RESEARCH ARTICLE | MARCH 15 2023

Exploring hidden flow structures from sparse data through deep-learning-strengthened proper orthogonal decomposition

Chang Yan (闫畅); Shengfeng Xu (许盛峰); Zhenxu Sun (孙振旭); Dilong Guo (郭迪龙); Shengjun Ju (鞠胜军); Renfang Huang (黄仁芳); Guowei Yang (杨国伟)

Physics of Fluids 35, 037119 (2023)
https://doi.org/10.1063/5.0138287

Physics of Fluids ARTICLE scitation.org/journal/phf

Exploring hidden flow structures from sparse data through deep-learning-strengthened proper orthogonal decomposition

Cite as: Phys. Fluids **35**, 037119 (2023); doi: 10.1063/5.0138287
Submitted: 10 December 2022 · Accepted: 23 February 2023 ·
Published Online: 15 March 2023

Chang Yan (闫畅),[1,2] Shengfeng Xu (许盛峰),[1,3] Zhenxu Sun (孙振旭),[1,a)] Dilong Guo (郭迪龙),[1,3] Shengjun Ju (鞠胜军),[1] Renfang Huang (黄仁芳),[1] and Guowei Yang (杨国伟)[1,3]

AFFILIATIONS

[1]Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

[a)]Author to whom correspondence should be addressed: sunzhenxu@imech.ac.cn

ABSTRACT

Proper orthogonal decomposition (POD) enables complex flow fields to be decomposed into linear modes according to their energy, allowing the key features of the flow to be extracted. However, traditional POD requires high-quality inputs, namely, high-resolution spatio-temporal data. To alleviate the dependence of traditional POD on the quality and quantity of data, this paper presents a POD method that is strengthened by a physics-informed neural network (PINN) with an overlapping domain decomposition strategy. The loss function and convergence of modes are considered simultaneously to determine the convergence of the PINN-POD model. The proposed framework is applied to the flow past a two-dimensional circular cylinder at Reynolds numbers ranging from 100 to 10 000 and achieves accurate and robust extraction of flow structures from spatially sparse observation data. The spatial structures and dominant frequency can also be extracted under high-level noise. These results demonstrate that the proposed PINN-POD method is a reliable tool for extracting the key features from sparse observation data of flow fields, potentially shedding light on the data-driven discovery of hidden fluid dynamics.

I. INTRODUCTION

The analysis of fluid dynamics often rests on the notion that the evolution of a flow field is primarily facilitated by a small number of coherent structures.[1] Various methods for extracting these structures from complex flow fields have been developed, such as proper orthogonal decomposition (POD),[2] dynamic mode decomposition (DMD),[3] Koopman analysis,[4] global linear stability analysis,[5] resolvent analysis,[6] and their variants. The POD method is widely used because its modes are linear, orthogonal, and ordered by eigenvalues. POD was first introduced to the fluid dynamics/turbulence community by Lumley *et al.*[2] as a mathematical technique for extracting coherent structures from turbulent flow fields. Delville *et al.*[7] studied the large-scale structures in a plane turbulent mixing layer through POD and demonstrated that streamwise-aligned vortices and quasi-two-dimensional spanwise structures were contained in the first mode. Liberge *et al.*[8] constructed a low-order dynamical system with POD to study fluid-structure interaction problems, while Muld *et al.*[9] decomposed the flow field of a surface-mounted cube using POD and DMD and investigated the convergence of POD modes. Liu *et al.*[10] identified the dominant coherent structures within cavitating flow around a hydrofoil through POD and DMD and observed large-scale cavity–vortex structures and re-entrant jets. Muld *et al.*[11] studied the wake field of a high-speed train and found that the dominant POD mode converges faster than the dominant DMD mode in broadband data. The near-wake field of a finite-length cylinder has been investigated based on the POD of particle image velocimetry (PIV) data,[12] which demonstrated that the wake is dominated by POD mode 1, corresponding to symmetrical vortex shedding. However, these previous applications typically required a high-resolution dataset from experimental or numerical spatiotemporal flow fields. In practical engineering applications, it is difficult to obtain a complete high-resolution flow field. The test equipment used for precise measurements of the flow field is usually expensive, such as laser Doppler velocimetry,[13] PIV,[14] and laser-induced fluorescence[15] apparatus. High-resolution numerical simulations are also costly in terms of preprocessing, computation, and storage.[16,17] Furthermore, the traditional turbulence models used in numerical simulations cannot accurately predict flow fields in which separation occurs.[18] Therefore, the application of POD to extract flow structures is extremely limited. In contrast, measurements can easily and cheaply be obtained from spatially sparse points in fluid mechanics experiments. If the original flow field can be regressed from a small amount of data, the application range of POD would be greatly enlarged.

With the continuous development of machine learning, deep learning techniques have been widely applied in the field of fluid mechanics.[19] Jin *et al.*[20] proposed a convolutional neural network-based data-driven method that establishes the relationship between wake structures and the pressure experienced on the wall of a cylinder. Xu *et al.*[21] introduced a machine-learning-assisted Reynolds-averaged Navier–Stokes equations (RANS) method to investigate the unsteady cavitating flow around a hydrofoil. Ma *et al.*[22] constructed a two-branch deep neural network model that improved the high-fidelity bubble migration results and reduced the dependence on the quantity of experimental data. Zheng *et al.*[23] explored active flow control strategies in suppressing vortex-induced vibration through reinforcement learning, resulting in an 82.7% reduction in the vibration amplitude. Zhang *et al.*[24] proposed a compressed sensing reduced-order modeling framework that combined a long short-term memory model with sparsity-promoting DMD, allowing unsteady flow fields to be reliably predicted. Peng *et al.*[25] developed an attention-enhanced neural network model and obtained various statistics and instantaneous spatial structures of turbulence, while Yuan *et al.*[26] reported a deconvolutional artificial neural network for subgrid-scale stress in the large-eddy simulation (LES) framework and showed that this network predicted subgrid-scale stress better than the velocity gradient model and conventional approximate deconvolution model.

Physics-informed deep learning exhibits excellent performance in the regression of flow fields from sparse observations. This approach has recently attracted extensive interest as a means of solving systems of partial differential equations (PDEs). The framework was first proposed by Lagaris *et al.*[27] 25 years ago, although it was only recently that Raissi *et al.*[28] refocused attention on this algorithm using a machine learning framework. The core design involves embedding physical laws into the framework of traditional deep learning to create a physics-informed neural network (PINN). This can be achieved by introducing the residual of the physical equations to the loss function of the neural network. During the training of the neural network, the PINN gradually approaches the solution of the physical equations as the loss function is minimized. Raissi *et al.*[28] took a two-dimensional (2D) incompressible laminar case as an example, in which the original fields were regressed from scattered data sampled throughout the spatiotemporal domain. In subsequent research,[29] the velocity and pressure fields were directly extracted from a flow visualization, and the PINN was extended to three-dimensional (3D) incompressible flow. Cai *et al.*[30] used a PINN to infer the instantaneous velocity and pressure fields from temperature measurements of the flow over an espresso cup, and Wang *et al.*[31] obtained a high-resolution velocity field from sparse PIV measurements using a PINN. Xu *et al.*[32] employed a PINN to regress the flow field from sparse data and infer missing data in a certain region, showing that the cosine annealing algorithm exhibits excellent performance in accelerating the convergence of the training stage. Qiu *et al.*[33] developed a phase-field PINN method for a 2D immiscible incompressible two-phase flow. This allowed them to obtain the interface shape with excellent accuracy and capture the dynamic behavior precisely. Compressible inviscid flows are also within the reach of PINNs. Mao *et al.*[34] embedded the Euler equations into a PINN to study supersonic aerodynamics, capturing the flow fields from only a few scattered points clustered randomly around the discontinuities.

Extensive studies have attempted to enhance the accuracy of PINNs. Rao *et al.*[35] proposed a PINN with a mixed-variable scheme to simulate steady and transient laminar flows and showed that this scheme improved the trainability and accuracy of the PINN. Zhu *et al.*[36] approached the Dirichlet boundary condition in a "hard" manner and chose the weights of distinct components of the loss function. Xu *et al.*[37] treated the physical equations as a parameterized constraint to explore the missing flow dynamics and unified the forms of the RANS equations and LES equations through an undetermined parameter ν_{eff}. Cheng and Tang[38] used Resnet blocks to enhance the stability of a PINN, while Sun *et al.*[39] designed a data-free PINN for incompressible flows and trained the network by minimizing the violation of flow governing equations, showing that "hard" boundary enforcement performs better than a "soft" boundary approach in data-free settings. Jin *et al.*[40] developed Navier–Stokes flow networks by encoding two different forms of the Navier–Stokes equations into neural networks and dynamically computed the data weights and components of the loss function to accelerate training and improve accuracy. Jagtap *et al.*[41] proposed a space–time domain decomposition method for PINNs. This extended PINN method efficiently lends itself to parallelized computation. The studies reviewed above prove that PINNs provide excellent tools for regressing flow fields from sparse observation data.

Given the advantages of PINNs, this paper presents a PINN-POD method that alleviates the dependence of POD on the quality and quantity of data. In our framework, the PINN acts as a preprocessor. The original flow fields are regressed by PINN from sparse measurements, and the regressed flow fields are then subjected to POD to extract the flow field structures. To efficiently fit a large training set formed by long-period observations, an overlapping temporal domain decomposition method is proposed. In this way, every decomposed time block is independently fitted by a subnet, with parallel training applied to accelerate the process.

The remainder of this paper is organized as follows. Section II introduces the detailed framework, settings, and parameters of the PINN-POD method. The PINN-POD method is then applied to flow fields with Reynolds numbers *Re* of 100, 3900, and 10 000 in Secs. III A–III C, respectively. The influence of noise in the observation data is investigated in Sec. III D. Finally, the conclusions to this study and prospects for future research are presented in Sec. IV.

II. METHODOLOGY

The framework of the PINN-POD method is illustrated in Fig. 1. The original sparsely sampled data are divided into *k* blocks covering equal periods of time. Each data block is then fitted by a subnet, and the equations governing the flow are inferred by automatic differentiation with backward propagation.[28] The learning rate is determined by the warm restart method.[42] At the end of each decay period, the convergence of the POD modes in the regressed flow fields and the value of the loss function are simultaneously evaluated, and the results are

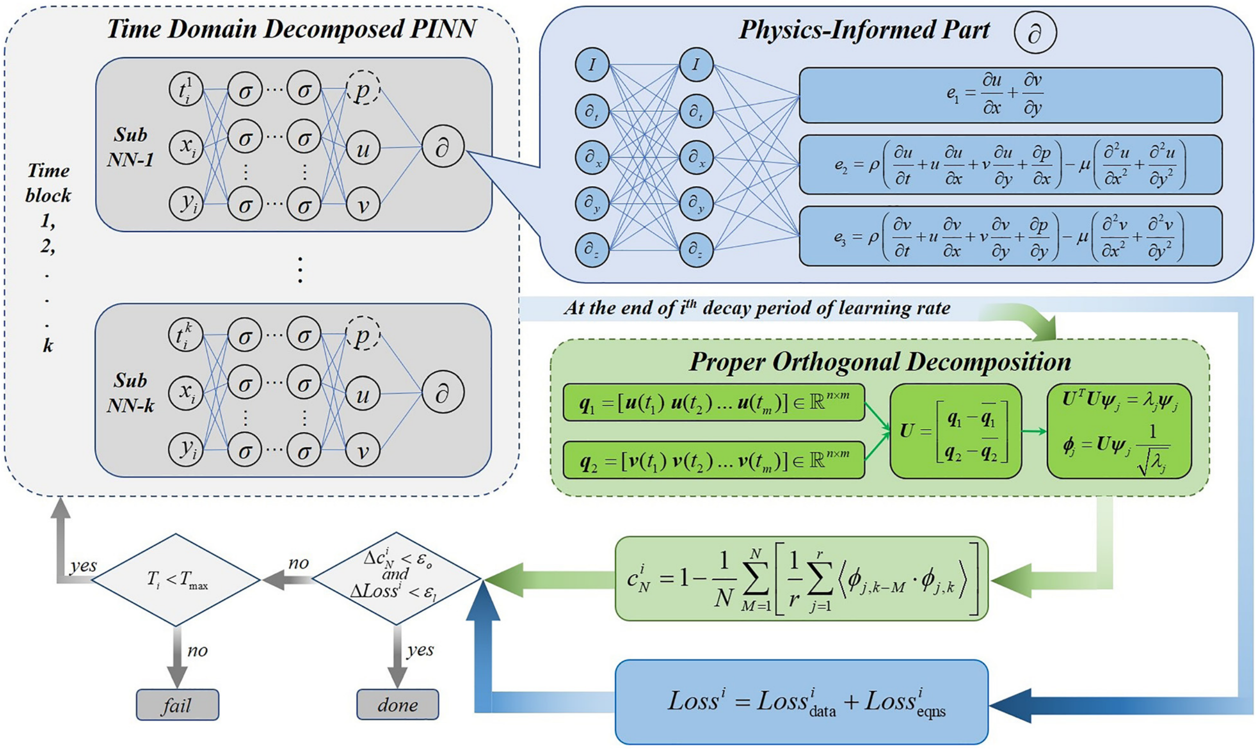

FIG. 1. Schematic of the PINN-POD framework. The blue parts show the route of the loss function, and the green parts show the route of POD. The learning rate is scheduled according to the warm restart method. The convergence of the POD modes and the value of the loss function are simultaneously evaluated at the end of each decay period.

used as stopping criteria for the training of the PINN-POD model. Once the stopping criteria have been satisfied, the full-spatiotemporal data can be regressed from the sparse measurements through the trained PINN, and then, the hidden correlated flow structures can be extracted from the regressed flow fields with POD.

A. Proper orthogonal decomposition

The POD method computes a series of dominant features and trends, known as "modes." The spatial modes correspond to coherent flow structures, and the mode coefficients reflect their temporal evolution.[11] The input to the POD method is a matrix $\boldsymbol{U}$ consisting of m column vectors $\{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_m}\}$. Each column vector $\boldsymbol{u_i} \in \mathbb{R}^n$ is a flattened snapshot of the flow field at n points from the ith moment in time. Specifically, the elements of $\boldsymbol{u_i}(\boldsymbol{x})$ are the fluctuating components of the quantity of interest (in this paper, velocity) in the flow field being studied at discrete spatial points $\boldsymbol{x}$ and discrete times t_i.

POD decomposes the flow field $\boldsymbol{q}(\boldsymbol{x}, t)$ into a set of basis functions and mode coefficients[11]

$$\boldsymbol{U} = [\boldsymbol{q}(\boldsymbol{x}, t) - \bar{\boldsymbol{q}}(\boldsymbol{x})] = \sum_j a_j(t)\boldsymbol{\phi}_j(\boldsymbol{x}), \quad t = t_1, t_2, \ldots, t_m, \tag{1}$$

where $[\boldsymbol{q}(\boldsymbol{x}, t) - \bar{\boldsymbol{q}}(\boldsymbol{x})]$ is the fluctuating component of the data vector $\boldsymbol{q}(\boldsymbol{x}, t)$ with its time-averaged value $\bar{\boldsymbol{q}}(\boldsymbol{x})$ removed, $\boldsymbol{\phi}_j(\boldsymbol{x})$ are the spatial basis functions or spatial modes, $a_j(t)$ are the mode coefficients, j is the order of modes, and m is the number of snapshots. The POD modes are orthonormal,[43] which means that the inner product between the modes satisfies

$$\langle \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \rangle = \begin{cases} 0, & j \neq k, \\ 1, & j = k, \end{cases} \tag{2}$$

where j and k denote the order of the modes. In this paper, the flow field data are decomposed using the snapshot POD method,[44] which relies on solving the following $m \times m$ eigenvalue problem

$$\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\psi}_j = \lambda_j\boldsymbol{\psi}_j, \quad \boldsymbol{\psi}_j \in \mathbb{R}^m, \tag{3}$$

where $\boldsymbol{U}^T\boldsymbol{U}$ is the temporal correlation matrix, and λ_j and $\boldsymbol{\psi}_j$ denote the eigenvalues and eigenvectors of $\boldsymbol{U}^T\boldsymbol{U}$, respectively. The POD modes $\boldsymbol{\phi}_j$ and temporal coefficients $a_j(t)$ can be written as

$$\boldsymbol{\phi}_j = \boldsymbol{U}\boldsymbol{\psi}_j\frac{1}{\sqrt{\lambda_j}} \in \mathbb{R}^n, \quad j = 1, 2, \ldots, m, \tag{4}$$

$$a_j(t) = \langle \boldsymbol{\phi}_j, \boldsymbol{U} \rangle, \tag{5}$$

or in matrix form as

$$\Phi = \boldsymbol{U}\Psi\boldsymbol{\Lambda}^{-1/2} \tag{6}$$

$$A = \Phi^T\boldsymbol{U}, \tag{7}$$

where $\boldsymbol{\Lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]$ is a diagonal matrix, and the eigenvalues λ_j are arranged in descending order. λ_j conveys how well the eigenvector $\boldsymbol{\psi}_j$ captures the original data in the L_2 sense. As the focus of this paper is the velocity field, the POD modes $\boldsymbol{\phi}_j$ are arranged in order of kinetic energy, meaning that the velocity field can be reconstructed by the first r modes

$$\boldsymbol{U} \approx \sum_{j=1}^{r} a_j(t)\boldsymbol{\phi}_j(\boldsymbol{x}). \tag{8}$$

Accordingly, the ratio of the kinetic energy in the reconstructed flow field to that in the original input flow field is defined as

$$\gamma = \sum_{j=1}^{r} \lambda_j \Big/ \sum_{j=1}^{m} \lambda_j, \tag{9}$$

where λ_j is the eigenvalue of the jth-order mode, and m is the total number of modes. When γ is close to 1, the primary characteristics of the original input flow field are well captured by the reconstructed flow field.

B. Physics-informed neural network

The inputs to the PINN are the temporal and spatial coordinates $(t, \boldsymbol{x})$, and the outputs are the variables of the solution vector for a system of PDEs, $\boldsymbol{u}(t, \boldsymbol{x})$. A time-dependent PDE system can be written as

$$\boldsymbol{u}_t + \mathcal{N}[\boldsymbol{u}] = 0, \quad \boldsymbol{x} \in \Omega, t \in [0, T], \tag{10}$$

where $\mathcal{N}[\cdot]$ denotes a nonlinear differential operator, $\boldsymbol{x}$ is the spatial coordinate vector defined over the domain Ω, $\boldsymbol{u}(t, \boldsymbol{x})$ is the solution vector of the PDE, and $\boldsymbol{u}_t$ is its derivative with respect to time t.[45] As shown in the physics-informed part of Fig. 1, the governing equations for the flow are the continuity equation and the 2D incompressible Naiver–Stokes equations

$$u_x + v_y = 0, \tag{11}$$

$$\rho(u_t + uu_x + vu_y + p_x) - \mu(u_{xx} + u_{yy}) = 0, \tag{12}$$

$$\rho(v_t + uv_x + vv_y + p_y) - \mu(v_{xx} + v_{yy}) = 0. \tag{13}$$

Thus, the inputs are (t, x, and y), and the outputs are (p, u, and v). The outputs are calculated through forward propagation according to the given inputs, which is the same as a classical fully connected neural network. Subsequently, the derivatives of the outputs with respect to the inputs can be calculated during backward propagation through automatic differentiation.[46] Therefore, the governing equations can be embedded into the loss function with the outputs and the calculated derivatives

$$\text{Loss} = \text{Loss}_{\text{data}} + \text{Loss}_{\text{eqns}}, \tag{14}$$

$$\text{Loss}_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i_d=1}^{N_{\text{data}}} \left[|u^{i_d} - u_{NN}^{i_d}|^2 + |v^{i_d} - v_{NN}^{i_d}|^2 \right], \tag{15}$$

$$\text{Loss}_{\text{eqns}} = \sum_{j=1}^{3} \frac{1}{N_{\text{eqns}}} \sum_{i_e=1}^{N_{\text{eqns}}} |e_j(t^{i_e}, x^{i_e}, y^{i_e})|, \tag{16}$$

where $\text{Loss}_{\text{data}}$ is the mean squared error between the measured velocity and the output velocity given by the PINN, and $\text{Loss}_{\text{eqns}}$ is the residual of the governing equations calculated in the physics-informed part. $\text{Loss}_{\text{data}}$ is used to quantify the difference between the PINN predictions and the real data $\{t^{i_d}, x^{i_d}, y^{i_d}, u^{i_d}, v^{i_d}\}_{i_d=1}^{N_{\text{data}}}$ measured by sensors. $\text{Loss}_{\text{eqns}}$ is a regularization mechanism that enforces the structure imposed by the governing equations at a finite set of equation points $\{t^{i_e}, x^{i_e}, y^{i_e}\}_{i_e=1}^{N_{\text{eqns}}}$. The number and position of the measuring points and equation points can be completely different.

C. PINN-POD

The PINN-POD method proposed in this paper extracts the POD modes from sparse velocity measurements. As shown in Fig. 1, the sparse measurements in k time segments are fitted by k sub-PINNs. High-resolution spatiotemporal data of the unsteady flow field is then inferred by the trained PINN. At the end of each decay period for the learning rate, the POD modes are extracted from the inferred flow field to evaluate the convergence of the PINN-POD model and the loss function. Once the stopping criteria are satisfied, the goal of extracting the POD modes from sparse measurements has been achieved.

PINN-POD provides excellent regression capabilities and obeys the implicit frequency principle in the training process of the neural network. That is, the target function is fitted from low to high frequencies during the training process.[47] Low-frequency features usually play important roles in flow fields, and the low-order, high-energy POD modes often have a low dominant frequency.[9,11,12] Therefore, PINN-POD has great potential to extract the main features of the flow field. To capture the key flow structures and their long-term evolution, multiple snapshots should be used for POD. However, a large training set that is formed by the long-term observations is difficult to train with a single PINN. Consequently, time-domain decomposition is introduced, as shown in the dashed gray box of Fig. 1, whereby k time blocks are fitted in parallel by k subnets. Considering the size of the observation data in this paper, k is set to 10.

To prevent poor performance by the PINN at the beginning and end of each time block, adjacent time blocks are overlapped. As illustrated in Fig. 2, the first time block t_{blk1} has 120 snapshots. Sub-NN-1 is trained using snapshots $[0, 120]$, while only snapshots $[10, 110]$ are chosen for POD. The second time block t_{blk2} overlaps with 20 snapshots at the end of t_{blk1}. The snapshots for training sub-NN-2 are $[100, 220]$, while only snapshots $[110, 210]$ are chosen for POD. After training, all regressed snapshots are reconnected to form the POD input.

The PINN parameters are initialized using Xavier's algorithm and optimized through the Adam adaptive optimizer.[48] The learning

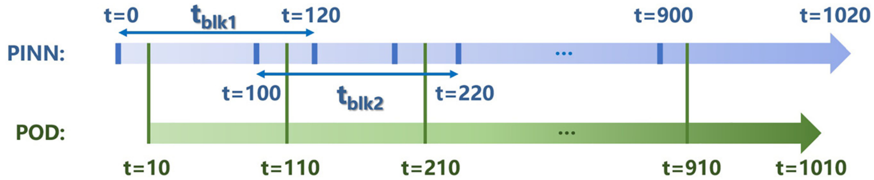

FIG. 2. Time block overlap. The adjacent time blocks used to train sub-PINNs share 20 snapshots. The middle 100 snapshots of each time block are combined to calculate POD modes.

rate is scheduled according to the warm restart method[42] to accelerate the convergence and improve the accuracy

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta^i_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{T_{\max}}{T_i}\pi\right)\right), \tag{17}$$

where η_t is the learning rate of a certain epoch, i indicates the ith decay period, $\eta^i_{\max}$ and $\eta_{\min}$ denote the maximum and minimum learning rates in a decay period, $T_{\max}$ is the number of epochs since the last restart, and T_i is the number of epochs in the ith decay period. $\eta^i_{\max}$ and T_i are determined by

$$\eta^{i+1}_{\max} = M_{\mathrm{mul}} * \eta^i_{\max}, \tag{18}$$

$$T_{i+1} = T_{\mathrm{mul}} * T_i. \tag{19}$$

Following previous numerical experiments and Xu *et al.*,[32] M_{mul}, T_{mul}, $\eta^0_{\max}$, $\eta_{\min}$, and T_0 are set to 1.0, 2.0, 10^{-3}, 10^{-8}, and 10^3, respectively. The resulting learning rate for each epoch is shown in Fig. 3.

To save computational resources, the convergence of POD modes is only evaluated at the end of every decay period of the learning rate. The convergence of modes is evaluated using

$$c^i_N = 1 - \frac{1}{N}\sum_{M=1}^{N}\left[\frac{1}{r}\sum_{j=1}^{r}\langle \boldsymbol{\phi}_{j,k-M}\cdot\boldsymbol{\phi}_{j,k}\rangle\right], \tag{20}$$

where $\boldsymbol{\phi}_{j,k}$ corresponds to the jth-order mode of snapshots of all k time blocks, $\boldsymbol{\phi}_{j,k-M}$ corresponds to the jth-order mode of snapshots of the first $k - M$ time blocks, $\langle\cdot\rangle$ denotes the scalar product, r denotes the first r modes, i corresponds to the ith decay period of the learning rate, and N denotes the number of sets used to evaluate the convergence. As POD provides an optimal low-rank approximation to a matrix U, the scalar product of perfectly converged modes with different snapshots should be 1. In practice, however, $\langle \boldsymbol{\phi}_{j,k-M}\cdot\boldsymbol{\phi}_{j,k}\rangle$ only approaches 1. Setting r and N to 6 and 3 in all examples means that the first 6 modes of the first 700 snapshots, 800 snapshots, and 900 snapshots are compared with the 1000 snapshots of the reference set. If $c^i_N \approx 0$, then sets with fewer than 700, 800, and 900 snapshots resemble the full set with 1000 snapshots, and the modes are considered to have converged. The convergent modes will not change, even when the number of snapshots increases. The loss function Loss^i represents the fitting error between the observations and the governing equations at the end of the ith decay period. The variation of c^i_N, Loss^i in two adjacent decay periods of the learning rate is defined as

$$\Delta c^i_N = c^i_N - c^{i-1}_N, \tag{21}$$

$$\Delta\mathrm{Loss}^i = \mathrm{Loss}^i - \mathrm{Loss}^{i-1}. \tag{22}$$

As shown in Fig. 1, Δc^i_N and $\Delta\mathrm{Loss}^i$ are evaluated at the end of every decay period. When Δc^i_N and $\Delta\mathrm{Loss}^i$ are less than ε_o, ε_l for two consecutive decay periods, the PINN-POD model is considered to have converged. The convergence criteria ε_o and ε_l are set to 10^{-2}, which balances accuracy against computational efficiency.

The PINN-POD model was developed in the open-source deep-learning framework TensorFlow. Considering the size of the time block and the training efficiency, each subnet consisted of 10 hidden layers and 50 neurons per layer. The training process of the 10 subnets was distributed and run in parallel on 10 NVIDIA Tesla V100 GPUs. Each subnet was assigned to a single GPU.

III. RESULTS AND DISCUSSION

The flow over a 2D circular cylinder was studied at $Re = 100$, 3900, and 10 000. The observation data used to train the PINN-POD model were sampled from numerical simulations. The computational domain, mesh, and measurement point distribution are shown in Fig. 4. The diameter of the cylinder $D = 1$ m. To ensure that y^+ is less than 1 in all cases, the height of the first layer near the cylinder is set to 10^{-4} m. The left and right sides of the domain are set as a velocity inlet and a pressure outlet, respectively, and the top and bottom of the domain are assigned as symmetric boundaries. The cylinder wall is assigned the no-slip condition. In Fig. 4, the crosses indicate the sparse measuring points, which are arranged along the yellow dotted line. The sampling time interval is 0.1 s, i.e., a sampling frequency of 10 Hz, which is easy to implement in experiments.

To verify the accuracy of the numerical simulations, Table I summarizes the results for the flow around a cylinder observed in experiments and given by numerical simulations at various Reynolds numbers. The results obtained in the present study are broadly consistent with the reference values, indicating that the sampled velocity data from the wake can be used to train the PINN-POD model.

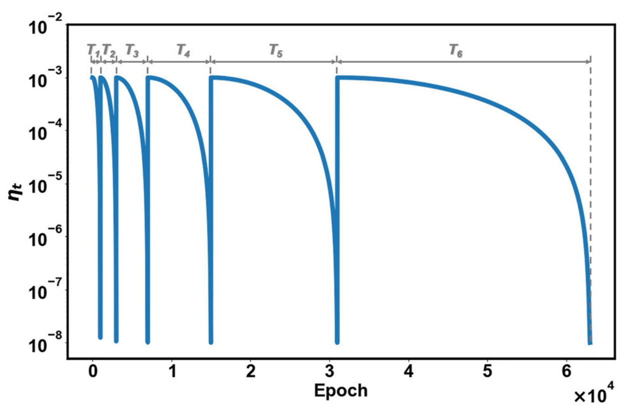

FIG. 3. Learning rate schedule. In each decay period, the learning rate decreases from 10^{-3} to 10^{-8}. The number of epochs contained in each decay period is doubled in the next period.

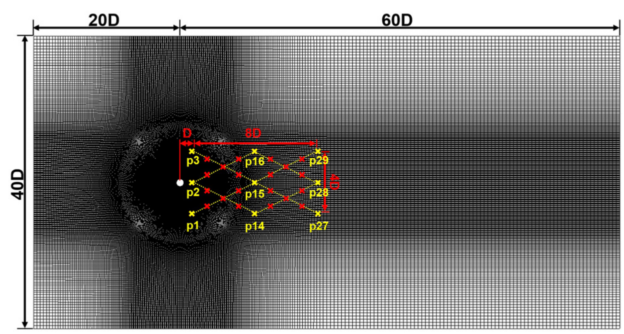

FIG. 4. Computational mesh of flow around a cylinder and measuring point distribution. Twenty-nine sensors, represented by crosses, are arranged over the domain $x \in [D, 9D]$, $y \in [-2D, 2D]$.

TABLE I. Drag coefficient and Strouhal number of flow around a cylinder at various Reynolds numbers.

Re	Case	Cd	St
100	Exp. (Williamson[49])	···	0.160
	CFD 3D DNS (Henderson[50])	1.349	0.166
	CFD 2D laminar (Rahman[51])	1.245	0.164
	Present CFD case 1 2D laminar	1.346	0.164
3900	Exp. (Norberg[52])	···	0.210
	CFD 3D LES (Lysenko[53])	0.970	0.209
	CFD 2D k-ε (Rahman[51])	0.997	0.200
	Present CFD case 2 2D k-ε	0.922	0.208
10 000	Exp. (Norberg[52])	···	0.201
	CFD 3D k-ω SST (Rosetti[54])	1.520	0.240
	CFD 2D k-ω SST (Stringer[55])	1.555	0.236
	Present CFD case 3 2D k-ω SST	1.587	0.242

Moreover, the POD modes of the velocity fields given by the numerical simulations can be used as references to assess the accuracy of the flow fields reconstructed by the proposed PINN-POD method. Considering the inherent noise in real measurements, we add different levels of Gaussian noise to the observation data in Sec. III D.

To analyze the accuracy of the PINN-POD method quantitatively, the relative L_2 error is introduced as

$$\varepsilon(u_{\text{reg}}, u_{\text{ref}}) = \frac{\frac{1}{N_s}\sum_{i=1}^{N_s}\left[u_{\text{reg}}(x_i) - u_{\text{ref}}(x_i)\right]^2}{\frac{1}{N_s}\sum_{i=1}^{N_s}\left[u_{\text{ref}}(x_i) - \frac{1}{N_s}\sum_{i=1}^{N_s}u_{\text{ref}}(x_i)\right]^2}, \tag{23}$$

where $\{x_i : i = 1, \ldots, N_s\}$ are N_s points scattered in the whole domain of interest, and u_{reg}, u_{ref} are the velocities reconstructed by the first r PINN-POD modes of the regressed fields and the traditional POD modes of the numerical simulation, respectively. The above definition has the favorable property that it is invariant under the shifting and scaling of both the reconstructed and reference functions; i.e., $\varepsilon(\beta u_{\text{reg}} + \alpha, \beta u_{\text{ref}} + \alpha) = \varepsilon(u_{\text{reg}}, u_{\text{ref}})$ for any constants α and $\beta \neq 0$.[29]

A. Case 1: $Re = 100$

First, a simple laminar flow over a cylinder at $Re = 100$ is studied. The flow fields are generated by laminar numerical simulations, and the velocities observed at various measuring points (yellow crosses in Fig. 4) are shown in Fig. 5. The measured values have been shifted by certain values for clarity. For each sensor, velocity data were collected over 1020 consecutive time steps. As illustrated in Fig. 2, the sampled data were split into 10 blocks with 20 snapshots overlapping, corresponding to $N^S_{\text{data}} = 29$ spatial points and $N^T_{\text{data}} = 120$ temporal points, and a total of $N_{\text{data}} = N^S_{\text{data}} \times N^T_{\text{data}} = 3480$ data points in one time block. Equation points were added to penalize the residual of the governing equations. For every time block, the equation points were distributed uniformly, with $N^S_{\text{eqns}} = 5000$ spatial points and $N^T_{\text{eqns}} = 120$ temporal points, giving a total of $N_{\text{eqns}} = N^S_{\text{eqns}} \times N^T_{\text{eqns}} = 6 \times 10^5$ equation points. The training set for each subnet included all the data points $\{t^{i_d}, x^{i_d}, y^{i_d}, u^{i_d}, v^{i_d}\}_{i_d=1}^{N_{\text{data}}}$ and equation points $\{t^{i_e}, x^{i_e}, y^{i_e}\}_{i_e=1}^{N_{\text{eqns}}}$ in the corresponding time block. Approximately 1.4 s was required to train the model for one epoch using this training set.

The loss function with respect to the training epoch is shown in Fig. 6. The loss value of the blue curve corresponds to the $Re = 100$ case. As shown in Fig. 6, the training process with $Re = 100$ terminates after the fourth cycle of learning rate decay. To evaluate the accuracy of the flow structures extracted from sparse data by PINN-POD, the traditional POD modes of the CFD results are taken as references. Comparisons are presented in Figs. 7–9.

Figure 7 compares λ_j and γ from the PINN-POD modes and traditional POD modes. According to the definition of λ_j and γ in Eq. (9), λ_j measures the kinetic energy of the jth-order mode, and γ represents the proportion of kinetic energy captured by the first r modes to the total kinetic energy of the flow field. The results indicate that the PINN-POD method captures the energy of the flow fields well, and more than 99% of the total energy is captured by the first six modes. In addition, the energy of each PINN-POD mode is basically in line with the reference value.

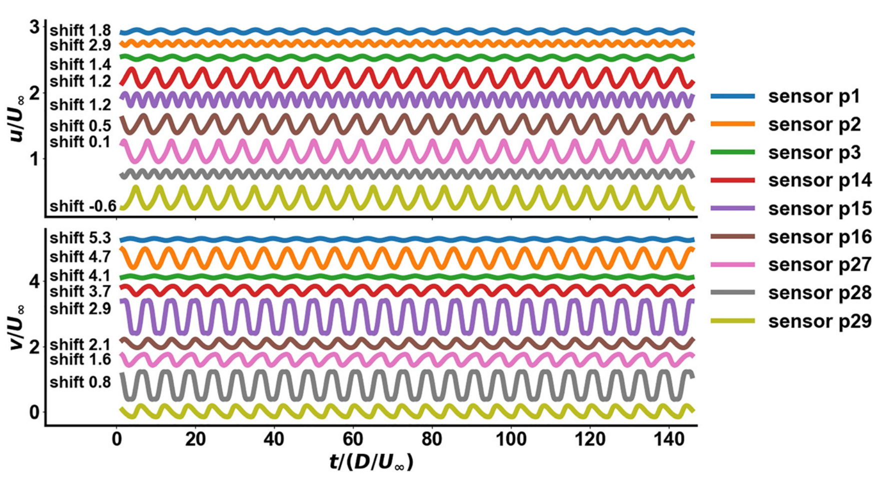

FIG. 5. Velocity at various measuring points ($Re = 100$). The values have been shifted by certain values.

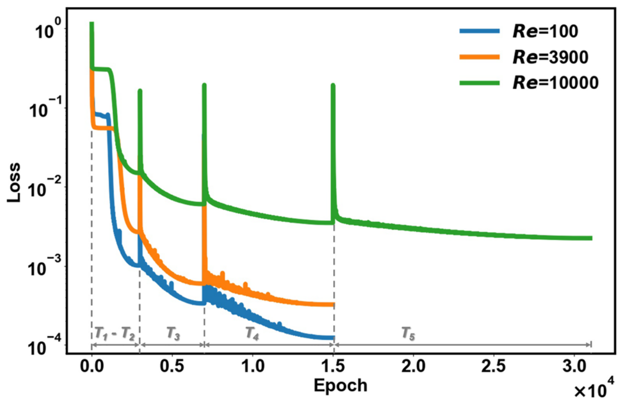

FIG. 6. Loss-epochs at different Reynolds numbers. The dotted line divides the learning rate decay period.

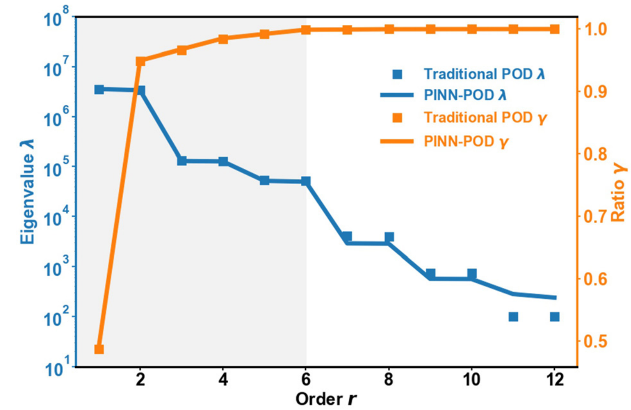

FIG. 7. Energy and ratio of cumulative energy to total energy of PINN-POD modes and referenced CFD results at $Re = 100$.

To evaluate the accuracy of the flow structures captured by PINN-POD, Fig. 8 intuitively shows that the first six modes extracted by the PINN-POD method are in good agreement with the traditional POD modes. To analyze the accuracy of the PINN-POD method quantitatively, the relative L_2 error given by Eq. (23) is evaluated. As the first six modes capture almost all of the energy in the velocity fields, we compare the relative L_2 error of the reconstructed flow fields between the PINN-POD modes and traditional POD modes with

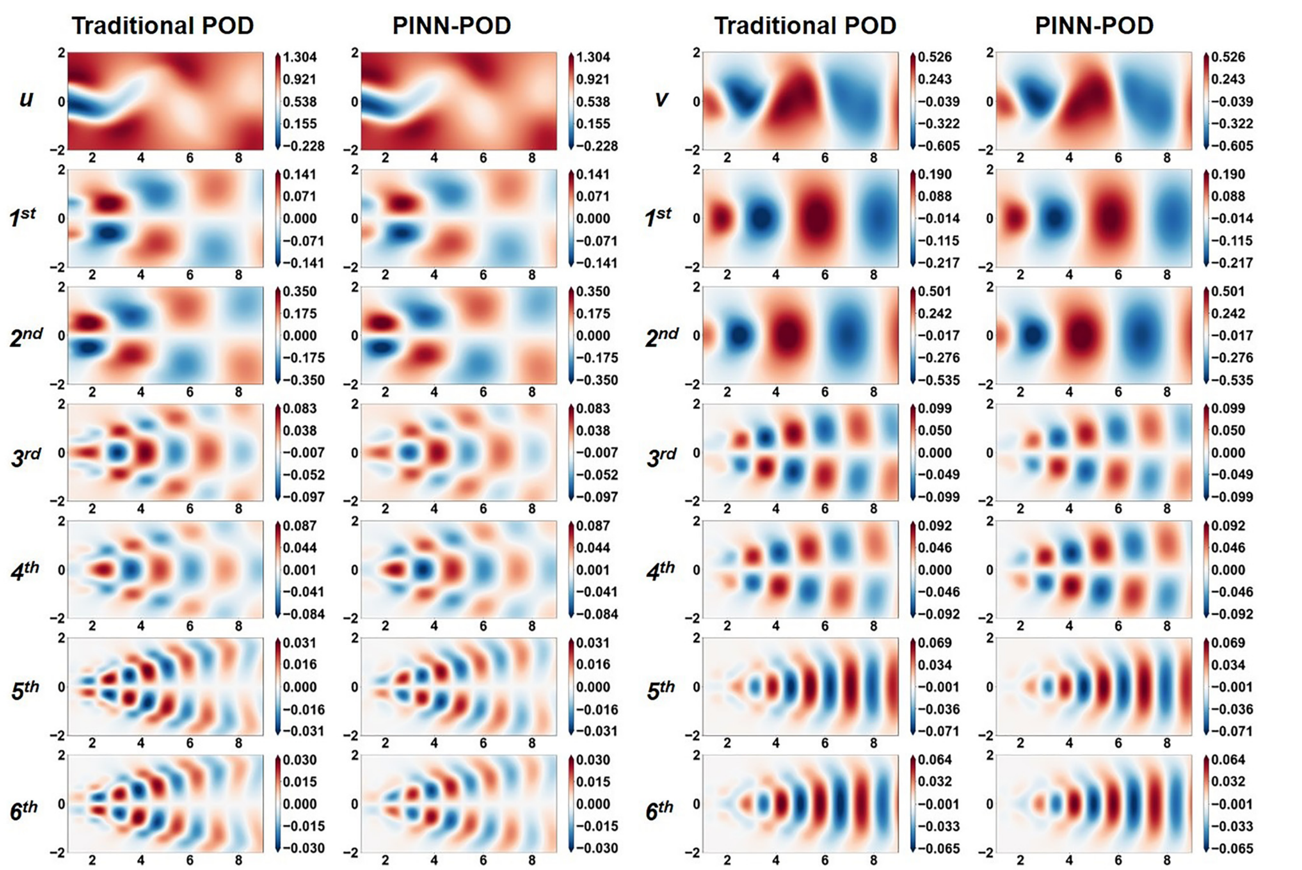

FIG. 8. Comparison of POD modes and PINN-POD modes at $Re = 100$, first snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

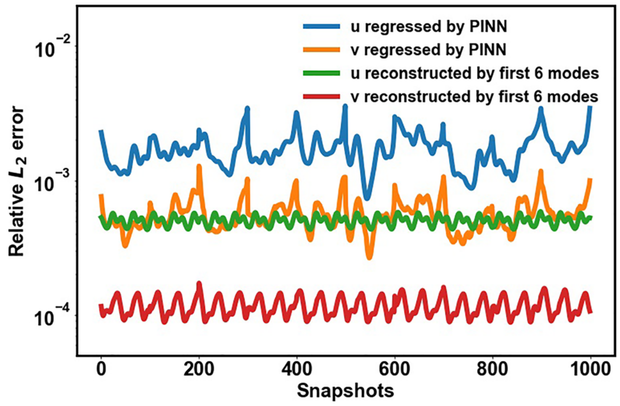

FIG. 9. Relative L_2 error of regressed and reconstructed velocity fields at $Re = 100$.The relative L_2 error of the regressed velocity is obtained by comparing the flow field of PINN regression with the flow field of CFD. The relative L_2 error of the reconstructed velocity compares the first six PINN-POD modes with the first six POD modes. PINN-POD modes are obtained from 29 sensor measurements by our PINN-POD method, while POD modes are obtained from full-spatiotemporal CFD data.

$r = 6$ in Eq. (8). As illustrated in Fig. 9, the relative L_2 error oscillates around a small value. These results demonstrate that the PINN-POD method successfully reconstructs the velocity fields and accurately extracts the fine spatial structures at $Re = 100$.

The time coefficient represents the evolution of spatial structures. As the time coefficients of the POD modes contain multiple frequencies, the fast Fourier transform is applied to extract the dominant frequency. To compare the Strouhal number (St) of the lift coefficient (see Table II), the dominant frequency is converted to the dimensionless frequency $St = fD/U_\infty$. For the first six modes, St is compared in Table II. The comparison demonstrates that the frequency characteristics of the spatial modes are accurately captured by the PINN-POD modes. In addition, the frequency of the first two modes matches the lift coefficient closely, which indicates that the first two modes are related to the vortex shedding on the cylinder surface. This provides further evidence that the first two modes capture the most significant features of the flow field.

TABLE II. Strouhal number of modes in various order. Inputs of traditional POD are full spatiotemporal data, whereas inputs of PINN-POD are the sparse measurements.

Re	Origin	Order of mode	St
100	Traditional POD	1 and 2	0.166
		3 and 4	0.332
		5 and 6	0.498
	PINN-POD	1 and 2	0.166
		3 and 4	0.332
		5 and 6	0.498
3900	Traditional POD	1 and 2	0.208
		3 and 4	0.424
		5 and 6	0.629
	PINN-POD	1 and 2	0.208
		3 and 4	0.424
		5 and 6	0.629
10 000	Traditional POD	1 and 2	0.256
		3 and 4	0.775
		5 and 6	0.512
	PINN-POD	1 and 2	0.256
		3 and 4	0.775
		5 and 6	0.512

B. Case 2: $Re = 3900$

The turbulent flow over a cylinder at $Re = 3900$ is now studied. The viscous model used for the numerical simulations is the k–ε model. The velocities observed at the same positions as for the $Re = 100$ case are shown in Fig. 10. The sampling settings and training set construction are the same as for the $Re = 100$ case. The orange curve in Fig. 6 shows the change in the loss function with respect to the epoch at $Re = 3900$. The training process of the $Re = 3900$ case terminates after the fourth cycle of learning rate decay.

Similar to Sec. III A, the results of the proposed PINN-POD method are compared with the reference traditional POD results. In terms of energy, Fig. 11 demonstrates that the PINN-POD method

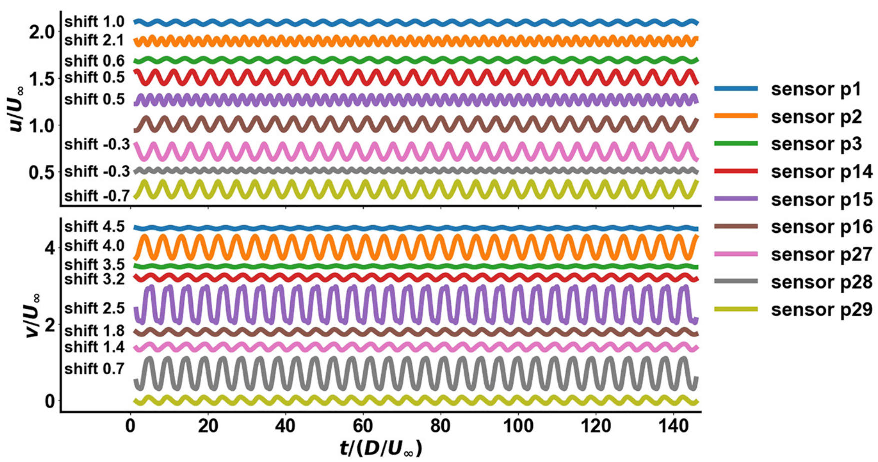

FIG. 10. Velocity at various measuring points ($Re = 3900$). The values have been shifted by certain values.

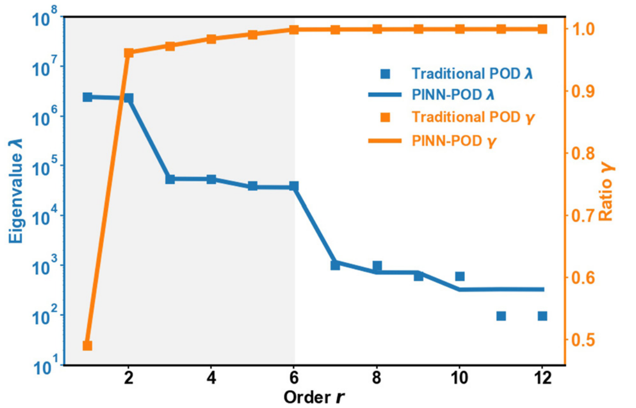

FIG. 11. Energy and ratio of cumulative energy to total energy of PINN-POD modes and referenced CFD results at $Re = 3900$.

also performs well at $Re = 3900$. More than 99% of the total energy is captured by the first six modes, and the energy of each PINN-POD mode is in good agreement with that of the traditional POD. The flow structures represented by the first six modes are shown in Fig. 12. These results clearly show that the position and symmetry of the vortices are accurately captured.

Quantitative analysis of the accuracy of the velocity fields reconstructed using the first six modes was performed using the relative L_2 error. Figure 13 compares the relative L_2 error of each time snapshot. The results demonstrate that the PINN-POD method reconstructs the velocity fields and extracts the flow structures well at $Re = 3900$. The relative L_2 error is slightly higher than that for case 1. As for the time coefficient, Table II demonstrates that the frequency characteristics of the spatial modes are accurately captured by the PINN-POD modes. The frequency of the first two modes is consistent with the lift coefficient. Therefore, the first two modes are again the dominant structures at $Re = 3900$. In summary, the main features of the flow field are accurately captured, and the PINN-POD method achieves good performance in the case of turbulent flow.

C. Case 3: $Re = 10\,000$

Finally, we consider the turbulent flow over a cylinder at $Re = 10\,000$. The viscous model used in the numerical simulations is the k-ω SST model, which is different from that used in case 2. The velocities observed at the same positions as for the $Re = 100$ case are shown in Fig. 14. The sampling settings and training set construction are the

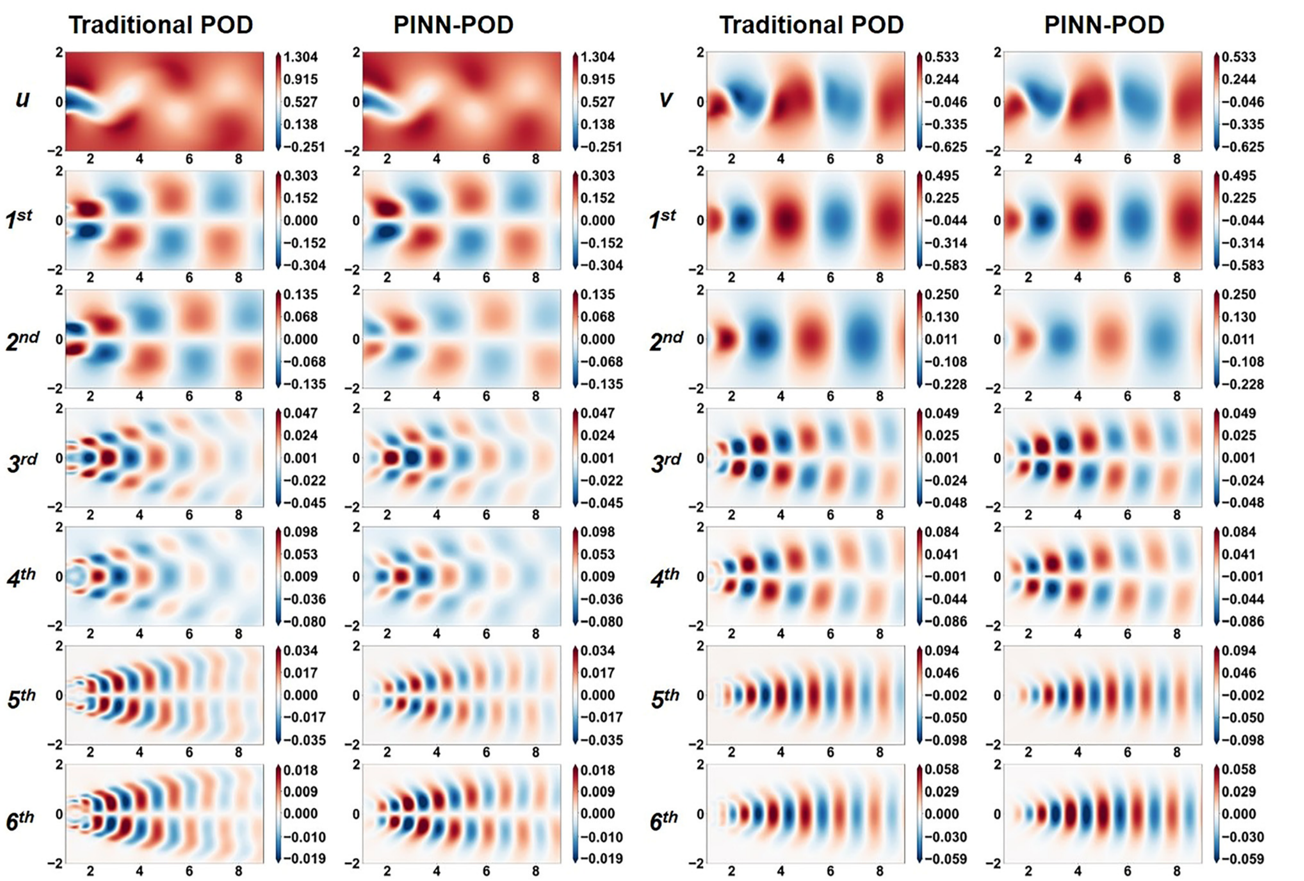

FIG. 12. Comparison of POD modes and PINN-POD modes at $Re = 3900$, 20th snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

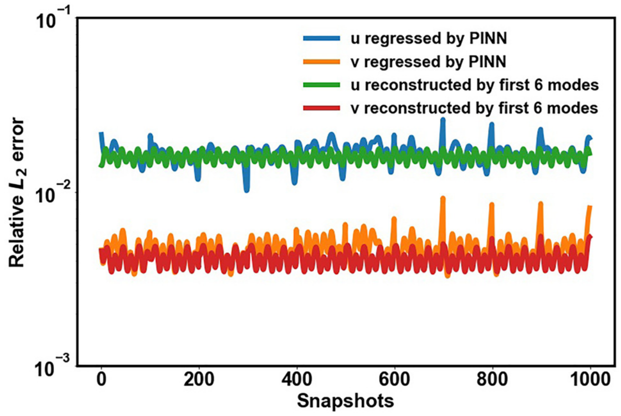

FIG. 13. Relative L_2 error of regressed and reconstructed velocity fields at $Re = 3900$. The relative L_2 error of the regressed velocity is obtained by comparing the flow field of PINN regression with the flow field of CFD. The relative L_2 error of the reconstructed velocity compares the first six PINN-POD modes with the first six POD modes. PINN-POD modes are obtained from 29 sensor measurements by our PINN-POD method, while POD modes are obtained from full-spatiotemporal CFD data.

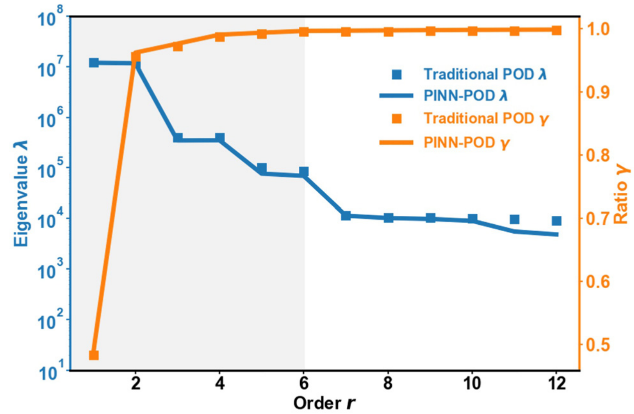

FIG. 15. Energy and ratio of cumulative energy to total energy of PINN-POD modes and referenced CFD results at $Re = 10\,000$.

same as for the $Re = 100$ case. The change in the loss function with respect to the number of epochs at $Re = 10\,000$ is described by the green curve in Fig. 6. The training process in the $Re = 10\,000$ case stops after the fifth cycle of learning rate decay; thus, more epochs are required for convergence than that with $Re = 100$ and $Re = 3900$. Figure 6 shows that the loss value decreases slowly in the $Re = 10\,000$ case, and the final loss value is larger than that in the other two cases. This is related to the implicit frequency property of the neural network, as mentioned in Sec. I.

We now compare the modes extracted from the sparse measurements by the trained PINN-POD model and the traditional POD modes extracted from the original numerical simulation results by the classical POD algorithm. In terms of energy, Fig. 15 shows that more than 99% of the total energy is captured by the first six modes, and the energy of each PINN-POD mode is relatively consistent with the reference values. The flow structures shown in Fig. 16 demonstrate that the position and symmetry of the vortices are captured relatively well. In the $Re = 10\,000$ case, the pattern is essentially different from that in the $Re = 100$ and $Re = 3900$ cases in terms of the location, size, and distribution of the vortex pairs. The vortex centers in the first- and second-order modes with $Re = 10\,000$ are closer to the centerline, while the third- and fourth-order modes have only two rows, indicating that the dissipation of the vortices through viscosity becomes weaker as Re increases.

The relative L_2 error of the velocity fields reconstructed from the first six modes is illustrated in Fig. 17. These results demonstrate that the PINN-POD method reconstructs the velocity fields and extracts the flow structures well at $Re = 10\,000$. Although the training loss is larger for $Re = 10\,000$ than that for $Re = 3900$, the relative L_2 error does not increase significantly compared with the lower-Re case. As seen in Table II, the dominant frequency of the spatial modes is excellently captured by the PINN-POD method. The frequency of the first two modes remains at the same level as the lift coefficient. Therefore, the proposed PINN-POD method achieves good accuracy in both laminar and turbulent cases.

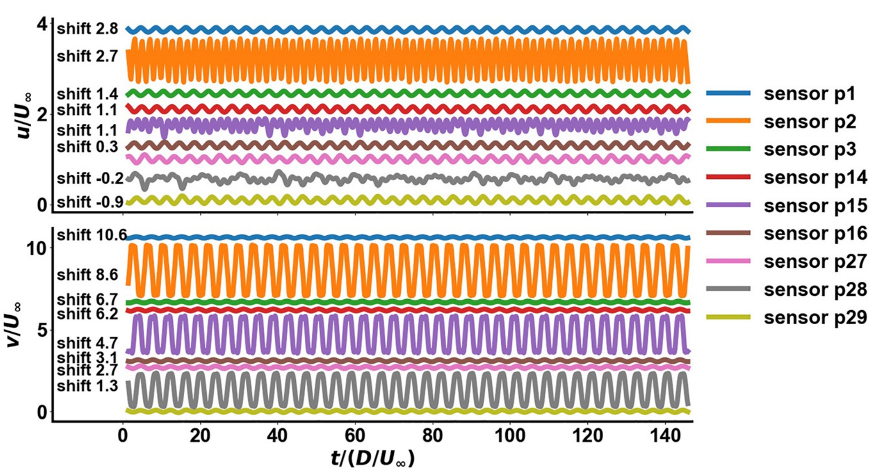

FIG. 14. Velocity at various measuring points ($Re = 10\,000$). The values have been shifted by certain values.

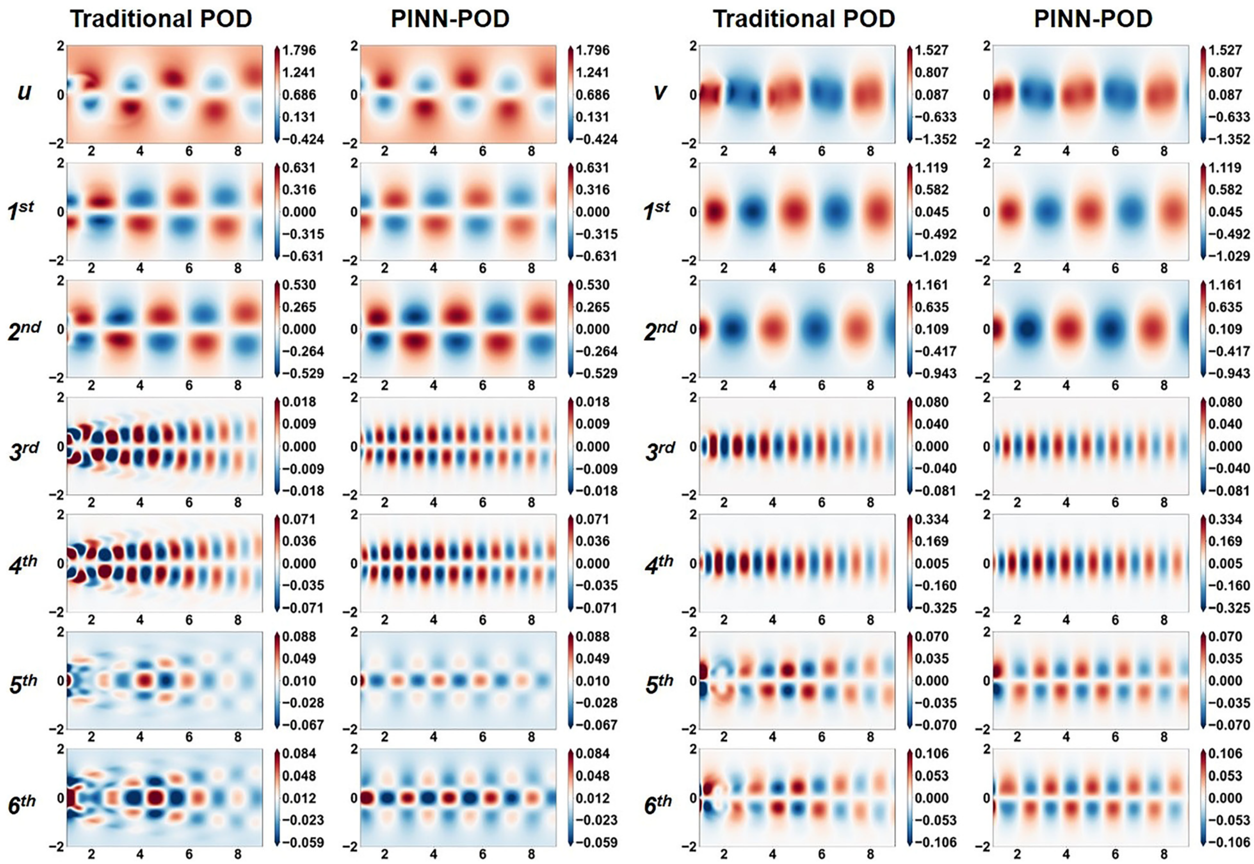

FIG. 16. Comparison of POD modes and PINN-POD modes at $Re = 10\,000$, fourth snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

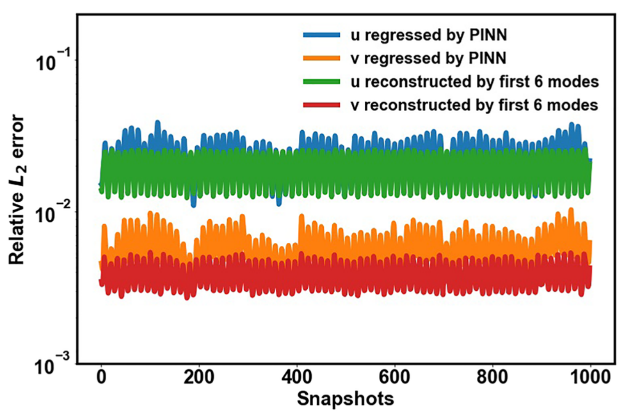

FIG. 17. Relative L_2 error of regressed and reconstructed velocity fields at $Re = 10\ 000$. The relative L_2 error of the regressed velocity is obtained by comparing the flow field of PINN regression with the flow field of CFD. The relative L_2 error of the reconstructed velocity compares the first six PINN-POD modes with the first six POD modes. PINN-POD modes are obtained from 29 sensor measurements by our PINN-POD method, while POD modes are obtained from full-spatiotemporal CFD data.

D. Influence of noisy data

Noise is almost unavoidable in measurements conducted in laboratory settings. Thus, to study the robustness of the proposed PINN-POD method, its performance with various noise levels added to the sensor measurements is now explored. The noise is assumed to follow a Gaussian distribution with zero mean and σ% standard deviation.[24] The samples obtained from the numerical simulations in Secs. III A–III C had noise added to give the noisy training set. Measurements with 20% noise added at various points are shown in Fig. 18.

The training settings are consistent with those described in the previous section. Compared with the cases without noise, the convergence speed is slower in the presence of noise. The variation in the loss during the training process for $Re = 100$, 3900, and 10 000 with various noise levels is shown in Figs. 19–21. There is a clear upward trend in the loss as the noise ratio increases. The accuracy of the captured modes also decreases to some extent. Figure 22 shows the mean relative L_2 error and the standard deviation of the velocity fields reconstructed using the first six modes at various noise ratios. As the noise ratio increases from 0% to 20%, the relative L_2 error in the $Re = 100$ case rises sharply by roughly an order of magnitude; in the $Re = 3900$

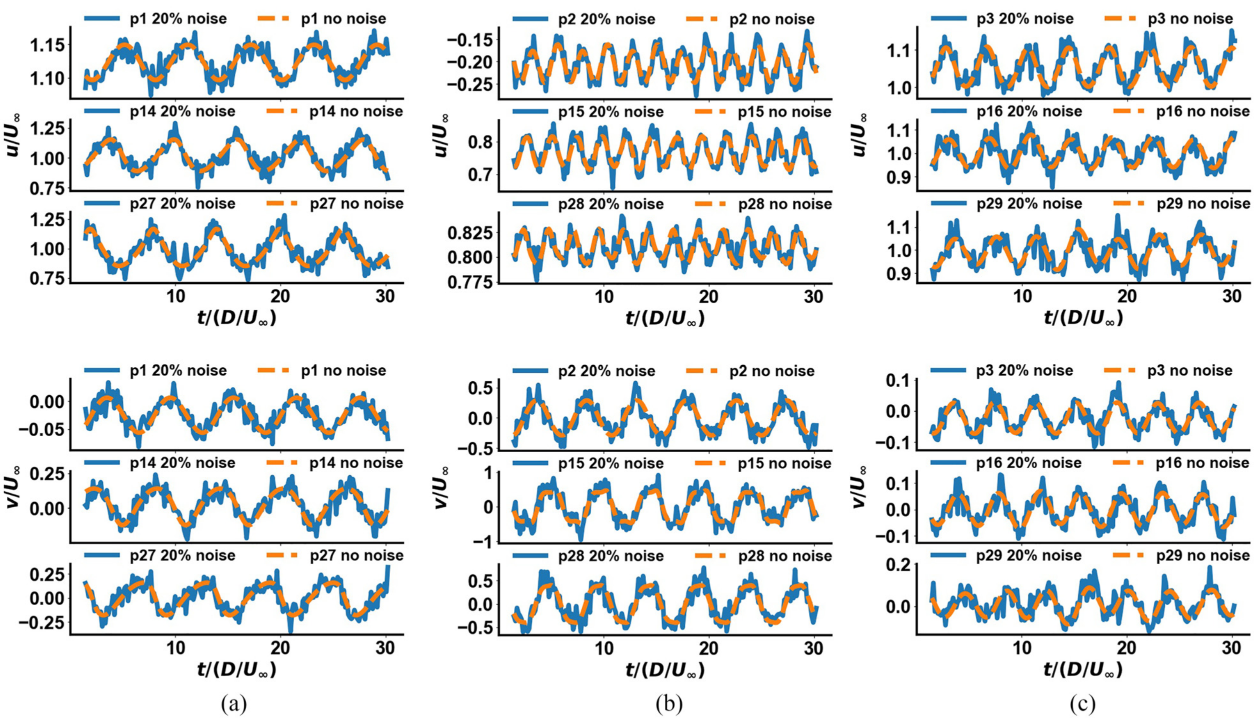

FIG. 18. Velocity at various measuring points with 20% noise for different *Re*. The first line represents the streamwise velocity, and the second line represents the transverse velocity. The columns represent different Reynolds numbers. (a) $Re = 100$, (b) $Re = 3900$, and (c) $Re = 10\,000$.

and 10 000 cases, the error increases only slightly. However, the relative L_2 error remains lower in the $Re = 100$ case than that in the other two cases. This is because low-order modes mainly capture low-frequency features, while high-frequency features come to play a dominant role as *Re* increases. The high-frequency features are harder to capture by neural networks than low-frequency features.

The flow structures at $Re = 100$, 3900, and 10 000 with the highest noise ratio are illustrated in Figs. 23–25. Compared with the cases without noise, the position of the vortex is broadly accurate, but the errors in the velocity amplitude have increased. This is in line with the relative L_2 error. The noisy data do not affect the accuracy of the time coefficient. Table III indicates that *St* corresponding to the dominant frequency of the PINN-POD modes extracted from the noisy data is in good agreement with that given by the traditional POD modes. These results demonstrate that the PINN-POD method provides an accurate and robust means of extracting the flow structures from noisy observations for both simple laminar flows and complex turbulent flows.

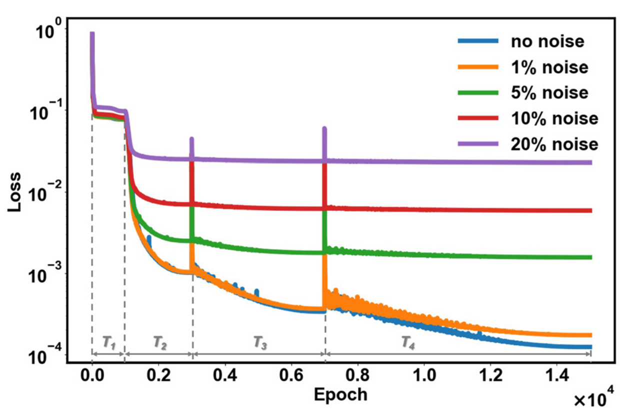

FIG. 19. Loss-epochs of $Re = 100$ with various noise ratios. The dotted line divides the learning rate decay period.

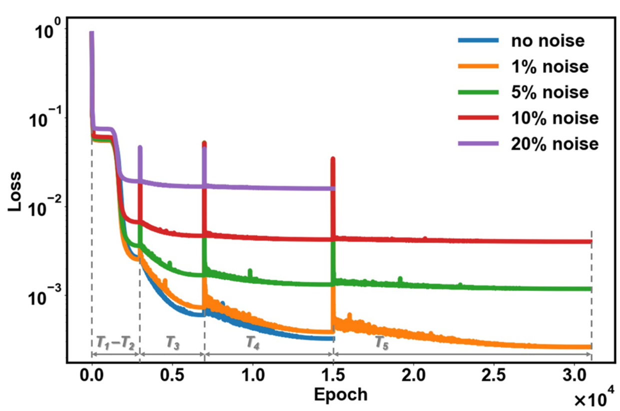

FIG. 20. Loss-epochs of $Re = 3900$ with various noise ratios. The dotted line divides the learning rate decay period.

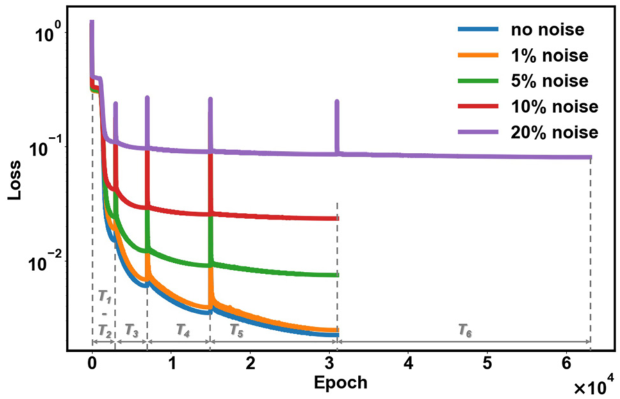

FIG. 21. Loss-epochs of $Re = 10\,000$ with various noise ratios. The dotted line divides the learning rate decay period.

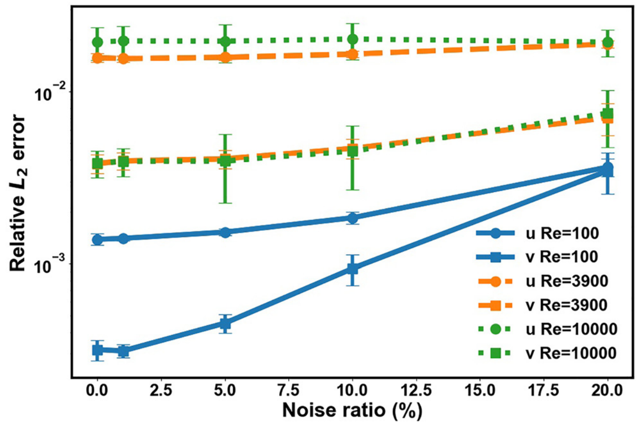

FIG. 22. Relative L_2 error of reconstructed velocity fields at various noise ratios. Values are expressed as mean ± standard deviation.

IV. CONCLUSIONS

The methodology proposed in this paper provides a way to extract hidden flow structures from sparse measurements. This method has been validated by examining flows at various Reynolds numbers. In contrast to the traditional POD method, PINN-POD requires only sparse observation data. The flow fields are regressed from these sparse observation data by several sub-PINNs. To determine the convergence of the proposed method, the convergence of

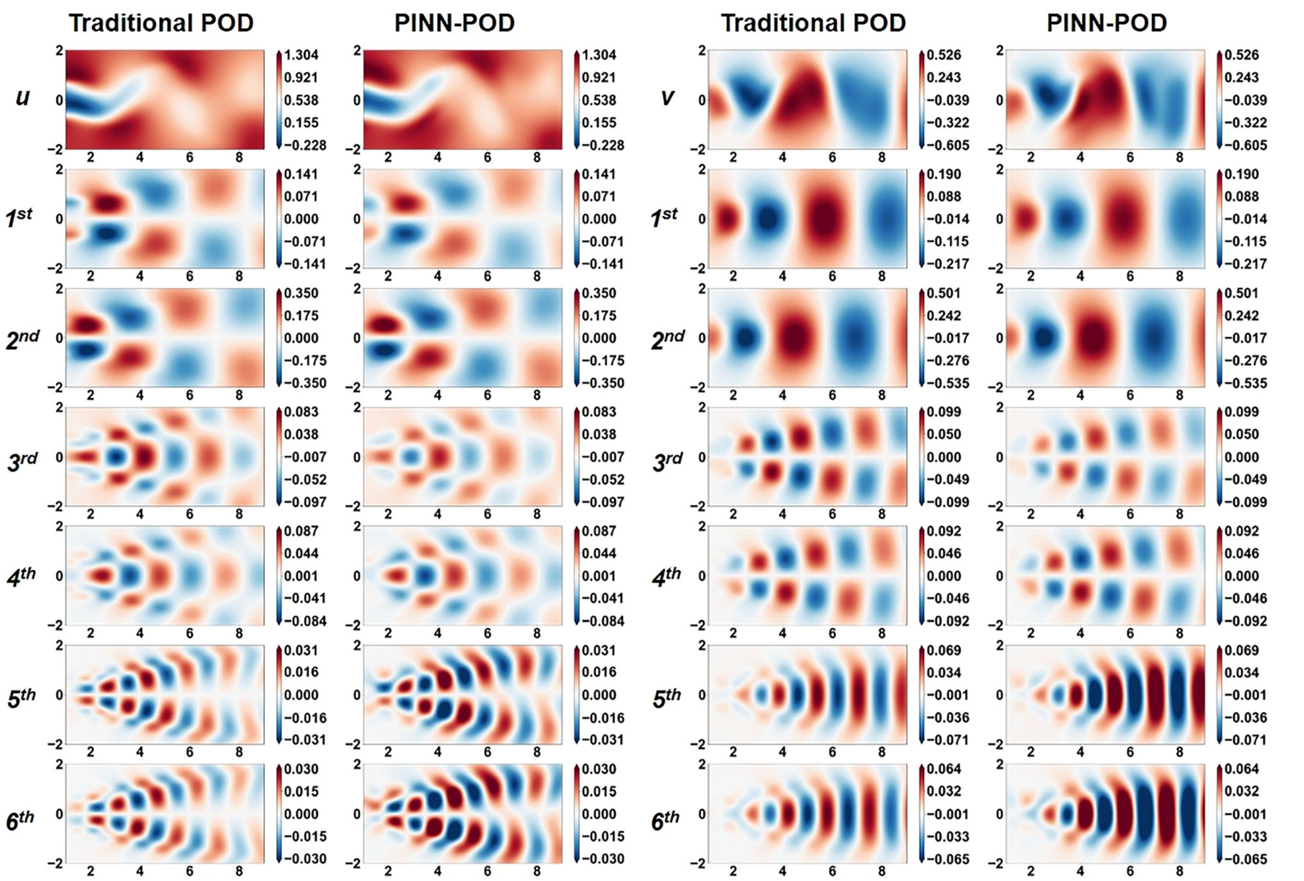

FIG. 23. Comparison of POD modes and PINN-POD modes with 20% noise ratio at $Re = 100$, first snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

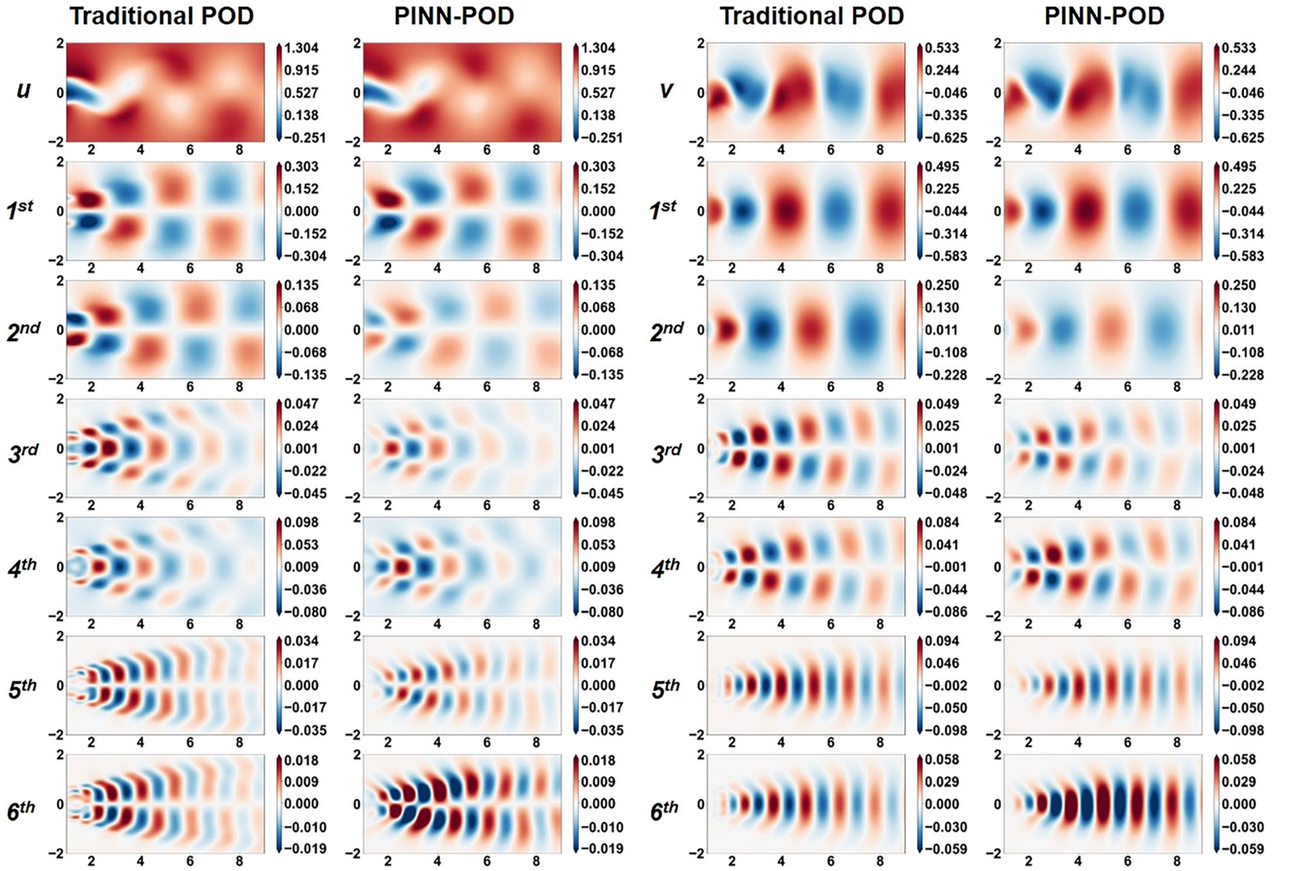

FIG. 24. Comparison of POD modes and PINN-POD modes with 20% noise ratio at $Re = 3900$, 20th snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

the PINN-POD modes is considered alongside the common loss function.

Unlike traditional POD, which requires high-resolution data, the flow patterns of a cylinder wake can be obtained by the proposed PINN-POD method from only 29 measuring points. The first two modes are usually representative of the dominant structures. Visualizations showed that the structures contained in the first six PINN-POD modes match the traditional POD results. Moreover, as the first six PINN-POD modes contain more than 99% of the energy in the flow domain, the velocity fields reconstructed by PINN-POD and traditional POD are highly consistent. The mean relative L_2 errors in u and v, as reconstructed by the first six modes, between PINN-POD and traditional POD are 1.385×10^{-3} and 3.166×10^{-4} at $Re = 100$, 1.577×10^{-2} and 3.827×10^{-3} at $Re = 3900$, and 1.946×10^{-2} and 3.841×10^{-3} at $Re = 10\,000$. As Re increases, the mean relative L_2 error gradually rises, which indicates that it is becoming less easy to train the PINN-POD model.

All extracted flow structures under various noise levels were in acceptable agreement with the traditional POD results. In the worst case ($Re = 10\,000$ with a noise ratio of 20%), the mean relative L_2 errors in u and v were 1.940×10^{-2} and 7.490×10^{-3}, respectively. In addition, the time coefficients of the PINN-POD modes are highly consistent with the traditional POD modes. For the first- and second-order PINN-POD modes, St is approximately equal to that of the lift coefficient, indicating that the evolution of the dominant structures is accurately captured by these PINN-POD modes. The training data sampled for the $Re = 100$, 3900, and 10 000 cases were generated by different viscosity models, demonstrating the robustness of the PINN-POD framework.

In summary, the proposed PINN-POD method provides an accurate and robust framework for extracting flow structures from sparse observation data. Compared with the classical POD method, the PINN-POD method has great potential for use in experimental fluid mechanics due to its low dependency on data. Although only 2D incompressible cases have been considered in this work, the proposed framework could be extended to a variety of flows, such as 3D incompressible or compressible flows, by changing the physics-informed part. The distribution of measurement points, learning rate scheduling strategy, and hyperparameters of the neural networks could also be further optimized. However, for flows at high Re

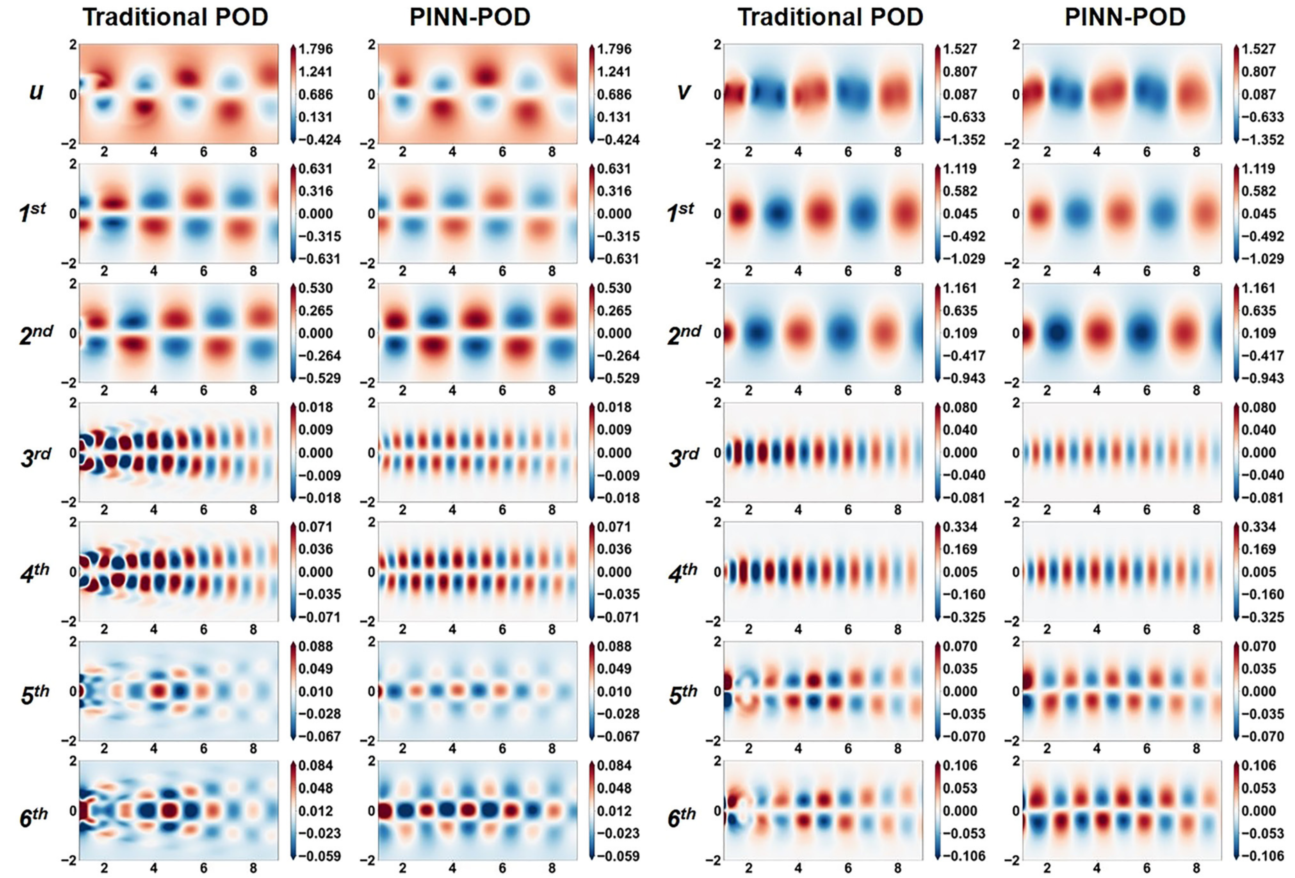

FIG. 25. Comparison of POD modes and PINN-POD modes with 20% noise ratio at $Re = 10\,000$, 40th snapshot. The two columns on the left represent the streamwise velocity, and the two columns on the right represent the transverse velocity. The first line represents the original flow field, and the remaining lines represent the modes of each order.

TABLE III. *St* of modes in various noisy cases.

Order	Noise ratio	$Re = 100$	$Re = 3900$	$Re = 10\,000$
1 and 2	1%	0.166	0.208	0.256
	5%	0.166	0.208	0.256
	10%	0.166	0.208	0.256
	20%	0.166	0.208	0.256
3 and 4	1%	0.332	0.422	0.775
	5%	0.332	0.422	0.775
	10%	0.332	0.422	0.775
	20%	0.332	0.422	0.775
5 and 6	1%	0.498	0.629	0.512
	5%	0.498	0.629	0.512
	10%	0.498	0.629	0.512
	20%	0.498	0.629	0.512

values, reducing the error of PINN-POD remains a challenging issue.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Youth Innovation Promotion Association CAS (Grant No. 2019020) and the National Natural Science Foundation of China (Grant No. 52006232).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Chang Yan: Conceptualization (lead); Data curation (lead); Formal analysis (equal); Investigation (equal); Methodology (lead); Software

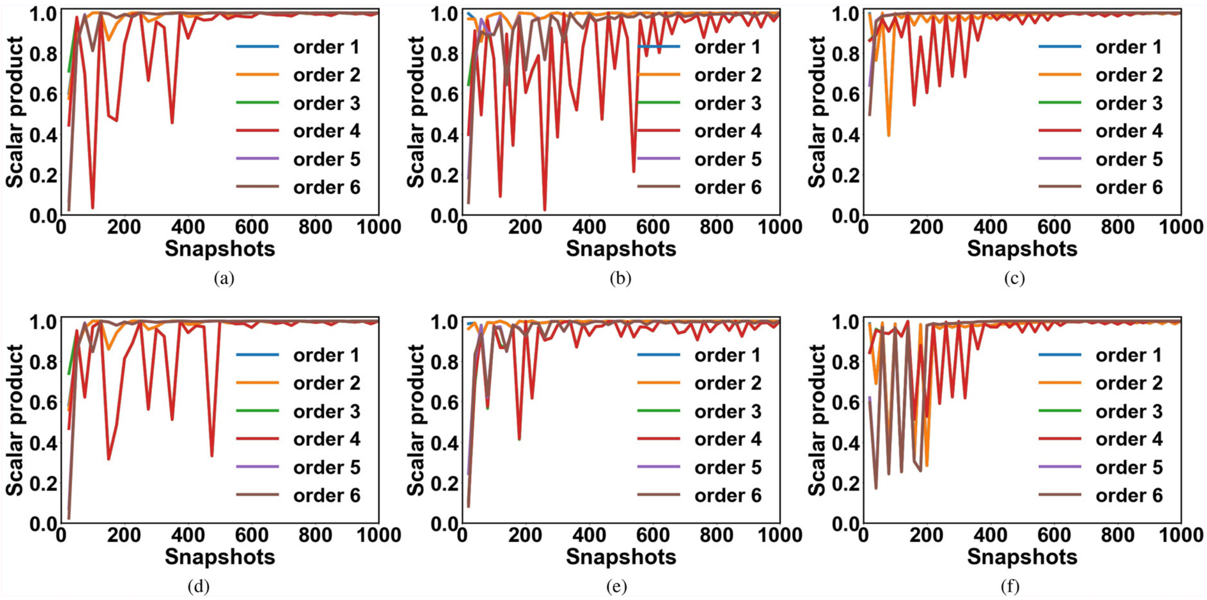

FIG. 26. Convergence of POD modes and PINN-POD modes with the number of snapshots at various Re: (a) $Re = 100$ POD modes, (b) $Re = 3900$ POD modes, (c) $Re = 10\,000$ POD modes, (d) $Re = 100$ PINN-POD modes, (e) $Re = 3900$ PINN-POD modes, and (f) $Re = 10\,000$ PINN-POD modes.

(lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Shengfeng Xu:** Formal analysis (equal); Investigation (equal); Software (equal); Validation (lead); Writing – review & editing (equal). **Zhenxu Sun:** Funding acquisition (equal); Project administration (lead); Supervision (equal). **Dilong Guo:** Funding acquisition (lead); Project administration (equal); Supervision (equal). **Shengjun Ju:** Formal analysis (equal); Funding acquisition (equal); Project administration (equal); Supervision (equal). **Renfang Huang:** Funding acquisition (equal); Project administration (equal); Writing – review & editing (equal). **Guowei Yang:** Formal analysis (equal); Funding acquisition (equal); Methodology (equal); Project administration (equal); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: CONVERGENCE VALIDATION OF MODES

The POD method is sensitive to the number of snapshots; this section studies the effect of snapshot numbers on convergence of modes. The convergence of modes is evaluated by scalar product.[9] The set of modes obtained from 1000 snapshots is selected as the base set. Then, the scalar product between the set of modes obtained from different snapshots and the base set is calculated to evaluate the convergence. As long as the scalar product approaches 1 as the number of snapshots increases, the mode is convergent. Figure 26 illustrates the change of scalar product of POD modes and PINN-POD modes as the number of snapshots increases at various Re. The results show that the convergence of POD modes and PINN-POD modes gradually improves with the increase in snapshots and reaches a good convergence after the number of snapshots is greater than 600. In addition, the first- and second-order mode converge fastest, and the higher order modes converge slower. Therefore, 1000 snapshots are used for the cases in this paper, which are sufficient to extract convergent modes.

APPENDIX B: CONVERGENCE CRITERIA OF PINN-POD

This section shows how the PINN-POD model meet the mode convergence criteria and loss function criteria during training. Tables IV–VI show the evaluation at the end of each learning rate decay period at $Re = 100$, $Re = 3900$, and $Re = 10\,000$, respectively.

TABLE IV. Criteria of mode convergence and loss function at the end of each learning rate decay period ($Re = 100$).

Noise ratio	Criterion	T_1	T_2	T_3	T_4	T_5	T_6
0%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		
1%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		
5%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		
10%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		
20%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		

TABLE V. Criteria of mode convergence and loss function at the end of each learning rate decay period ($Re = 3900$).

Noise ratio	Criterion	T_1	T_2	T_3	T_4	T_5	T_6
0%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		
1%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓	✓	
5%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓	✓	
10%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓	✓	
20%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓		
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓		

TABLE VI. Criteria of mode convergence and loss function at the end of each learning rate decay period ($Re = 10\,000$).

Noise ratio	Criterion	T_1	T_2	T_3	T_4	T_5	T_6
0%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✓	✓	✓	
1%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✗	✓	✓	
5%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✗	✓	✓	
10%	$\Delta c_N^i < 10^{-2}$	✗	✗	✓	✓	✓	
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✗	✓	✓	
20%	$\Delta c_N^i < 10^{-2}$	✗	✗	✗	✗	✓	✓
	$\Delta \mathrm{Loss}^i < 10^{-2}$	✗	✗	✗	✓	✓	

The maximum number of learning rate decay periods is 6, and the criterion is satisfied if the value is less than 10^{-2} in this paper. The mark ✓ means that the criterion is met, and the mark ✗ means that the criterion is not met. When the criteria are met in two consecutive decay periods, the PINN-POD model is considered to have converged. If the criteria are still not satisfied after reaching the maximum decay period (6 here), it means that the case is divergent. The divergent case needs to be adjusted, such as resetting parameters, increasing the number of sensors, etc.

APPENDIX C: INFLUENCE OF SENSOR PLACEMENT AND NUMBER

The position and number of sensors do have a significant impact on the accuracy of capturing the feature of wake flow.[24] In this paper, the PINN is employed to regress the full-spatiotemporal flow fields. Therefore, accuracy of the regressed flow field is essential, which is closely related to the placement strategy of the sensors.[32] In this section, the accuracy of four groups of placement

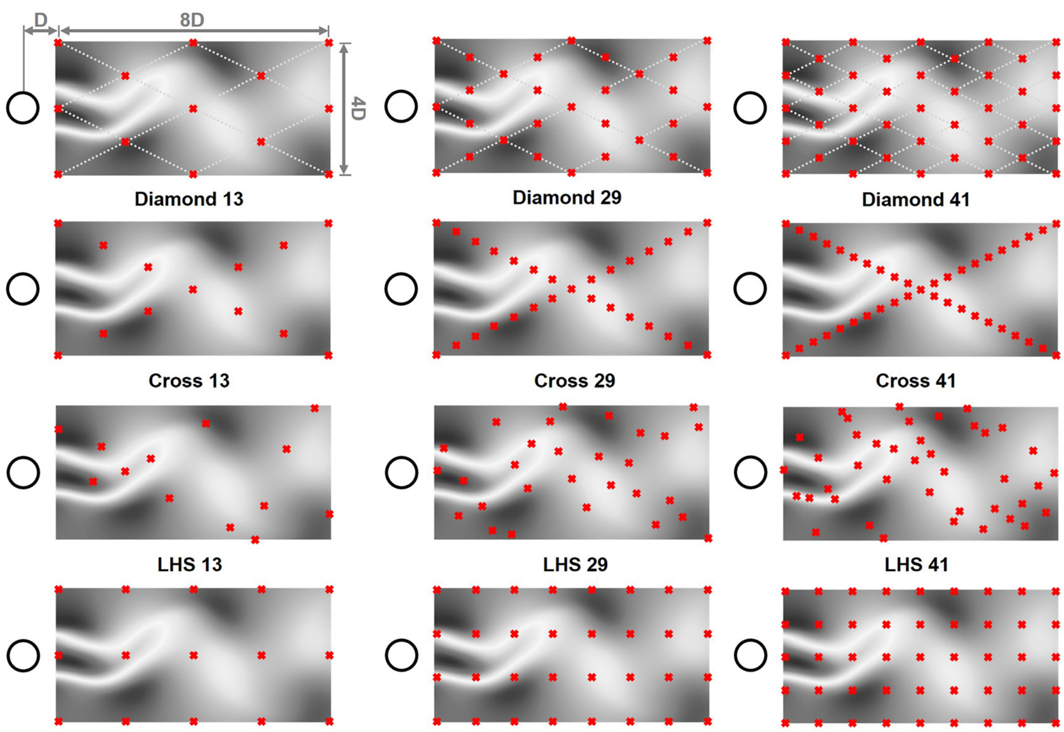

FIG. 27. Various placement methods and quantities of sensor.

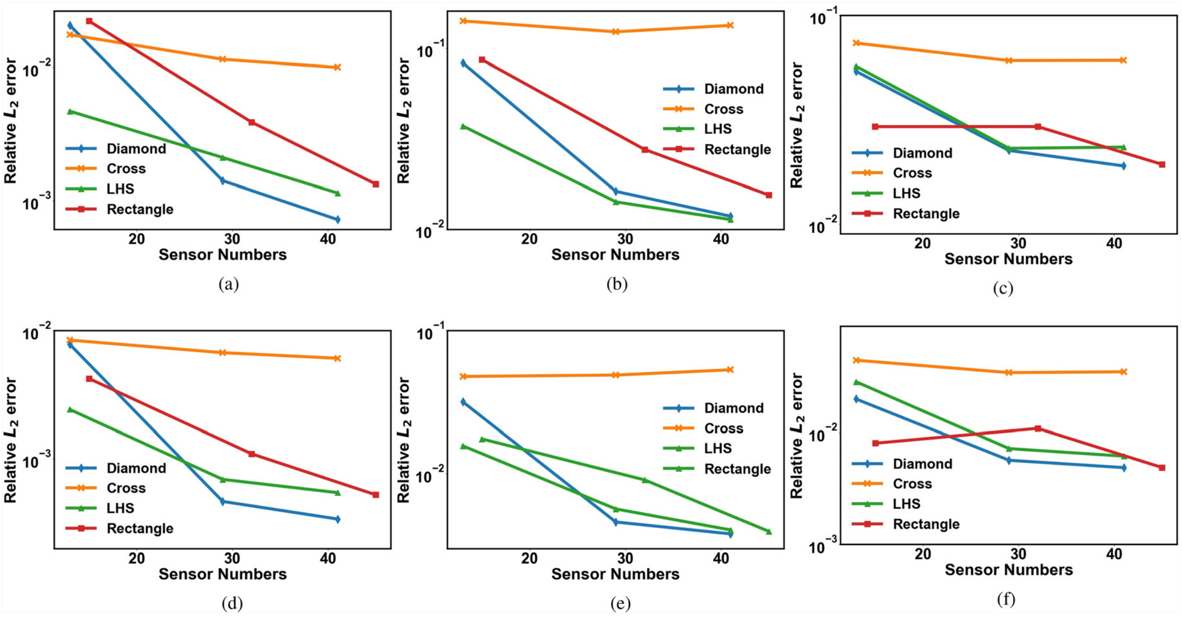

FIG. 28. Relative L_2 error of regressed velocity by PINN from different sensor placement strategies: (a) streamwise velocity u at $Re = 100$, (b) streamwise velocity u at $Re = 3900$, (c) streamwise velocity u at $Re = 10\,000$, (d) transverse velocity v at $Re = 100$, (e) transverse velocity v at $Re = 3900$, and (f) transverse velocity v at $Re = 10\,000$.

strategy is studied, and each group has three quantities of sensors. As illustrated in Fig. 27, the four placement strategies are diamond, cross, Latin hypercube sampling (LHS), and rectangle, respectively. The relative L_2 error of regressed velocity with various placement strategies and numbers of sensor is compared in Fig. 28. The results show that the accuracy of the regressed flow field improves with the increase in the number of sensors generally. Moreover, the accuracy of diamond placement is the best in most cases, and the accuracy will enhance stably with the increase in measuring points, while the cross placement gets the worst accuracy. Considering the need to test our PINN-POD framework to explore hidden flow structure from the data as sparse as possible, the diamond placement with 29 sensors is adopted in this paper.

APPENDIX D: TRAINING A PINN-POD MODEL USING LES DATA

In this section, to investigate the performance of the PINN-POD framework in dealing with more complex turbulence, the PINN-POD model was trained by the wake flow field data obtained from the LES of a circular cylinder at $Re = 3900$. The mesh and computation settings are referenced from Jiang *et al.*[56] However, while LES are typically three-dimensional, our PINN-POD framework has been designed to be two-dimensional at present. To obtain a 2D wake field, the span-average of the target wake region was performed. The streamwise velocity of the original and span-averaged wake is shown in Fig. 29. The wake field obtained by span-averaging the 3D LES data is more turbulent than that obtained by simulating with a 2D k-ε turbulent model in Sec. III B. The velocity data sampled from span-averaged wake were employed to train our PINN-POD model. To make a comparison with the case at the same Re presented in Sec. III B, velocities at the same 29 positions were sampled for 1020 time instances, which were divided into ten blocks with 20 snapshots overlapping between adjacent blocks. The parameters of PINN-POD mode were consistent with those of other cases in Sec. III B. The model reached convergence after four annealing cycles.

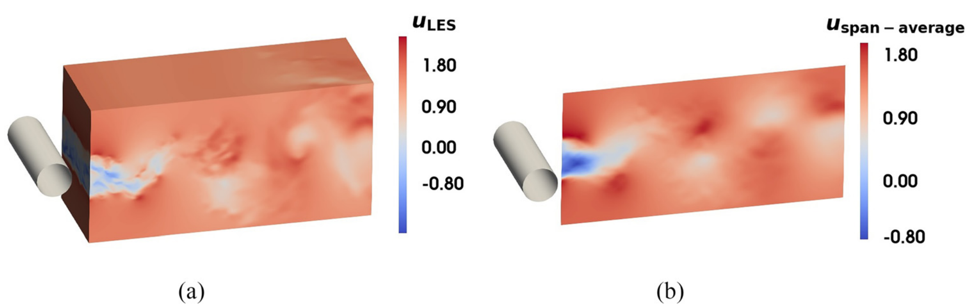

FIG. 29. Streamwise velocity u_{LES} and its span-averaged value $u_{span-average}$ obtained from LES of the target wake region: (a) original 3D wake and (b) span-averaged wake.

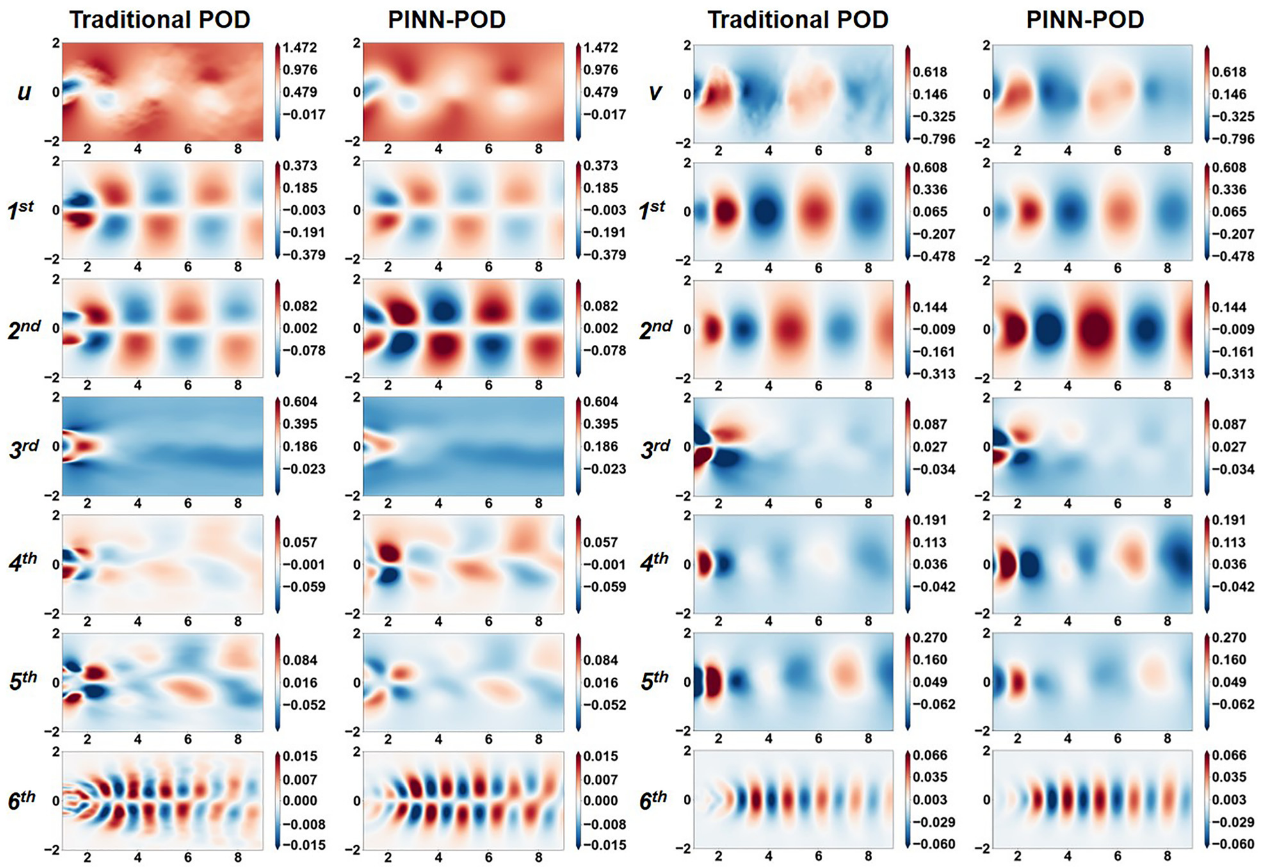

FIG. 30. Comparison of POD modes and PINN-POD modes from LES data at $Re = 3900$, 20th snapshot.

Figure 30 demonstrates that the first six modes obtained by the PINN-POD method are in good concurrence with the traditional POD modes. The comparison between Fig. 30 and 12 shows that the primary flow pattern of the Karman vortex shedding in the wake of the cylinder at $Re = 3900$ is captured by the first and second PINN-POD modes, regardless of whether the training data are based on RANS or LES simulations. However, there are some differences in the higher-order modes, which is due to the fact that the LES data contain abundant high-frequency fluctuations. LES is capable of resolving smaller-scale vortices, unlike RANS data which can only resolve large-scale vortices, the energy distribution of the modes in the LES flow field is less concentrated than that in the RANS data.

The energy and ratio of cumulative energy to total energy of PINN-POD modes and referenced LES results are compared in Fig. 31. The results indicate that the PINN-POD method captures the energy of the flow fields well, with the first six modes accounting for over 90% of the total energy, which is slightly lower than the 99% captured in case 2. However, the energy of each PINN-POD mode is basically in line with the reference value. Furthermore, the relative L_2 error of the velocity fields reconstructed from the first six modes is illustrated in Fig. 32. The PINN-POD modes obtained from LES data reconstruct the velocity field well, but no better than those obtained from RANS data in Sec. III B. This suggests that to explore the hidden flow structures from more complex 3D turbulence, it is necessary to extend the proposed 2D PINN-POD framework to 3D and to study in more detail the effects of various parameters in 3D cases.

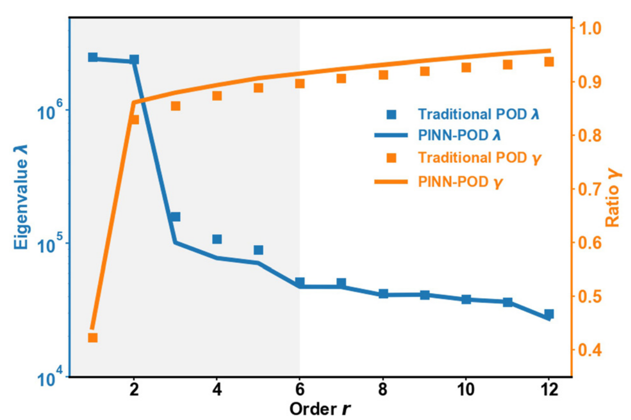

FIG. 31. Energy and ratio of cumulative energy to total energy of PINN-POD modes and referenced LES results at $Re = 3900$.

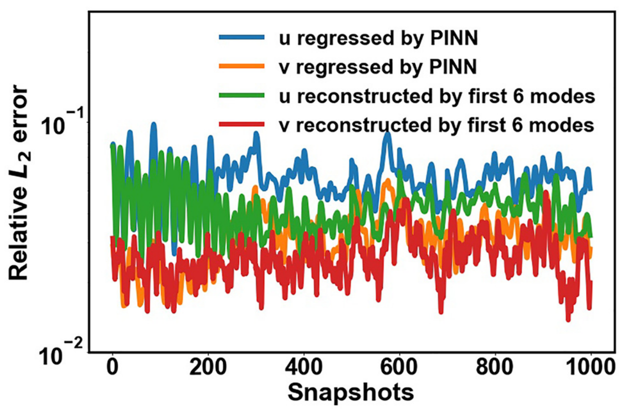

FIG. 32. Relative L2 error of regressed and reconstructed velocity fields at $Re = 3900$.

REFERENCES

[1]P. J. Schmid, "Dynamic mode decomposition and its variants," Annu. Rev. Fluid Mech. **54**, 225 (2022).
[2]J. L. Lumley, "The structure of inhomogeneous turbulent flows," Atmos. Turbul. Radio Wave Propag. **790**, 166–178 (1967).
[3]P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," J. Fluid Mech. **656**, 5 (2010).
[4]B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," Proc. Natl. Acad. Sci. U. S. A. **17**, 315 (1931).
[5]V. Theofilis, "Global linear instability," Annu. Rev. Fluid Mech. **43**, 319–352 (2011).
[6]P. J. Schmid and D. S. Henningson, "*Stability and transition in shear flows*" (Springer, New York, 2001).
[7]J. Delville, L. Ukeiley, L. Cordier, J.-P. Bonnet, and M. Glauser, "Examination of large-scale structures in a turbulent plane mixing layer. Part 1. Proper orthogonal decomposition," J. Fluid Mech. **391**, 91 (1999).
[8]E. Liberge and A. Hamdouni, "Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder," J. Fluids Struct. **26**, 292 (2010).
[9]T. W. Muld, G. Efraimsson, and D. S. Henningson, "Mode decomposition on surface-mounted cube," Flow Turbul. Combust. **88**, 279 (2012).
[10]Y. Liu, J. Long, Q. Wu, B. Huang, and G. Wang, "Data-driven modal decomposition of transient cavitating flow," Phys. Fluids **33**, 113316 (2021).
[11]T. W. Muld, G. Efraimsson, and D. S. Henningson, "Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition," Comput. Fluids **57**, 87 (2012).
[12]H. Wang, H. Cao, and Y. Zhou, "POD analysis of a finite-length cylinder near wake," Exp. Fluids **55**, 1–5 (2014).
[13]Y. Yeh and H. Cummins, "Localized fluid flow measurements with an He–Ne laser spectrometer," Appl. Phys. Lett. **4**, 176 (1964).
[14]F. D. Felice and F. Pereira, "Developments and applications of PIV in naval hydrodynamics," in *Particle Image Velocimetry* (Springer, 2007), pp. 475–503.
[15]C. Dewey, Jr., "Qualitative and quantitative flow field visualization utilizing laser-induced fluorescence," in *Proceedings of the Applications of Non-Intrusive Instrumentation in Fluid Flow Research, Saint-Louis, France, 1 May 1976* (1976).
[16]J. P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis, "CFD vision 2030 study: A path to revolutionary computational aerosciences," Technical Report No. NASA/CR-2014-218178 (NASA, 2014).
[17]A. W. Cary, J. Chawner, E. P. Duque, W. Gropp, W. L. Kleb, R. M. Kolonay, E. Nielsen, and B. Smith, "CFD vision 2030 road map: Progress and perspectives," in *AIAA Aviation 2021 Forum* (AIAA, 2021), p. 2726.
[18]Y. Liu, W. Cao, W. Zhang, and Z. Xia, "Analysis on numerical stability and convergence of Reynolds averaged Navier–Stokes simulations from the perspective of coupling modes," Phys. Fluids **34**, 015120 (2022).
[19]S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**, 477 (2020).
[20]X. Jin, P. Cheng, W.-L. Chen, and H. Li, "Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," Phys. Fluids **30**, 047105 (2018).
[21]M. Xu, H. Cheng, and B. Ji, "RANS simulation of unsteady cavitation around a Clark-Y hydrofoil with the assistance of machine learning," Ocean Eng. **231**, 109058 (2021).
[22]X. Ma, C. Wang, B. Huang, and G. Wang, "Application of two-branch deep neural network to predict bubble migration near elastic boundaries," Phys. Fluids **31**, 102003 (2019).
[23]C. Zheng, T. Ji, F. Xie, X. Zhang, H. Zheng, and Y. Zheng, "From active learning to deep reinforcement learning: Intelligent active flow control in suppressing vortex-induced vibration," Phys. Fluids **33**, 063607 (2021).
[24]X. Zhang, T. Ji, F. Xie, H. Zheng, and Y. Zheng, "Unsteady flow prediction from sparse measurements by compressed sensing reduced order modeling," Comput. Methods Appl. Mech. Eng. **393**, 114800 (2022).
[25]W. Peng, Z. Yuan, and J. Wang, "Attention-enhanced neural network models for turbulence simulation," Phys. Fluids **34**, 025111 (2022).
[26]Z. Yuan, C. Xie, and J. Wang, "Deconvolutional artificial neural network models for large eddy simulation of turbulence," Phys. Fluids **32**, 115106 (2020).
[27]I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," IEEE Trans. Neural Networks **9**, 987 (1998).
[28]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686 (2019).
[29]M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science **367**, 1026 (2020).
[30]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).
[31]H. Wang, Y. Liu, and S. Wang, "Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network," Phys. Fluids **34**, 017116 (2022).
[32]S. Xu, Z. Sun, R. Huang, D. Guo, G. Yang, and S. Ju, "A practical approach to flow field reconstruction with sparse or incomplete data through physics informed neural network," Acta Mech. Sin. **39**, 322302 (2023).
[33]R. Qiu, R. Huang, Y. Xiao, J. Wang, Z. Zhang, J. Yue, Z. Zeng, and Y. Wang, "Physics-informed neural networks for phase-field method in two-phase flow," Phys. Fluids **34**, 052109 (2022).
[34]Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020).
[35]C. Rao, H. Sun, and Y. Liu, "Physics-informed deep learning for incompressible laminar flows," Theor. Appl. Mech. Lett. **10**, 207 (2020).
[36]Q. Zhu, Z. Liu, and J. Yan, "Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks," Comput. Mech. **67**, 619 (2021).
[37]H. Xu, W. Zhang, and Y. Wang, "Explore missing flow dynamics by physics-informed deep learning: The parameterized governing systems," Phys. Fluids **33**, 095116 (2021).
[38]C. Cheng and G.-T. Zhang, "Deep learning method based on physics informed neural network with Resnet block for solving fluid flow problems," Water **13**, 423 (2021).
[39]L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," Comput. Methods Appl. Mech. Eng. **361**, 112732 (2020).
[40]X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," J. Comput. Phys. **426**, 109951 (2021).
[41]A. D. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations," in AAAI Spring Symposium: MLPS, 2021.

[42]I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," arXiv:1608.03983 (2016).
[43]K. Taira, S. L. Brunton, S. T. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," AIAA J. **55**, 4013 (2017).
[44]L. Sirovich, "Turbulence and the dynamics of coherent structures, Parts I, II and III," Q. Appl. Math. 561 **45**, 561 (1987).
[45]H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, "Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations," Phys. Fluids **34**, 075117 (2022).
[46]A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," J. Mach. Learn. Res. **18**, 1–43 (2018).
[47]Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," arXiv:1901.06523 (2019).
[48]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).
[49]C. H. K. Williamson, "Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers," J. Fluid Mech. **206**, 579 (1989).
[50]R. D. Henderson, "Nonlinear dynamics and pattern formation in turbulent wake transition," J. Fluid Mech. **352**, 65 (1997).
[51]M. M. Rahman, M. M. Karim, and M. A. Alim, "Numerical investigation of unsteady flow past a circular cylinder using 2D finite volume method," J. Naval Archit. Mar. Eng. **4**, 27–42 (1970).
[52]C. Norberg, "An experimental investigation of the flow around a circular cylinder: Influence of aspect ratio," J. Fluid Mech. **258**, 287 (1994).
[53]D. A. Lysenko, I. S. Ertesvåg, and K. E. Rian, "Large-eddy simulation of the flow over a circular cylinder at Reynolds number 3900 using the OpenFOAM toolbox," Flow Turbul. Combust. **89**, 491 (2012).
[54]G. F. Rosetti, G. Vaz, and A. L. Fujarra, "URANS calculations for smooth circular cylinder flow in a wide range of Reynolds numbers: Solution verification and validation," J. Fluids Eng. **134**, 121103 (2012).
[55]R. Stringer, J. Zang, and A. Hillis, "Unsteady rans computations of flow around a circular cylinder for a wide range of Reynolds numbers," Ocean Eng. **87**, 1–9 (2014).
[56]H. Jiang and L. Cheng, "Large-eddy simulation of flow past a circular cylinder for Reynolds numbers 400 to 3900," Phys. Fluids **33**, 034119 (2021).