W_5)$ in Conv5 to one-dimensional data in Layer6 with the shape of $(N_6 = N_5)$ for the baseline network. Then, we add another fully connected layer as the middle layer to compress the data dimension and further extract features in Layer7 with $N_7 = 50$. Finally, we use the last fully connected layer to estimate the force coefficient C_d^{CNN} from the CNN model.

Being different from the image recognition, an appropriate incorporation of physical information can be crucial to improve the performance of neural networks in fluid dynamics.[48,71,72] First, we can add critical flow parameters, such as the angle of attack of the plate, into the CNN. The scalar α can be directly introduced into the fully connected layer remote from the network output,[49] for example, Layer6 with $N_6 = C_5 + 1$ in Fig. 9. Second, the force estimation in the theoretical model is mainly determined by the information in the local effective region with large velocity gradient (see Fig. 3), so the CNN model could achieve better performance by focusing on the local effective region. This inspires us to incorporate the attention module[51,73] into the CNN, which can be effective to improve the CNN performance in image classification and object detection.[52,53]

The attention mechanism can discover the crucial and irrelevant spatial regions. Figure 10 sketches the architecture of the attention

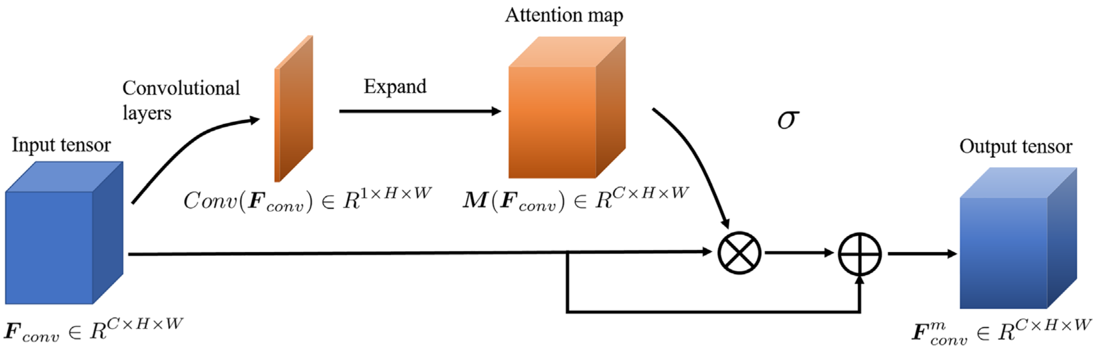

Attention map
Convolutional layers
Expand
Input tensor
σ
Output tensor
$Conv(\boldsymbol{F}_{conv}) \in R^{1\times H\times W}$
$\boldsymbol{M}(\boldsymbol{F}_{conv}) \in R^{C\times H\times W}$
$\boldsymbol{F}_{conv} \in R^{C\times H\times W}$
$\boldsymbol{F}^m_{conv} \in R^{C\times H\times W}$

FIG. 10. Schematic of the architecture of spatial attention module. Given the input tensor $\boldsymbol{F}_{conv}$, the module generates the attention map $\boldsymbol{M}(\boldsymbol{F}_{conv})$ and then obtains the output tensor $\boldsymbol{F}^m_{conv}$ via element-wise multiplication.

module. This module is inserted after the convolution layer. Given the input feature map $\boldsymbol{F}_{conv}$ in the CNN, the generated feature map

$$\boldsymbol{F}^m_{conv} = \boldsymbol{F}_{conv} + \sigma(\boldsymbol{M}(\boldsymbol{F}_{conv})) \otimes \boldsymbol{F}_{conv} \tag{33}$$

is obtained, where $\boldsymbol{M}(\boldsymbol{F}_{conv})$ is the spatial attention map generated by several convolutional layers, $\otimes$ denotes the element-wise multiplication, and $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. The network learns the spatial weight at different positions, so that it focuses on the spatial position with large weights in subsequent feature learning. As sketched in Fig. 9, we add the attention module after three convolutional layers (Conv2–Conv4) in the improved CNN architecture,[52,53] which can extract the most effective spatial information for force estimation.

The performance of CNN is related to the network input and architecture. For comparison, three types of CNN models are constructed and trained. As listed in Table IV, CNN($\boldsymbol{u}$) is a baseline CNN model, in which only the velocity $\boldsymbol{u}$ is used to train the network (see Fig. 9). In CNN($\boldsymbol{u}, \alpha$), α is an additional input to improve model performance. In CNN($\boldsymbol{u}, \alpha$, att.), the attention module in Fig. 10 is integrated with convolutional layers to further improve model performance.

C. Training CNN models

The CNN models are constructed by training DNS data via CNN. First, cross-sectional velocity data $\boldsymbol{u}_{CNN}$ are collected in the wake region at $[x_{\min}, x_{\max}]$ with an interval $d = 0.02$. For a DNS case, the number of data samples is $N_d = (x_{\max} - x_{\min})/d$, where we set $x_{\min} = 1.0$ and $x_{\max} = 15$ to cover near and far wakes with $N_d = 700$. The spatial region $S_{CNN} \in [-4, 4] \times [-6, 2]$ of $\boldsymbol{u}_{CNN}$ in the y–z plane is selected to cover the effective spatial region with large velocity gradient, as shown in Fig. 3(b). Then, $\boldsymbol{u}_{CNN}$ with the shape (201, 201) is interpolated from Ω by the Shepard method.[74] For eight DNS cases in Table I, 5600 data samples are collected.

TABLE IV. Types of CNN models.

Network type	Input	Attention module
CNN ($\boldsymbol{u}$)	$\boldsymbol{u}$	No
CNN ($\boldsymbol{u}, \alpha$)	$\boldsymbol{u}, \alpha$	No
CNN ($\boldsymbol{u}, \alpha$, att.)	$\boldsymbol{u}, \alpha$	Yes

Second, the loss function

$$\text{Loss} = \frac{1}{N_b}\sum_{i=1}^{N_b} (C^{CNN}_{d,i} - C^{DNS}_{d,i})^2 \tag{34}$$

is defined for training CNN, where N_b is the batch size of the training data. For the ith training sample, $C^{DNS}_{d,i}$ is the force coefficient from DNS, and $C^{CNN}_{d,i}$ is the predicted value from CNN.

Finally, the open-source library PyTorch[75] is applied to train CNN. The data $\boldsymbol{u}_{CNN}$ are first resized to the shape of (H_0, W_0) for the network input. In general, the performance of CNN models increases with H_0 and W_0 for $H_0 = W_0 = \{64, 128, 201, 256, 320\}$, while the computational cost also grows with the data size, so we set $(H_0, W_0) = (256, 256)$ by balancing the computational cost and prediction accuracy. The network parameters, including $\boldsymbol{k}_{conv}$ in convolutional layers and $\boldsymbol{w}_{fc}$ and $\boldsymbol{b}_{fc}$ in fully connected layers, are initialized using the robust method of.[76] The gradient descent method is used to minimize Loss in Eq. (34) and update network parameters for each iteration. The gradient of network parameters is obtained by the chain rule[38] and automatically computed in the PyTorch framework. The learning rate is 0.002 initially, and it is reduced by a factor of five in every five epochs. The total number of epochs and the batch size are set to be 25 and 32, respectively. The CNN is trained on a GPU (NVIDIA Tesla K80) in about one hour. Figure 11 plots the variation of Loss in Eq. (34) vs the epoch. The training loss decays from $O(10^{-1})$ and converges to $O(10^{-4})$, indicating effective network training.

D. Assessment of CNN models

A data-driven model is constructed by the training set and evaluated by the validation set. In general, the data-driven model can be assessed through the interpolation and extrapolation of training data.[50,57] We split the eight DNS cases in Table I into training and validation sets in terms of the angle of attack, which significantly influences the force and wake structures (see Fig. 4 and Table II). Table V lists three dataset classifications for training and assessing CNN models. In classification D_1, the angle of attack α_{val} in validation datasets is within the range of the angle of attack α_{train} in training datasets, corresponding to data interpolation. In classification D_2, α_{val} is outside of the

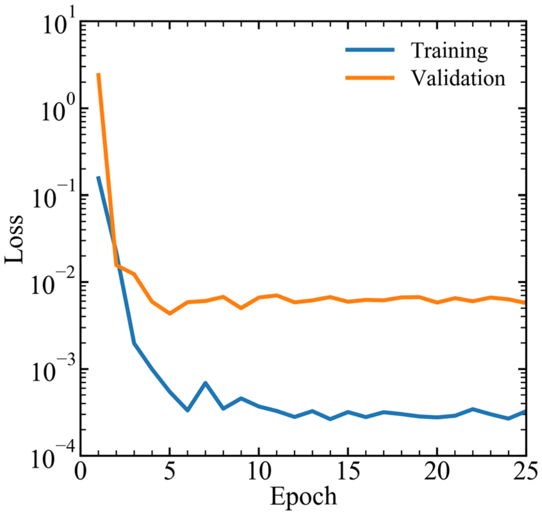


FIG. 11. Training and validation losses vs epoch in the training of model CNN($\boldsymbol{u}$) for data classification D_1.

range of α_{train}, as data extrapolation. Additionally, classification D_3 is used to study the effect of training data size on the CNN performance.

The CNN models are assessed by the relative error

$$\varepsilon(\alpha_{\text{val}}) = \frac{1}{N_{\text{val}}}\sum_{i=1}^{N_{\text{val}}}\frac{|C_{d,i}^{CNN}(\alpha=\alpha_{\text{val}}) - C_{d,i}^{DNS}(\alpha=\alpha_{\text{val}})|}{C_{d,i}^{DNS}(\alpha=\alpha_{\text{val}})}, \qquad (35)$$

over the cases with α_{val} in the validation set, where N_{val} is the number of cases with different α_{val}, $C_{d,i}^{DNS}$ is the force coefficient calculated from DNS, and $C_{d,i}^{CNN}$ is predicted from CNN based on the training set.

Based on data classification D_1, we compare model performances of CNN($\boldsymbol{u}$), CNN($\boldsymbol{u}, \alpha$), and CNN($\boldsymbol{u}, \alpha$, att.) for data interpolation. As shown in Table VI, the average error of the baseline model CNN($\boldsymbol{u}$) is 12.9%, larger than 6.58% of the theoretical model. The predictability of the model CNN($\boldsymbol{u}$) is further explained in Appendix C. By introducing α into the model CNN($\boldsymbol{u}, \alpha$), $\varepsilon(\alpha_{\text{val}})$ is significantly reduced to 4.4%, superior to the theoretical model, so the incorporation of physical information can remarkably improve the model performance.[48,49] The introduction of the attention mechanism causes CNN($\boldsymbol{u}, \alpha$, att.) to focus on the effective spatial region for force prediction, and $\varepsilon(\alpha_{\text{val}})$ is further reduced. The incorporation of physical information and attention mechanism in CNN models is further discussed in flows past a flapping plate in Appendix A.

The detailed comparison of the theoretical and CNN models is presented in Fig. 12. The performance of the theoretical model depends on the streamwise position, and it can only predict forces well in the wake around $5 \leq x \leq 15$. The force estimation of CNN models is not sensitive to the position. The performance of the baseline model CNN($\boldsymbol{u}$) is generally comparable with the theoretical model, and the improved model CNN($\boldsymbol{u}, \alpha$, att.) with the maximum $\varepsilon(\alpha_{\text{val}})$ under 10% in the entire wake outperforms the theoretical model. Thus, the data-driven model can be robust and effective with sufficiently large training data covering a broad range of flow parameters.[16,18]

Based on data classification D_2, we train the model with $15° \leq \alpha_{\text{train}} \leq 30°$ and assess it with $\alpha_{\text{val}} \leq 10°$ and $\alpha_{\text{val}} \geq 35°$ to investigate the performance of CNN for data extrapolation. Table VI shows $\varepsilon(\alpha_{\text{val}})$ of three CNN models for four cases with different α. We find that $\varepsilon(\alpha_{\text{val}})$ of D_2 is much larger than that of D_1 for data interpolation. The model CNN($\boldsymbol{u}$) has $\varepsilon(\alpha_{\text{val}}) > 10\%$ for all cases. As discussed in Appendix C, the baseline CNN model is not suitable for the extrapolation task, and it can only work well with proper training dataset.[15,50,58]

In particular, $\varepsilon(\alpha_{\text{val}})$ for $\alpha = 5°$ is very large in both CNN($\boldsymbol{u}$) and CNN($\boldsymbol{u}, \alpha$), due to the large difference between the velocity data of $\alpha = 5°$ and α_{train} in D_2. The relatively small effective region for $\alpha = 5°$ can further increase $\varepsilon(\alpha_{\text{val}})$ when the network does not focus on the effective region. Compared with the model CNN($\boldsymbol{u}, \alpha$), the model CNN ($\boldsymbol{u}, \alpha$, att.) with the attention module greatly improves the model performance for $\alpha = 5°$ by reducing about 80% error and also slightly improves the performance in other three cases. This improvement is further explained with the visualization of CNN training in Appendix D.

The models are further compared in the case of $\alpha = 40°$ in Fig. 13. The baseline CNN($\boldsymbol{u}$) is only good at data interpolation cases. Although it can identify that the data with $\alpha = 40°$ is close to the learned data with $\alpha = 20°$ and 25°, the estimated value of C_d (green dash-dotted line) is restricted to the values in the training set (green-shaded region). This is a general weakness of the methods of supervised machine learning.[11,15,50] Thus, if the value of C_d in the validation set is outside those in the training set, CNN($\boldsymbol{u}$) can have large errors. If the model is trained with the physical information α, the improved CNN can learn the trend that C_d grows with α, so that the CNN($\boldsymbol{u}, \alpha$, att.) is able to extrapolate the estimated value from those in the training set and achieve much better performance than that of the baseline CNN model.

Based on data classification D_3, we examine the dependence of CNN model performance on the size of learning data. In Table VI, $\varepsilon(\alpha_{\text{val}})$ for $\alpha = 20°$ and $\alpha = 25°$ from D_3 is generally reduced by half of each CNN model error from D_1. The best force estimation of the CNN($\boldsymbol{u}, \alpha$, att.) has very small error $\varepsilon(\alpha_{\text{val}}) = 2\%$. Hence, the estimation of the CNN models can be further improved by training larger datasets in practical applications.[49,77]

V. COMPARISON OF THEORETICAL AND DATA-DRIVEN MODELS

Appropriate criteria are necessary to assess the model performance of theoretical and data-driven models in practical application. Pope[78] proposed principal criteria to assess theoretical models for

TABLE V. Different data classifications of training and validation sets.

Data classification	Training set	Validation set
D_1	$\alpha_{\text{train}} = \{5°, 10°, 35°, 40°\}$	$\alpha_{\text{val}} = \{15°, 20°, 25°, 30°\}$
D_2	$\alpha_{\text{train}} = \{15°, 20°, 25°, 30°\}$	$\alpha_{\text{val}} = \{5°, 10°, 35°, 40°\}$
D_3	$\alpha_{\text{train}} = \{5°, 10°, 15°, 30°, 35°, 40°\}$	$\alpha_{\text{val}} = \{20°, 25°\}$

TABLE VI. Comparison of $\varepsilon(\alpha_{\text{val}})$ for different CNN models and data classifications. In this comparison, we only consider velocity data in the far wake at $5 \le x \le 15$ consistent with the theoretical model.

D_1	Model	$\alpha = 15°$	$\alpha = 20°$	$\alpha = 25°$	$\alpha = 30°$	$\alpha = \{15°, 20°, 25°, 30°\}$
	CNN $(\boldsymbol{u})$	9.4%	9.2%	17.3%	16.0%	12.9%
	CNN $(\boldsymbol{u}, \alpha)$	2.5%	8.3%	3.9%	2.8%	4.4%
	CNN $(\boldsymbol{u}, \alpha, \text{att.})$	1.4%	5.1%	4.2%	4.1%	3.7%
D_2	Model	$\alpha = 5°$	$\alpha = 10°$	$\alpha = 35°$	$\alpha = 40°$	$\alpha = \{5°, 10°, 35°, 40°\}$
	CNN $(\boldsymbol{u})$	68.3%	11.6%	11.4%	16.0%	28.7%
	CNN $(\boldsymbol{u}, \alpha)$	102.8%	10.0%	4.1%	7.2%	31.0%
	CNN $(\boldsymbol{u}, \alpha, \text{att.})$	23.7%	7.0%	3.3%	6.2%	10.1%
D_3	Model	$\alpha = 20°$	$\alpha = 25°$	$\alpha = \{20°, 25°\}$		
	CNN $(\boldsymbol{u})$	10.3%	7.3%	8.8%		
	CNN $(\boldsymbol{u}, \alpha)$	6.3%	3.5%	4.9%		
	CNN $(\boldsymbol{u}, \alpha, \text{att.})$	2.0%	2.0%	2.0%		

turbulent flows. The performance of data-driven models is usually assessed from the dimension of interpretability, generalization ability, and accuracy.[18,50]

Based on the previous works, we use six criteria to compare theoretical and data-driven models in the present study.

1. Level of description. It characterizes the level of information of physical quantities that a model can predict. The particular theoretical model in Eq. (26) can only predict drag, whereas the data-driven model can predict both drag and lift simultaneously with given necessary training data.[49]
2. Completeness. A model is complete if it is free from flow-dependent specifications. The theoretical model in Eq. (26) is complete except for the selection of S_e, while the CNN model is dependent on the particular dataset for model optimization.
3. Cost and ease of use. The algebraic model in Eq. (26) comes from physical laws, and its cost in application is negligible. The implementation of CNN models requires a moderate computational cost and the technical experience in using data-driven models.
4. Range of applicability. It evaluates if a model is applicable to different flows. The theoretical model is only applicable to velocity data in the far wake, while the CNN model can process data in the whole wake and its requirement on data quality less demanding than that of the theoretical model (see Appendix E). On the other hand, the theoretical model can be applied to various flows, while the application of the CNN model is highly dependent on the difference between training and testing datasets (see Appendix B).
5. Accuracy. The theoretical model in Eq. (26) is relatively accurate (generally $\bar{\varepsilon} < 10\%$) for various flows. The accuracy of CNN models depends on the training data. The data-driven model trained by enough data with the same flow configuration for model validation can be more accurate than the theoretical model.
6. Interpretability. The theoretical model can be well explained by underlying physics; for example, the relation between the drag and velocity defect is clear. The data-driven model appears to be a black box, and the relation between the network input and output is hard to be well interpreted by simple laws. However, in some sense, we can identify the important spatial region determining the model output through visualization of CNN (see

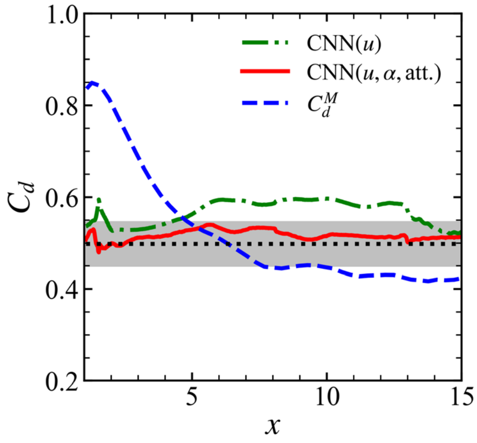


FIG. 12. Comparison of the CNN models in Table IV and theoretical model in Eq. (26) for estimating C_d in DNS case with $\alpha = 30°$. The gray shade denotes 10% relative error.

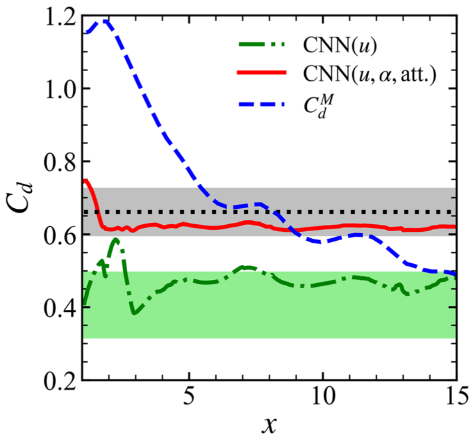


FIG. 13. Comparison of the CNN models in Table IV and theoretical model in Eq. (26) for estimating C_d in DNS case with $\alpha = 40°$. The gray shade denotes 10% relative error, and the green shade denotes the range of C_d in the training set.

Appendix D), and the network can also learn some empirical laws such as the growth of drag with angle of attack.

For the present cases, the theoretical model is superior to the data-driven model in terms of the completeness, cost, and interpretability, and the data-driven model with appropriate training data and optimized model architecture has better description and accuracy. The applicability of the two types of models is generally fair and flow-dependent.

VI. CONCLUSIONS

We develop and assess theoretical and data-driven models for estimating forces from cross-sectional data in the wake of flows past a plate. A series of datasets of three-dimensional flows past a flat plate with a range of angles of attack are calculated using the immersed boundary method. They are used for further model development and validation.

First, we develop a theoretical model to estimate drag of a flat plate based on the cross-sectional velocity data in the far wake. This model of the streamwise force coefficient in Eq. (26) incorporates the local momentum deficit and pressure variation. In the momentum term in Eq. (27), the force contribution from the stress on the plate surface is converted into a surface integral on a cross-sectional plane downstream. This force contribution is dominated by the velocity defect concentrated in an effective local region. In the pressure term in Eq. (28), Bernoulli's equation is used to calculate the pressure change using velocity data. This contribution is usually ignored in former two-dimensional studies and is important in the present three-dimensional study. We assess the theoretical model in Eq. (26) using eight DNS cases with different angles of attack in the far wake. The spatially mean error is generally less than 10%.

Second, we develop several CNN models for force estimation by regarding the velocity field in a cross section as a RGB image. To analyze the model performance, we design three CNN architectures for integrating different physical mechanisms and three training datasets for interpolation and extrapolation tasks. The baseline CNN architecture is adapted from the ResNet. We introduce the physical information, the angle of attack of the plate, into the CNN, so that the network can learn hidden physical laws between the force and angle of attack to improve the CNN performance. In addition, we add the attention module to make the CNN focus on the effective region for force estimation. For the data interpolation, the CNN model performs well. The performance of the optimized CNN model is superior to the theoretical model and can be further increased by training larger datasets. For the data extrapolation, the performance of the baseline model CNN($\boldsymbol{u}$) is not satisfactory due to the difference between the training and test datasets. The CNN performance can be improved in models CNN($\boldsymbol{u}, \alpha$) and CNN($\boldsymbol{u}, \alpha$, att.) by incorporating physical information into the network architecture.

The performances of the theoretical and data-driven models are compared by multiple criteria. In the present cases, the algebraic model in Eq. (26) is superior to the CNN model in terms of the completeness, cost, and interpretability, and the CNN model with appropriate training data and optimized network architecture has better description and accuracy.

Although the discussion on modeling is mainly for the flow past a stationary plate, Appendixes A and B demonstrate that both the present theoretical and CNN models can be extended to more complex flows past a moving body, and Appendix E shows that both models are robust for moderate noisy or under-resolved data. In the future work, the data-driven models are expected to be further informed by theoretical models and generalized to various flow configurations[79–81] and larger Re by utilizing data assimilation,[13,82] transfer learning,[83] and generating much larger training datasets.

ACKNOWLEDGMENTS

Numerical simulations were carried out on the TH-2A supercomputer in Guangzhou, China. This work has been supported in part by the National Natural Science Foundation of China (Nos. 91952108, 11925201, and 11988102), the National Numerical Wind Tunnel Project (No. NNW2021ZT6-B36), the National Key R&D Program of China (No. 2020YFE0204200), and the Xplore Prize.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Wenwen Tong: Conceptualization (supporting); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Shizhao Wang:** Conceptualization (supporting); Formal analysis (equal); Investigation (equal); Writing – review & editing (equal). **Yue Yang:** Conceptualization (lead); Formal analysis (equal); Funding acquisition (lead); Investigation (equal); Methodology (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in CNN Wake Model at https://github.com/YYgroup/wakemodel, Ref. 84.

APPENDIX A: MODELING FOR THE WAKE OF A FLAPPING PLATE

We extend the theoretical and CNN models to estimate thrust in the flow past a flapping plate. The kinematics of the flapping plate is described by

$$\alpha(t) = \alpha_m \cos(2\pi f t) \tag{A1}$$

and

$$z_c(t) = A \sin(2\pi f t), \tag{A2}$$

where $\alpha(t)$ denotes the time-varying angle of attack, α_m the pitching amplitude, $z_c(t)$ the instantaneous vertical position of the plate center, A the heaving amplitude, and f the flapping frequency. Motivated by the investigation of flow past low-aspect-ratio flapping wings,[19,22,59] we conduct a series of DNS cases of the flow past a flapping plate with $Re = 200$, $R_A = 1$, $A = 0.5$, and varying f and

TABLE VII. Parameters for the DNS of flows past a flapping plate and relative errors $\bar{\varepsilon}(C_d^M)$ of theoretical models for thrust estimation.

Case	1	2	3	4	5	6	7	8	9	10	11	12	13
f	0.6	0.6	0.6	0.7	0.7	0.7	0.8	0.8	0.8	0.65	0.65	0.75	0.75
α_m (°)	30	15	45	15	30	45	15	30	45	22.5	37.5	22.5	37.5
St	0.6	0.6	0.6	0.7	0.7	0.7	0.8	0.8	0.8	0.65	0.65	0.75	0.75
$\bar{\varepsilon}(C_d^M)$	6.9%	8.2%	14.4%	5.2%	5.9%	8.1%	9.2%	8.8%	7.0%	4.7%	5.4%	8.0%	5.4%

α_m listed in Table VII. For Strouhal numbers $St = 2Af/U = 0.6 \sim 0.8$, the wake in the present simulations has two sets of vortices.[22,59,60,85]

Note that the force exerted on the flapping plate is the thrust with $u_x > U$ in the wake (see Fig. 14), instead of the drag on the stationary plate with velocity defect [see Fig. 3(b)]. Accordingly, the definition of S_e in Eq. (25) is replaced by

$$S_e(x) = \{(y(x), z(x)) | u_x(y(x), z(x)) > u_{th}\}, \qquad \text{(A3)}$$

with $u_{th} = 1.01$ for the wake of a flapping plate. We apply the theoretical model in Eq. (26) to estimate the time-averaged thrust based on the time-averaged cross-sectional data for 13 cases in Table VII. The relative error $\bar{\varepsilon}(C_d^M)$ is generally less than 10%, so the theoretical model is applicable to the wake of a flapping plate.

The data-driven model is also applied to the wake velocity data of the flapping plate. Since the area of S_e for the flapping plate is larger than that for the stationary plate, the spatial region for CNN training and validation is enlarged to $S_{CNN} \in [-8, 8] \times [-8, 8]$. As shown in Fig. 15, we split 13 DNS cases into training and validation datasets as data interpolation. For the flapping plate, the flapping frequency is another important factor affecting the CNN performance, so we incorporate f, similar to α, into Layer6 in Fig. 9. Thus, we have five CNN models listed in Table VIII. The CNN models predict thrust well with relative errors generally less than 8%, comparable to that of the theoretical model in Table VII. Since the thrust is not linearly increased with $\alpha \in [15°, 45°]$, the prediction improvement of CNN$(\boldsymbol{u}, \alpha)$ is not as good as for cases of the stationary plate. On the other hand, the incorporation of f in the model CNN$(\boldsymbol{u}, f)$ can improve the performance due to the positive correlation between thrust and f. Since the attention mechanism is effective to focus on a relatively small spatial region illustrated in Fig. 14, the model CNN$(\boldsymbol{u}, \alpha, f, \text{att.})$ shows the best performance.

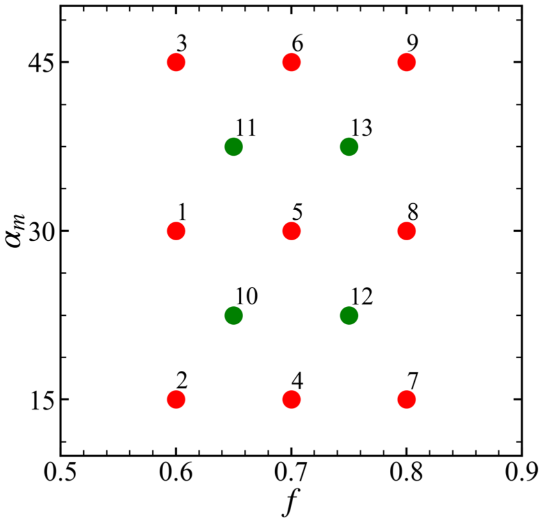


FIG. 15. Parameter distribution of training set (red dots) and validation set (green dots) for cases of flows past a flapping plate.

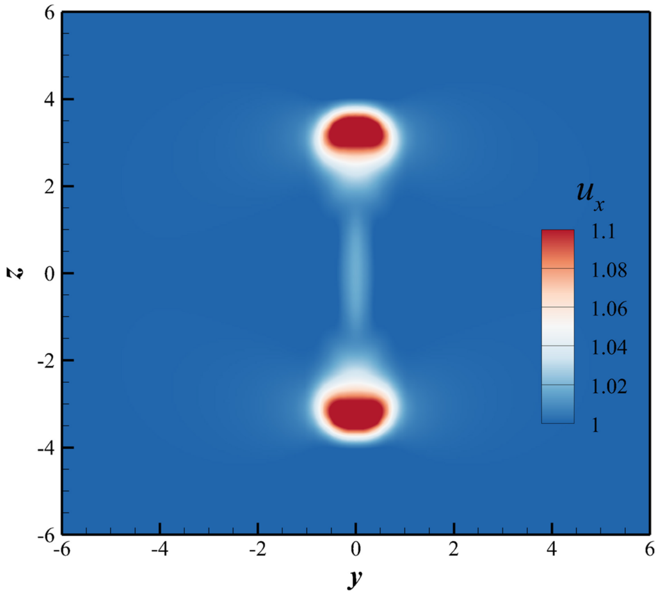


FIG. 14. Contour of u_x on the cross section at $x = 10$ for case 1 in Table VII.

TABLE VIII. Comparison of $\varepsilon(\alpha_{\text{val}})$ for different CNN models on the validation set.

Model	Case 10	Case 11	Case 12	Case 13	Cases 10–13
CNN $(\boldsymbol{u})$	5.4%	6.3%	9.3%	8.0%	7.2%
CNN $(\boldsymbol{u}, \alpha)$	5.7%	5.9%	8.6%	7.6%	6.9%
CNN $(\boldsymbol{u}, f)$	7.7%	10.2%	5.0%	4.5%	6.8%
CNN $(\boldsymbol{u}, \alpha, f)$	5.9%	8.5%	3.4%	2.3%	5.0%
CNN $(\boldsymbol{u}, \alpha, f, \text{att.})$	5.2%	3.3%	4.5%	3.7%	4.1%

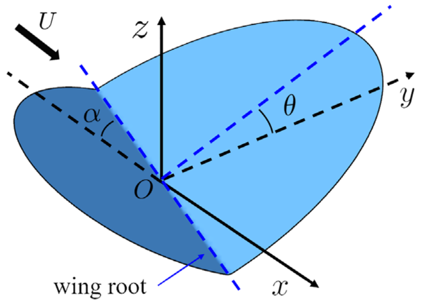


FIG. 16. Schematic of the flow past a butterfly-like elliptical plate with the angle of attack α and flapping angle θ.

APPENDIX B: MODELING FOR THE WAKE OF A BUTTERFLY-LIKE ELLIPTICAL PLATE

We extend the theoretical and CNN models to a complex wake for the flow past a butterfly-like elliptical plate.[86,87] As sketched in Fig. 16, the geometry of the elliptical plate is defined by $x^2/a^2 + y^2/b^2 = 1$ with $a = 0.5c$, $b = c$, and the maximum chord length c. The plate can be folded along the wing root in the x–z plane. The kinematics of the elliptical plate is characterized by

$$\alpha(t) = \alpha_m[1 - \cos(2\pi f t)] \tag{B1}$$

and

$$\theta(t) = \theta_m \cos(2\pi f t), \tag{B2}$$

where $\alpha(t)$ denotes the instantaneous angle of attack of the wing root, α_m the pitching amplitude, $\theta(t)$ the instantaneous flapping angle of the semi-elliptic wing, and θ_m the flapping amplitude. We set $Re = 200$, $f = 0.6$, $\alpha_m = 30°$, and $\theta_m = 30°$. The wake vortical structure in Fig. 17 illustrates much more complex wake topology with interconnected structures than the ring-like structures for the flow past a flat plate. This significant difference is used to test the robustness of force models.

The theoretical model in Eq. (26) is directly applied to estimate the time-averaged drag of the elliptical plate. It achieves excellent performance with the error around 3.3% in the far wake with $5 \le x \le 15$. The data-driven models developed from the flat-plate data are also applied to the wake flow of the elliptical plate. As listed in Table IX, we construct four CNN models based on different training datasets and network architectures. We applied the warm up strategy[44] to stabilize the training process and avoid overfitting from various training datasets. The learning rate is increased from 0 to 0.002 in the first training epoch, and then, it is decreased by a factor of five in every five epochs. The model CNN($\boldsymbol{u}$) with training data from the stationary plate has the maximum error about 20%. The error is reduced to 10% with the increase in training data and the optimization of network architectures.

The detailed model comparison is presented in Fig. 18. The theoretical model predicts the drag well in the far wake with a robust performance. The performance of the CNN model trained by data of simpler flow cases is less satisfactory than that of the model trained by the same type of flow data in Fig. 12. Therefore, the performance of CNN models strongly depends on the range of training data. The CNN model can generalize to unseen data only if training data samples are sufficient and test data are not too far from the training data.[50]

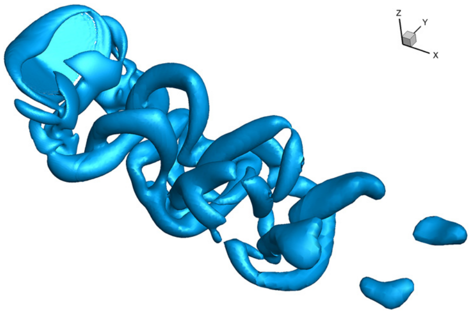


FIG. 17. Isosurface of $Q = 1$ for the flow past a butterfly-like elliptical plate.

TABLE IX. Comparison of relative errors of theoretical and CNN models for estimating thrust of the flapping elliptical plate.

Model	Training dataset	$\bar{\varepsilon}$
CNN ($\boldsymbol{u}$)	Stationary plate	19.2%
CNN ($\boldsymbol{u}, \alpha$)	Stationary plate	16.8%
CNN ($\boldsymbol{u}, \alpha$, att.)	Stationary plate	14.8%
CNN ($\boldsymbol{u}$)	Stationary plate+flapping plate	10.7%
C_d^M	⋯	3.3%

APPENDIX C: PROBABILITY INTERPRETATION OF THE CNN MODEL

We explain the mechanism of the CNN model for estimating forces from the viewpoint of a data classification task. The Bayesian classifier, the probability $P(c|\boldsymbol{x}_{in})$ of a given class c conditioned on the input data $\boldsymbol{x}_{in}$, is often used to classify the input data.[88] For the

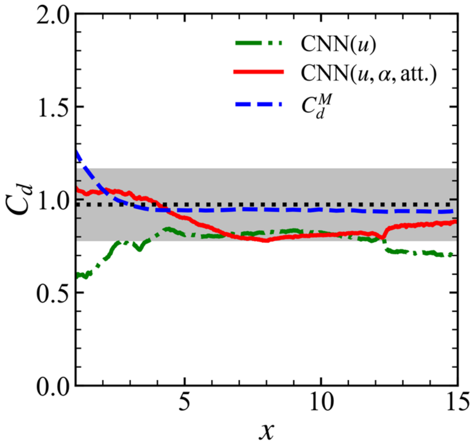


FIG. 18. Comparison of theoretical and CNN models for estimating thrust of the flapping elliptical plate at different x. The gray shade denotes 20% relative error.

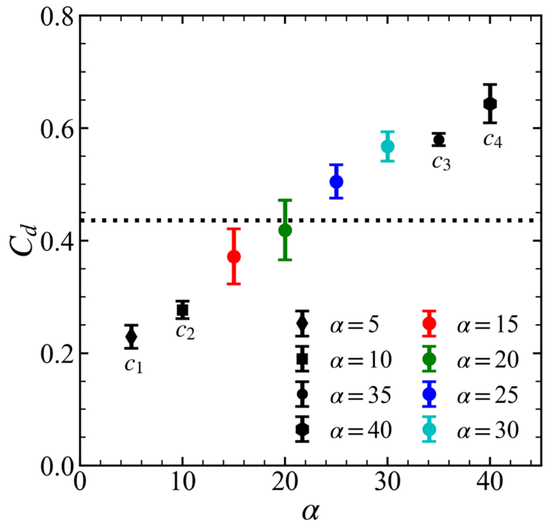


FIG. 19. Data of C_d in the training set (in black) and estimations of C_d from the model CNN($\boldsymbol{u}$) and the validation set (in color) of D_1 for the stationary plate, where $c_1 \sim c_4$ are labels of the four cases in the training set. The error bar denotes the standard deviation. The dotted line denotes the modeling result from the random classifier.

dataset D1 in Table V, we regard the training set as four classes labeled by c_i with $i = 1, 2, 3, 4$, corresponding to four α as illustrated in Fig. 19. In the estimation of C_d from input image data $\boldsymbol{x}_{in}$, the CNN model first learns $P(c_i|\boldsymbol{x}_{in})$ from the training set and then calculates the output as

$$C_d^{CNN}(\boldsymbol{x}_{in}) = \sum_{i=1}^{i=4} P(c_i|\boldsymbol{x}_{in}) C_d^{DNS}(c_i), \qquad \sum_{i=1}^{i=4} P(c_i|\boldsymbol{x}_{in}) = 1, \qquad \text{(C1)}$$

from the validation set, where $C_d^{DNS}(c_i)$ is the force coefficient from DNS in training data with label c_i. Taking the training set in data classification D_1 for the stationary plate as an example, Eq. (C1) with a random classifier $P(c_i|\boldsymbol{x}_{in}) = 0.25$ has $C_d^{CNN}(\boldsymbol{x}_{in}) = 0.436$, and its relative error for the validation set is 18.7%. Training of the baseline model CNN($\boldsymbol{u}$) generates more complex and appropriate $P(c_i|\boldsymbol{x}_{in})$ than that of the random guess, so its relative error is reduced to 12.9%.

In Fig. 19, estimating C_d from the model CNN($\boldsymbol{u}$) for the stationary plate in D_1 in Table V is explained as the classification in Eq. (C1). For example, the distributions of the CNN model output for the input data with $\alpha = 25°$ and $\alpha = 30°$ are closer to the data in the learning set with class labels c_3 and c_4. For $\boldsymbol{x}_{in}$ with $\alpha = 30°$, $P(c_3|\boldsymbol{x}_{in})$ and $P(c_4|\boldsymbol{x}_{in})$ are larger than $P(c_1|\boldsymbol{x}_{in})$ and $P(c_2|\boldsymbol{x}_{in})$ due to the similarity of the datasets among $\alpha = 30°$, $\alpha = 35°$, and $\alpha = 40°$. Thus, the CNN prediction for data with $\alpha = 30°$ leans to the data values with labels c_3 and c_4. Moreover, Eq. (C1) and Fig. 19 imply that the model CNN($\boldsymbol{u}$) has a limited data extrapolation capability due to the nature of classifying known training data.

APPENDIX D: VISUALIZATION OF TRAINING CNN

We interrogate the CNN training process to explain different performances of CNN models by visualizing some intermediate steps.[54,55] A typical case with $\alpha = 5°$ from the validation set in classification D_2 is chosen to visualize the middle convolutional layer. Typical cross-sectional velocity data at $x = 10$, as a network input, are shown on the leftmost of Fig. 20. First, we visualize the feature map of intermediate convolutional layers in CNN.[89] The convolutional layer Conv5, as a representative one in the CNN architecture in Fig. 9, is visualized. The composite feature map is obtained by integrating C_5 channels in Conv5 as

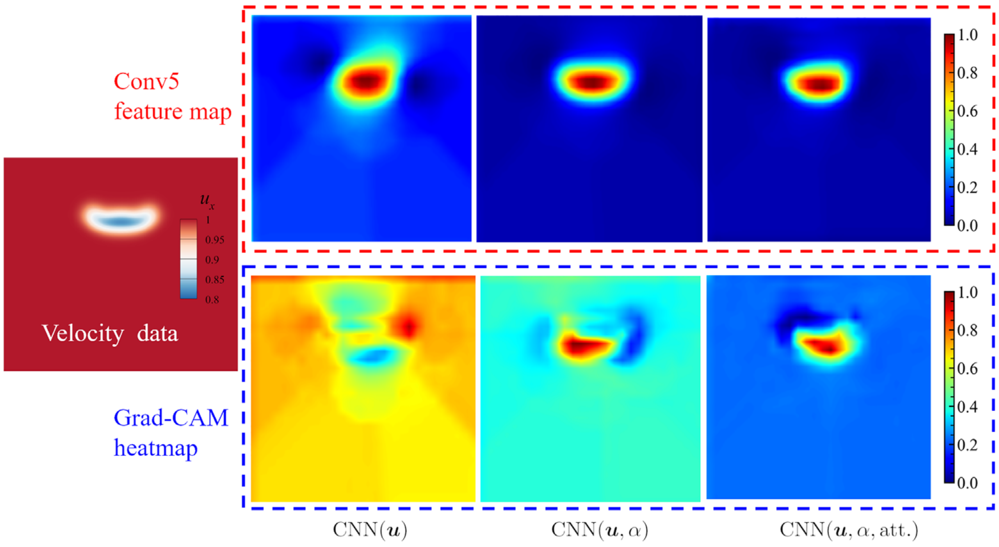


FIG. 20. Visualization of training CNN. The leftmost figure is the contour of u_x at $x = 10$ for the flow past a stationary plate with $\alpha = 5°$. The upper right three figures are feature maps of Conv5 in the three CNN architectures listed in Table IV. The lower right three figures are heat maps obtained by the Grad-CAM.

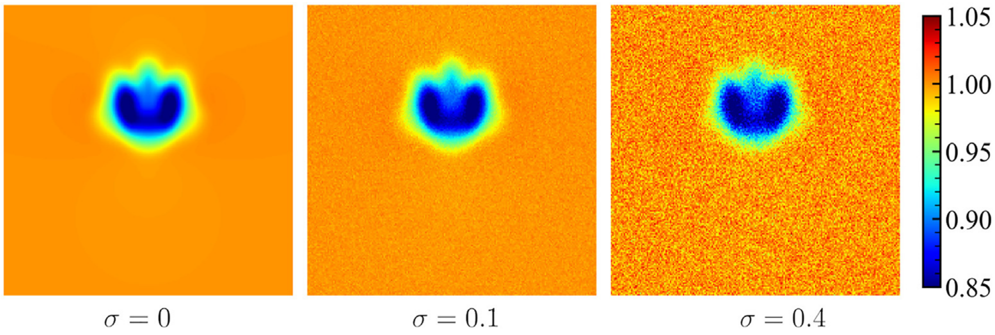


FIG. 21. Comparison of contours $u_{s,x}$ of synthetic data with different noise levels at $x = 10$ for the flow past a stationary plate with $\alpha = 30°$.

TABLE X. Comparison of relative errors of theoretical and CNN models for estimating drag based on synthetic data with different noise levels for the flow past a stationary plate with $\alpha = 30°$.

Model	$\sigma = 0$	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
C_d^M	10.7%	10.6%	10.6%	10.4%	13.2%	24.4%
CNN $(\boldsymbol{u}, \alpha)$	2.8%	7.0%	4.7%	7.0%	1.1%	4.0%

$$f_{Conv5}(j,k) = \sum_{i=1}^{C_5} \boldsymbol{x}_{con}(i,j,k), \tag{D1}$$

where $\boldsymbol{x}_{con}$ is the extracted feature in Conv5 for the given input image. The normalized feature maps of $\boldsymbol{f}_{Conv5}$ within $[0, 1]$ from different CNN models are compared in Fig. 20. The feature map of the improved CNN more focuses on the effective local region with large velocity defect than that of the baseline CNN. Note that the feature map is not symmetric owing to the asymmetry of convolutional kernel for feature extraction.

Second, we visualize the network using the gradient-weighted class activation mapping (Grad-CAM)[90,91] to find the important spatial location deciding the network output. In Grad-CAM, the gradient of network output is back propagated into the last convolutional layer to generate a localization heat map to highlight the significant region for network prediction (see Ref. 91). In Fig. 20, the heat map for CNN($\boldsymbol{u}$) is not localized, indicating that the baseline CNN cannot utilize the most effective spatial information to predict force. The heat map of CNN($\boldsymbol{u}$, α, att.) focuses on the local region with the maximum response to make decision with the attention mechanism, which improves model performance (see Table VI).

APPENDIX E: MODEL PERFORMANCE FOR NOISY OR UNDER-RESOLVED DATA

The theoretical and data-driven models work well on the DNS data, but the data can be noisy or under-resolved in experimental measurements and practical applications. These effects on the model performance are studied below.

The noisy data u_n are generated from accurate data u by[92,93]

$$u_n \sim \sigma\sigma_u N(0, 1), \tag{E1}$$

where the noise-to-signal ratio σ indicates the noise level, σ_u is the standard deviation of u, and $N(0, 1)$ is the normal distribution. The synthetic data are

$$u_s = u + u_n. \tag{E2}$$

We generate the synthetic velocity $\boldsymbol{u}_s = (u_{s,x}, u_{s,y}, u_{s,z})$ with components $u_{s,x}$, $u_{s,y}$, and $u_{s,z}$ obtained from Eqs. (E1) and (E2). Figure 21 shows $u_{s,x}$ in a cross section at $x = 10$ for the DNS case with $\alpha = 30°$ and different σ.

We compare the theoretical and CNN models based on synthetic velocity data $\boldsymbol{u}_s$ with different noise levels in Table X.

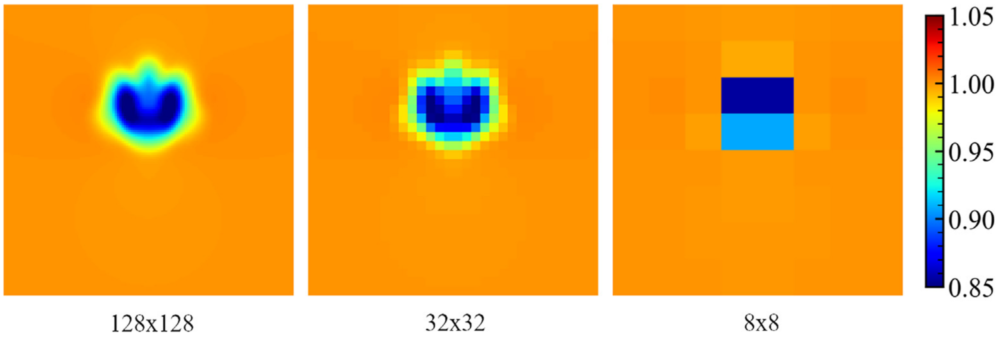


FIG. 22. The comparison of u_x with different mesh resolutions at $x = 10$ for the DNS case with $\alpha = 30°$.

TABLE XI. Comparison of relative errors of theoretical and CNN models for estimating drag based on data with different spatial resolutions for the flow past a stationary plate with $\alpha = 30^\circ$.

Model	256×256	128×128	64×64	32×32	16×16	8×8
C_d^M	10.8%	10.3%	9.4%	7.7%	4.6%	17.2%
CNN $(\boldsymbol{u}, \alpha)$	2.8%	2.7%	3.8%	7.6%	2.6%	9.4%

The relative error $\bar{\varepsilon}(C_d^M)$ of the theoretical model grows with $\sigma > 0.1$ due to inaccurate identification of the effective region for force calculation in Eq. (25). By contrast, the effect of noise on the performance of CNN models (trained by data classification D_1) is almost negligible. Therefore, the CNN model can be more robust than the theoretical model for predicting forces from noisy velocity data.

The under-resolved data are generated with different resolutions from the DNS data by the linear interpolation. The resolution of the data is in the shape of (H, W) with $H = W = \{256, 128, 64, 32, 16, 8\}$. The low-resolution velocity fields are shown in Fig. 22. As compared in Table XI, both models perform well with the errors under 10% for under-resolved data, except for the theoretical model for very coarse data with resolution 8×8. In addition, the model based on limited measurements[94,95] and super-resolution reconstruction[40,96] can facilitate improving force estimation from under-resolved flow data.

REFERENCES

[1] A. Hedenström, L. C. Johansson, M. Wolf, R. von Busse, Y. Winter, and G. R. Spedding, "Bat flight generates complex aerodynamic tracks," Science **316**, 894–897 (2007).

[2] T. Y. Hubel, D. K. Riskin, S. M. Swartz, and K. S. Breuer, "Wake structure and wing kinematics: The flight of the lesser dog-faced fruit bat, *Cynopterus brachyotis*," J. Exp. Biol. **213**, 3427–3440 (2010).

[3] S. Wang, G. He, and T. Liu, "Estimating lift from wake velocity data in flapping flight," J. Fluid Mech. **868**, 501–537 (2019).

[4] K. Olasek and M. Karczewski, "Velocity data-based determination of airfoil characteristics with circulation and fluid momentum change methods, including a control surface size independence test," Exp. Fluids **62**, 108 (2021).

[5] W. M. van Rees, M. Gazzola, and P. Koumoutsakos, "Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers," J. Fluid Mech. **722**, R3 (2013).

[6] D. B. Quinn, G. V. Lauder, and A. J. Smits, "Maximizing the efficiency of a flexible propulsor using experimental optimization," J. Fluid Mech. **767**, 430–448 (2015).

[7] G. R. Spedding, M. Rosén, and A. Hedenström, "A family of vortex wakes generated by a thrush nightingale in free flight in a wind tunnel over its entire natural range of flight speeds," J. Exp. Biol. **206**, 2313–2344 (2003).

[8] G.-J. Li and X.-Y. Lu, "Force and power of flapping plates in a fluid," J. Fluid Mech. **712**, 598–613 (2012).

[9] L. L. Kang, L. Q. Liu, W. D. Su, and J. Z. Wu, "Minimum-domain impulse theory for unsteady aerodynamic force," Phys. Fluids **30**, 016107 (2018).

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

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

[12] A. Chizfahm and R. Jaiman, "Data-driven stability analysis and near-wake jet control for the vortex-induced vibration of a sphere," Phys. Fluids **33**, 044104 (2021).

[13] S. Li, C. He, and Y. Liu, "A data assimilation model for wall pressure-driven mean flow reconstruction," Phys. Fluids **34**, 015101 (2022).

[14] Y. J. Lee, K.-B. Lua, T. T. Lim, and K. S. Yeo, "A quasi-steady aerodynamic model for flapping flight with improved adaptability," Bioinspiration Biomimetics **11**, 036005 (2016).

[15] K. Fukami, K. Fukagata, and K. Taira, "Assessment of supervised machine learning methods for fluid flows," Theor. Comput. Fluid Dyn. **34**, 497–519 (2020).

[16] X. Cai, D. Kolomenskiy, T. Nakata, and H. Liu, "A CFD data-driven aerodynamic model for fast and precise prediction of flapping aerodynamics in various flight velocities," J. Fluid Mech. **915**, A114 (2021).

[17] J. Wu, L. Liu, and T. Liu, "Fundamental theories of aerodynamic force in viscous and compressible complex flows," Prog. Aerosp. Sci. **99**, 27–63 (2018).

[18] J. Kou and W. Zhang, "Data-driven modeling for unsteady aerodynamics and aeroelasticity," Prog. Aerosp. Sci. **125**, 100725 (2021).

[19] J. H. J. Buchholz and A. J. Smits, "On the evolution of the wake structure produced by a low-aspect-ratio pitching panel," J. Fluid Mech. **546**, 433–443 (2005).

[20] K. Taira and T. Colonius, "Three-dimensional flows around low-aspect-ratio flat-plate wings at low Reynolds numbers," J. Fluid Mech. **623**, 187–207 (2009).

[21] C. Li, H. Dong, and Z. Liang, "Proper orthogonal decomposition analysis of 3D wake structures in a pitching-rolling plate," in *54th AIAA Aerospace Sciences Meeting, San Diego, California* (AIAA, 2016), p. 2071.

[22] W. Tong, Y. Yang, and S. Wang, "Estimating thrust from shedding vortex surfaces in the wake of a flapping plate," J. Fluid Mech. **920**, A10 (2021).

[23] Q. Mao, J. Zhao, Y. Liu, and H. J. Sung, "Hydrodynamic benefits of pectoral fins in a self-propelled flexible plate," Phys. Fluids **34**, 021909 (2022).

[24] J. H. J. Buchholz and A. J. Smits, "The wake structure and thrust performance of a rigid low-aspect-ratio pitching panel," J. Fluid Mech. **603**, 331–365 (2008).

[25] A.-J. Buchner, N. Buchmann, K. Kilany, C. Atkinson, and J. Soria, "Stereoscopic and tomographic PIV of a pitching plate," Exp. Fluids **52**, 299–314 (2012).

[26] L. Mendelson and A. H. Techet, "Quantitative wake analysis of a freely swimming fish using 3D synthetic aperture PIV," Exp. Fluids **56**, 135 (2015).

[27] H. Choi, J. Lee, and H. Park, "Wake structures behind a rotor with superhydrophobic-coated blades at low Reynolds number," Phys. Fluids **31**, 015102 (2019).

[28] G. Pavia, M. Varney, M. Passmore, and M. Almond, "Three dimensional structure of the unsteady wake of an axisymmetric body," Phys. Fluids **31**, 025113 (2019).

[29] F. Noca, D. Shiels, and D. Jeon, "Measuring instantaneous fluid dynamic forces on bodies, using only velocity fields and their derivatives," J. Fluid Struct **11**, 345–350 (1997).

[30] F. Noca, D. Shiels, and D. Jeon, "A comparison of methods for evaluating time-dependent fluid dynamic forces on bodies, using only velocity fields and their derivatives," J. Fluid Struct. **13**, 551–578 (1999).

[31] G. V. Lauder and E. G. Drucker, "Forces, fishes, and fluids: Hydrodynamic mechanisms of aquatic locomotion," News Physiol. Sci. **17**, 235–240 (2002).

[32] J.-Z. Wu, H.-Y. Ma, and M.-D. Zhou, *Vortical Flows* (Springer, 2015).

[33] P. Henningsson and A. Hedenström, "Aerodynamics of gliding flight in common swifts," J. Exp. Biol. **214**, 382–393 (2011).

[34] E. Gutierrez, D. B. Quinn, D. D. Chin, and D. Lentink, "Lift calculations based on accepted wake models for animal flight are inconsistent and sensitive to vortex dynamics," Bioinspiration Biomimetics **12**, 016004 (2016).

[35] G. R. Spedding and A. Hedenström, "PIV-based investigations of animal flight," Exp. Fluids **46**, 749–763 (2009).

[36] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," Annu. Rev. Fluid Mech. **51**, 357–377 (2019).

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

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1986).

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, Quebec, Canada* (Curran Associates, 2014), pp. 2672–2680.

[40]H. Kim, J. Kim, S. Won, and C. Lee, "Unsupervised deep learning for super-resolution reconstruction of turbulence," J. Fluid Mech. **910**, A29 (2021).

[41]Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**, 2278–2324 (1998).

[42]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, Nevada* (Curran Associates, 2012), pp. 1097–1105.

[43]K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in 3rd International Conference on Learning Representations (ICLR), San Diego, California, 2015.

[44]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada* (IEEE, 2016), pp. 770–778.

[45]J. Kim and C. Lee, "Prediction of turbulent heat transfer using convolutional neural networks," J. Fluid Mech. **882**, A18 (2020).

[46]L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa, "Convolutional-network models to predict wall-bounded turbulence from wall quantities," J Fluid Mech **928**, A27 (2021).

[47]K. Fukami, K. Fukagata, and K. Taira, "Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows," J. Fluid Mech. **909**, A9 (2021).

[48]Y. Zhang, W. Sung, and D. Mavris, "Application of convolutional neural network to predict airfoil lift coefficient," in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, Florida* (AIAA, 2018), p. 1903.

[49]M. Morimoto, K. Fukami, K. Zhang, A. G. Nair, and K. Fukagata, "Convolutional neural networks for fluid flow analysis: Toward effective meta-modeling and low dimensionalization," Theor. Comput. Fluid Dyn. **35**, 633–658 (2021).

[50]M. Morimoto, K. Fukami, K. Zhang, and K. Fukagata, "Generalization techniques of neural networks for fluid flow estimation," Neural Comput. Appl. **34**, 3647–3669 (2022).

[51]L. Itti and C. Koch, "Computational modelling of visual attention," Nat. Rev. Neurosci. **2**, 194–203 (2001).

[52]J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, BAM: Bottleneck attention module, in *29th British Machine Vision Conference (BMVC)* (Northumbria University, United Kingdom, 2018).

[53]S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany* (Springer, 2018), pp. 3–19.

[54]Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural network see the world—A survey of convolutional neural network visualization methods," Math. Found. Comput. **1**, 149–180 (2018).

[55]G. D. Portwood, B. T. Nadiga, J. A. Saenz, and D. Livescu, "Interpreting neural network models of residual scalar flux," J. Fluid Mech. **907**, A23 (2021).

[56]E. Jagodinski, X. Zhu, and S. Verma, "Uncovering dynamically critical regions in near-wall turbulence using 3D convolutional neural networks," arXiv:2004.06187 (2020).

[57]N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, "Shallow neural networks for fluid flow reconstruction with limited sensors," Proc. R. Soc. A **476**, 20200097 (2020).

[58]K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata, "Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes," Theor. Comput. Fluid Dyn. **34**, 367–383 (2020).

[59]H. Dong, R. Mittal, and F. M. Najjar, "Wake topology and hydrodynamic performance of low-aspect-ratio flapping foils," J. Fluid Mech. **566**, 309–343 (2006).

[60]S. Wang, X. Zhang, G. He, and T. Liu, "A lift formula applied to low-Reynolds-number unsteady flows," Phys. Fluids **25**, 093605 (2013).

[61]W. Tong, Y. Yang, and S. Wang, "Characterizing three-dimensional features of vortex surfaces in the flow past a finite plate," Phys. Fluids **32**, 011903 (2020).

[62]S. Wang and X. Zhang, "An immersed boundary method based on discrete stream function formulation for two- and three-dimensional incompressible flows," J. Comput. Phys. **230**, 3479–3499 (2011).

[63]C. S. Peskin, "The immersed boundary method," Acta Numer. **11**, 479–517 (2002).

[64]P. Sooraj, A. Agrawal, and A. Sharma, "Measurement of drag coefficient for an elliptical cylinder," J. Energy Environ. Sustainability **5**, 1–7 (2018).

[65]J. C. R. Hunt, A. A. Wray, and P. Moin, "Eddies, streams, and convergence zones in turbulent flows," in *Proceedings of Summer Program* (Stanford University, 1988), pp. 193–208.

[66]S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning (ICML), Lille, France* (PMLR, 2015), pp. 448–456.

[67]A. F. Agarap, "Deep learning using rectified linear units (ReLU)," arXiv:1803.08375 (2018).

[68]M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah (IEEE, 2018), pp. 4510–4520.

[69]M. Gamahara and Y. Hattori, "Searching for turbulence models by artificial neural network," Phys. Rev. Fluids **2**, 054604 (2017).

[70]T. Nakamura, K. Fukami, and K. Fukagata, "Identifying key differences between linear stochastic estimation and neural networks for fluid flow regressions," Sci. Rep. **12**, 3726 (2022).

[71]M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex-induced vibrations," J. Fluid Mech. **861**, 119–137 (2019).

[72]S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, "Flow over an espresso cup: Inferring 3D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks," J. Fluid Mech. **915**, A102 (2021).

[73]V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, Quebec, Canada* (Curran Associates, 2014), pp. 2204–2212.

[74]R. Franke, "Scattered data interpolation: Tests of some methods," Math. Comput. **38**, 181–200 (1982).

[75]A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, Canada* (Curran Associates, 2019), pp. 8026–8037.

[76]K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV), Santiago, Chile* (IEEE, 2015), pp. 1026–1034.

[77]S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," Comput. Mech. **64**, 525–545 (2019).

[78]S. B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).

[79]D. Zhang, Q.-G. Huang, G. Pan, L.-M. Yang, and W.-X. Huang, "Vortex dynamics and hydrodynamic performance enhancement mechanism in batoid fish oscillatory swimming," J. Fluid Mech. **930**, A28 (2022).

[80]L. Wang, F.-B. Tian, and H. Liu, "Numerical study of three-dimensional flapping wings hovering in ultra-low-density atmosphere," Phys. Fluids **34**, 041903 (2022).

[81]A. Menzer, Y. Gong, F. E. Fish, and H. Dong, "Bio-inspired propulsion: Towards understanding the role of pectoral fin kinematics in manta-like swimming," Biomimetics **7**, 45 (2022).

[82]Y. Wang, Z. Yuan, C. Xie, and J. Wang, "Temporally sparse data assimilation for the small-scale reconstruction of turbulence," Phys. Fluids **34**, 065115 (2022).

[83]S. Lee, J. Yang, P. Forooghi, A. Stroh, and S. Bagheri, "Predicting drag on rough surfaces by transfer learning of empirical correlations," J. Fluid Mech. **933**, A18 (2022).

[84]W. Tong and Y. Yang, see https://github.com/YYgroup/wakemodel for "CNN Wake Model (2022)."

[85]G. Liu, H. Dong, and C. Li, "Vortex dynamics and new lift enhancement mechanism of wing-body interaction in insect forward flight," J. Fluid Mech. **795**, 634–651 (2016).

[86]K. Suzuki, T. Aoki, and M. Yoshino, "Effect of wing mass in free flight of a two-dimensional symmetric flapping wing–body model," Fluid Dyn. Res. **49**, 055504 (2017).

[87] K. Suzuki, I. Okada, and M. Yoshino, "Effect of wing mass on the free flight of a butterfly-like model using immersed boundary–lattice Boltzmann simulations," J. Fluid Mech. **877**, 614–647 (2019).

[88] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," Neural Comput. **3**, 461–483 (1991).

[89] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland* (Springer, 2014), pp. 818–833.

[90] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada* (IEEE, 2016), pp. 2921–2929.

[91] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision (ICCV), Venice, Italy* (IEEE, 2017), pp. 618–626.

[92] S. Scharnowski and C. J. Kähler, "Estimation and optimization of loss-of-pair uncertainties based on PIV correlation functions," Exp. Fluids **57**, 23 (2016).

[93] S. Scharnowski and C. J. Kähler, "On the loss-of-correlation due to PIV image noise," Exp. Fluids **57**, 119 (2016).

[94] J. L. Callaham, K. Maeda, and S. L. Brunton, "Robust flow reconstruction from limited measurements via sparse representation," Phys. Rev. Fluids **4**, 103907 (2019).

[95] K. Fukami, R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira, "Global field reconstruction from sparse sensors with Voronoi tessellation-assisted deep learning," Nat. Mach. Intell. **3**, 945–951 (2021).

[96] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," J. Fluid Mech. **870**, 106–120 (2019).